aws migration evaluator

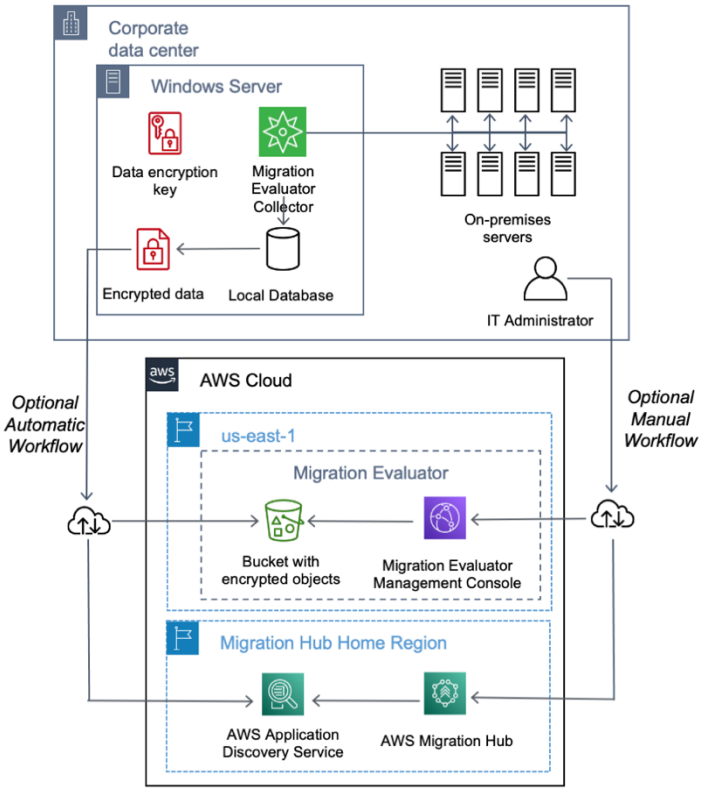
Overview for Agentless Collector

Version 2022-06-07

Formerly TSO Logic

Overview

The following document outlines the flow of data during an engagement with Migration Evaluator (Formerly TSO Logic) in which the customer chooses to acquire provisioning, utilization patterns, and network connections via the on-premises agentless collector. The network traffic is broken into two logical phases: data synchronization and data collection. All data collected is encrypted at rest (via a customer specific certificate) and transport (via HTTPS).



Note: the selected AWS Migration Hub Home Region may be different than the region used by Migration Evaluator. AWS Application Discovery Service and AWS Migration Hub support: US East (N. Virginia), US West (Oregon), Asia Pacific (Sydney), Asia Pacific (Tokyo), Europe (Frankfurt), Europe (Ireland), and Europe (London). Migration Evaluator supports US East (N. Virginia).

Data Synchronization

The Migration Evaluator Collector supports collection for both a migration assessment (via Migration Evaluator) and network visualization and tracking (via AWS Migration Hub and AWS Discovery Service). The following section outlines the different paths available.

Automatic Workflow to Migration Evaluator

If enabled, the data collected each day is exported from the local encrypted database instance, re-encrypted using a customer specific certificate, and sent via HTTPS to a private, encrypted Amazon S3 bucket folder provided by Migration Evaluator. The Amazon S3 bucket leverages AES-256 Server-Side Encryption with AWS Key Management Service (SSE-KMS).

Bucket folders are not shared between clients. Amazon S3 resources are hosted in US East (Northern Virginia).

Manual Workflow to Migration Evaluator

When manually providing files for a business case, the customer authenticates with the Migration Evaluator Management Console using their personal username, password and optional multi-factor authentication token. Files uploaded are stored in an Amazon S3 bucket managed by Migration Evaluator and leverages AES-256 Server-Side Encryption with AWS Key Management Service (SSE-KMS). Amazon S3 resources are hosted in US East (Northern Virginia).

Examples of files to be uploaded include: exports from a content management database (CMDB), and/or existing performance monitoring system. An export from the Migration Evaluator collector into an Excel workbook is also supported. This allows for inspection, as well as obfuscation, if need be, before being transported.

Automatic Workflow to AWS Application Discovery Service

If enabled, data used for creating server to server network visualization in AWS Migration Hub is sent to the customer's AWS Application Discovery Service (ADS) account. Authorization is provided by an IAM user created by the customer. All data sent is stored in the customer's AWS Migration Hub home region.

Manual Workflow to AWS Application Discovery Service

To make server provisioning and utilization discovered by the Migration Evaluator Collector available within AWS Migration Hub, the customer must manually import into AWS Migration Hub a pre-populated template exported from the Migration Evaluator Collector. All data imported is stored in the customer's AWS Migration Hub home region.

Data Retention

Use of Migration Evaluator is subject to the terms of the AWS Customer Agreement and AWS Service Terms. Customer data processed by AWS during your use of Migration Evaluator is subject to the Data Protection Terms in Section 1 of the AWS Service Terms and will be stored in AWS US East (N. Virginia). By using Migration Evaluator, you are authorizing an AWS solutions architect to access your data in order to provide the service to you. You can share your data in Migration Evaluator with your AWS account team by using the Enhanced Migration Assistance feature.

You can configure data collected from the Migration Evaluator Collector to be sent to AWS Application Discovery Service (ADS). Please note that the AWS region for AWS Application Discovery Service may be different than the AWS region for Migration Evaluator or your source server, which could result in data being sent cross-region.

Data Collection

All information collected by Migration Evaluator within a client's data center is persisted to a local encrypted database instance. The encryption key for the database is only available to the Windows administrator.

Configuration of the collector can be done only via a locally hosted website from within the customer's data center. All communication is encrypted.

It is the customer's responsibility to take the desired protections to this data including:

- Windows User Accounts/ACLs
- Replacing the default self-signed certificate used for HTTPS

Monitoring VMware Infrastructure

For monitoring VMware, the collector communicates with each vSphere virtual appliance (not the actual virtual machine). Monitoring adds no additional load on the host system or virtual machines. Communication is done via the vSphere SOAP API over HTTPS (TCP 443). Usernames and passwords are encrypted at rest leveraging DPAPI and are not included in the data synchronised to AWS.

The following vSphere SOAP API calls are made by the collector:

- RetrieveEntityPermissions
- CreatePropertyCollector
- CreateContainerView
- RetrieveServiceContent
- CurrentTime
- CreateFilter
- DestroyPropertyFilter
- QueryPerf
- RetrievePropertiesEx
- ContinueRetrievePropertiesEx
- WaitForUpdatesEx

The following are the polling intervals:

- compute and storage provisioning every one hour
- compute utilization every 15 minutes
- power state every 15 minutes
- storage utilization every six hours

VMware infrastructure provisioning is persisted by the agentless collector along with their relationships and time-series usage. (i.e., a VirtualMachine runs on a HostSystem or a HostSystem runs within this ClusterComputeResource)

ClusterComputeResource

Attribute	Example Value
Key	domain-c518
Name	PROD-SHARED-SVC

ComputeResource

Attribute	Example Value
Key	domain-s173
Name	192.168.0.226

Datacenter

Attribute	Example Value
Key	datacenter-2
Name	PROD

Datastore

Attribute	Example Value
Accessible	TRUE
Key	datastore-126041
MultipleHostAccess	TRUE
Name	PRD-APX-G-V-CDP-002-PRODLNX30660
Type	VMFS
Url	ds:///vmfs/volumes/59c283f4-8e0be3ba-31ee-0025b50aa00f/

Folder

Attribute	Example Value
Key	group-v73828
Name	Colleague Portal

HostSystem

Attribute	Example Value

AssetTag	
CpuMhz	2799
CpuModel	Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
Key	host-356
MemorySizeInBytes	17037066240
Model	UCSB-B200-M3
Name	3005.bc.abcd.net
NormalizedUuId	b52500000b000000000000000000005
NumCpuCores	20
NumCpuPkgs	2
NumCpuThreads	40
ServiceTag	FCH2020JFBP
UuId	b5250000-0b00-0000-0000-000000000005
Vendor	Cisco Systems Inc

VirtualApp

Attribute	Example Value
Key	resgroup-v819
Name	VIPR SRM

VirtualDisk

Attribute	Example Value
BackingContentId	
BackingFilename	
BackingLunUuid	
BackingThinProvisioned	FALSE
BackingType	
BackingUuid	
CapacityInKb	0
DeviceId	4003
Label	
VirtualMachineKey	vm-231571

VirtualMachine

Attribute	Example Value
CpuAllocationLimit	-1
CpuAllocationReservation	0
FullName	Red Hat Enterprise Linux 6 (64-bit)

GuestId	rhel6_64Guest
GuestState	notRunning
HostName	prodlnx3028.bc.abcd.net
InstanceUuId	502efe96-0150-b395-682f-adb11f482945
IpAddress	
IpAddresses	
Key	vm-886
MemoryMb	16384
Name	prodlnx3028
NumCpu	8
Template	FALSE
ToolsRunningStatus	guestToolsNotRunning
UuId	422e44c8-5caa-4648-387c-ecc6e6ddb546

Monitoring Hyper-V Infrastructure

For monitoring Hyper-V, the collector communicates with each Hyper-V host (not the actual virtual machine). Monitoring adds no additional load on the virtual machines. Communication is done via WMI over TCP port 135 + ephemeral TCP port range (49152 - 65535). *Note: WMI can be problematic through firewalls due to maintaining contracts in the ephemeral port range.* Usernames and passwords are encrypted at rest leveraging DPAPI and are not included in the data synchronised to AWS.

The following are the polling intervals:

- compute and storage provisioning every one hour
- compute and storage utilization every nine minutes
- power state every nine minutes

Hyper-V infrastructure provisioning is persisted by the agentless collector along with their relationships and time-series usage. (ie, a VirtualMachine runs on a HostSystem)

HostSystem

Attribute	Example Value
AllocatedDisk	254008094720
AverageProcessorClockSpeed	3600
CredentialProfile	hyperV
FQDN	HYPERV4.WORKGROUP
HostName	192.168.0.50
IdentifyingNumber	3F4C9M2
Model	OptiPlex 7050

NumberOfCores	4
NumberOfCpus	1
NumberOfLogicalProcessors	8
OperatingSystemVersion	6.3.9600
ProcessorId	BFEBFBFF000906E9
ProcessorName	Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
TotalPhysicalMemory	17037066240
Uuid	4C4C4544-0046-3410-8043-B3C04F394D32
Vendor	Dell Inc.
WmiCredentialsUsed	1e40b1ba-0a63-4f12-9669-31c6ca166d32

VirtualMachine

Attribute	Example Value
AllocatedDisk	53687091200
AllocatedMemory	512
FQDN	W2012-2
IpAddresses	192.168.0.66
Key	548922cb-35b1-4db0-863b-47f57625b8ac
LastReplicationTime	16010101000000.000000-000
LastReplicationType	0
Name	W2012-2
OperationalStatus	
OSName	Windows Server 2012 Standard
ProcessorCores	1
ProcessorLimit	100000
ProcessorReservation	0
ProcessorWeight	100
ReplicationHealth	0
ReplicationMode	0
ReplicationState	0

Monitoring Bare Metal Infrastructure

For monitoring bare metal servers, the collector communicates with each server directly. For Windows servers, collection can be configured to leverage WMI, SNMP v2c or SNMP v3. For Linux servers, SNMP v2c or v3 is available. Communication via SNMP is done over UDP port 161. Communication via WMI is over TCP port 135 + ephemeral TCP port range (49152 - 65535). *Note: WMI can be problematic through firewalls due to maintaining contracts in the ephemeral port range.* Usernames and passwords are encrypted at rest leveraging DPAPI and are not included in the data synchronised to AWS.

The following WMI namespaces are queried:

- \default\StdRegProv (HKEY_USERS)
- \cimv2\Win32_PerfFormattedData_PerfOS_Processor
- \cimv2\Win32_PerfFormattedData_PerfOS_Memory
- \cimv2\Win32_ComputerSystem
- \cmiv2\Win32_LogicalDisk
- \cimv2\Win32_PerfFormattedData_Tcpip_TCPv4
- \cimv2\Win32_OperatingSystem
- \cimv2\Win32_Processor

The following SNMP OIDs are queried:

Description	Linux	Windows
CPU Utilization	1.3.6.1.2.1.25.3.3.1.2	1.3.6.1.2.1.25.3.3.1.2
Memory Utilization	1.3.6.1.4.1.2021.4	1.3.6.1.2.1.25
CPU Provisioning	1.3.6.1.2.1.25.3.2	N/A
Memory Provisioning	1.3.6.1.2.1.25.2.3.*	N/A
Storage Provisioning	1.3.6.1.2.1.25.2.3.*	N/A

The following are the polling intervals:

- compute and storage utilization every nine minutes
- compute provisioning every nine minutes
- power state every nine minutes.

Bare metal infrastructure provisioning is persisted by the agentless collector along with time-series usage.

Device

Attribute	Example Value
FQDN	
IpAddress	192.168.0.63
Local Storage Size	
Location	
MacAddress	
MachineType	
Name	Server-1
OperatingSystem	Windows Server 2012 Standard
Physical Memory	
Processor String	Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
Uuid	

Monitoring Network Connections

For monitoring TCP network connections, the collector communicates with each server directly. For Windows servers, collection can be configured to leverage WMI, SNMP v2c or SNMP v3. For Linux servers, SNMP v2c or v3 is available. Communication via SNMP is done over UDP port 161. Communication via WMI is over TCP port 135 + ephemeral TCP port range (49152 - 65535). *Note: WMI can be problematic through firewalls due to maintaining contracts in the ephemeral port range.* Usernames and passwords are encrypted at rest leveraging DPAPI and are not included in the data synchronised to AWS.

The following WMI namespace is queried:

- \root\standardcimv2\MSFT_NetTCPConnection (Windows Server 2012 or greater)

The following SNMP OID is queried:

Description	Linux	Windows
TCP Connections	1.3.6.1.2.1.6.13.*	1.3.6.1.2.1.6.13.*

The polling interval for TCP connections is 60 seconds. Each collection cycle is limited up to 1000 servers. If more that 1000 servers are available, a random set of servers will be selected each collection cycle.

Network connection data is only held in memory and is not persisted disk on the on-premises collector.

Discovering SQL Server Instances

For discovering which servers are running Microsoft SQL Server, the collector communicates with each server directly. For Windows servers, collection can be configured to leverage WMI or T-SQL. For Linux servers, only T-SQL is supported. Communication via T-SQL is over TCP port 1433. Communication via WMI is over TCP port 135 + ephemeral TCP port range (49152 - 65535). *Note: WMI can be problematic through firewalls due to maintaining contracts in the ephemeral port range.* Usernames and passwords are encrypted at rest leveraging DPAPI and are not included in the data synchronised to AWS.

The following WMI namespace is queried:

- \root\Microsoft\SqlServer

The following T-SQL queries are run:

- SELECT @@SERVICENAME, @@VERSION, SERVERPROPERTY('productversion'), SERVERPROPERTY('edition')
- SELECT * FROM sys.databases
- SELECT * FROM sys.master_files

The polling interval for scanning all servers is 24 hours. Scanning can be manually initiated from the collector UI.

Via WMI, the following metadata is persisted about each installed SQL Server component.

Component

Attribute	Example Value
Instance	MSSQLSERVER1
ServiceName	MSSQL$MSSQLSERVER1
Version	15.0.1102.911
FileVersion	2019.150.2000.5
Edition	Developer
State	4
ServicePack	2
Type	1
ClusterName	Cluster-1
PortNumber	1433

Via WMI, the following metadata is persisted about Reporting Server Instances.

Component

Attribute	Example Value
ServiceName	MSSQLSERVER

Version	15.0.1102.911
Edition	Developer
State	4
DatabaseName	ReportServer
DatabaseServerName	SQLFSX2\MSSQLSERVER2

Via T-SQL, the following metadata is persisted about the Microsoft SQL Server database instance bound to TCP port 1433.

Database Instance

Attribute	Example Value
Instance Name	MSSQLSERVER1
SQL Server version	15.0.1102.911
SQL Server edition	Standard Edition (64-bit)

Via T-SQL, the following metadata is persisted for each database running on the database instance.

Database Information

Attribute	Example Value
Database Name	master
Description	LOG
Size	The current size in MB
Max Size	The maximum size in MB