Microsoft Modernization Practice

Modernization Viability Assessment

Data Collection Instructions

Updated: 08/07/2025

Table of Contents

Summary

This PowerShell script will scan your SQL Server instances (and associated storage) and CloudWatch metrics to collect data points on general configurations and performance statistics and export that into CSV files.

Initial Setup

1. Unzip MVA-Data-Collection.zip to a local drive such as `C:\MVA\`
2. The Batch (BAT) and PowerShell (PS1) scripts will need to be renamed and ".rename" extension removed.
3. The config.txt file will need to be edited to include the resources in scope for this data collection. See section "Editing the config file" for details.

Editing the Config File

The config file is loaded by the PowerShell script to provide the resources in scope for the data collection. Each parameter in the file is of the format "name:value;".

CustomerName

- This should be your company name or initials with no spaces.
- Example - `customername:AcmeInc;`

Region

- List of which region(s) should be scanned when locating EC2 instances, RDS instances, Storage, and CloudWatch data
- Example – `region:us-east-1;us-east-2;`

Server

- List of servers by domain name using the format server\instance,port
- These should be resolvable by DNS from the workstation or server running the script
- If the server is only running a default instance then an entry in the Instances list is sufficient, if there is also named instances then they should be added here. The script will filter for duplicates between Instances and Servers based on IP.
- Example - `server:SQL1.acme.com,1433;SQL2.lab.com;SQL2.lab.com\Prod,51403;`

Instance

- List of InstanceIDs to be collected from
- The IDs will be needed for CloudWatch data collection

- These instances will be mapped to their primary IP and that IP will be used for SQL connectivity.
- Example - `instance:i-0abcdef1234567890;`

Rds

- List of RDS DB Identifiers to be collected from
- Example - `rds:sql-express-2022;sql-prod;`

FSx

- List of FSx file system IDs to be collected from
- Example - `fsx:fs-xxxxxxxxxxxxxxxx;`

Database

- List of database names to be ***Included*** or ***Excluded*** in scan, if no names provided all user databases will be analyzed. Format is "INCLUDE;" or "EXCLUDE;" followed by database names.
- Example - `database:INCLUDE;StackOverflow2013;WideWorldImporters;`
- Example - `database:EXCLUDE;StackOverflow2013;WideWorldImporters;`

Authentication

For AWS authentication the script can leverage an EC2 associated IAM role, a profile configured locally, SSO login using the referenced profile.

For SQL Server authentication the default will be pass-through authentication where the current domain account will be used. The script, .bat or PS session can be launched using the "Run As" option to specify an alternate domain account. A SQL login account can also be specified.

Note the same credentials will be used for all resources identified in the config file, if multiple credentials are needed separate data collection runs will be needed for each credential.

Collecting the Data

The "Validate Resources" process should be run first to confirm DNS, permissions, and any access issues. When this step is completed without error the data collection step can be started.

Example Script: ***MVA-Data-Collection-ValidateResourcesOnly.bat*** *– this may need to be edited to meet your environments needs.*

One Time Collection

There are three primary parameters to include in running the data collection script that are needed for a complete evaluation.

- CollectTsqlData - This runs a series of T-SQL scripts against all SQL instances provided and exports them to CSV files.
- ExportDacPacs – This will create a Schema Only DacPac export for all databases in scope. This entails an automated download and install of the latest version of the SqlPackage software from Microsoft that will be placed in a \Dacpac\ directory alongside the PowerShell script location.
- CollectCloudWatchData – This will collect 30 days of cloud watch data on compute and storage utilization and export to CSV files.

There are also several ancillary parameters that may need to be set dependent on your environment.

- SqlServerConnectionTimeout – This connection timeout defaults to 5 seconds but may need to be increased.
- SqlServerQueryTimeout – The query timeout defaults to 300 seconds but may need to be increased.
- SqlUser & SqlPassword – The default behavior is Pass Through Authentication, but SQL Authentication can be provided here. The password can also be supplied at run time.
- AWSProfile – This can be used to set the environment profile for AWS Authentication.
- UseSSOLogin – This will call the SSO login Process, this feature is currently in preview and additional support may be needed to use this.
- ExportPath – The Export directory structure defaults to the same directory as the PowerShell script; an alternate Local or remote path can be specified here.

Example Script: ***MVA-Data-Collection-CollectEnvironmentData.bat*** *– this may need to be edited to meet your environments needs.*

Recurring Collection

Optionally, to build an application dependency model Evolve recommends using the script to create a SQL Agent Job that will scrape the active connections, create a temporary database MVA-Data-Collection, and write them to a ConnectionInfo table every five minutes using the CollectConnectionsOnly parameter.

The first execution will create the SQL Agent Job and database; a second execution will be needed to collect all data from the ConnectionInfo table at the end of the collection period.

The appropriate period for a sufficient sample of connections depends on the characteristics of your workload but Evolve recommends 24hrs as a minimum.

When the connection collection process is no longer needed the job **CollectConnections** can be manually disabled/deleted or the CleanUpEnvironment parameter can be used to remove the Job and database from all servers specified in the config file.

Note: Ensure the connections data has been collected and exported to a CSV (using the CollectConnectionsOnly parameter) prior to clean up as all connections data is permanently deleted.

Example Script: ***MVA-Data-Collection-CollectConnectionsOnly.bat*** *– this may need to be edited to meet your environments needs.*

Multi Region Collection

The PowerShell script will process multiple regions to locate the AWS resources added to the config file so within a single account one collection process can handle multiple regions.

Multi Account Collection

The PowerShell script is not configured to connect to multiple accounts, if AWS Organizations are configured and the account executing the collection script has access across all accounts then a one collection process can be used otherwise its recommended to run a collection effort per account.

When running multiple data collection processes, it's recommended to unzip the scripts into multiple separate directories and keep a separate config file per collection.

Reporting and ETL

Within your SQL Server environment you may have Integration Services (SSIS) and Reporting Services (SSRS) in use. We do scrape the SSISDB and ReportServer databases for basic information on those objects, if those SSIS and SSRS databases reside on a central or separate server from the user databases those servers can also be included in the config (database filtering recommended), collected as a separate data collection run, or info on the SSIS and SSRS objects can be manually submitted separately.

How to Share Results

When the PowerShell script successfully completes there will be a ***-MVA-Export-ALL-mmddyyhhmmss.zip file created within the \Export\ directory. This Export directory will either reside in the same directory as the PS script (default behavior) or at the -ExportPath location if the override parameter is used.

The CSV files contained within this zip will have metadata such as servernames etc. but will not contain any user data. We highly encourage you to review the contents internally before sharing with Evolve.

Methods to securely send the zip(s) to Evolve will be dependent on your own security policies, common approaches we see are:

- Password protected Zip to email – mva@evolvecloudservices.com
    - *Note: the zip file created in this script is not protected, a third-party utility will be required for this*
- Shared via Amazon S3 bucket using a Pre-Signed URL
- File sharing solutions such as DropBox, Box or similar

Environment Clean Up

When all data has been collected and sent to Evolve the final step is to run the script with the CleanUpEnvironment parameter. This will remove all SQL Agent Jobs, databases and DacPac agent install. The final ***-MVA-Export-ALL-mmddyyhhmmss.zip(s) created will not be deleted and will need to be removed manually.

Example Script: ***MVA-Data-Collection-CleanUpEnvironment.bat*** *– this may need to be edited to meet your environments needs.*

Zip File Content

The zip file MVA-Data-Collection.zip should be downloaded onto a server/workstation that has PowerShell 2.0 or higher (PowerShell is standard on Windows 2008 R2 and above) and the SQL Server client tools.

- MVA-Data-Collection.ps1
  - PowerShell Script
- MVA-Data-Collection-CleanUpEnvironment.bat
  - Preconfigured script to remove all objects created by this script
- MVA-Data-Collection-CollectConnectionsOnly.bat
  - Preconfigured script to run connections collection
- MVA-Data-Collection-ValidateResourcesOnly.bat
  - Preconfigured script to validate resources added to config.txt
- MVA-Data-Collection-CollectEnvironmentData.bat
  - Preconfigured script to collect SQL data, Schema only DacPacs and CloudWatch data
- config.txt
  - Configuration file
- Evolve-MVA-Data-Collection-Instructions.docx
  - This document