Migrate SQL Server to SQL Server on Azure Virtual Machines

Pre-migration overview

As you prepare for migrating your SQL Server database to SQL Server on Azure VMs, be sure to consider the versions of SQL Server that are supported and to address any prerequisites. This will help to ensure an efficient and successful migration.

Important: For scenarios in which you need to meet certain regulatory, policy, or compliance requirements that prevent you from using SQL Server on Azure VMs, you can use Azure Stack. You can run Azure Stack at the edge for remote locations, run it completely disconnected from the internet, or create a hybrid solution. The guidance in this scenario applies to migrations from SQL Server on-premises to SQL Server on Azure VMs as well as migrations to Azure Stack Virtual Machines.

Supported versions

SQL Server on Azure VMs supports migration of the following versions of SQL Server:

- SQL Server 2005
- SQL Server 2008 and SQL Server 2008 R2
- SQL Server 2012 and SQL Server 2012 SP1 CU2
- SQL Server 2014
- SQL Server 2016
- SQL Server 2017

Additional resources

- See the white paper Choosing your database migration path to Azure for additional information and recommendations.
- For a matrix of the Microsoft and third-party services and tools that are available to assist you with various database and data migration scenarios as well as specialty tasks, see the article Service and tools for data migration.

Prerequisites

Before beginning your migration project, it is important to address the associated prerequisites.

- Download and install the latest version of the MAP Toolkit.
- Download and install the latest version of the Database Experimentation Assistant.
- Download and install the Data Migration Assistant v3.3 or later.
- Create an instance of SQL Server on Azure VMs by following the detail in the article How to provision a Windows SQL Server virtual machine in the Azure portal.

Discover

The goal of the Discover phase is to identify existing data sources and details about the features that are being used to get a better understanding of and plan for the migration. This process involves scanning the network to identify all your organization's SQL instances together with the version and features in use.

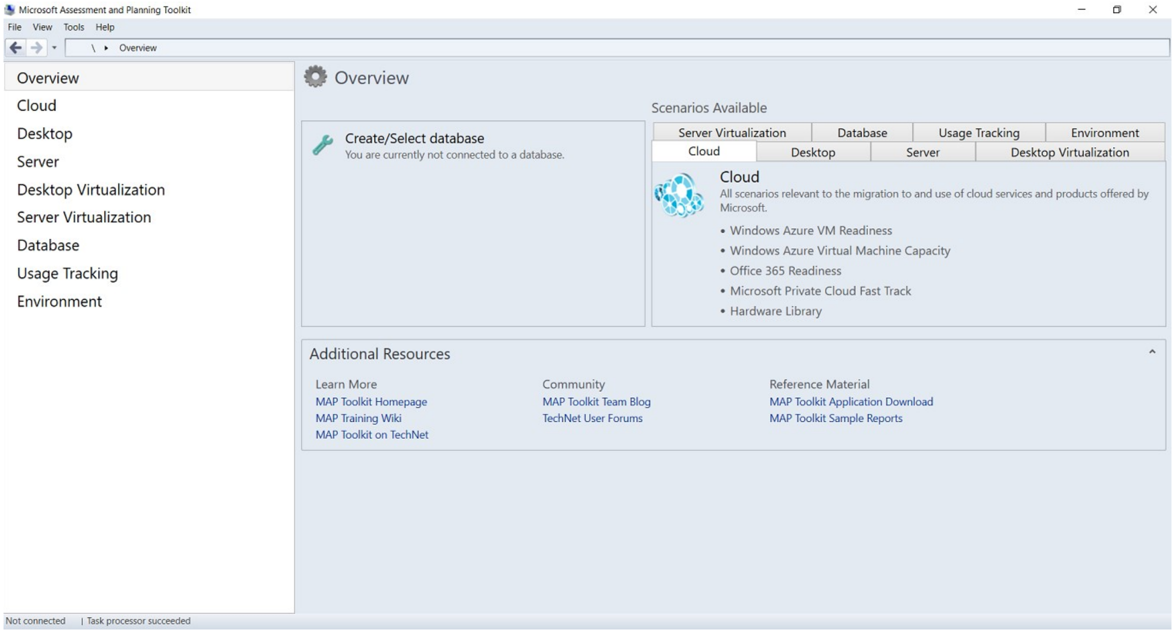
To use the MAP Toolkit to perform an inventory scan, perform the following steps.

Steps

1. Run the MAP Toolkit.

   a. Open the MAP Toolkit, and then on the left pane, select Database.

   You will be on the following screen:

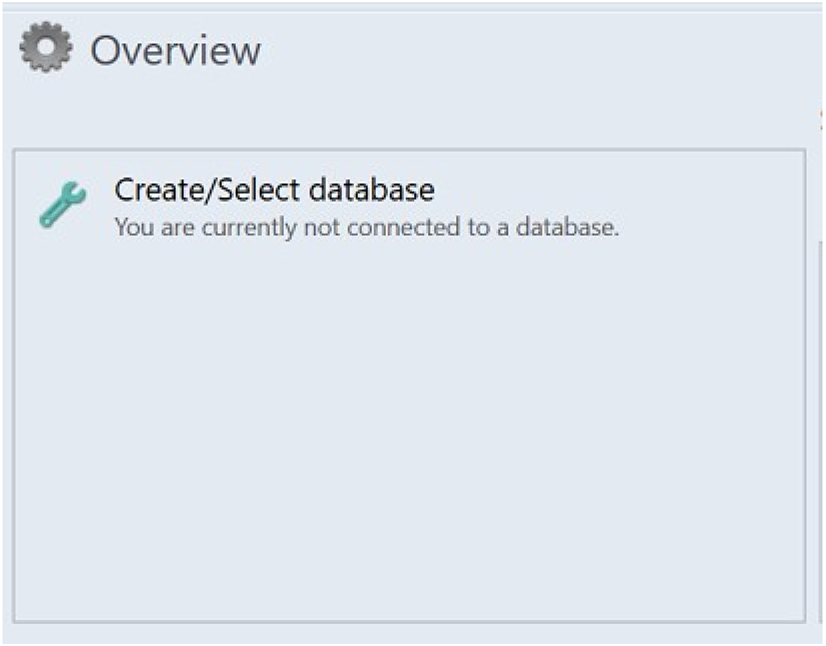


You are currently not connected to a database.

All scenarios relevant to the migration to and use of cloud services and products offered by Microsoft.

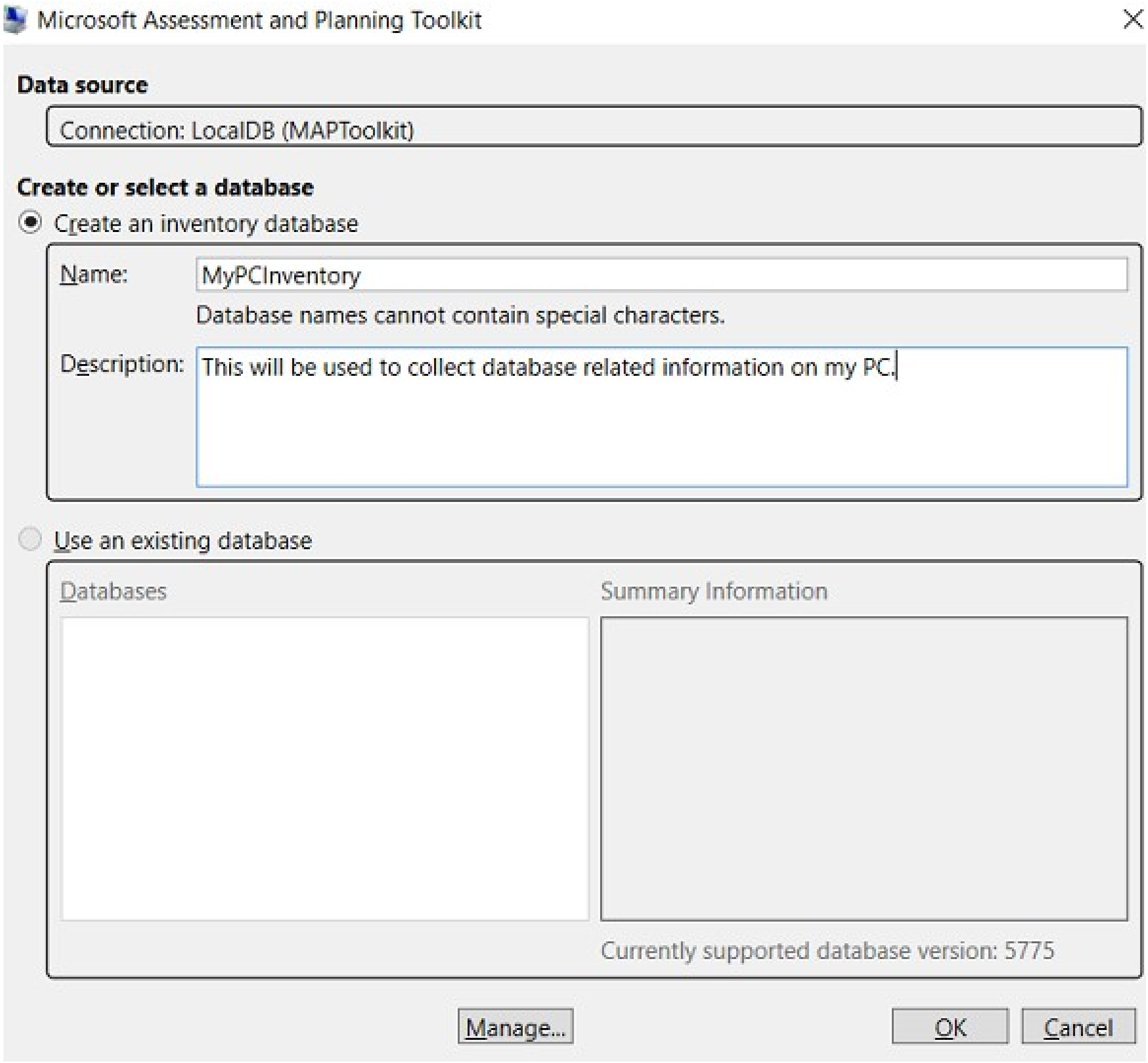
- Windows Azure VM Readiness
- Windows Azure Virtual Machine Capacity
- Office 365 Readiness
- Microsoft Private Cloud Fast Track
- Hardware Library

b. Select Create/Select database.



You are currently not connected to a database.

c. Ensure that Create an inventory database is selected, enter a name for the database, a brief description, and then select OK.



Microsoft Assessment and Planning Toolkit

Data source

Connection: LocalDB (MAPToolkit)

Create or select a database

● Create an inventory database

Name: MyPCInventory

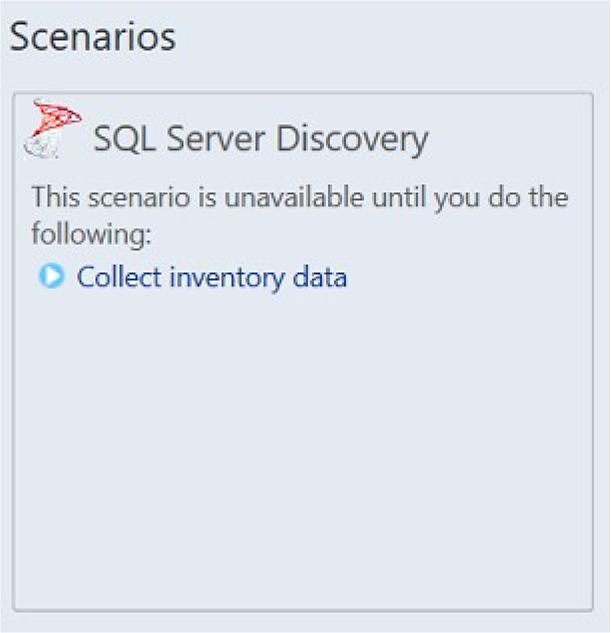
Database names cannot contain special characters.

Description: This will be used to collect database related information on my PC.

○ Use an existing database

The next step is to collect data from the database created.

d. Select Collect inventory data.



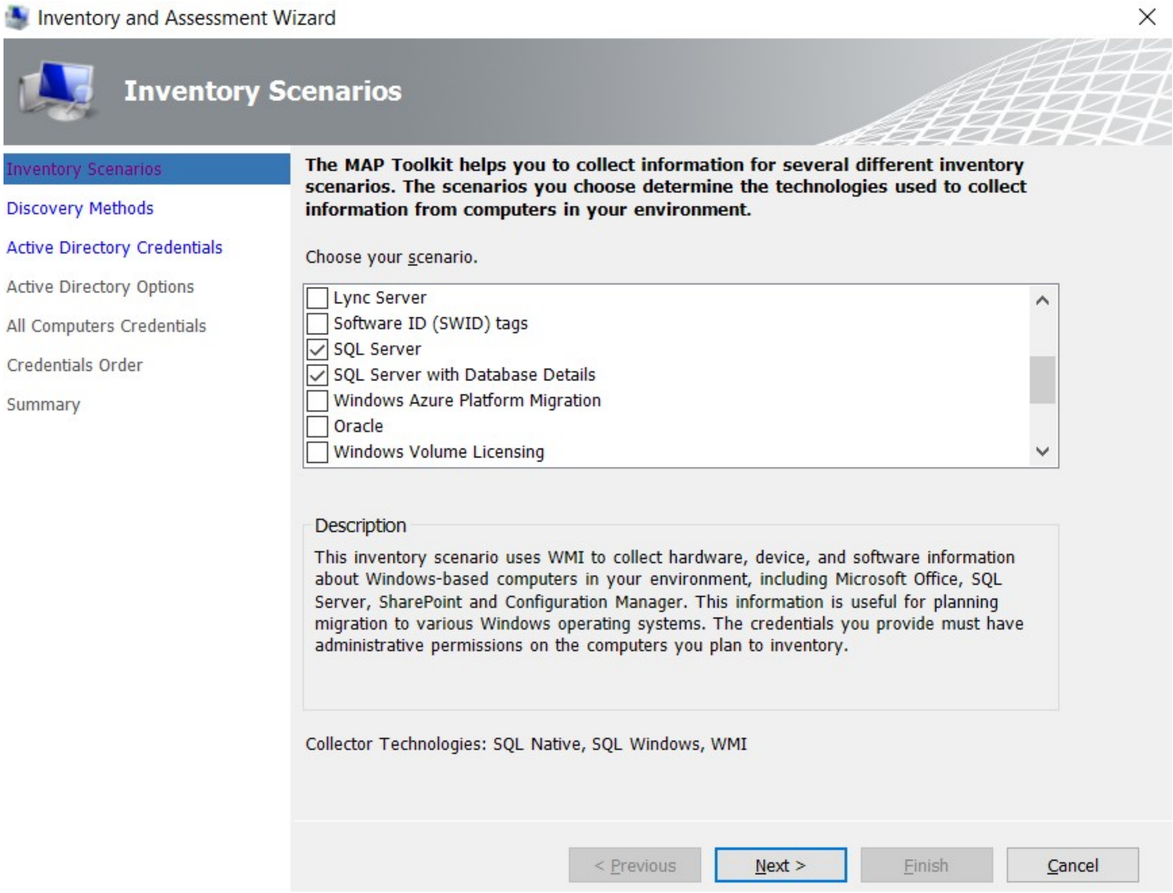
Scenarios

SQL Server Discovery

This scenario is unavailable until you do the following:

Collect inventory data

e. In the Inventory and Assessment Wizard, select SQL Server and SQL Server with Database Details, and then select Next.



The MAP Toolkit helps you to collect information for several different inventory scenarios. The scenarios you choose determine the technologies used to collect information from computers in your environment.

Choose your scenario.

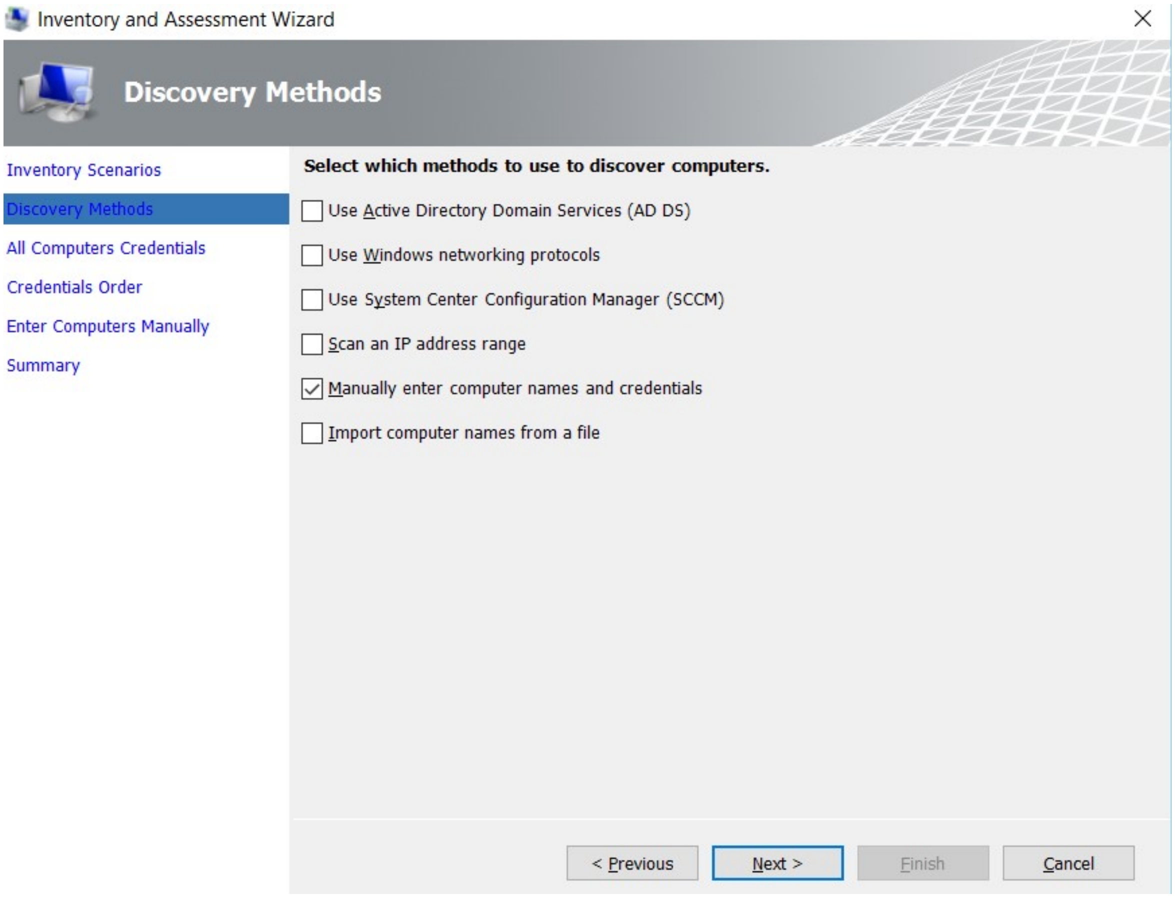
- ☐ Lync Server
- ☐ Software ID (SWID) tags
- ☑ SQL Server
- ☑ SQL Server with Database Details
- ☐ Windows Azure Platform Migration
- ☐ Oracle
- ☐ Windows Volume Licensing

Description

This inventory scenario uses WMI to collect hardware, device, and software information about Windows-based computers in your environment, including Microsoft Office, SQL Server, SharePoint and Configuration Manager. This information is useful for planning migration to various Windows operating systems. The credentials you provide must have administrative permissions on the computers you plan to inventory.

Collector Technologies: SQL Native, SQL Windows, WMI

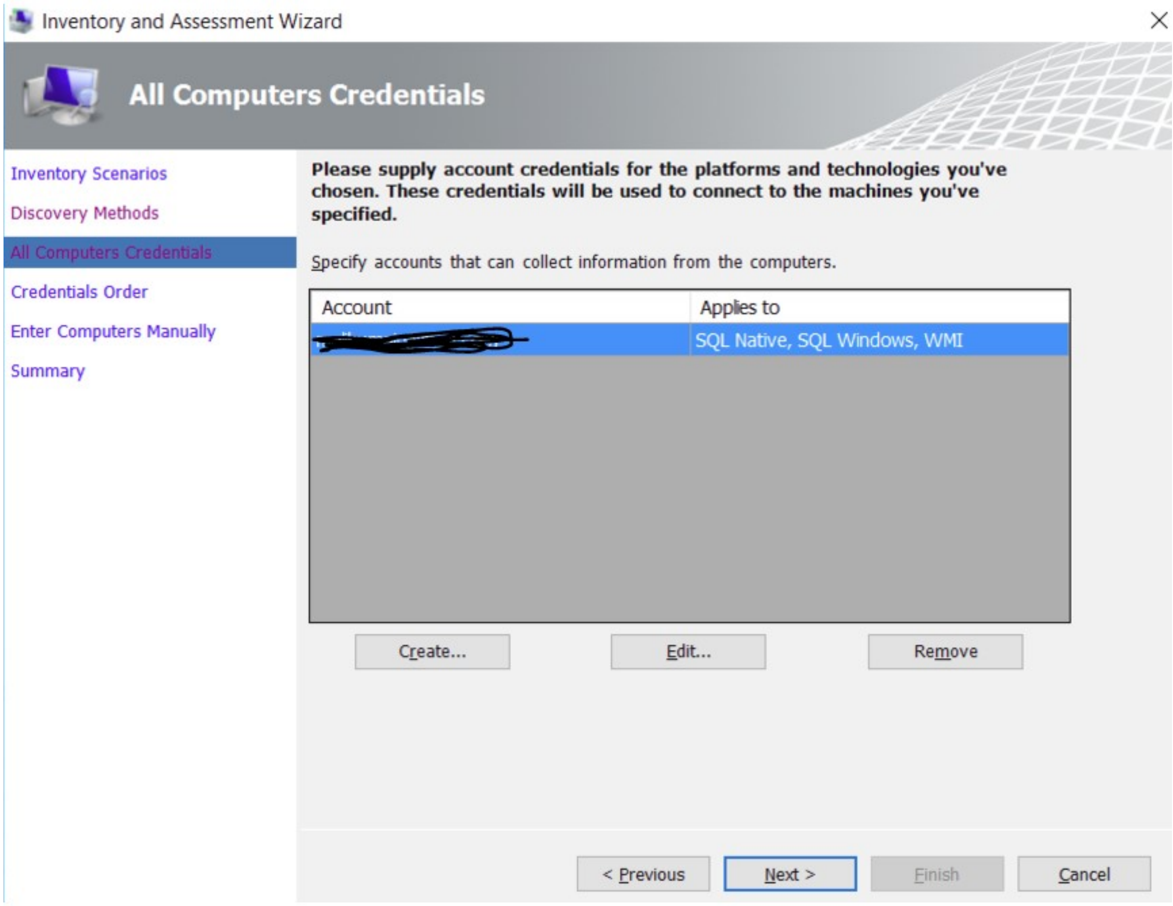
f. Select the best method option to search the computers on which Microsoft Products are hosted, and then select Next.



Select which methods to use to discover computers.

- ☐ Use Active Directory Domain Services (AD DS)
- ☐ Use Windows networking protocols
- ☐ Use System Center Configuration Manager (SCCM)
- ☐ Scan an IP address range
- ☑ Manually enter computer names and credentials
- ☐ Import computer names from a file

g. Enter credentials or create new credentials for the systems that you want to explore, and then select Next.

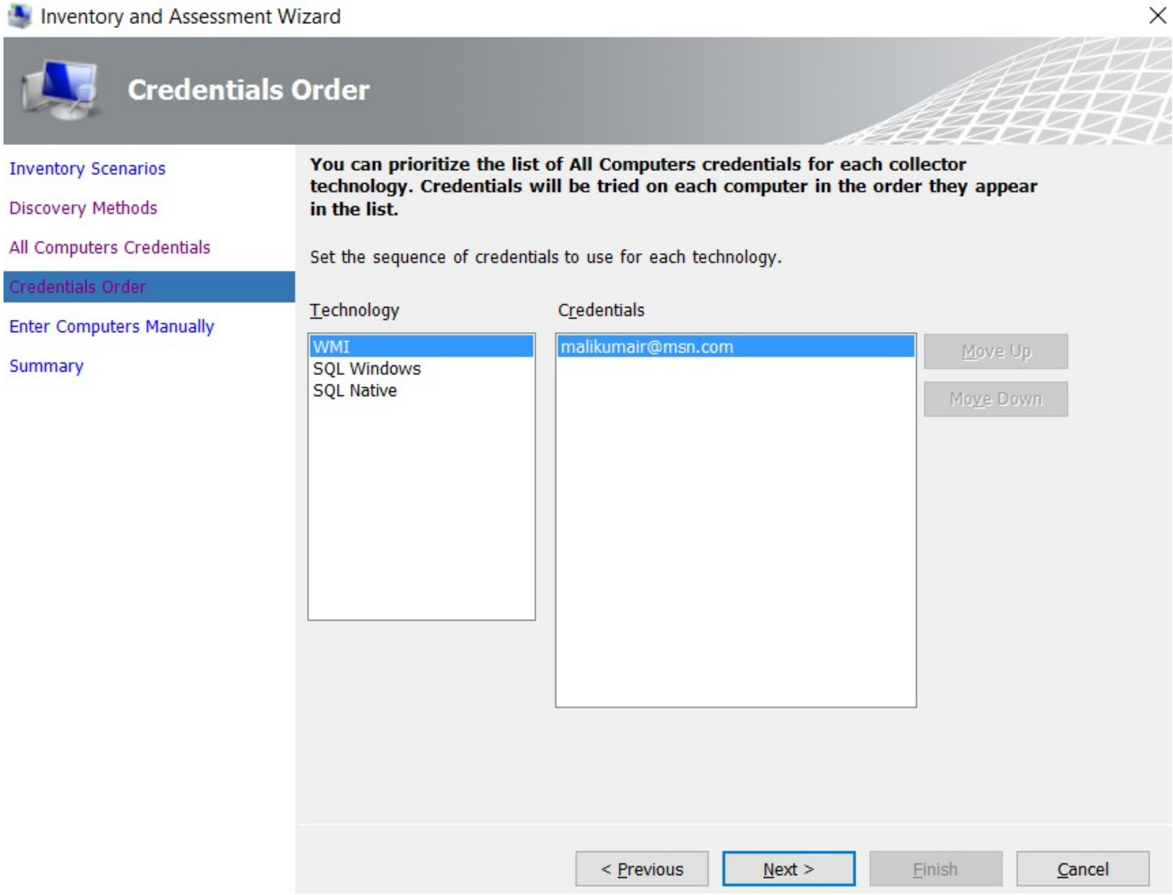


Please supply account credentials for the platforms and technologies you've chosen. These credentials will be used to connect to the machines you've specified.

Specify accounts that can collect information from the computers.

Account	Applies to
[redacted]	SQL Native, SQL Windows, WMI

h. Set the order of the credentials, and then select Next.


Inventory and Assessment Wizard

Credentials Order

You can prioritize the list of All Computers credentials for each collector technology. Credentials will be tried on each computer in the order they appear in the list.

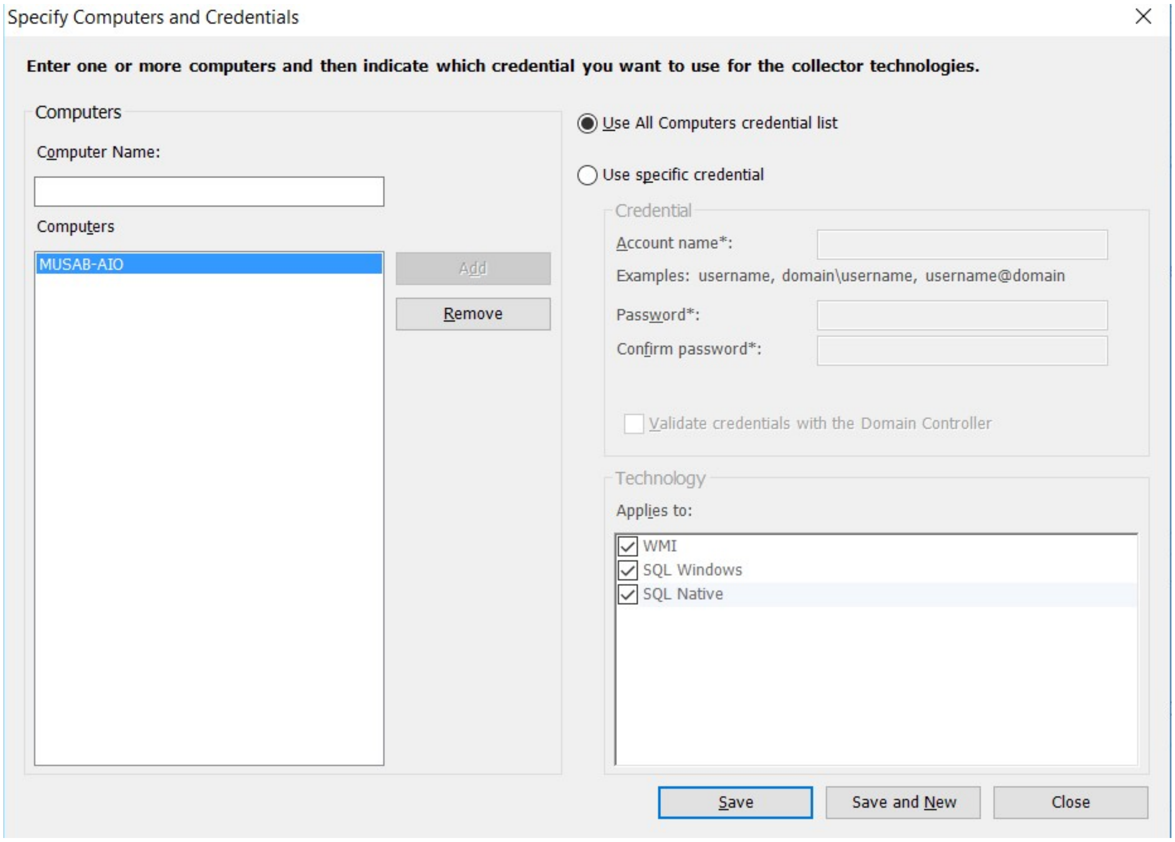
Set the sequence of credentials to use for each technology.

Technology: WMI, SQL Windows, SQL Native

Credentials: malikumair@msn.com

Now, you need to specify the credentials for each computer that you want to discover. You can use unique credentials for each computer/machine, or you can choose to use the All Computer Credentials list.

i. After setting up the credentials, select Save, and then select Next.



Specify Computers and Credentials

Enter one or more computers and then indicate which credential you want to use for the collector technologies.

Computers

Computer Name:

Computers

MUSAB-AIO

● Use All Computers credential list

○ Use specific credential

Credential

Account name*:

Examples: username, domain\username, username@domain

Password*:

Confirm password*:

□ Validate credentials with the Domain Controller

Technology

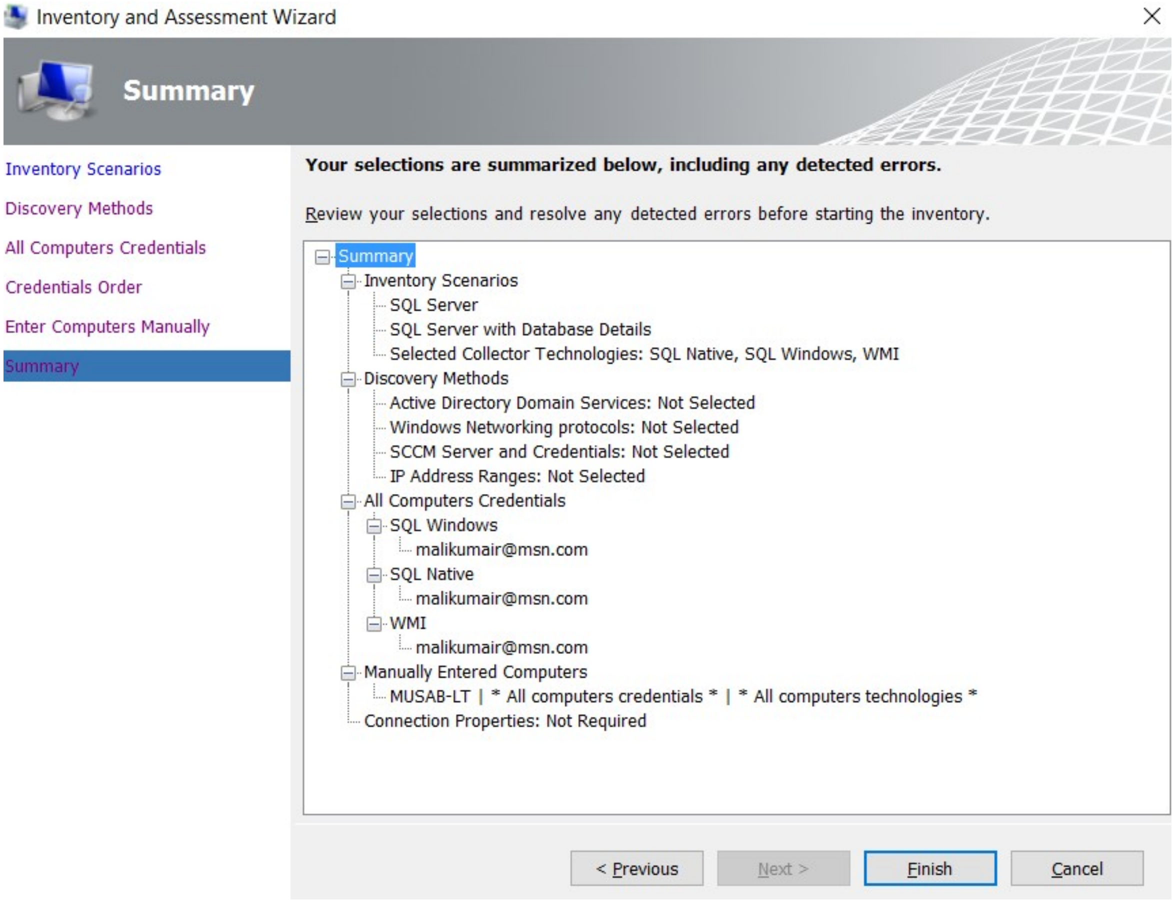
Applies to:

☑ WMI

☑ SQL Windows

☑ SQL Native

j. Verify your selection summary, and then select Finish.

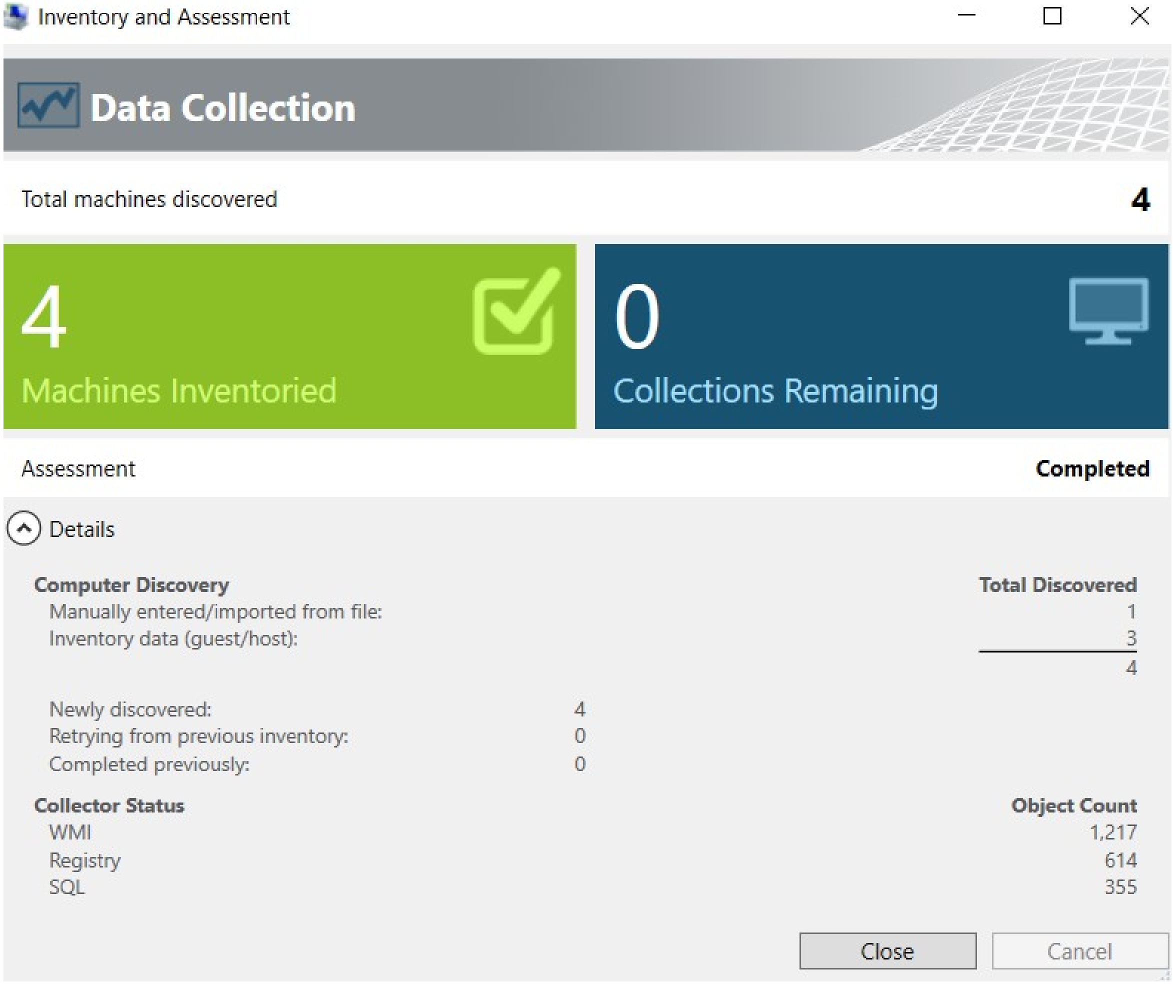


Your selections are summarized below, including any detected errors.

Review your selections and resolve any detected errors before starting the inventory.

```
Summary
  Inventory Scenarios
    SQL Server
    SQL Server with Database Details
    Selected Collector Technologies: SQL Native, SQL Windows, WMI
  Discovery Methods
    Active Directory Domain Services: Not Selected
    Windows Networking protocols: Not Selected
    SCCM Server and Credentials: Not Selected
    IP Address Ranges: Not Selected
  All Computers Credentials
    SQL Windows
      malikumair@msn.com
    SQL Native
      malikumair@msn.com
    WMI
      malikumair@msn.com
  Manually Entered Computers
    MUSAB-LT | * All computers credentials * | * All computers technologies *
  Connection Properties: Not Required
```

k. Wait for a few minutes (depending on the number of databases) for the Data Collection summary report.



Inventory and Assessment

Data Collection

Total machines discovered 4

4 Machines Inventoried

0 Collections Remaining

Assessment Completed

Details

Computer Discovery		Total Discovered
Manually entered/imported from file:		1
Inventory data (guest/host):		3
		4
Newly discovered:	4	
Retrying from previous inventory:	0	
Completed previously:	0	

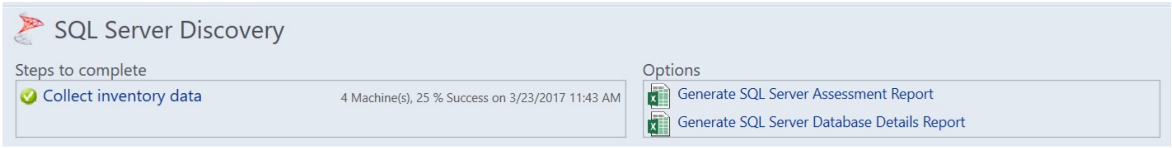
Collector Status	Object Count
WMI	1,217
Registry	614
SQL	355

l. Select Close.

The Main window of the tool appears, showing a summary of the Database Discovery completed so far.

2. Generate a report.

On the top-right corner of the tool, an Options page appears, which you can use to generate report about the SQL Server Assessment and the Database Details.



SQL Server Discovery

Steps to complete

✓ Collect inventory data 4 Machine(s), 25 % Success on 3/23/2017 11:43 AM

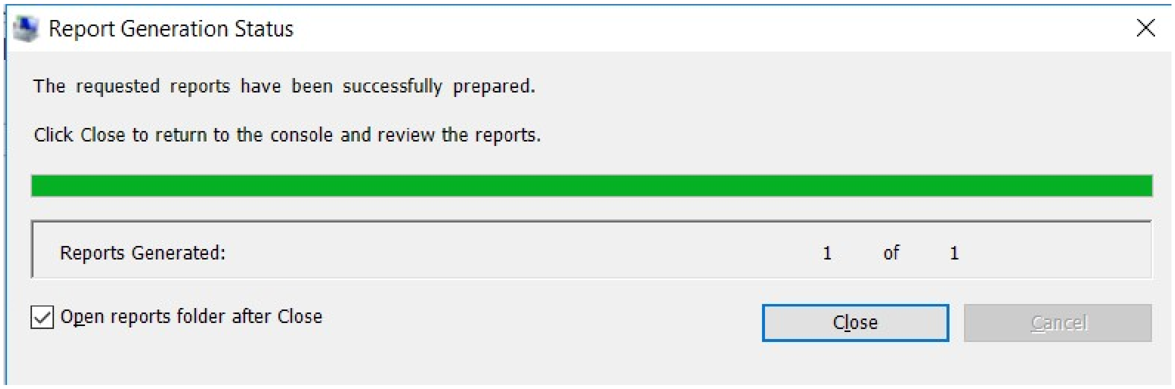
Options

Generate SQL Server Assessment Report

Generate SQL Server Database Details Report

a. Select both options (one by one) to generate the report.

This will take a couple of seconds to a few minutes depending on the size of the inventory completed during discovery.



Report Generation Status

The requested reports have been successfully prepared.

Click Close to return to the console and review the reports.

Reports Generated: 1 of 1

☑ Open reports folder after Close

Assess

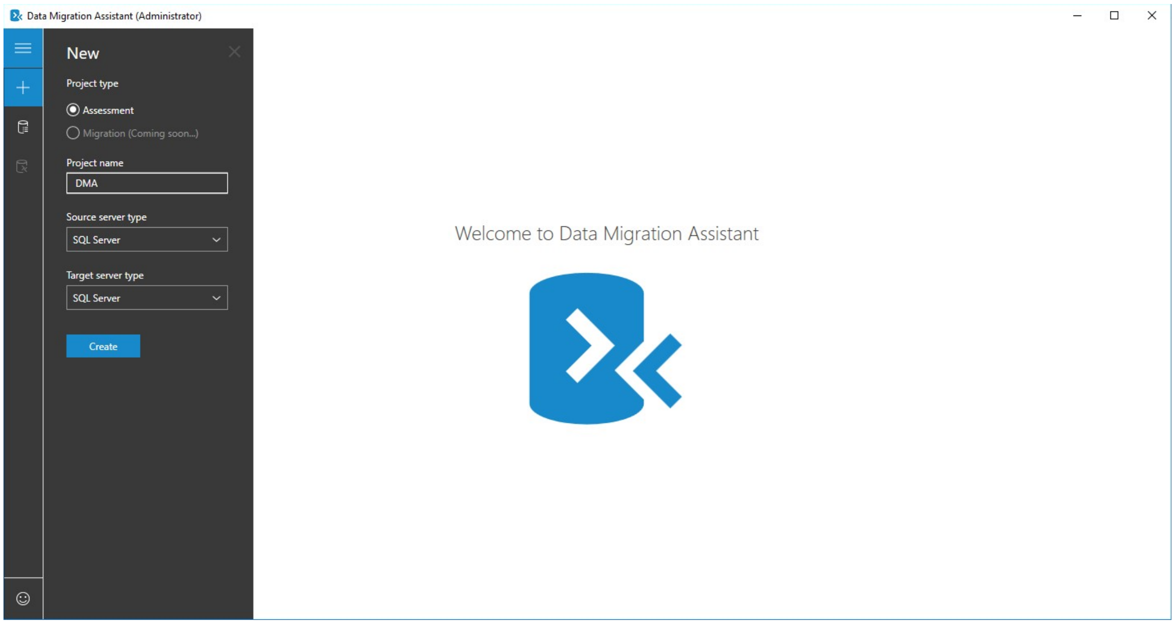
After identifying the data sources, the next step is to assess the on-premises SQL Server instance(s) migrating to a specific version of SQL Server on Azure VMs so that you understand the gaps between the source and target instances. Use the Data Migration Assistant (DMA) to assess your source database before migrating your SQL Server instance.

Steps

To use DMA to create an assessment, perform the following steps.

1. Create a new assessment project.

   a. Launch DMA, select the New (+) icon, select the Assessment project type, specify a project name, select SQL Server as the source and target, and then select Create.



Data Migration Assistant (Administrator)

New

Project type

● Assessment

○ Migration (Coming soon...)

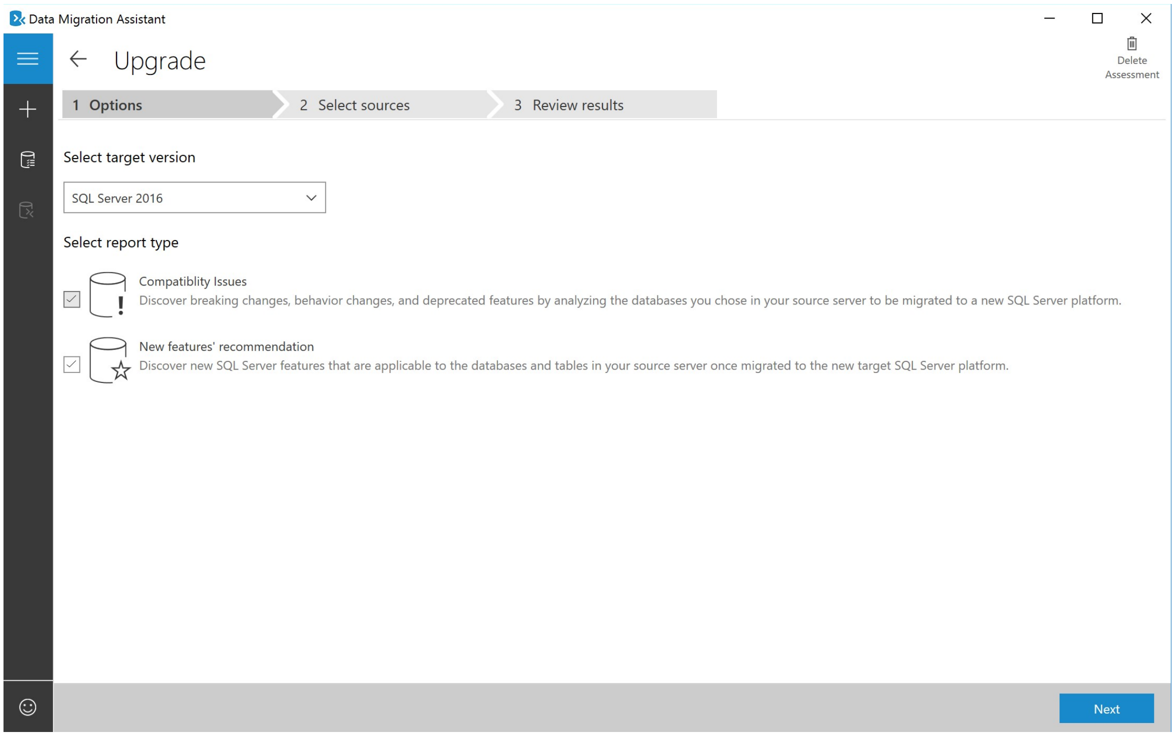
Project name: DMA

Source server type: SQL Server

Target server type: SQL Server

Welcome to Data Migration Assistant

b. Select the target SQL Server version that you plan to migrate to and against which you need to run an assessment, select one or both of the assessment report types (Compatibility Issues and New features' recommendation), and then select Next.



Select target version

SQL Server 2016

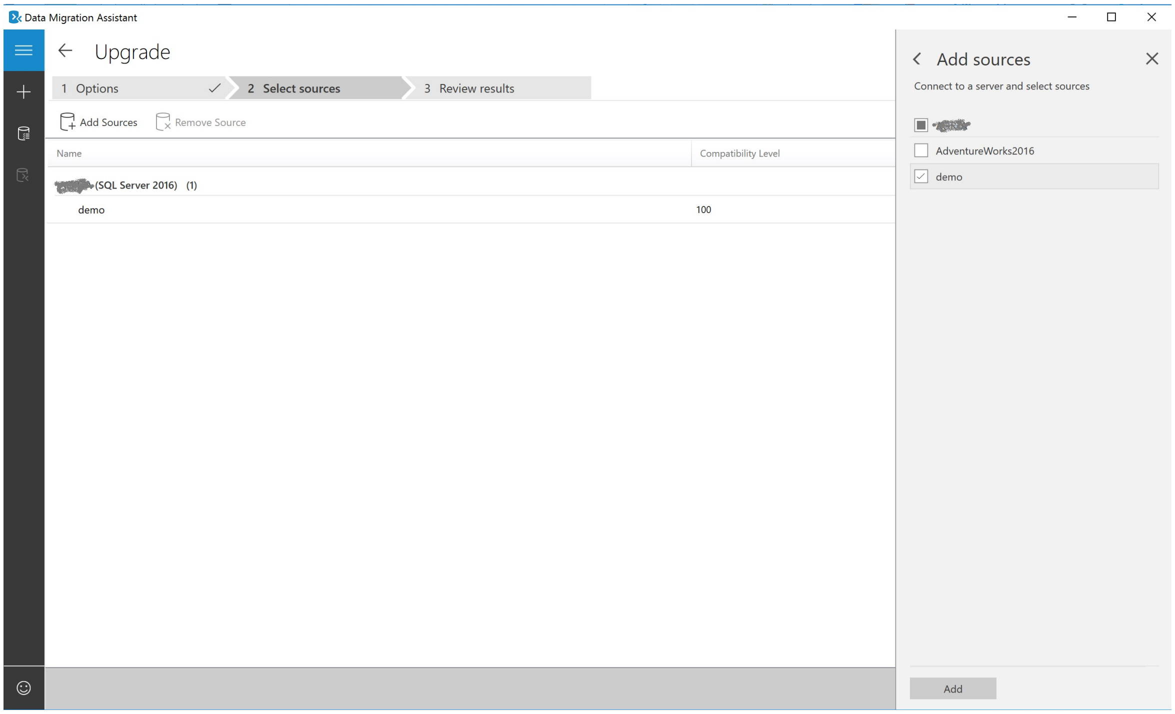
Select report type

☑ Compatiblity Issues
Discover breaking changes, behavior changes, and deprecated features by analyzing the databases you chose in your source server to be migrated to a new SQL Server platform.

☑ New features' recommendation
Discover new SQL Server features that are applicable to the databases and tables in your source server once migrated to the new target SQL Server platform.

c. In the Connect to a server fly-out, specify the name of the SQL Server instance to connect to, specify the Authentication type and Connection properties, and then select Connect.

d. In the Add Sources fly-out, select the database(s) you that want to assess, and then select Add.



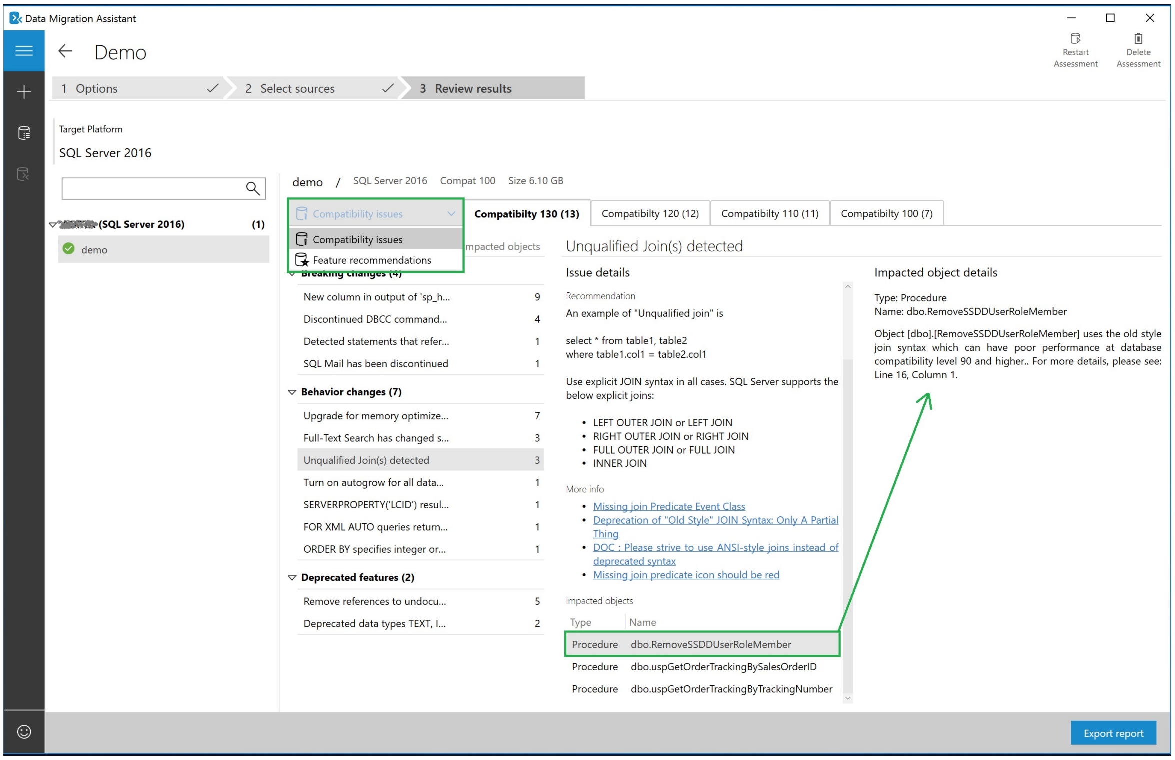
Name	Compatibility Level
[redacted] (SQL Server 2016) (1)	
demo	100

■ [redacted]
□ AdventureWorks2016
☑ demo

e. Select Start Assessment.

Now wait for the assessment results; the duration of the assessment depends on the number of databases added and the schema size of each database. Results will be displayed per database as soon as they are available.

f. Select the database that has completed assessment, and then switch between Compatibility issues and Feature recommendations by using the switcher.



New column in output of 'sp_h...	9
Discontinued DBCC command...	4
Detected statements that refer...	1
SQL Mail has been discontinued	1
Upgrade for memory optimize...	7
Full-Text Search has changed s...	3
Unqualified Join(s) detected	3
Turn on autogrow for all data...	1
SERVERPROPERTY('LCID') resul...	1
FOR XML AUTO queries return...	1
ORDER BY specifies integer or...	1
Remove references to undocu...	5
Deprecated data types TEXT, I...	2

Unqualified Join(s) detected

Issue details

Recommendation

An example of "Unqualified join" is

```
select * from table1, table2
where table1.col1 = table2.col1
```

Use explicit JOIN syntax in all cases. SQL Server supports the below explicit joins:

- LEFT OUTER JOIN or LEFT JOIN
- RIGHT OUTER JOIN or RIGHT JOIN
- FULL OUTER JOIN or FULL JOIN
- INNER JOIN

More info

- Missing join Predicate Event Class
- Deprecation of "Old Style" JOIN Syntax: Only A Partial Thing
- DOC : Please strive to use ANSI-style joins instead of deprecated syntax
- Missing join predicate icon should be red

Impacted objects

Type	Name
Procedure	dbo.RemoveSSDDUserRoleMember
Procedure	dbo.uspGetOrderTrackingBySalesOrderID
Procedure	dbo.uspGetOrderTrackingByTrackingNumber

Impacted object details

Type: Procedure
Name: dbo.RemoveSSDDUserRoleMember

Object [dbo].[RemoveSSDDUserRoleMember] uses the old style join syntax which can have poor performance at database compatibility level 90 and higher.. For more details, please see: Line 16, Column 1.

g. Review the compatibility issues by analyzing the impacted object and its details for every issue identified under Breaking changes, Behavior changes, and Deprecated features.

h. Review feature recommendations across the Performance, Storage, and Security areas.

Feature recommendations cover a variety of features such as In-Memory OLTP and Columnstore, Stretch Database, Always Encrypted (AE), Dynamic Data Masking (DDM), and Transparent Data Encryption (TDE).

2. Review the assessment results.

a. After all database assessments are complete, select Export report to export the results to either a JSON or CSV file for analyzing the data at your own convenience.

Convert

After assessing the source database instance(s) you are migrating, for heterogenous migrations, you need to convert the schema to work in the target environment. Since this is a homogeneous migration, the Convert phase is unnecessary.

Migration overview

After you have the necessary prerequisites in place and have completed the tasks asociated with the Pre-migration phase, you are ready to perform the actual migration.

There are several methods for migrating a SQL Server database to an instance of SQL Server on Azure Virtual Machines (Azure VMs).

Overview of migration methods

The list below identifies the primary methods for migrating from SQL Server to SQL Server on Azure VMs.

- Use the Data Migration Assistant (DMA) to migrate the schema and data into an Azure VM.
- Perform an on-premises backup using compression, and then manually copy the backup file into an Azure VM.
- Perform a backup to URL, and then restore into an Azure VM from the URL.
- Detach and copy the data and log files to Azure Blob storage, and then attach to an Azure VM from the URL.
- Convert an on-premises VW to Hyper-V VHDs, upload to Azure Blob storage, and then deploy a new Azure VM using the uploaded VHD.
- Ship a hard drive using Windows Import/Export Service.
- Use the Add Azure Replica Wizard.
- Use SQL Server transactional replication.

A couple of key points to consider when reviewing the migration methods to determine the best option for your business scenario:

- For optimum data transfer performance, migrate database files into the SQL Server on Azure VM instance using a compressed backup file.
- To minimize downtime during the database migration process, use either the AlwaysOn option or the transactional replication option.

Tip: There is no supported way to upgrade the version/edition of an existing SQL Server on Azure VM instance. Instead, create a new SQL Server on Azure VM instance running the desired version/edition, and then use one of the techniques in the table above to move your databases.

If it is not possible to use the above methods, migrate your database manually. Using this method, you will generally start with a database backup followed by a copy of the database backup into Azure and then perform a database restore. You can also copy the database files themselves into Azure and then attach them.

Note: When upgrading from older versions of SQL Server to SQL Server 2014 or SQL Server 2016, we recommend that you address all dependencies on features that are not supported by the later version of SQL Server as part of the migration project. For more information on the supported editions and scenarios, see the article Upgrade SQL Server.

The following section provides details about using DMA to migrate the schema and data to an Azure VM.

For additional information about the other options for migrating from SQL Server on-premises to SQL Server on Azure VMs, see the article Migrate a SQL Server database to SQL Server in an Azure VM.

Migrate schema and data

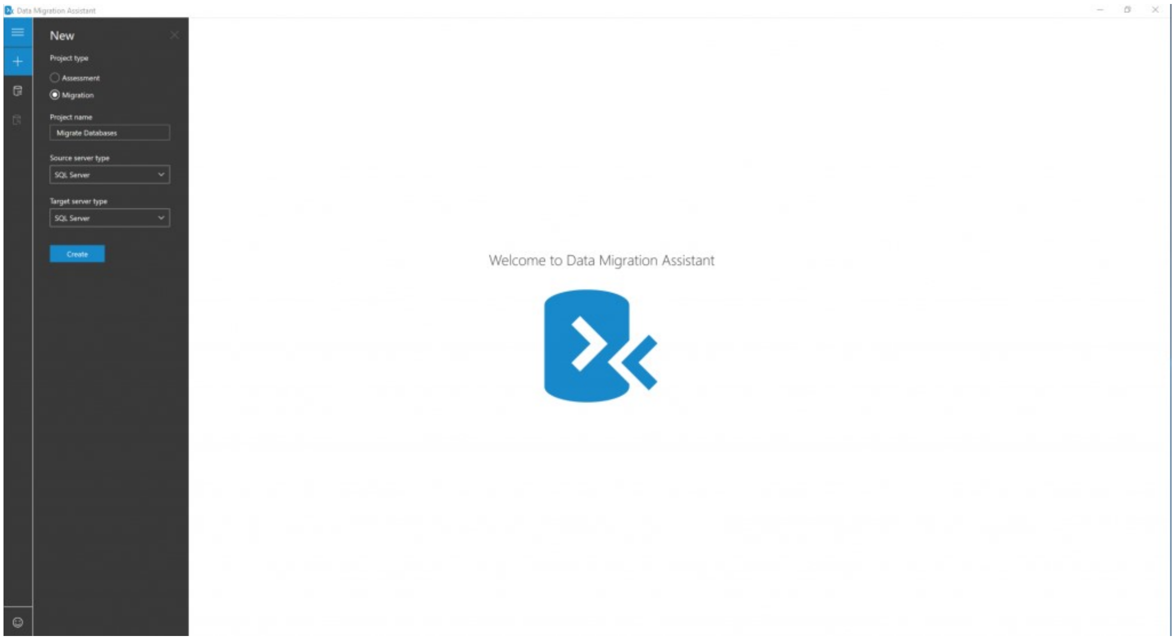
After assessing your databases, the next step is to begin the process of migrating the schema and database by using DMA.

Steps

To use DMA to create a migration project, perform the following steps.

1. Create a New Migration project

   a. Select the New icon, select the Migration project type, select SQL Server as source and target types, and then select Create.


Data Migration Assistant

New

Project type

○ Assessment

● Migration

Project name

Migrate Databases

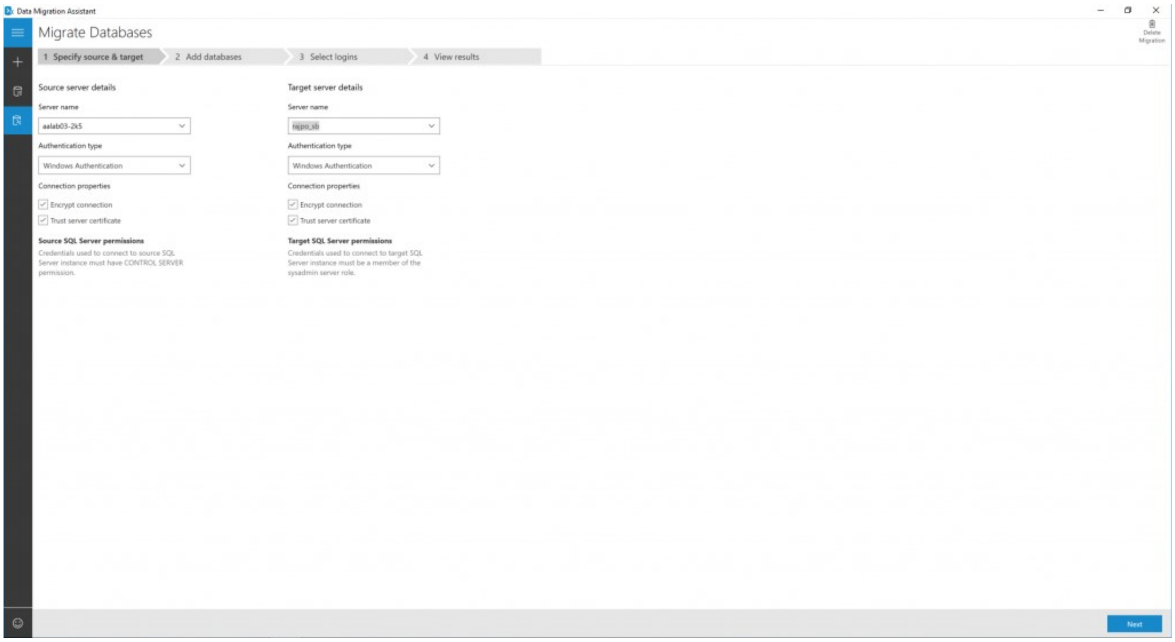
Source server type

SQL Server

Target server type

SQL Server

b. Provide source and target SQL server conenction details, and then select Next.


Data Migration Assistant

Migrate Databases

1 Specify source & target › 2 Add databases › 3 Select logins › 4 View results

Source server details

Server name

aalab03-2k5

Authentication type

Windows Authentication

Connection properties

☑ Encrypt connection

☑ Trust server certificate

Source SQL Server permissions

Credentials used to connect to source SQL Server instance must have CONTROL SERVER permission.

Target server details

Server name

rajpo_sb

Authentication type

Windows Authentication

Connection properties

☑ Encrypt connection

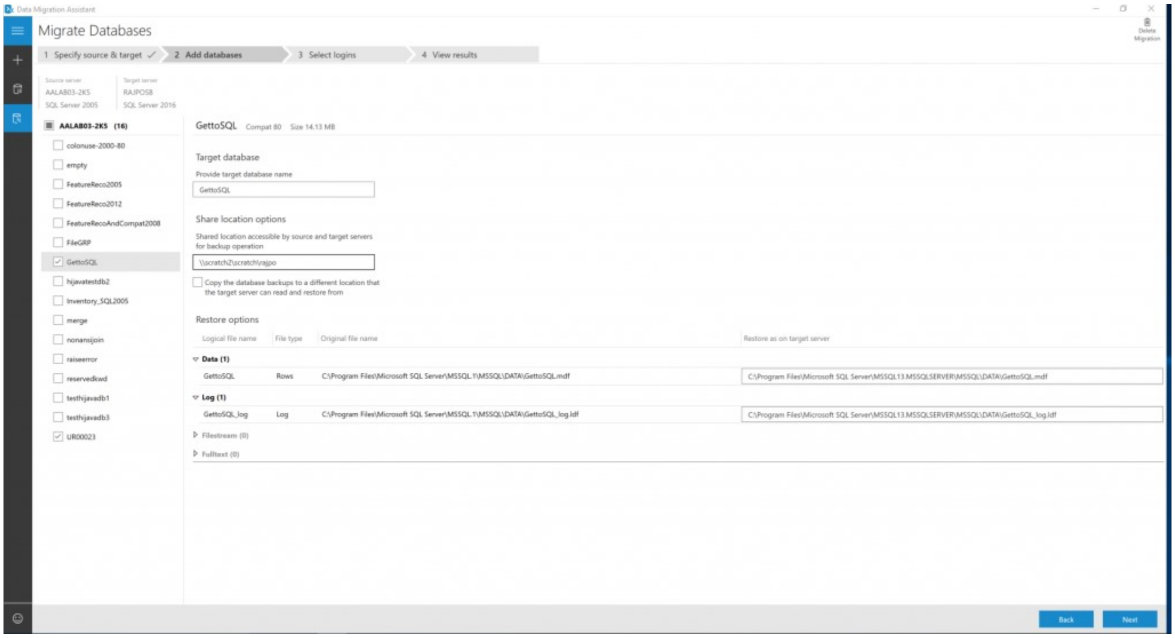
☑ Trust server certificate

Target SQL Server permissions

Credentials used to connect to target SQL Server instance must be a member of the sysadmin server role.

c. Select databases from the source to migrate, and then specify the Shared location accessible by source and target SQL servers for backup operation.

Note: Be sure that the service account running the source SQL Server instance has write privileges on the shared location and that the target SQL Server service account has read privileges on the shared location.



Migrate Databases

1 Specify source & target ✓ | 2 Add databases | 3 Select logins | 4 View results

Source server: AALAB03-2K5, SQL Server 2005
Target server: RAJPOSB, SQL Server 2016

■ AALAB03-2K5 (16)

- ☐ colonuse-2000-80
- ☐ empty
- ☐ FeatureReco2005
- ☐ FeatureReco2012
- ☐ FeatureRecoAndCompat2008
- ☐ FileGRP
- ☑ GettoSQL
- ☐ hijavatestdb2
- ☐ Inventory_SQL2005
- ☐ merge
- ☐ nonansijoin
- ☐ raiseerror
- ☐ reservedkwd
- ☐ testhijavadb1
- ☐ testhijavadb3
- ☑ UR00023

GettoSQL Compat 80 Size 14.13 MB

Target database

Provide target database name: GettoSQL

Share location options

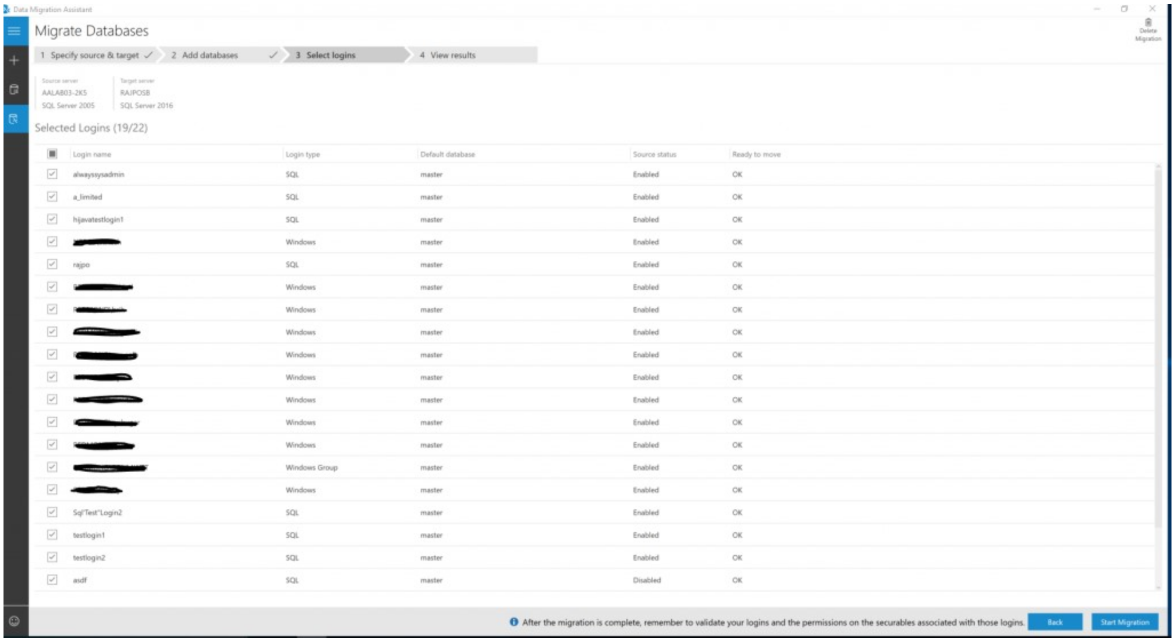
Shared location accessible by source and target servers for backup operation: \\scratch2\scratch\rajpo

☐ Copy the database backups to a different location that the target server can read and restore from

Restore options

Logical file name	File type	Original file name	Restore as on target server
▽ Data (1)			
GettoSQL	Rows	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\GettoSQL.mdf	C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\GettoSQL.mdf
▽ Log (1)			
GettoSQL_log	Log	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\GettoSQL_log.ldf	C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\GettoSQL_log.ldf
▷ Filestream (0)			
▷ Fulltext (0)			

d. Select Next, select the logins that you want to migrate, and then select Start Migration.



Migrate Databases

1 Specify source & target ✓ | 2 Add databases ✓ | 3 Select logins | 4 View results

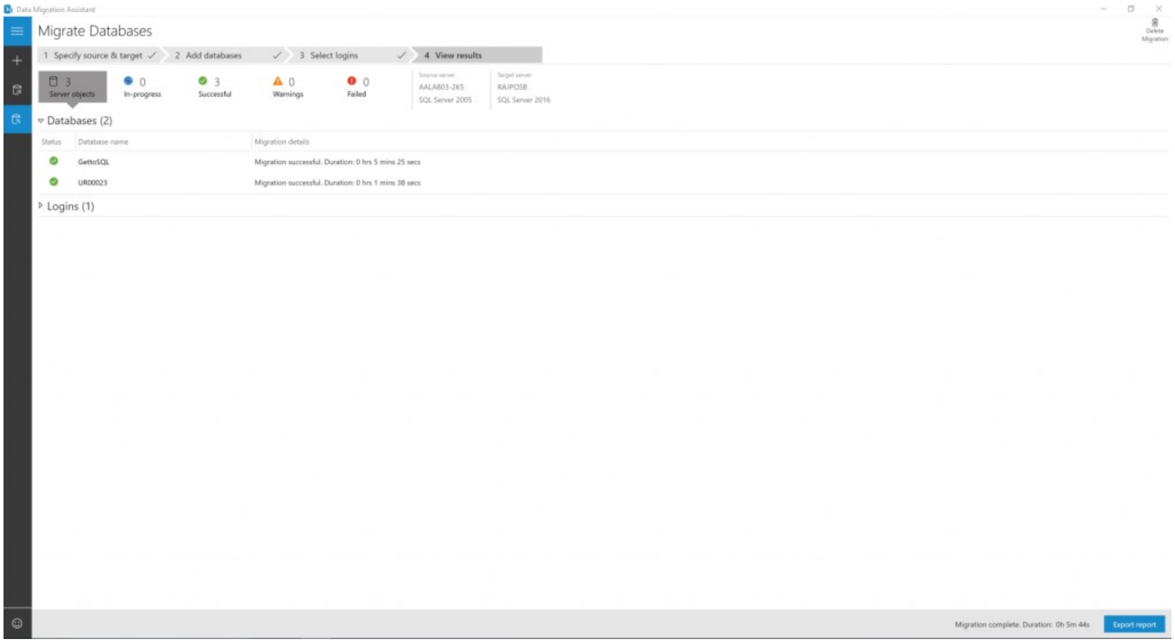
Source server: AALAB03-2K5, SQL Server 2005
Target server: RAJPOSB, SQL Server 2016

Selected Logins (19/22)

■	Login name	Login type	Default database	Source status	Ready to move
☑	alwayssysadmin	SQL	master	Enabled	OK
☑	a_limited	SQL	master	Enabled	OK
☑	hijavatestlogin1	SQL	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	rajpo	SQL	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	[redacted]	Windows Group	master	Enabled	OK
☑	[redacted]	Windows	master	Enabled	OK
☑	Sql'Test"Login2	SQL	master	Enabled	OK
☑	testlogin1	SQL	master	Enabled	OK
☑	testlogin2	SQL	master	Enabled	OK
☑	asdf	SQL	master	Disabled	OK

After the migration is complete, remember to validate your logins and the permissions on the securables associated with those logins.

e. Now, monitor the progress of migration in the View Results screen.



Source server: AALAB03-2K5, SQL Server 2005

Target server: RAJPOSB, SQL Server 2016

Databases (2)

Status	Database name	Migration details
✓	GettoSQL	Migration successful. Duration: 0 hrs 5 mins 25 secs
✓	UR00023	Migration successful. Duration: 0 hrs 1 mins 38 secs

Logins (1)

Migration complete. Duration: 0h 5m 44s

2. Review Migration Results

   a. Select Export report to save the migration results to a .csv or .json file.

   b. Review the saved file for details about data and logins migration and verify successful completion of the process.

Note: If you are interested in participating in a Private preview of a one-time migration for SQL Server to Azure SQL Database using the Azure Database Migration Service, submit a nomination on the Azure Database Migration Service Preview page.

Data sync and Cutover

For minimal-downtime migrations, the source you are migrating continues to change after the one-time migration occurs, and it will drift from the target in terms of data and schema. During this phase, you need to ensure that all changes in the source are captured and applied to the target in near real time. After you verify that all changes in source have been applied to the target, you can cutover from the source to the target environment.

Support for minimal-downtime migrations is not yet available for this scenario, so the Data sync and Cutover phases are not currently applicable.

Post-migration overview

After you have successfully completed the Migration stage, you need to go through a series of post-migration tasks to ensure that everything is functioning as smoothly and efficiently as possible.

Remediate applications

After the data is migrated to the target environment, all the applications that formerly consumed the source need to start consuming the target. Accomplishing this will in some cases require changes to the applications.

Perform tests

After the data is migrated to target, perform tests against the databases to verify that the applications perform well against the after the migration. Use the Database Experimentation Assistant (DEA) to assist with evaluating the target SQL Server on Azure VM environment.

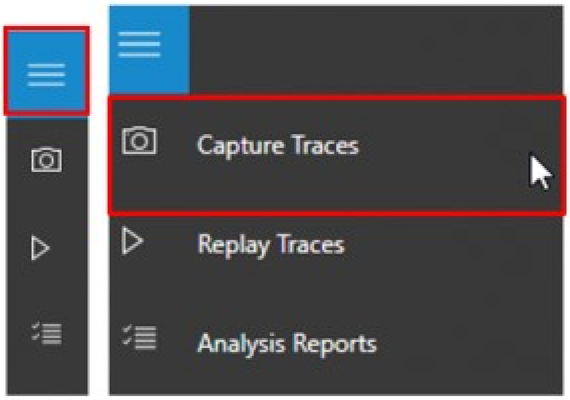
Note: For assistance with developing and running post-migration validation tests, also consider using the Data Quality Solution available from the partner QuerySurge.

Steps

To use DEA for database migration testing, perform the following steps.

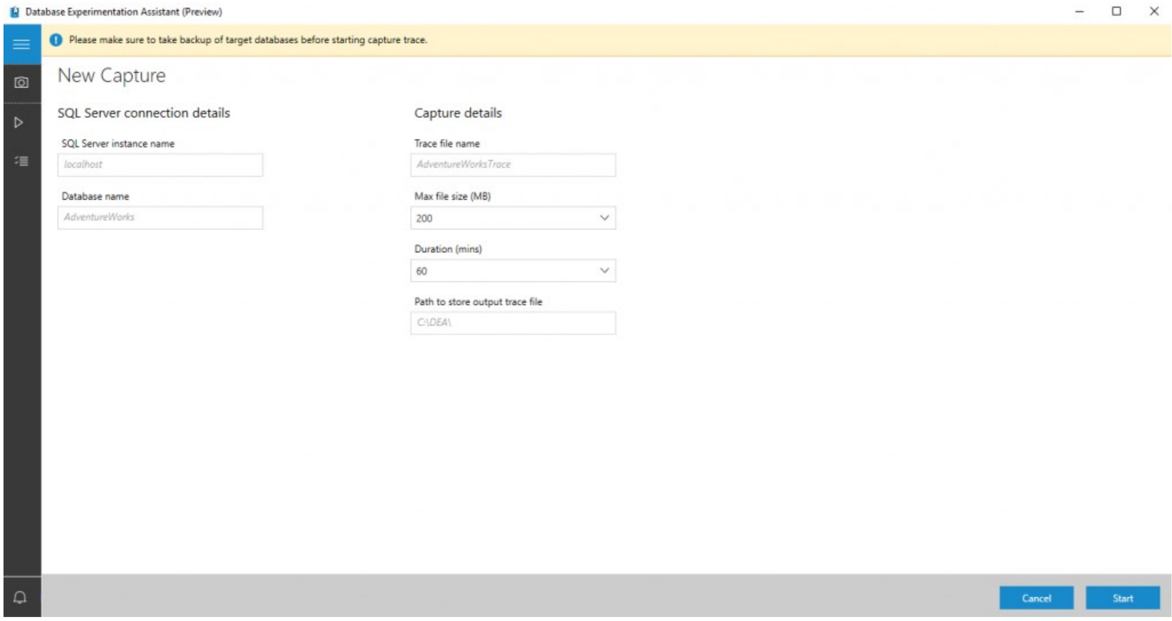
1. Run a trace capture.

   a. On the left navigation tree, select the camera icon the go to All Captures.



   b. To start a new capture, select New Capture.

c. To configure the capture, speficy the trace name, duration, SQL Server instance name, database name, and the share location for storing the trace file on the computer running SQL Server.



Database Experimentation Assistant (Preview)

Please make sure to take backup of target databases before starting capture trace.

New Capture

SQL Server connection details

SQL Server instance name: localhost

Database name: AdventureWorks

Capture details

Trace file name: AdventureWorksTrace

Max file size (MB): 200

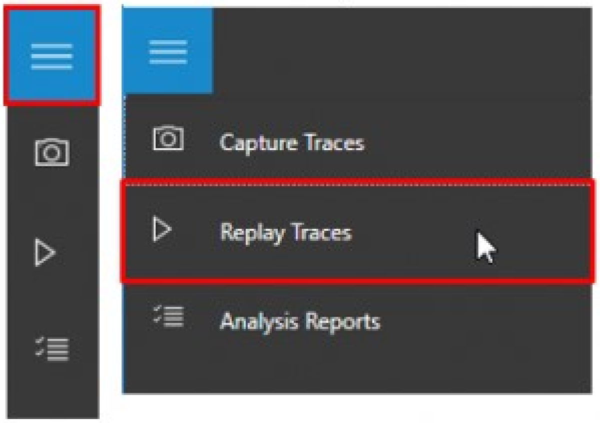
Duration (mins): 60

Path to store output trace file: C:\DEA\

d. Select Start to begin trace capture.

2. Run a trace replay.

a. On the left navigation tree, select the play icon the go to All Replays.



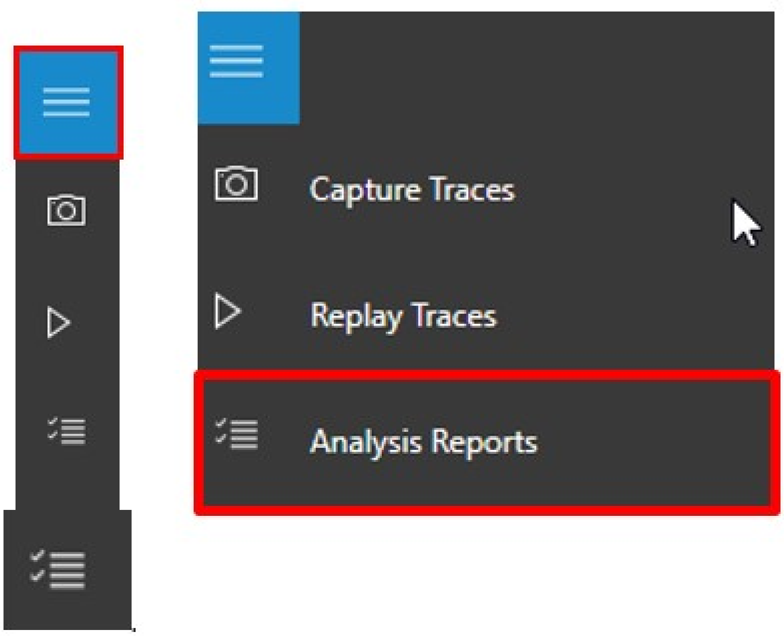
b. To start a new replay, select New Replay.

c. To configure the replay, specify the replay name, controller machine name, path to source trace file on controller, SQL Server instance name, and the pathfor storing the target trace file on the computer running SQL Server.

d. Select Start to begin replay of your capture.

3. Create an analysis report.

a. On the left navigation tree, select the checklist icon to go to Analysis Reports.

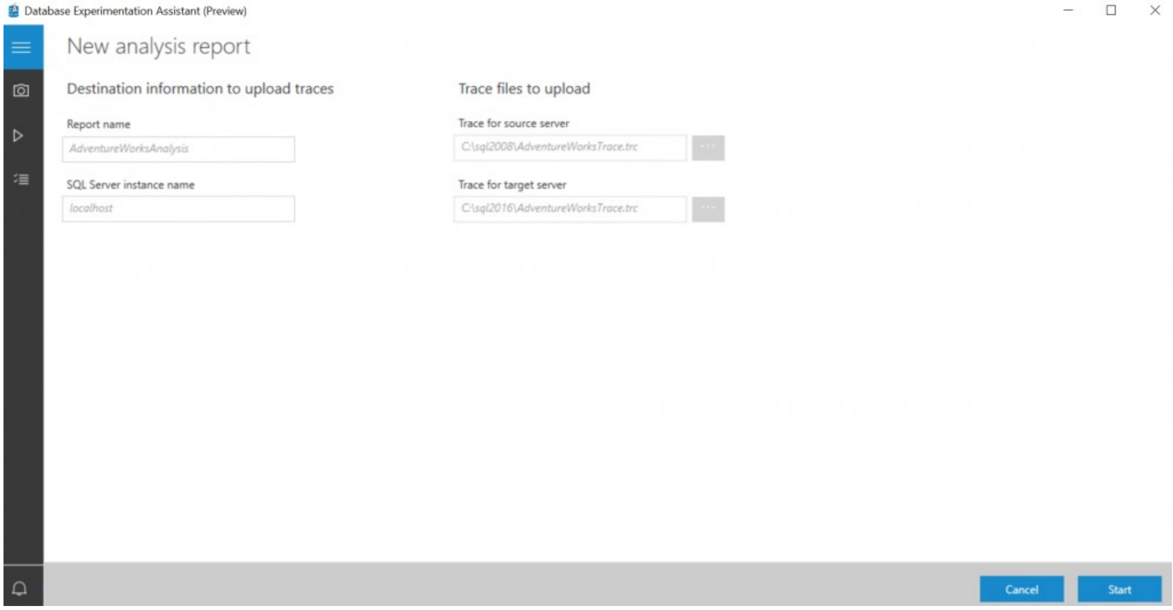


b. Connect to the SQL Server on which you will store your report databases.

You will see the list of all reports in the server.

c. Select New Report.

d. To configure the report, specify the report name, and specify paths to the traces fon the source and target SQL Server instances.


Database Experimentation Assistant (Preview)

New analysis report

Destination information to upload traces

Report name: AdventureWorksAnalysis

SQL Server instance name: localhost

Trace files to upload

Trace for source server: C:\sql2008\AdventureWorksTrace.trc

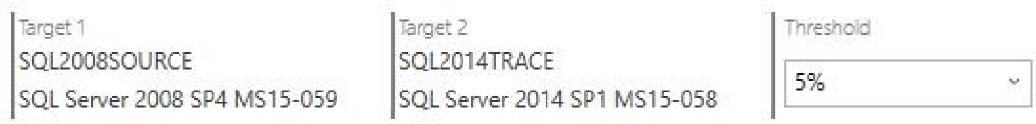
Trace for target server: C:\sql2016\AdventureWorksTrace.trc

4. Review the analysis report.

a. On the first page of the report, the version and build information for the target servers on which the experiment was run displays.

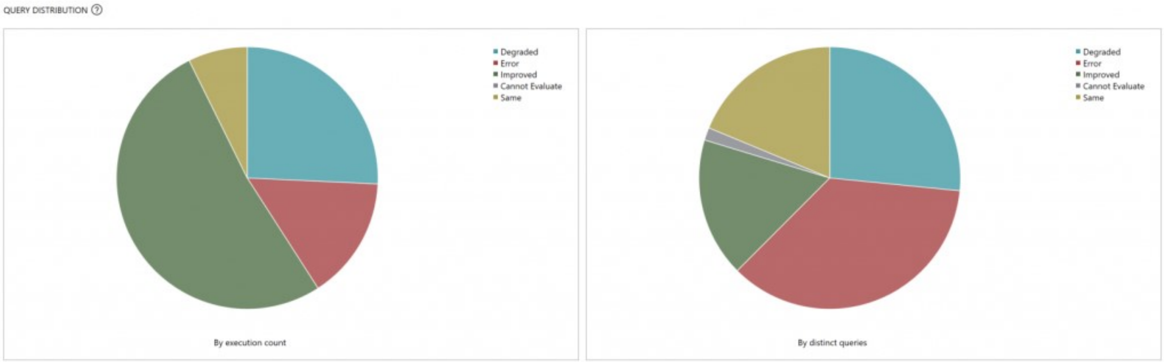
Threshold allows you to adjust the sensitivity or tolerance of your A/B Test analysis.

Note: By default, threshold is set to 5%; any performance improvement that is >= 5% is categorized as 'Improved'. The drop-down selector allows you to evaluate the report using different performance thresholds.

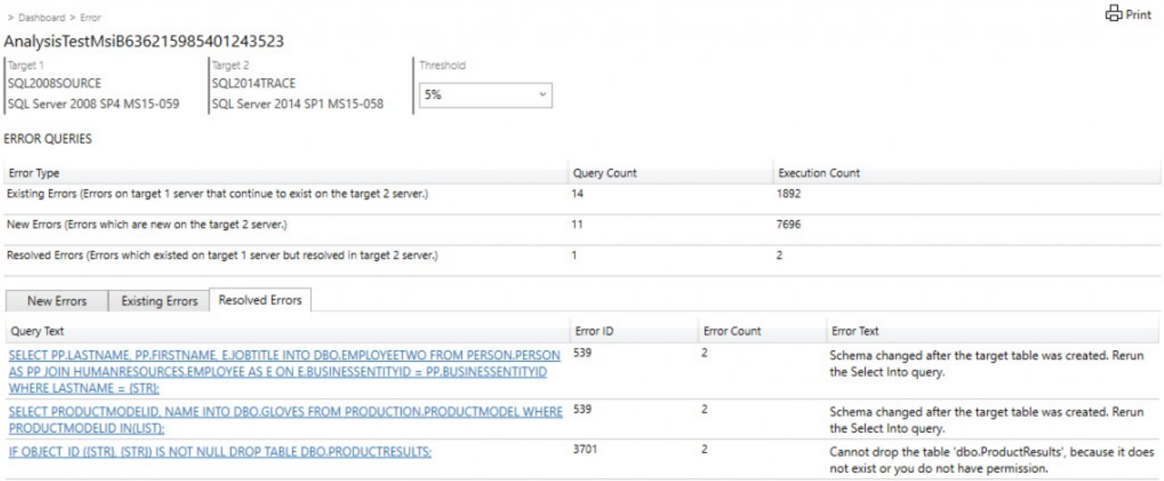


Target 1	Target 2	Threshold
SQL2008SOURCE	SQL2014TRACE	5%
SQL Server 2008 SP4 MS15-059	SQL Server 2014 SP1 MS15-058	

b. Select the individual slices of the pie chart to view drill-down metrics on performance.



On the drill-down page for a performance change category, you will see a list of queries in that category.



AnalysisTestMsiB636215985401243523

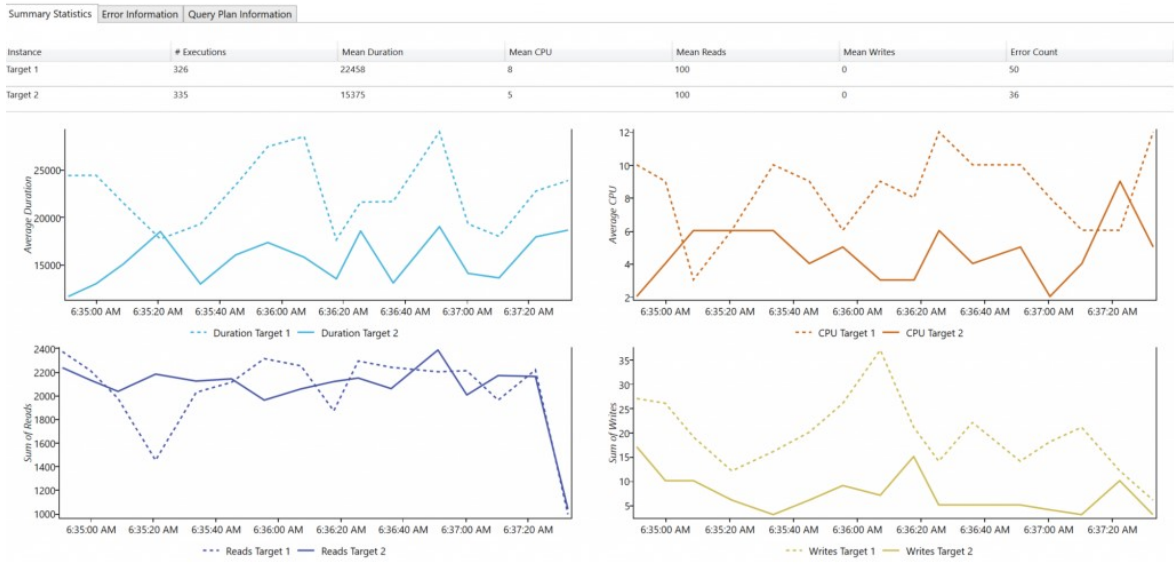
Target 1	Target 2	Threshold
SQL2008SOURCE	SQL2014TRACE	5%
SQL Server 2008 SP4 MS15-059	SQL Server 2014 SP1 MS15-058	

ERROR QUERIES

Error Type	Query Count	Execution Count
Existing Errors (Errors on target 1 server that continue to exist on the target 2 server.)	14	1892
New Errors (Errors which are new on the target 2 server.)	11	7696
Resolved Errors (Errors which existed on target 1 server but resolved in target 2 server.)	1	2

Query Text	Error ID	Error Count	Error Text
SELECT PP.LASTNAME, PP.FIRSTNAME, E.JOBTITLE INTO DBO.EMPLOYEETWO FROM PERSON.PERSON AS PP JOIN HUMANRESOURCES.EMPLOYEE AS E ON E.BUSINESSENTITYID = PP.BUSINESSENTITYID WHERE LASTNAME = {STR};	539	2	Schema changed after the target table was created. Rerun the Select Into query.
SELECT PRODUCTMODELID, NAME INTO DBO.GLOVES FROM PRODUCTION.PRODUCTMODEL WHERE PRODUCTMODELID IN(LIST);	539	2	Schema changed after the target table was created. Rerun the Select Into query.
IF OBJECT_ID ({STR}, {STR}) IS NOT NULL DROP TABLE DBO.PRODUCTRESULTS;	3701	2	Cannot drop the table 'dbo.ProductResults', because it does not exist or you do not have permission.

c. Select an individual query to get performance summary statistics, error information, and query plan information.



Instance	# Executions	Mean Duration	Mean CPU	Mean Reads	Mean Writes	Error Count
Target 1	326	22458	8	100	0	50
Target 2	335	15375	5	100	0	36

Optimize

The post-migration phase is crucial for reconciling any data accuracy issues and verifying completeness, as well as addressing performance issues with the workload.

Note: For additional detail about these issues and specific steps to mitigate them, see the following resource:

- The Post-migration Validation and Optimization Guide.