COHESITY

Version 1.1

July 2024

Protect SQL Databases with Cohesity BaaS Best Practices Guide

A Strategy for SaaS Connectors

ABSTRACT

Cohesity BaaS for SQL provides visibility across all your backups and allows you to select which backups to recover. DMaaS applies storage efficiency to minimize storage costs, it is application aware, has built in data security, and allows you to restore across regions.

Table of Contents

Figures

No table of figures entries found.

Tables

No table of figures entries found.

Why SQL BaaS

Cloud providers like AWS provide infrastructure; they do not focus on backup and recovery. Native AWS protection capabilities are the bare minimum of what is acceptable in the Data Protection industry.

If you are just doing EC2 snapshots, those are just volume level backups. EC2 snapshots don't know anything about SQL Server or other applications running inside.

AWS does not apply storage efficiency, neither does it provide visibility across all your backups, nor does AWS allow restores across regions.

Sometimes our assumptions about AWS backups are not accurate. For example:

- AWS applies storage efficiency.
- AWS protects my backups with tight security controls
- AWS protects my data from accidental deletion.
- AWS protects my encryption keys from accidental deletion

These assumptions are incorrect and for all these reasons a proper data protection solution is needed in the cloud even more so than On-Prem.

Today, Cohesity DMaaS for SQL provides visibility across all your backups and allows you to select which backups to recover. Cohesity also applies storage efficiency to minimize storage costs, has data security built-in, DMaaS is application aware, and allows you to restore across regions.

Links in a Chain

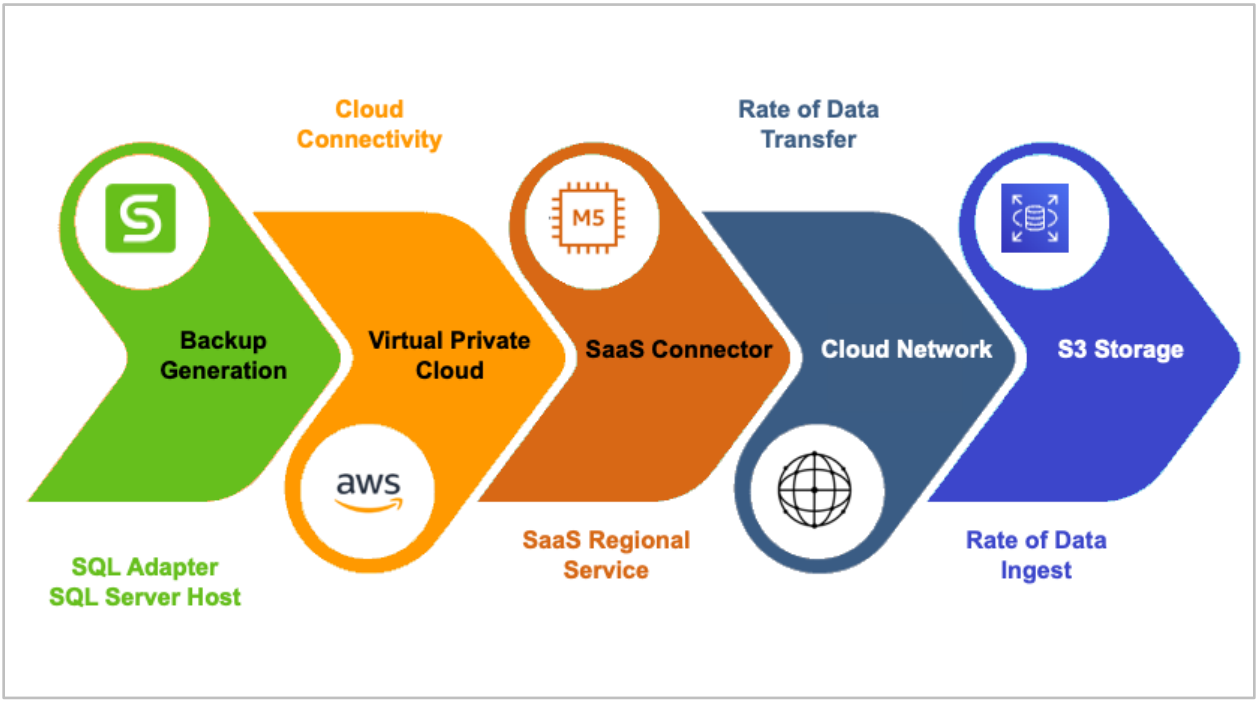
A BaaS backup of a SQL Server database is a series of technical links in a chain. This means that the most efficient backup requires all the links in the chain to work optimally.

The major links are:

- Backup Generation
- Cloud Connectivity
- SaaS Regional Service
- Rate of Transfer
- Rate of Data Ingest

An efficient SQL DMaaS backup and recovery means no link in the chain creates a data flow bottleneck.

Diagram 1: Links between SQL server and DMaaS S3 Storage

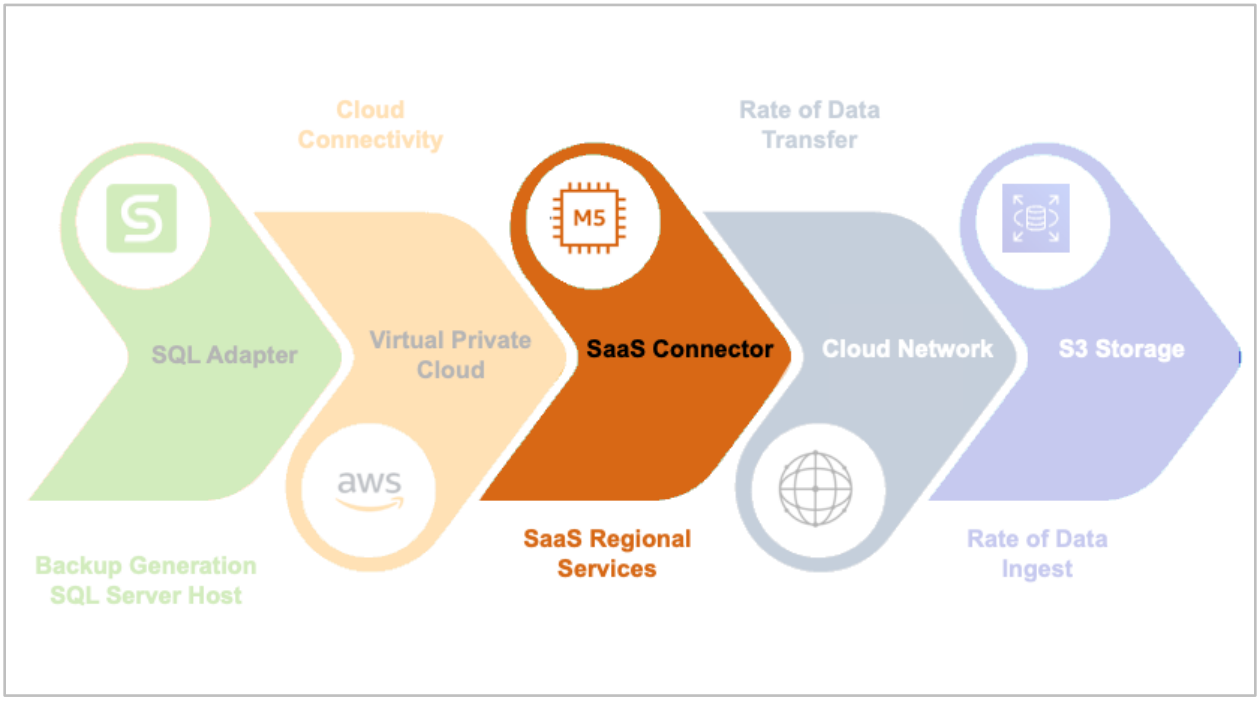


We will discuss the dynamics of the SaaS Connector in the next sections - let's go.

SaaS Connector

All the links in the DMaaS chain are important. The SaaS Connecter is the most important link because it is where data meets the cloud. It functions as an air traffic controller to orchestrate the tasks needed to efficiently store your data.

Diagram 2: DMaaS SaaS Connector



Design Decisions and Deployment Methodologies

Deciding which SQL databases can be migrated to DMaaS is an important step.

Table 1 shows the kinds of databases that DMaaS can protect.

Identify the Type of Database

Before proceeding to DMaaS for SQL data protection, Cohesity recommends you correctly identify the type of SQL database you want to protect.

Below is a table of SQL database types that are supported by DMaaS.

Table 1: MS SQL Server Database Types

DB TYPES	DEFINITION	PROTECTION METHOD
		SQL DMAAS
Stand Alone	A SQL Server User database	**Yes**
AG	A database that belongs to an availability group (AG). For each availability database, the availability group maintains a single read-write copy (the primary replica) and one to eight read-only copies (secondary replicas). For details, see Always On availability groups: a high-availability and disaster-recovery solution in the Microsoft documentation.	SQL Server AAG Backup is currently not supported with Cohesity DataProtect Service. The Selected databases will be treated as if the databases are deployed on a stand-alone SQL Server Instance for backup and restore operations.
FCI	A database that belongs to a Failover Cluster Instance (FCI), which is a single instance of SQL Server that is installed across Windows Server Failover Clustering (WSFC) nodes.	**No**
CDC	A change data capture (CDC) enabled SQL database records activity and is applied to an internal table. This makes the details of the changes available in an easily consumed relational format. For details, see About Change Data Capture (SQL Server) in the Microsoft documentation.	**Yes**

DB TYPES	DEFINITION	PROTECTION METHOD
		SQL DMAAS
TDE	A Transparent Data Encryption (TDE) enabled SQL database has its data files encrypted, known as encrypting data at rest. TDE performs real-time I/O encryption and decryption of the data and log files. For details see Transparent Data Encryption (TDE) in the Microsoft documentation. **NOTE**: Cohesity does not manage the keys associated with the TDE database.	**Yes**
File Stream	A Filestream enabled SQL Database is a database which contains a table that stores its data outside the database on the host file system. SQL Filestream leverages the OS file system's ability to handle large objects BLOBS like image ¨PDFs¨ Binary ¨Bmp¨ Docs¨ and other unstructured data. For details, see FILESTREAM (SQL Server) in the Microsoft documentation.	**No**
SQL System	SQL Server maintains a set of four system-level databases (*master, model, msdb, tempdb*), which are essential for the operation of a server instance. System databases must be backed up after every significant update. The system databases that you must always back up include msdb, master, and model. For details, see Backup & restore: system databases (SQL Server) in the Microsoft documentation.	**No**
SQL Log Shipping	SQL Server Log shipping allows you to automatically send transaction log backups from a primary database on a primary server instance to one or more secondary databases on separate secondary server instances. The transaction log backups are applied to each of the secondary databases individually.	**No**
SQL Mirroring	Database mirroring maintains two copies of a single database that must reside on different server instances of SQL Server Database Engine. Typically, these server instances reside on computers in different locations.	**No**

Now that you can Identify the type of SQL database you want to protect, you can properly register it as a Cohesity source.

Important Prerequisite: The Cohesity SQL Adapter must be installed on your SQL Server host and registered properly as a Cohesity Source.

SaaS Connector Recommendations

It is important to know the performance and sizing of the SaaS Connector so that you can deploy it properly. Below are deployment recommendations so that you can efficiently use the SaaS Connector.

Recommendations:

- A separate connection group for SQL
- Install a minimum of 2 Saas Connectors for each SQL group
- SaaS Connector Load
- SQL Backups via BaaS are all VDI-based databases
- 4 backups in parallel
- Default 3 SQL threads per database backup

Each of these is discussed below.

- A separate connection group for SQL is recommended.

  NOTE: This gives you flexibility in managing the backup load.

  NOTE: This gives you flexibility in managing the backup load.

- Install a minimum of 2 Saas Connectors for each SQL group.

  NOTE: Having more than one SaaS Connector adds redundancy in the event of upgrade glitches or connectivity issues. This will keep your backup jobs from failing in the event one of the SaaS Connectors isn't working properly.

- SaaS Connector load should be configured with less than or equal to 160 SQL databases or 16 TB of data.

  Important: Full backups are a significantly bigger load than the subsequent incremental backups. Cohesity recommends that you stagger the full backups.

- SQL Backups via BaaS are all VDI-based databases.

  IMPORTANT: A VDI-based backup allows for FULL, DIFF, and LOG backups. This type of backup does not cause the SQL Instance to go through a freeze/thaw. As per Microsoft, a restore from a VDI based backup *requires* having a periodic FULL backup.

  IMPORTANT: You must specify a Periodic FULL backup in the Cohesity Policy.

  IMPORTANT: All Microsoft VDI backups, their differentials, and their logs are dependent on a full backup to perform a database restore. Microsoft requires that in order to restore a SQL Database,

you must start with a full backup, then its transaction logs can be applied. This means your backup retention policy must keep a full backup along with its log backups in order to successfully restore a database.

Cohesity recommends retaining two sets of full backups with their differential and log backups.

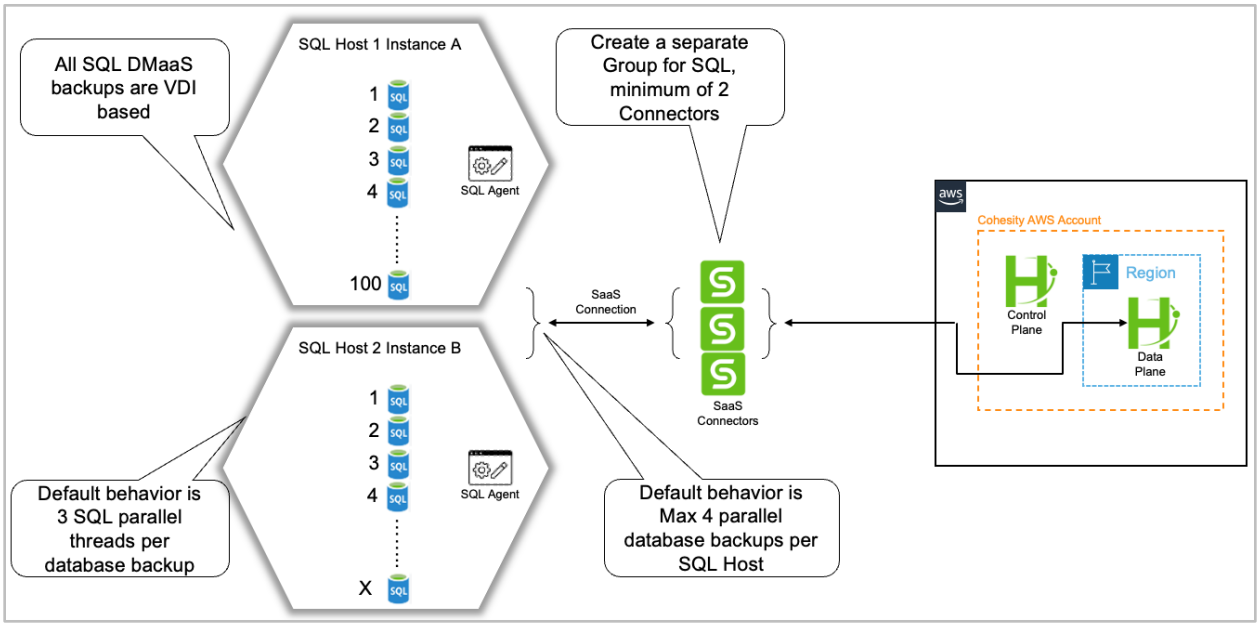
TIP: A good backup plan always includes a combination of full, differential, and log backups.

- The SaaS Connector will maintain a maximum of 4 parallel database backups per SQL host. As each backup completes another will start. Backups will continue until all databases in the backup job are protected.

  NOTE: Cohesity recommends that you begin by a Policy that protects sets of small databases. You can avoid queuing in the SaaS Connector by protecting larger databases in their own protection job.

- SQL Server backup threading is used for each database backup. The default is three SQL parallel threads for each database backup.

Diagram 3: SQL SaaS Connector Deployment Recommendations



Scaling and Sizing for Performance

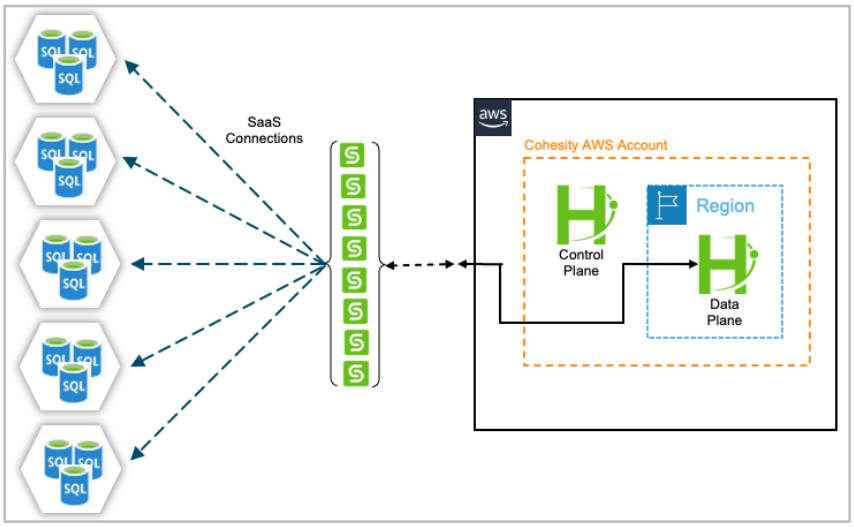
Scaling can be achieved by adding SaaS Connectors, adding SaaS Connections,

and Policy management.

Load Balancing with Multiple SaaS Connectors.

Increasing the number of SaaS Connectors for a connection increases the Connector's capacity to handle more database backups. Additionally, database backup threads are load balanced across SaaS Connectors.

Diagram 4: Adding SaaS Connectors to Balance Load

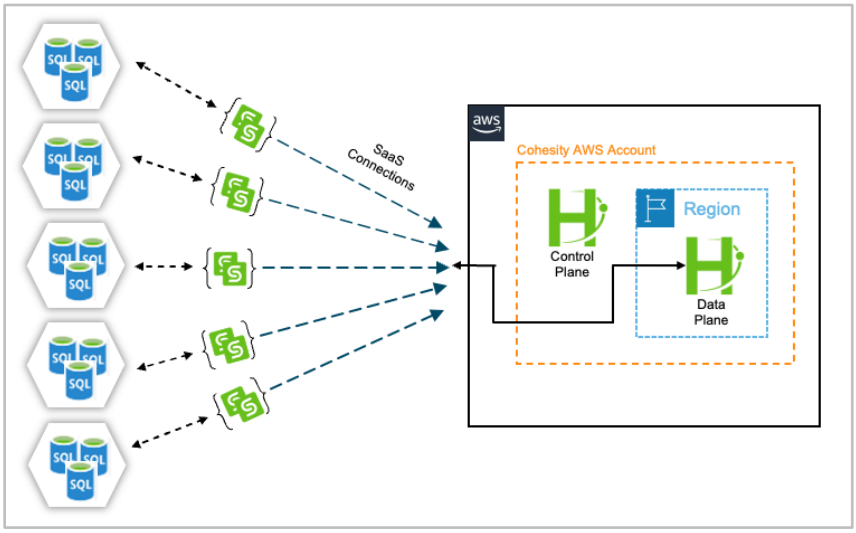


Load Balancing with Multiple Connections

Sources are associated with a specific SaaS connection. Multiple SaaS connectors can be deployed per SaaS connection.

NOTE: Deploying many SaaS Connectors in your environment to protect your SQL databases gives you two important advantages; 1) a high level of granular management of your backup jobs, and 2) a high level of granular management of your connections.

Diagram 5: Adding SaaS Connections to Balance Load



Load Balancing Using Cohesity Policy

When you protect a SQL database in Cohesity DMaaS, you assign it to a Cohesity Policy.

The Policy specifies all the options of the backup including which SaaS Connection the backup will use. You may miss your backup SLAs by choosing too many databases under one Policy. Changing the number of databases under a Policy you change the amount of data flowing over the SaaS Connection.

TIP: Production environments rarely have balanced data loads across SQL Server databases and across applications. Cohesity recommends employing a combination of load balancing techniques to achieve your backup SLAs.

In the case of large databases, you can tune backup performance by:

- Increasing the number of streams for VDI or the number of subtasks for file-based backup
- Putting the large DB in a separate job
- Avoiding overlapping jobs when the large DBs backup is running

Backup Considerations

Three factors are critical in determining which SQL database can be protected with DMaaS.

In priority order, those factors are:

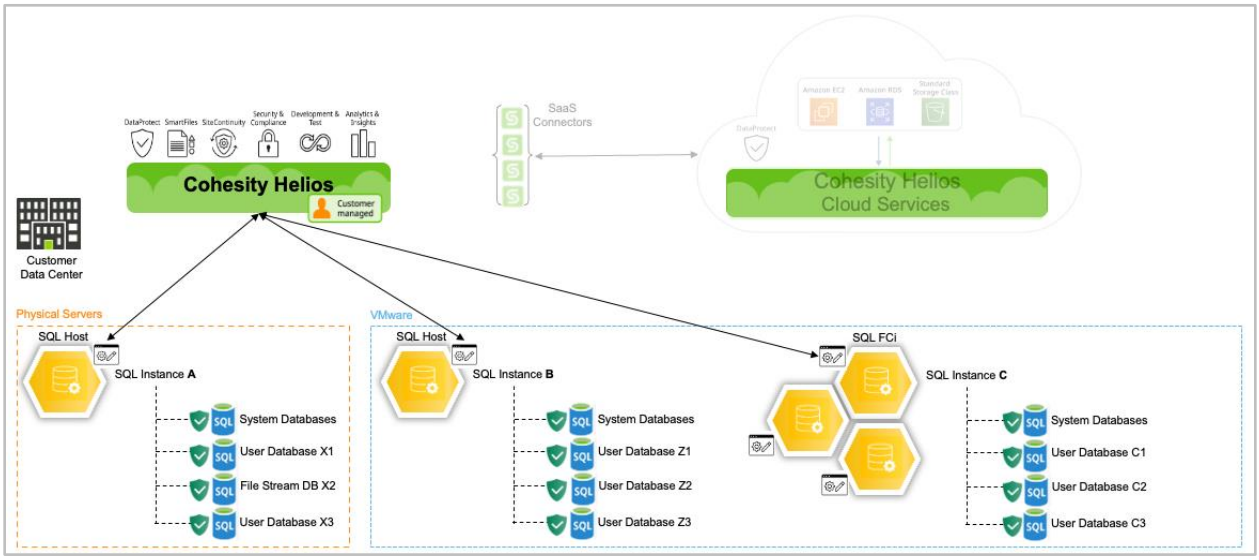
1. SQL Instance configuration
2. SQL database type
3. Database size

Cohesity recommends you evaluate your SQL instances and database to determine which databases to protect with DMaaS.

The following charts and diagrams will help you develop a strategy for migrating your SQL backups to DMaaS.

By comparison, Cohesity On-Prem SQL Data Protection supports all SQL instance configurations and database types. Cohesity SQL Adapter employs several backup technologies, which have the ability to support all the SQL instance/database type combinations.

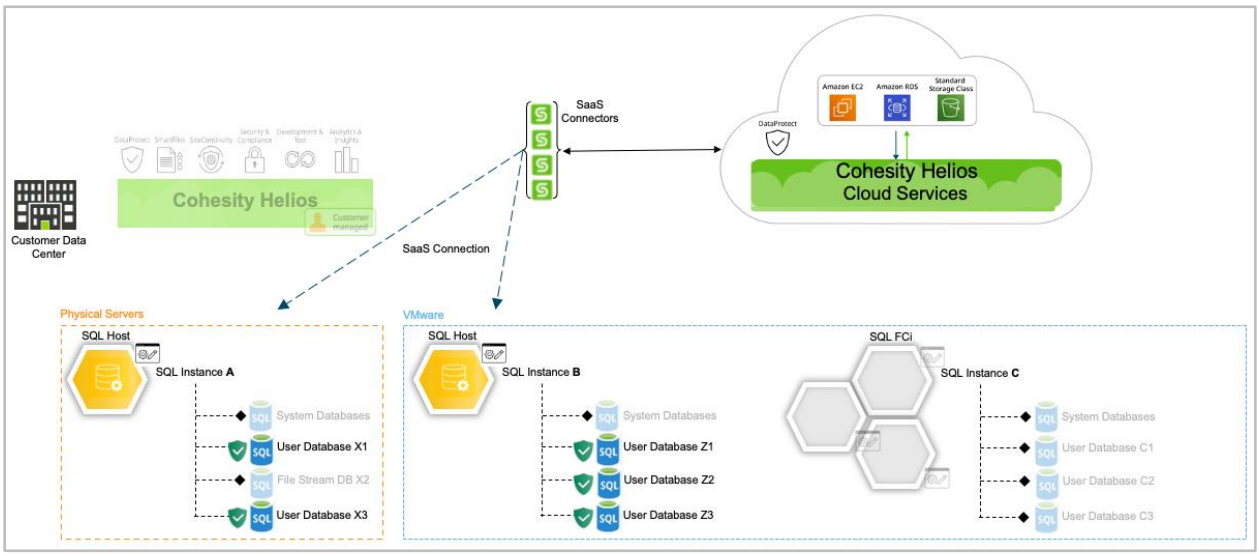
Diagram 6: SQL Protection Coverage using Cohesity On-Prem



Tip: You can still take advantage of Cohesity's Cloud Archive and tiering to reduce your overall storage costs.

By contrast, Cohesity SQL DMaaS employs VDI-based backup technology, which supports a subset of SQL instance/database type combinations. This means that some SQL instances must remain under Cohesity On-Prem DataProtect.

Diagram 7: SQL Protection Coverage Using Cohesity DMaaS



Notice this diagram shows you that the SQL FCI instance, system databases, and the Filestream database are not supported by SQL DMaaS.

SQL Instances and databases that are not candidates for DMaaS protection.
These databases should be protected with Cohesity On-Prem.

- SQL Databases on a Windows Failover Cluster
- SQL AAG Databases
- SQL Filestream Databases
- SQL System Databases

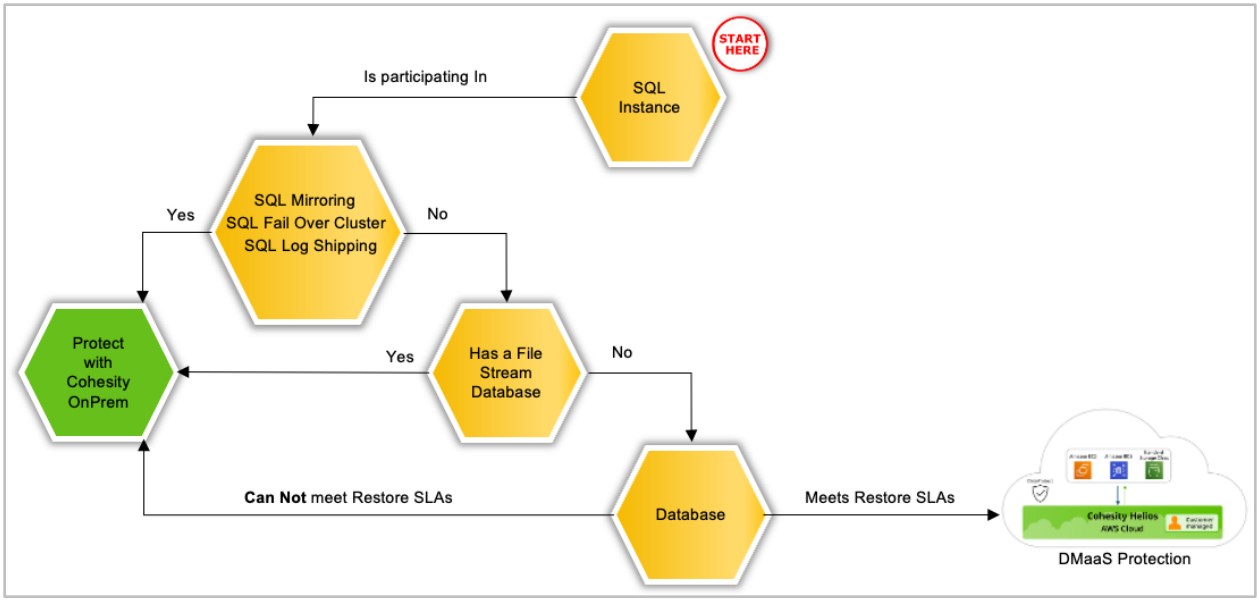
Note: The advantage of a VDI-based backup is its portable SQL native format. It can be restored to any qualified SQL Server Instance. This is not so with cloud snap backups or exports.

Which SQL Databases to Migrate to DMaaS?

Three factors are critical in determining which SQL database can be protected with DMaaS.

They are (1) SQL Instance configuration, (2) SQL database type, and (3) database size. Below is a decision tree to help you evaluate your SQL databases.

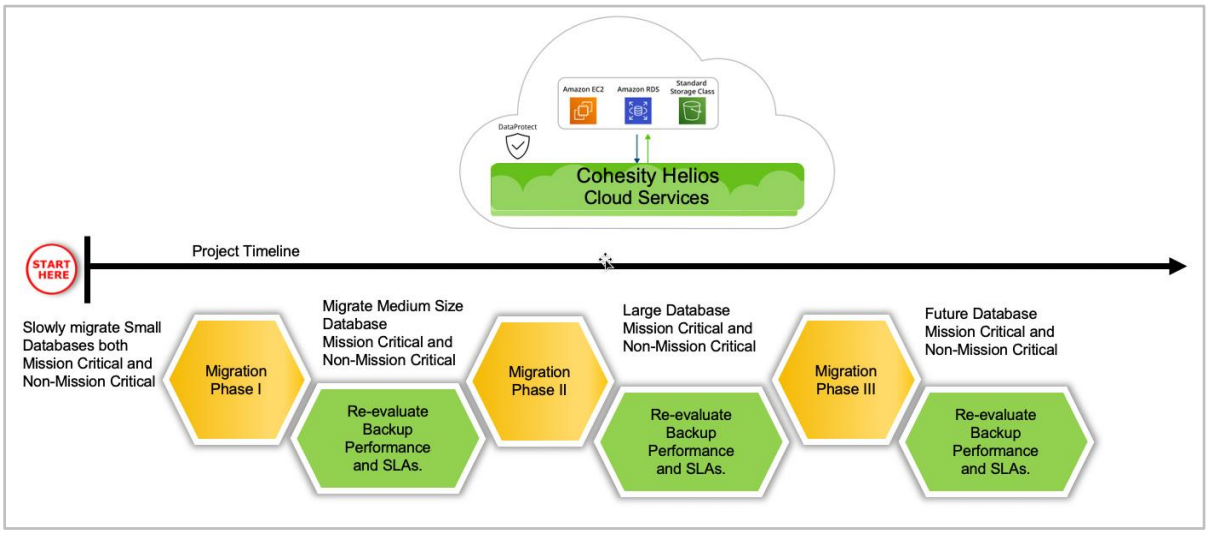
Diagram 8: DMaaS database Decision Tree



A DMaaS Migration Method

Switching protection of qualified SQL databases to DMaaS should be methodical and deliberate so as to avoid overloading the SaaS Connection.

Diagram 9: DMaaS database Migration Method



Restore Considerations

To ensure a smooth database restoration operation with Cohesity, follow these best practices and guidelines.

Restore Prerequisites.

- Compatible SQL Server version
- Sufficient disk space on the target server
- SQL Server instance is up and running
- SQL Adapter is up and running
- SQL Adapter has proper permissions

NOTE: Cohesity recommends periodically restoring a sample database to validate restore workflows, restore SLAs, and backup integrity.

Retention for VDI Backups

The aim of the SQL DBA is to have a combination of backups so that the database can be *restored* to any point in time. A good combination of backups consists of having a full backup, differential (in Cohesity, *incremental*) backups, and log backups.

For example, all SQL database restores must begin with a full database backup,a differential can then be applied to the full, and then finally, log backups can be applied in sequence to complete the database restore.

When the backups are applied during the database restore process, you are sequentially adding the captured changes to the database: FULL+DIFF+Log1+Log2+Log3 = Restored Database.

IMPORTANT: All Microsoft VDI backups, their differentials, and their logs are dependent on a full backup to perform a database restore. Microsoft requires that in order to restore a SQL Database, you must start with a full backup, then its transaction logs can be applied. This means your backup retention policy must keep a full backup along with its log backups in order to successfully restore a database.

Simply put, a SQL VDI-based database restore requires a full backup to seed the database, then a differential and/or log backups are applied to the specified point in time.

Cohesity recommends retaining two sets of full backups with their differential and log backups.

IMPORTANT: You must specify a Periodic FULL backup in the Cohesity Policy.

TIP: A good recovery plan always includes a combination of full, differential, and log backups.

Archiving and Replication

If you are performing or planning to archive or replicate your SQL VDI-based backups be sure that a FULL backup is present in the backup chain. Ensure that your FULL backups are available and current by setting a retention that preserves them together with your incremental and logs.

Technical Considerations

The SaaS Connector for BaaS will work with many Source Types and Platforms.

Table 2: SQL SaaS Connector Deployment Recommendations

DMAAS SOURCE TYPE	SAAS CONNECTOR REQUIRED?	SAAS CONNECTOR PLATFORM
Virtual Machines (VMware | Hyper-V)	Yes	VMware source > VMware OVA Hyper-V source > Hyper-V VHD
NAS (Generic | NetApp | Isilon)	Yes	VMware or Hyper-V
M365 (Mail | OneDrive | Sites | Teams)	No (uses proxy with agent)	N/A
MS SQL Server * Cohesity SQL Agent required	Yes	VMware or Hyper-V
Oracle	Yes	VMware or Hyper-V
AWS Snapshot (EC2 | RDS) (on-prem = Snapshot Manager)	No (direct AWS API calls)	N/A
AWS Cohesity Snapshot (EC2) (on-prem = Native Snapshot)	Yes	AWS
Physical (Windows | LINUX)	Yes	VMware or Hyper-V

Configuration and Resources

Resources requirements are the same for all supported platforms:

4 x CPUs, 10GB memory, 20GB disk space, 100Mbps throughput, 100 IOPS

- VMware SaaS Connector
  4 x vCPU, 10GiB Memory, 22GiB HDD

- Hyper-V SaaS Connector

  4 x vCPU, 10GiB Memory, 22GiB HDD
  Note that 4 x vCPU is required but only 1 vCPU gets provisioned when the VM is deployed.

- AWS SaaS Connector
  M5.xlarge instance type
  4 x vCPU, 16GiB Memory, 22GiB gp2 EBS Volume

Network

All SaaS Connectors are deployed with 2 10G NICs. These are configured to use the same network or they can be assigned each to different networks.

Network bandwidth usage is automatically balanced among the SaaS Connectors within each SaaS Connection. However, if you need to contain the amount of network bandwidth consumed by your backup and recovery tasks at different times and days of the week, the Cohesity DataProtect service allows you to throttle your bandwidth consumption in your SaaS Connections.

The bandwidth usage options in each SaaS Connection allow you to choose the days of the week and set the start and end times to limit bandwidth usage to a specific value in bytes per second.

Note: If your VMs are running on a hypervisor configured with a 10G port then the SaaS Connector will also run at 10G. If your hypervisor runs a 1G port, then the SaaS Connector will be limited to 1G.

Security

A SaaS connector can be used to bridge a private network, which can be seen as a security risk.

Customers adopting an “as-a-service” model will require public internet access – this is required regardless of the service being used.

The Cohesity SaaS connector poses a very limited security risk.

The SaaS connector UI only contains the ability to register the SaaS connection.

SSH login will always be refused – no authorized user.

RT access is disabled by default – but can be enabled temporarily by the customer if needed

Only direct console access can be used to connect to the CLI (password is not shared/secured).

Firewall Ports

Table 3: Firewall Ports for DMaaS

PORT	PROTOCOL	TARGET	DIRECTION (FROM CONNECTOR)	PURPOSE
443	TCP	helios.cohesity.com	Outgoing	Connection used for control path
443	TCP	helios-data.cohesity.com	Outgoing	Used to send telemetry data
22, 443	TCP	rt.cohesity.com	Outgoing	Support channel
11117	TCP	*.dmaas.helios.cohesity.com	Outgoing	Connection used for data path
29991	TCP	*.dmaas.helios.cohesity.com	Outgoing	Connection used for data path
443	TCP	*.cloudfront.net	Outgoing	To download upgrade packages
443	TCP	*.amazonaws.com	Outgoing	For S3 data traffic
123, 323	UDP	ntp.google.com or internal NTP	Outgoing	Clock sync
53	TCP & UDP	8.8.8.8 or internal DNS	Bidirectional	Host resolution

These firewall rules allow outgoing traffic from a SaaS Connector to the DataProtect service endpoint. The SaaS Connector opens a secure encrypted gRPC tunnel to the endpoint and uses it for both backup and recovery traffic.

Upgrades

SaaS connector upgrades are triggered by DMaaS data plane upgrades.

Cohesity (CloudOps) uses the CIM service in the DMaaS control plane to push upgrades.

SaaS connectors are generally upgraded within a day (depends on time to upgrade all data plane clusters, SaaS connector availability, and more).

Upgrades are briefly disruptive (transparent to user)

Protection and recovery operations will not fail as a result of an upgrade.

It is possible some subtasks may briefly queue during a SaaS connector upgrade.

Multiple SaaS connectors

Rolling upgrades are used when multiple SaaS connectors are present.

Each SaaS connector will be unavailable for a brief time during upgrade.

Subtasks will continue to execute on SaaS connectors that are currently available.

Other Tunables

Cohesity managed cloud environments are monitored for performance and efficiency. They alert on connectivity problems and protection jobs that do not meet their SLAs.

Even though the DMaaS Operations team is working behind the scenes, there are some things you can do to ensure the best possible backup job performance.

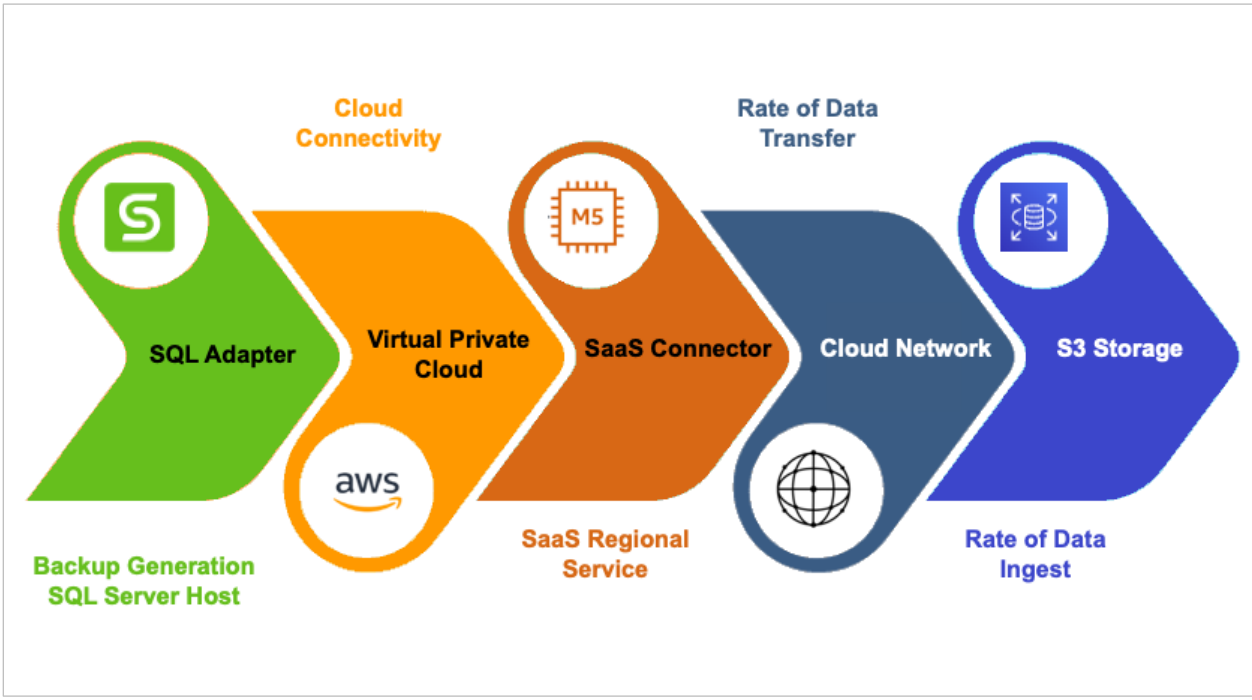
All the Links in the Chain

A DMaaS backup of a SQL Server database is a series of technical links in a chain. This means that the most efficient backup requires all the links in the chain to work optimally.

The major links are:

- Backup Generation
- Cloud Connectivity
- SaaS Regional Service
- Rate of Transfer
- Rate of Data Ingest

Having an efficient SQL DMaaS backup and recovery means no link in the chain creates a bottleneck for the flow of data.



Backup Generation

Performance of the SQL host is the first link in the chain.
To ensure that your backups run with the highest possible performance, review and optimize each component in the backup process, starting with the SQL Server host:

- MS SQL Server
- Network
- The Cohesity platform

Gauging MS SQL Server Host performance (Physical or VM) is important, as a SQL Server that is underpowered, equipped with poor resources, or overloaded will adversely affect any backup process.

Cohesity recommends that MS SQL Server hosts meet all Microsoft best practices.

SQL Host CPU

Cohesity recommends that MS SQL Server hosts meet Microsoft SQL Server CPU recommendations and best practices. For a good reference on this, see Storage and SQL Server capacity planning and configuration.

For example, a SQL Server host (physical or VM) can have an average CPU usage between 1%-15%, with swells of 16%-30% and spikes of 65%-85% during high transaction volumes.

MS SQL Server consumes CPU resources equally across all processors on the host. It derives its transactional power from CPU cycles. If not managed, the operating system and MS SQL Server will compete for CPU cycles.
High CPU usage means fewer cycles to spend on taking SQL backups, leading to longer waits to release worker threads and resolve transactional commits to the database.

SQL Host Memory

Cohesity recommends that MS SQL Server hosts meet Microsoft SQL Server memory recommendations and best practices. For a good reference on this, see Storage and SQL Server capacity planning and configuration.

MS SQL Server assumes that host memory exists solely for its own purpose. If not managed, the operating system and MS SQL Server will compete for host memory. For example, when a production Windows host will have 16GB-64GB of memory.
MS SQL Server aggressively attempts to push all database objects into memory — tables, indexes, stored code, and procedure caches — into memory, it has a large impact on total available memory. In turn, inefficient memory usage indirectly degrades backup generation.

A well provisioned production Windows Server host has a lot of memory. Approximately 6-10 GB is allocated for the operating system, and the rest is reserved for MS SQL Server. A properly configured MS

SQL Server uses all available host memory. MS SQL Server automatically moves the most frequently used objects into memory. If memory pressure is too great, the OS will start writing to the system pagefile and this will have a significant impact on performance.

SQL Server Database settings

Cohesity recommends the SQL database RECOVERY_MODEL property be set to FULL.

The Full Recovery Model requires log backups to be taken on the database to offset the growth of the database log file. Taking log backups allows recovery of the database to any specific point in time.

NOTE: With the *Simple* Recovery Model, the database log file remains at one size, and recovery of the database is only available for the last backup point in time. The changes that occur after the most recent backups are left unprotected. **Cohesity recommends *against*** using the Simple Recovery Model in a production environment.

Deploy the Cohesity SQL Adapter

You need to install the Cohesity Windows Agent on your SQL Server host. The Windows agent is designed to work specifically with the Windows operating system and is compatible with Windows versions 2008R2 and above. If there are multiple SQL Server hosts, you will need to install the agent on each host you wish to protect.

You manage the Cohesity Agent through the Sources page in Cohesity. When an upgrade becomes available for any agent you've installed, an **Upgrade Agent** button appears next to your SQL Server source. You can upgrade the agent from there.

> **IMPORTANT**: For physical servers you need to install the agent on each SQL Server host you wish to protect.

Under Cohesity Cloud Services, upgrades to the SQL agent are performed automatically via the SaaS Connector.

Your Feedback

Was this document helpful? Send us your feedback!

About the Authors

Scott Lorenz is a Staff Solutions Engineer at Cohesity. In his role, Scott focuses on business-critical applications, MS SQL Server databases, cloud storage, and enterprise data protection. Scott has over 26 years' experience as an enterprise DBA.

Other essential contributors include:

- Jay White, Technical Director - ATSO

Document Version History

VERSION	DATE	DOCUMENT HISTORY
1.1	July 2024	Republishing
1.0	Mar 2023	First release

ABOUT COHESITY

Cohesity is a leader in AI-powered data security and management. Aided by an extensive ecosystem of partners, Cohesity makes it easier to protect, manage, and get value from data – across the data center, edge, and cloud. Cohesity helps organizations defend against cybersecurity threats with comprehensive data security and management capabilities, including immutable backup snapshots, AI-based threat detection, monitoring for malicious behavior, and rapid recovery at scale. Cohesity solutions are delivered as a service, self-managed, or provided by a Cohesity-powered partner. Cohesity is headquartered in San Jose, CA, and is trusted by the world's largest enterprises, including six of the Fortune 10 and 44 of the Fortune 100.

Visit our website and blog, follow us on Twitter and LinkedIn and like us on Facebook.