

Version 1.1

July 2024

Protect SQL Server Failover Cluster with Cohesity

Use Cohesity Solution for Backup and Restore of SQL Server Databases Configured with Microsoft Failover Cluster

ABSTRACT

As databases continue to grow in number and size, data centers in today's organizations need different methods and strategies to protect their databases and manage their growing data. Cohesity has added another backup method to your toolbox, a SQL adapter that is failover cluster aware. With Cohesity's cluster-aware adapter, protecting databases on Microsoft's highly redundant failover cluster configuration is now easy.

Table of Contents

Complexity Forces Us to Sink or Swim	3
How Cohesity Protects Your SQL Failover Cluster	4
Features and Benefits of Cohesity Protection	4
Use Cohesity Protection for SQL Databases	6
Deploy the Cohesity Windows Agent	7
Register SQL Server in Cohesity	11
Create a Database Protection Group	17
Recover SQL Database	22
Upgrade Your Disaster Recovery Preparedness	24
Take Local Snapshots	24
Replicate Backups Off-Site.....	25
Archive Backups to the Cloud	25
Best Practices for Cohesity SQL Server Protection	26
Appendix A: Terminology	28
Appendix B: Product Documentation	29
Your Feedback	30
About the Authors.....	30
Document Version History.....	30

Figures

Figure 1: Cohesity's Protection for SQL Server Failover Clusters	4
Figure 2: Set Up SQL Server Data Protection with Cohesity	6
Figure 3: SQL Backups in Cohesity are Available to Replicate and Archive	24
Figure 4: Cohesity CloudArchive, Cloud Recover, and CloudRetrieve Provide Disaster Recovery	25

Complexity Forces Us to Sink or Swim

In complex data center environments, there is no easy way to manage all the backups and recoveries that comprise the foundation of their protection. Administrators often describe the task as similar to keeping many plates spinning at the same time; without an enterprise-level data management solution, one is left to either sink or swim.

Three essential factors push the demand for efficient data management:

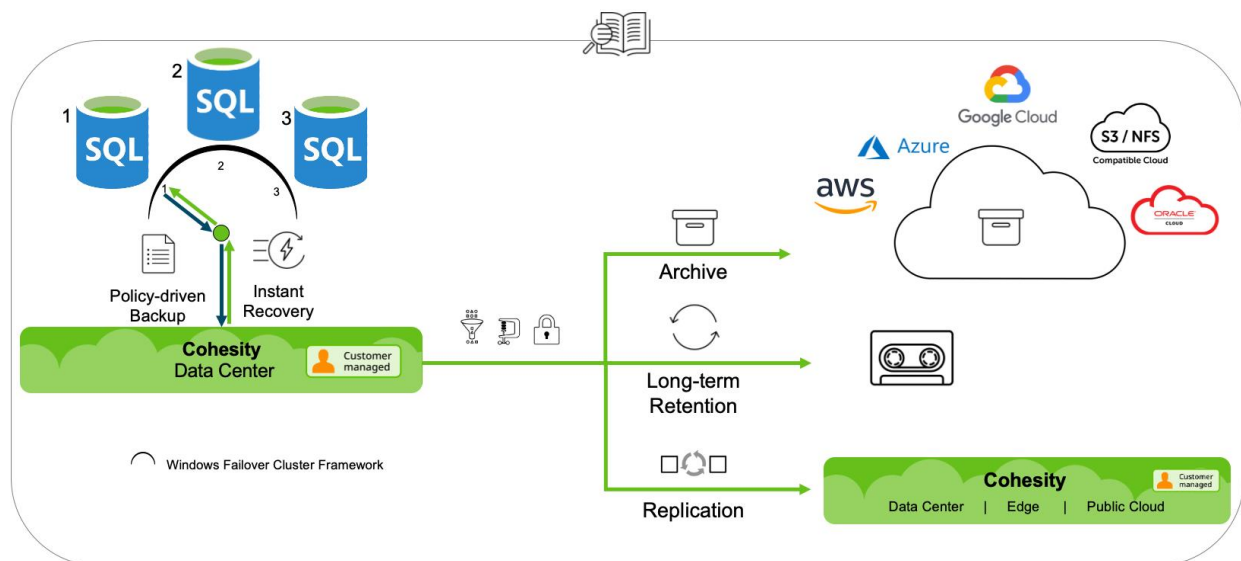
- **The growing number of databases.** The number and kind of backups increases as the demand for protection continues to grow.
- **The increasing size of databases.** Larger databases result in longer backup windows that make it harder to meet your SLAs.
- **The ever-increasing duration and cost of retention.** Managing the storage for backups becomes more difficult as company standards and government regulations increase retention requirements.

How Cohesity Protects Your SQL Failover Cluster

Cohesity is a modern, software-defined platform for data management. Taking inspiration from its web-scale architecture and leveraging its unique distributed file system ([SpanFS®](#)), Cohesity offers high scalability and reliability, with extensive features to help you meet your business and compliance needs, SLAs, long-term retention, recovery, and preparedness requirements.

Cohesity's flexible architecture makes expansion easy by increasing operational simplicity and improving total cost of ownership (TCO). Cohesity's solution for SQL Server works on-premises, in the public cloud, and in your remote and branch offices. You can choose how to back up your data, where to keep your backups, and for how long.

Figure 1: Cohesity's Protection for SQL Server Failover Clusters



Features and Benefits of Cohesity Protection

Cohesity's solution for SQL Server includes many features that make your backups much more valuable, including:

- **Flexibility.** Cohesity gives you the ability to browse and search across all your snapshots, and to restore to different locations on different servers.
- **Performance to Meet Your SLAs.** Cohesity gives you the backup performance you need to protect your SQL Server databases efficiently and securely.
- **Scalability.** Cohesity protection for SQL Server is scalable from a single database to several SQL failover cluster instances (FCI) and even an entire data center with hundreds of SQL instances.
- **Compression.** Data compression significantly reduces storage usage and data transmission. Efficient storage means you have room for more backups and other important data. By default, Cohesity performs compression on all the data it stores.

If you also enable *inline compression*, the process occurs as Cohesity is saving the data to storage, instead of after saving it.

- **Encryption at rest, in flight, and in the cloud.** It is vital to protect your data from unauthorized access.
 - **Data at Rest.** The Cohesity [SpanFS](#) file system provides full at-rest encryption based on the strong AES-256 CBC (Cipher Block Chaining) standard.
 - **Data in Flight.** Cohesity encrypts all data while it is being transmitted.
 - **Data in Cloud.** Cohesity's CloudArchive provides encryption for data stored in the cloud.
For details, see [Cohesity Security Features](#) in the online Help.
- **Archive to cloud.** Cohesity's policy-based ability to archive to public clouds like AWS, Azure, and Google Cloud, as well as to any S3-compatible storage, makes it easy to leverage lower-cost, long-term retention and protect your data from regional disasters. Cohesity makes it easy to retrieve your organization's information to different geographical locations, whenever you need to.
- **Disaster Recovery.** Protect your SQL Server universe from disaster by replicating your Cohesity backups to another location that can be ready to failover (and failback, after repairs) as soon as disaster strikes.
- **Database Cloning.** SQL databases can be cloned for several use cases like testing and development. A clone of a SQL database can be brought up and taken down on demand. The SQL clone is safe since it is completely separate from production and backups. It is very efficient since it uses no storage on the host machine.

In addition, you can use Cohesity to protect SQL Servers that are deployed with different configurations. SQL Server might be on a Windows Failover Cluster, or configured for Always On Availability Groups (AG), or might be running on a Virtual Machine (VM). In such cases, you can use Cohesity for other *SQL Server backups* and even *create new backup strategies*. For example, you can protect the SQL Server database and the VM it is running on, or you can protect SQL databases across different data centers.

Use Cohesity Protection for SQL Databases

Cohesity offers a policy-based, highly scalable data protection infrastructure for your SQL Server data. With a few steps, you can set up Cohesity to meet all your data protection requirements.

Cohesity uses three methods to perform backups of databases on any SQL Server instance that is registered with Cohesity, and they are:

- File-based backup
- Volume-based backup
- Virtual Device Interface (VDI) backup

Cohesity supports full database server backups, differential SQL Server database backups (via Cohesity incremental backups), and SQL Server T-log backups.

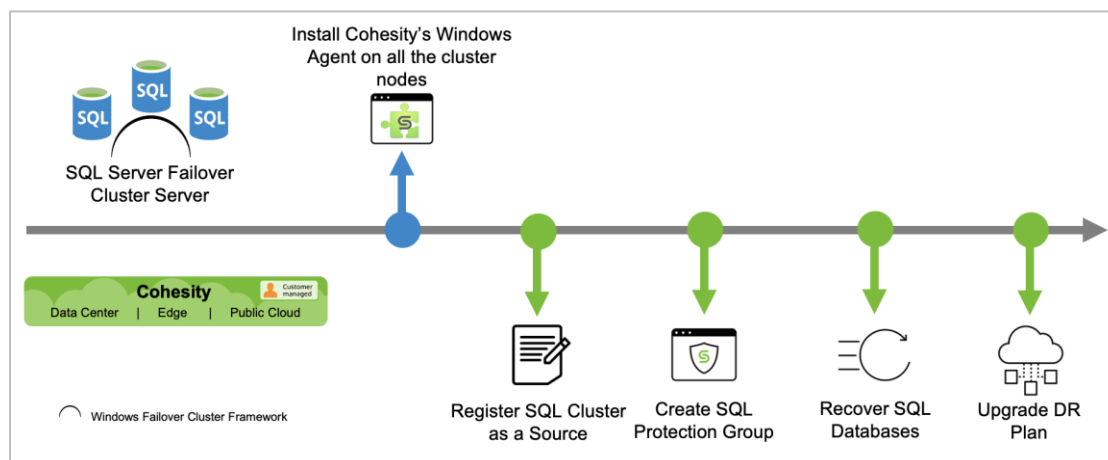
It is important to understand the process that makes a backup strategy successful. For example, it is important to have a second copy of backup data in case the original copy fails. But that is only part of the story. When you need to recover your SQL databases, it is crucial that you be able to find those backups and restore them quickly. To take full advantage of many features of Cohesity's solution, be sure you understand each step of the implementation.

To protect your SQL databases using Cohesity:

1. [Install and deploy Cohesity's Windows Agent on your SQL Server.](#)
2. [Register your SQL Server as a Cohesity source.](#)
3. [Create a Cohesity Protection Group to specify the SQL data you need to protect.](#)
4. [Recover protected SQL Server databases.](#)
5. [Upgrade your disaster recovery \(DR\) plan to improve your enterprise's readiness and resilience.](#)

NOTE: For more background, see [Appendix A: Terminology](#) and [Appendix B: Product Documentation](#).

Figure 2: Set Up SQL Server Data Protection with Cohesity



Complete these steps to protect your SQL databases. Get started by deploying Cohesity's Windows Agent next!

Deploy the Cohesity Windows Agent

To start, 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.

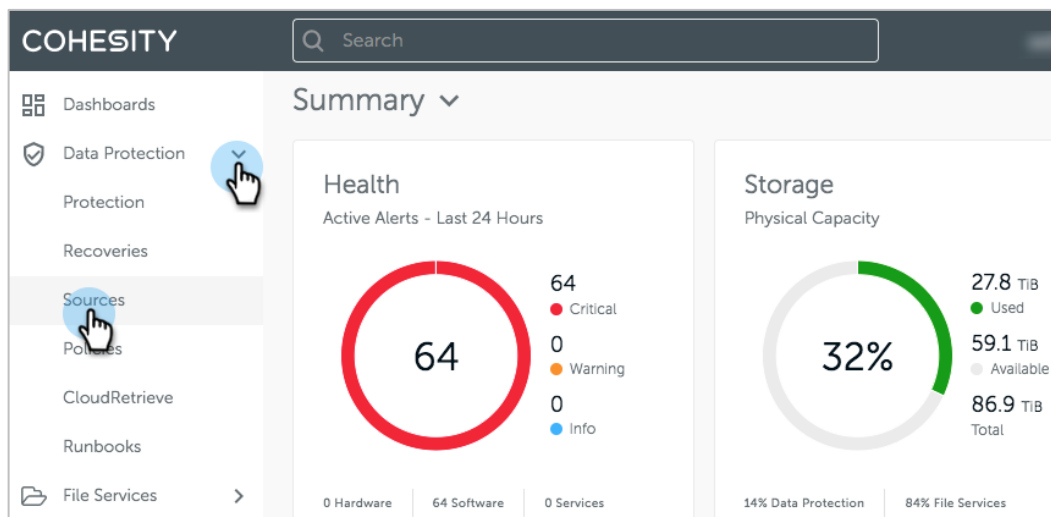
The agent is lightweight and has a small memory footprint. It carries out the tasks you define in your Cohesity Protection Groups and ties together technologies and capabilities already in Windows, like Windows VSS, and new technologies, like Cohesity Changed Block Tracker (CBT), so that you can tackle data management efficiently.

You can 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.

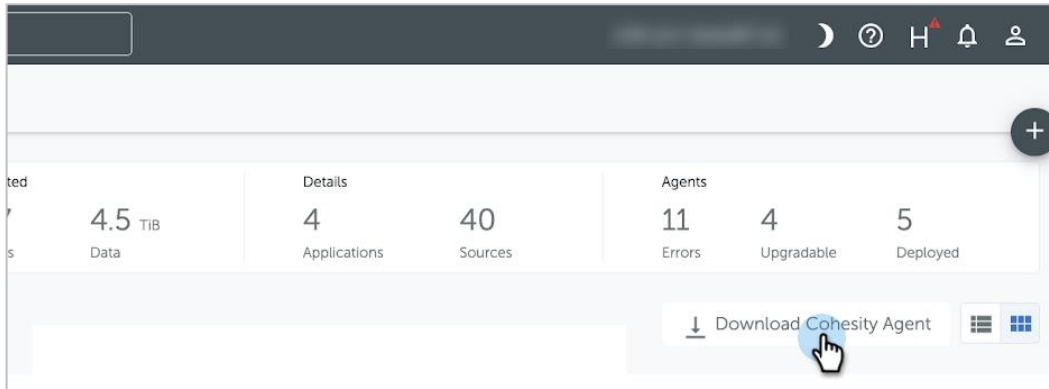
IMPORTANT: You need to install the agent on each node of the Windows cluster.

To install the agent:

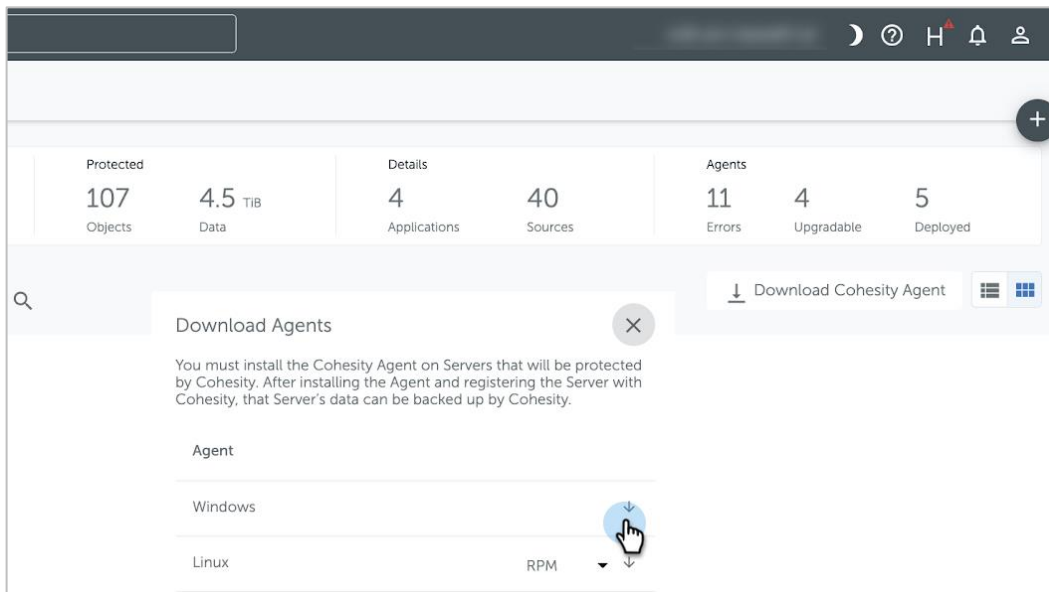
1. Log in to Cohesity and navigate to **Data Protection > Sources**.



2. Click **Download Cohesity Agent**.

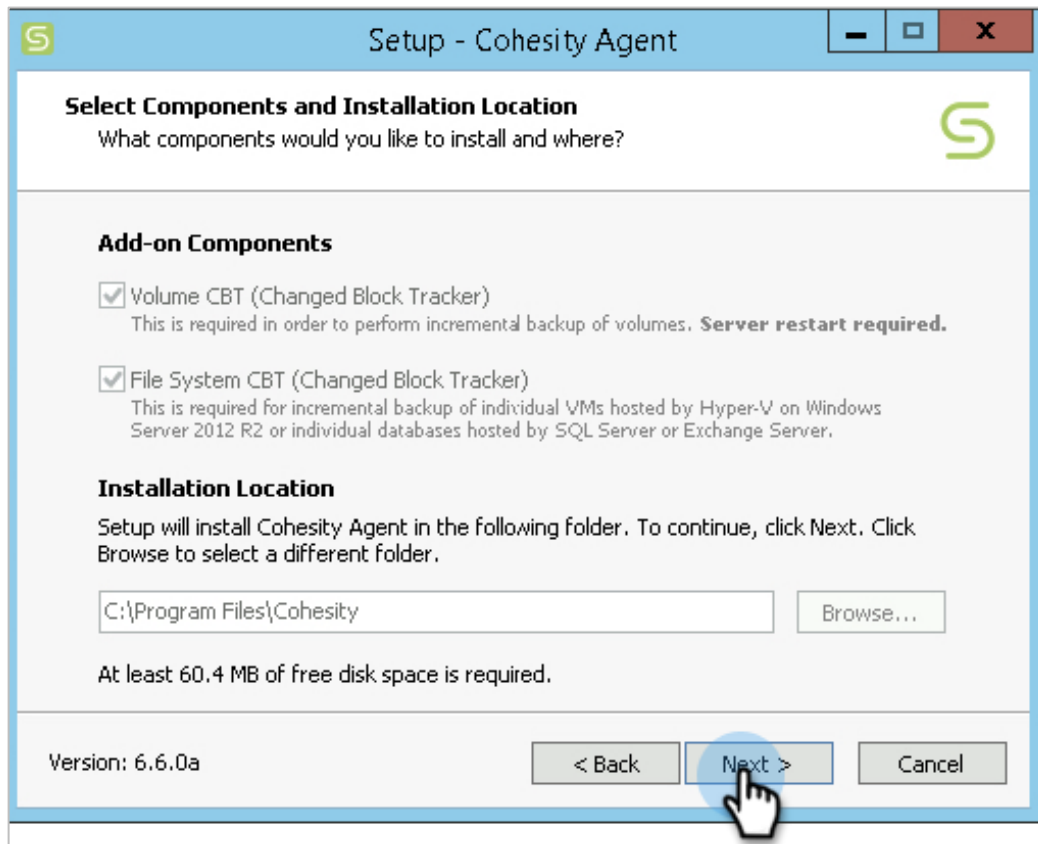


3. Click **Download (↓)** button for **Windows**.



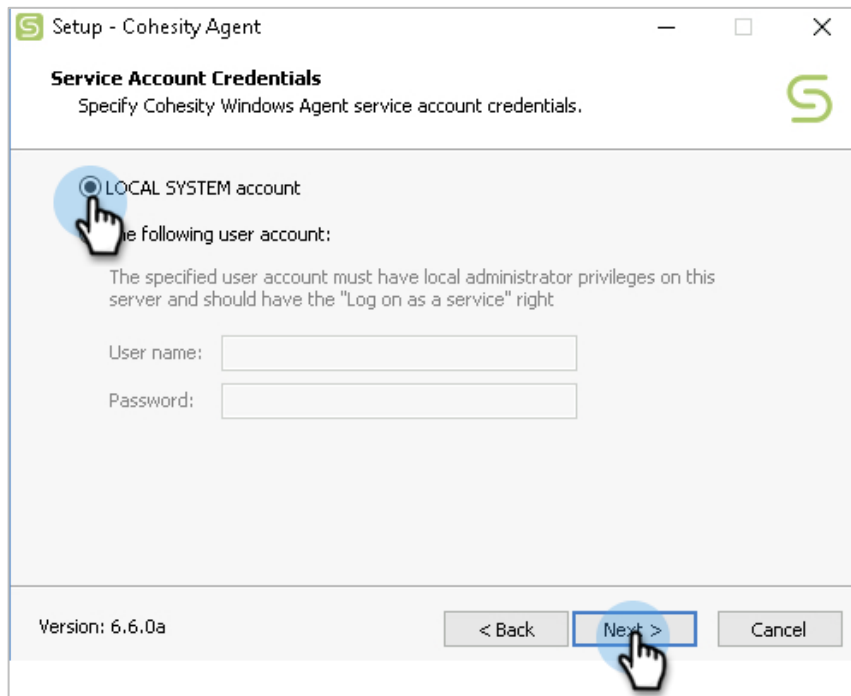
4. Download or copy the agent to all nodes of the failover cluster. On each node, install the agent and run the installer.

- When you launch the Cohesity Windows Agent installer, under **Add-on Components**, ensure that **Volume CBT** and **File System CBT** (the default) are selected and click **Next**.



TIP: The Cohesity volume snapshot captures all the data on the SQL Server host. That means it captures all the changes on the volume and changes to other, non-SQL files. To keep your SQL backups running efficiently, keep the volume that SQL Server is using clear of lower-priority and unrelated files.

6. Select the type of service account to use for control over your systems. The **LOCAL SYSTEM account** gives you direct control, but you can also enter an Active Directory domain admin user account.



TIP: If you are unsure which account to use, don't let that slow you down — choose the **LOCAL SYSTEM account**. This account already has most of the required permissions. You can change the agent service account later if you change your mind.

For more, see [Protect SQL Server with Cohesity — Deployment Considerations Guide](#).

Your SQL FCI is now ready to be registered as a Cohesity source in the next chapter.

Register SQL Server in Cohesity

To protect your SQL Server databases with Cohesity and take advantage of its many features, you need to register it as a Cohesity source. Once it's registered in Cohesity, you will be able to add it to a Protection Group and configure the settings for your environment.

To register SQL Server as a Source in Cohesity:

1. Before you register SQL Server as a source, you need to join AD to your Cohesity cluster. Log in to Cohesity, navigate to **Settings > Access Management > Active Directory**, and click **Add Active Directory**.

The screenshot shows the Cohesity web interface. The left sidebar contains navigation options: Dashboards, Data Protection, Infrastructure, File Services, Test & Dev, Marketplace, System, Reporting, Settings, Summary, Access Management, Network, and SNMP. The 'Access Management' section is selected. The main content area is titled 'Access Management' and features a search bar and a navigation bar with tabs: Users & Groups, Roles, Active Directory, Kerberos, LDAP, Organizations, SSO, Keystone, and NIS. The 'Active Directory' tab is active. Below the tabs is a table with columns: Domain, Organization, Organizational Unit, Workgroup, Machine Accounts, and Mapped Provider. The table contains four rows of data. A blue button labeled 'Add Active Directory' is located in the top right corner of the main content area. Hand icons with numbers 1, 2, 3, and 4 indicate the steps: 1 points to the 'Settings' menu item, 2 points to 'Access Management', 3 points to the 'Active Directory' tab, and 4 points to the 'Add Active Directory' button.

Domain	Organization	Organizational Unit	Workgroup	Machine Accounts	Mapped Provider
qa01.eng.cohesity.com	-	-	QA01	haswell6-p1	-
pmad02.pm.cohesity.com	testorg	-	PMAD02	sac01-haswell6	-
jarvis.dc	jarvis	-	JARVIS	cohesity-hyx	-
nastarget.com	-	-	NASTARGET	haswelnew6	-

2. In the **Join Active Directory** form, enter the AD administrator **Username** and **Password** and click **Join**.

× Back to Active Directory

Join Active Directory

Username* administrator

Password*

Preferred Domain Controllers (Optional)

Machine Accounts

+ Add

Machine Account	DNS Hostname	Encryption Types
haswell6-vip	-	2 Selected

Use the above Machine Accounts even if they already exist in AD Domain.

Mapped Provider

None LDAP NIS

Organizational Unit (Optional)
Format - OUName or OUName/SubOUName

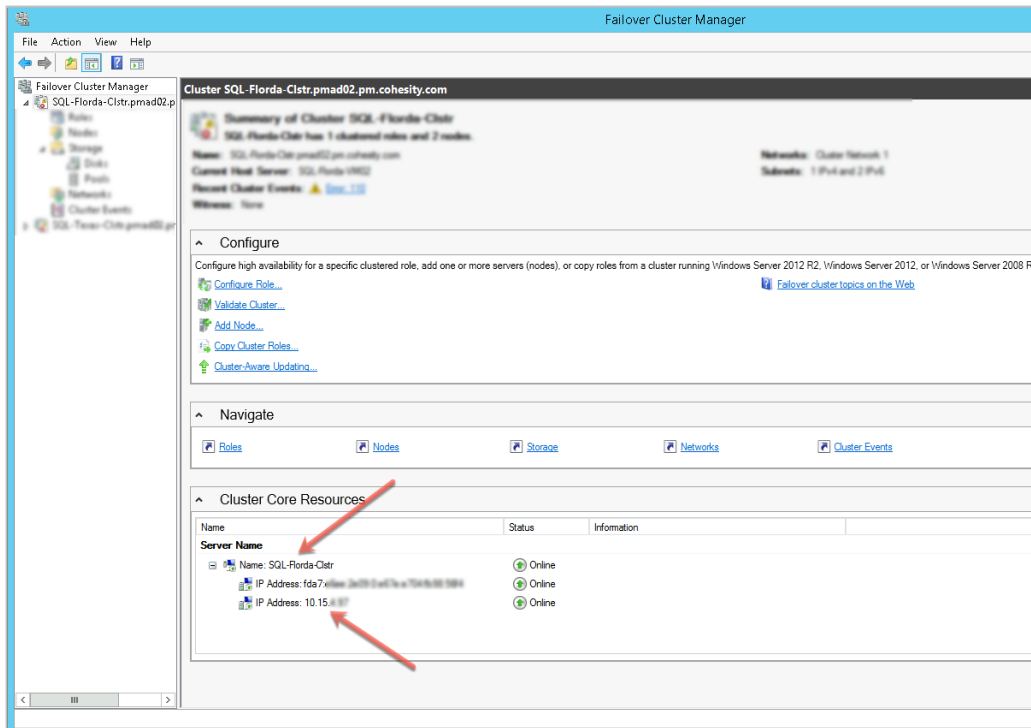
AD Workgroup / NetBIOS Name (Optional)

Discover Trusted Domains

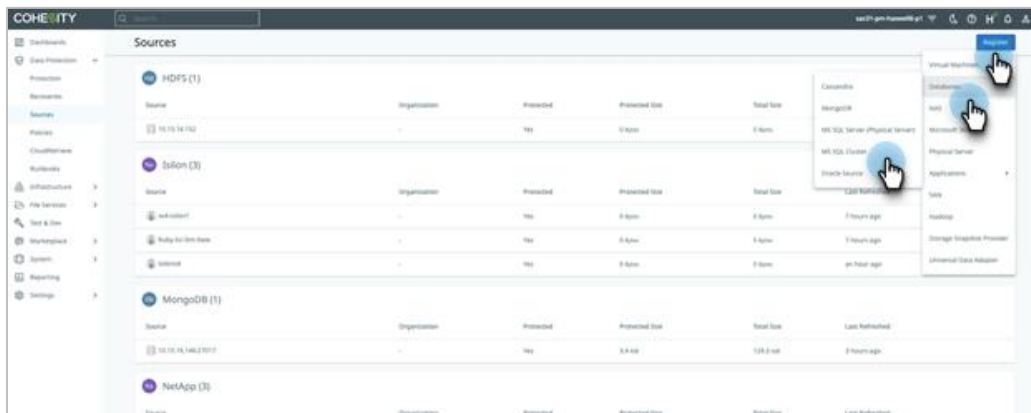
Join Cancel

For details on the other options here, see [Join the Cluster to an AD Domain](#) in the online Help.

- Before you can register SQL Server as a source, you need the SQL Server cluster FQDN or IP address. You can get the **FQDN** or **IP Address** from the Windows **Failover Cluster Manager**.



- Navigate to **Data Protection > Sources**. Then click **Register > Databases > MS SQL Cluster**.



5. In the **Register MS SQL Cluster** form, enter the SQL Server cluster FQDN or IP address and click **Register**.

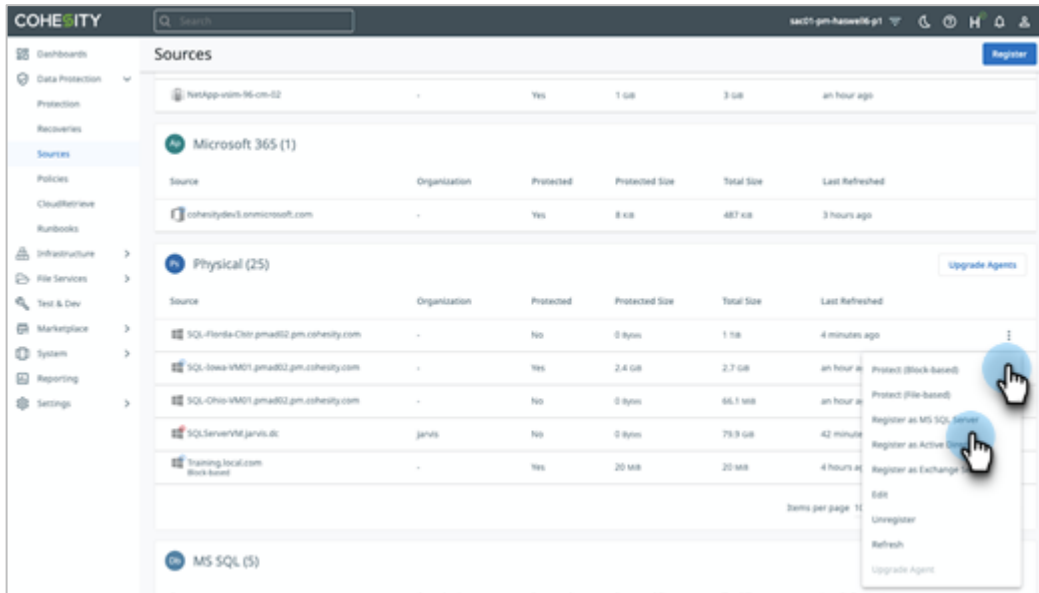


Cohesity recommends you register an FCI by using the virtual IP (SQL VIP) for the SQL cluster. Properly registering an FCI as a Cohesity MS SQL cluster source enables Cohesity to follow the SQL instance during a failover to new nodes, and then continue to take backups.

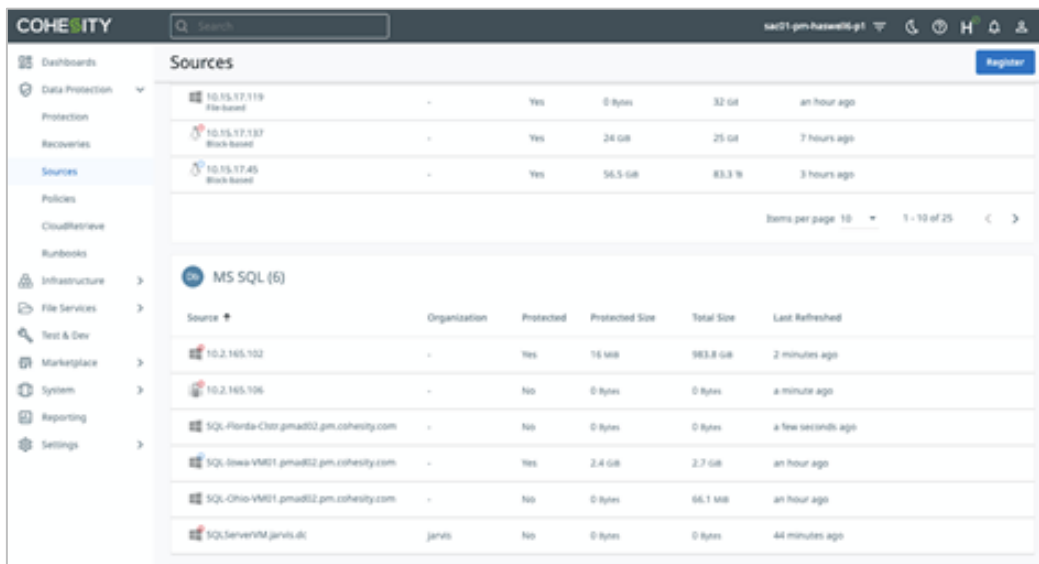
IMPORTANT: A common mistake is to register a SQL FCI using the physical IPs or the Windows cluster IP instead of the SQL FCI's *virtual IP*.

NOTE: Cohesity does not support Clustered Shared Volumes (CSV). CSV refers to the storage underneath the SQL FCI. Clustered Shared Volumes enable multiple nodes in a failover cluster to simultaneously have read-write access to the same LUN (disk) that is provisioned as an NTFS volume.

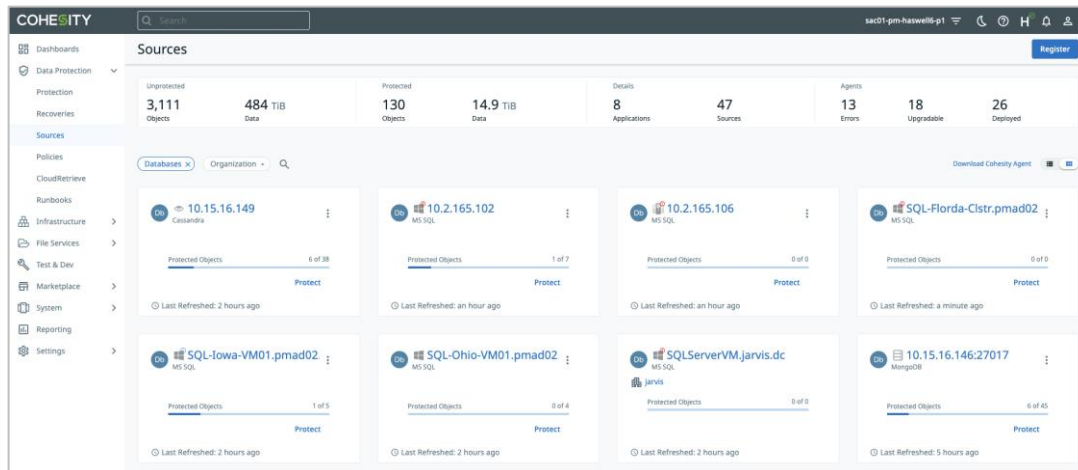
- In the **Sources** form, navigate to your new source, click the ellipsis (⋮) and click **Register as MS SQL Server**.



NOTE: In this step you register SQL Server as an application. Cohesity recognizes the host and the SQL Server application so that it has full application awareness.



- The **Sources** page now includes your SQL Server (in our example, **SQL-Florda-Clstr**), available for immediate protection.



To protect your newly registered SQL source, you'll create a Cohesity Protection Group for it in the next chapter.

Create a Database Protection Group

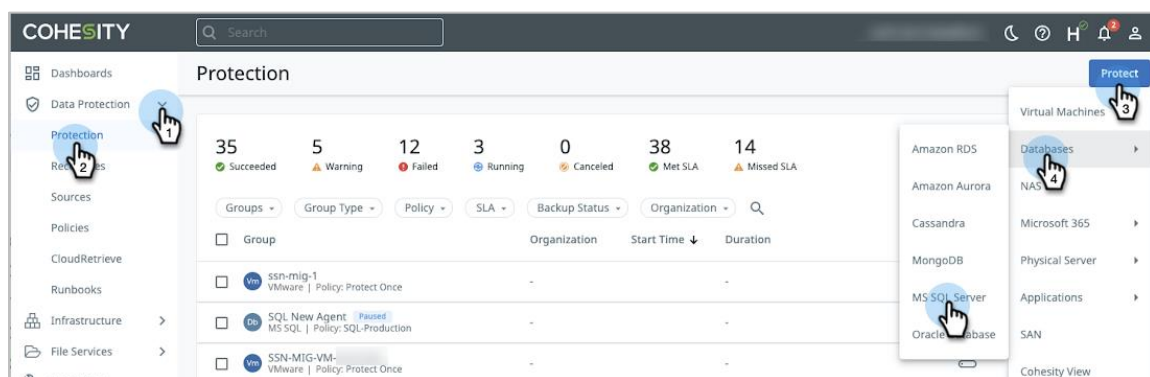
Automation is the only way to stay ahead of the demand for backups and data management. In Cohesity, Protection Groups combine operational requirements (which objects to protect, indexing, alerts, exclusions, inclusions, etc.) with the business requirements that are defined in a Protection Policy (scheduling, retention, etc.). Multiple Protection Groups can use the same Protection Policy, but each Group can have only one Policy. For more, see [About Policies and Protection Groups](#) in the online Help.

Automate your SQL database backups by building a Protection Group and assigning the SQL host you [registered](#) earlier and applying the Protection Policy that meets your business requirements.

NOTE: Because SQL Server configurations can be implemented across multiple hosts, or can be part of a Windows cluster, it's important that you identify all the SQL Server hosts so that they can be included in your Protection Group.

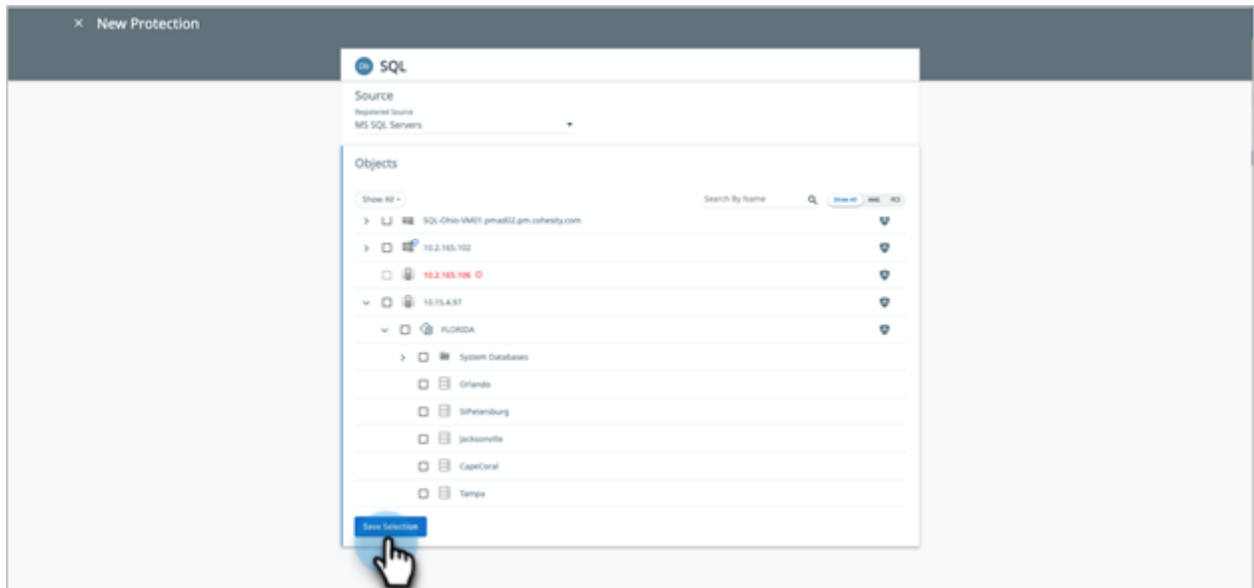
To create a Protection Group:

1. Log in to Cohesity and navigate to **Data Protection > Protection**. Then click **Protect > Databases > MS SQL Server**.



TIP: You can add or remove more SQL sources to the Protection Group later if you like. In this way, you can build onto the Protection Group to manage all your SQL servers.

2. In the **New Protection** form, under **Source**, select the SQL hosts you registered earlier, make your selections, and click **Save Selection**.



3. In the same form, enter a Protection Group **Name** and select the appropriate **Policy**. Under **Settings**, select the **Storage Domain** and set the **Start Time**.

The screenshot shows the 'New Protection' configuration form. The form is divided into several sections:

- Source:** Registered Source: MS SQL Servers.
- Objects:** 2 MS SQL Objects, 2 Manually Protected.
- Protection Group:** Radio buttons for 'New Group' (selected) and 'Existing Group'. The name field contains 'SQL-Florida-Cistr Production Data'.
- Policy:** 'SQL Production' is selected. A list of policies is shown: 'Backup' (Policy: 1 hour, Retain: 4 days), 'Retry Options' (No retries enabled, 1000 retries, 12h), and 'Log Backup (Databases)' (Policy: 10 minutes, Retain: 4 days).
- Settings:** 'Storage Domain' is set to 'DefaultStorageDomain' (Deduplication: online, Compression: online). 'Start Time' is set to 'Not Applicable'.

Hand icons indicate the fields being interacted with: the Protection Group name, the Policy selection, the Storage Domain dropdown, and the Start Time field.

TIPS:

- Give your Protection Group a descriptive name that identifies the kind of data being protected and how it is managed. This will help you identify and manage your SQL backups as your environment grows. Use descriptors like: production (PROD), critical, infrastructure (INFRA), financial, sales, primary, secondary, and employees (EMP).

For example:

- Production_Sales
 - DataCenter_Dallas_Production
 - Critical_Infrastructure
 - Production_ReplicatedTo_DRsite
 - Archive_LongRetention
 - Development_User_Data
- Once you set the Policy for a Protection Group, all the sources assigned to that Protection Group will be conveniently managed the same way.
 - To create a custom Protection Policy to meet specific scheduling, retry options, log backup, replication, and archiving needs, see [Create or Edit a Standard Policy](#) in the online Help.
 - For maximum space savings and security, choose a Storage Domain with compression, deduplication, and encryption enabled. For details, see [Create or Edit Storage Domains](#) in the online Help.

4. In the same form, configure any **MS SQL Settings** and **Additional Settings** that you need to and then click **Protect**. For details on the **Additional Settings**, see [Protection](#) in the online Help to decide which type of backup you need.

The screenshot shows the 'New Protection' configuration window for a SQL database. The window is titled 'New Protection' and has a close button (X) in the top left corner. The main content area is divided into two sections: 'MS SQL Settings' and 'Additional Settings'. The 'MS SQL Settings' section includes the following fields:

Start Time	Not Applicable
MS SQL Settings	File-based
Backup Type	File-based
Make Full Backups Copy-only	Off
Databases to Backup	All user and system databases. Use server preferences for AAG databases.

The 'Additional Settings' section includes the following fields:

Additional Settings	Never
End Date	Never
QoS Policy	Backup HDD
Cancel Runs at Quiet Time Start	No
Alerts	Alert On: Failure
Priority	Medium
Pre & Post Scripts	None
SLA	Full: 120 minutes Incremental: 60 minutes
Pause Future Runs	No
Description	None

At the bottom of the window, there are two buttons: 'Protect' and 'Cancel'. The 'Protect' button is highlighted with a mouse cursor.

Your new SQL Protection Group is now active and running, and appears on the **Protection** page. For more on optimizing your protection, see [Cohesity MS SQL Best Practices](#) in the online Help.

Now that you have created a Protection Group for your SQL Server databases, you can add other SQL sources, or change the Protection Policy and settings. This way, all your SQL sources in this Protection Group will be managed the same way. For example, to replicate all the backups to an off-site target, simply add replication to the Policy that is assigned to this Protection Group.

TIP: In larger environments, build two or three different Protection Groups with different configurations to manage several SQL Server hosts. This helps keep your data management simpler even as your environment grows.

IMPORTANT: After a Windows failover event, Cohesity will recognize the SQL instance on the active node. The next run of the Cohesity Protection Job will take a full backup of the database and a new log chain will start.

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.

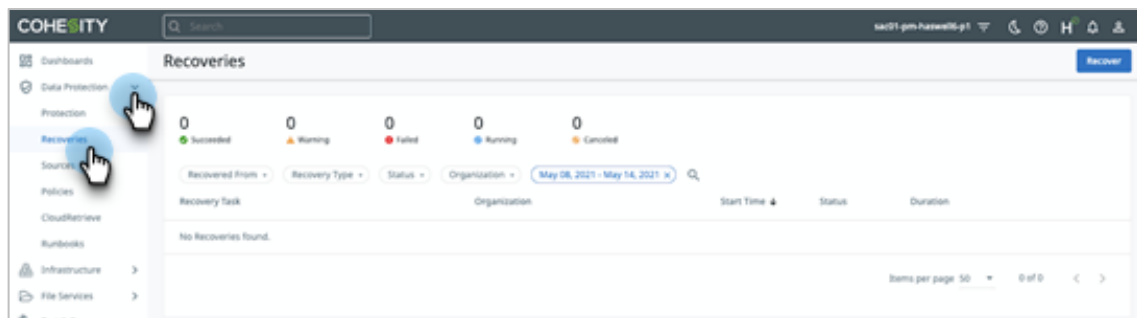
TIP: A good backup plan always includes a combination of full, differential, and log backups.

Recover SQL Database

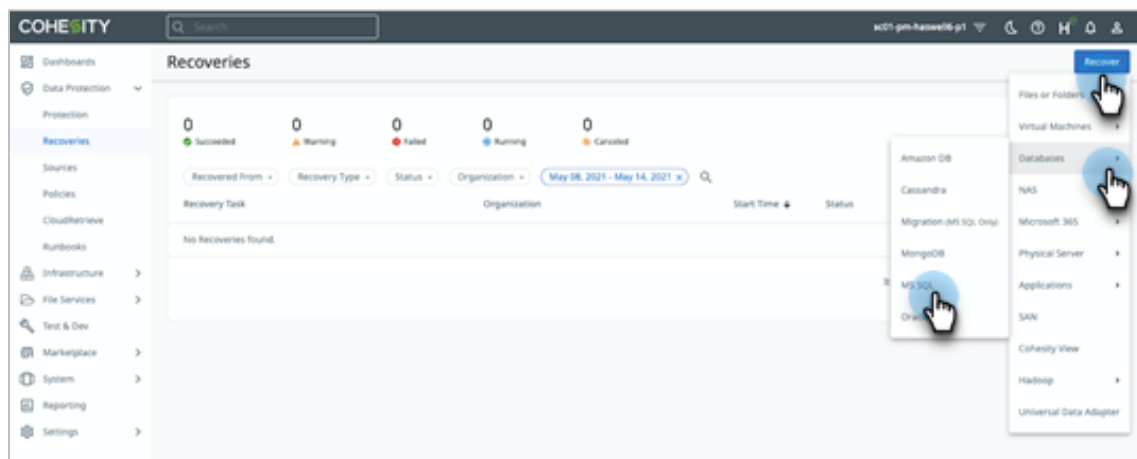
For the DBA, restoring the database starts with a restore of the full backup and then a differential and/or hundreds of individual log files. This is a slow process and very time-consuming. In the same way a SQL DBA manually performs the restore sequence, Cohesity knows to look for the full backup, the differential (if any) in between the full and log backups, and then the logs necessary to walk the restore forward to a specific point in time. Cohesity makes the SQL database restore process easy.

To recover SQL Server databases:

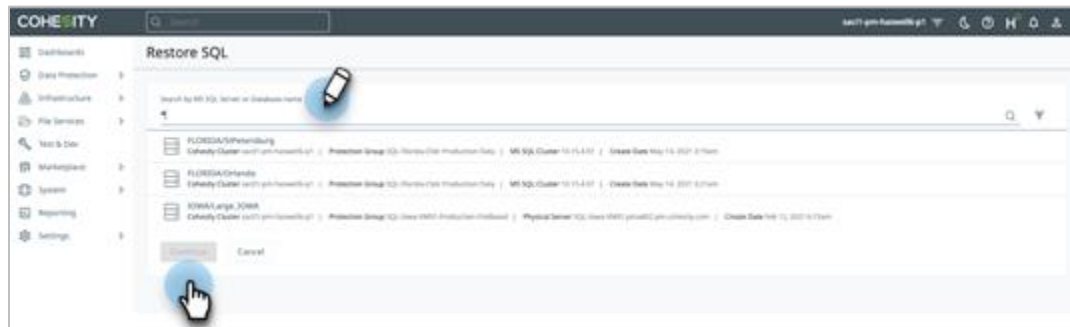
1. Log in to Cohesity and navigate to **Data Protection > Recoveries**.



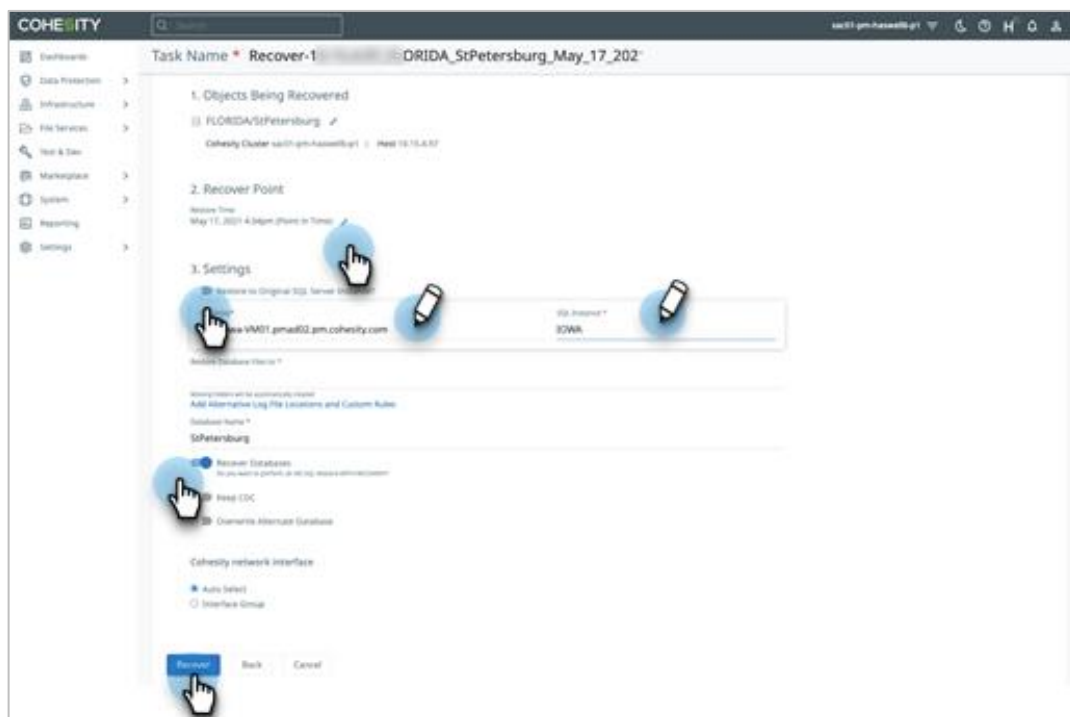
2. On the **Recoveries** page, click **Recover > Databases > MS SQL**.



- In the **Restore SQL** form, enter a search term to locate your database backup. When you locate it, click **Continue**.



- In the **Task Name** form, if you need a different backup (point in time), click the **Recover Point** to select an earlier backup. Under **Settings**, select the **Restore to Original SQL Server Instance** and enter the **SQL Host** and **SQL Instance** names where you need to restore the backup. If you need a different backup (point in time), click the backup listed under **Recover Point** to select an earlier backup. Click **Recover**.



For more information on the other SQL DB recovery options, see [Restore MS SQL Databases](#) in the online Help.

Your recovery task launches and appears on the **Recoveries** page.

Upgrade Your Disaster Recovery Preparedness

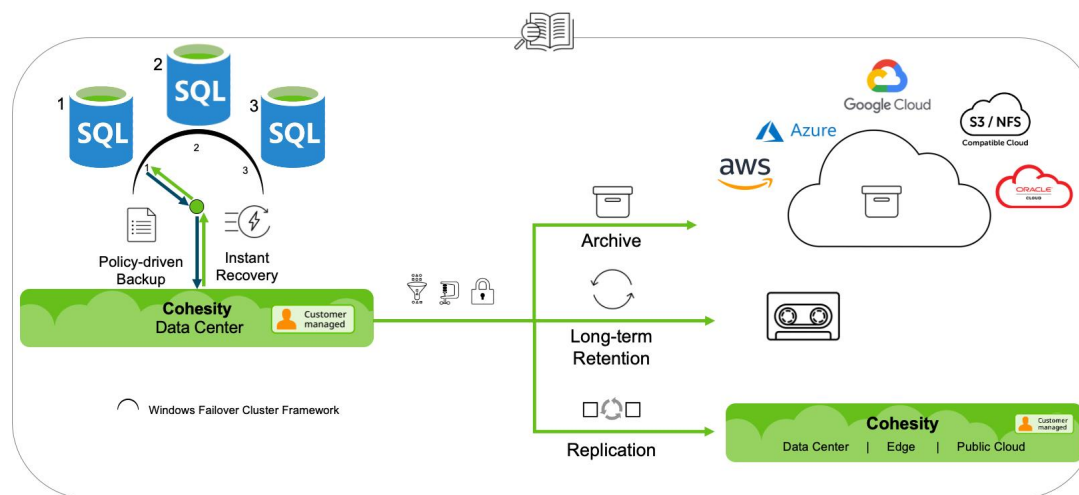
Disaster Recovery (DR) and business continuity are closely related plans designed to proactively protect a business's infrastructure and data. Taking SQL backups is only one part of protecting your business; you must also protect the data from corruption and catastrophic disaster. You can achieve this by keeping a series of backups, replicating those backups off-site, and archiving them under a long-term retention plan.

Cohesity gives you the foundation to build a DR plan to protect your business:

- **Capture and Store.** Protect your SQL databases from loss and corruption with regularly scheduled backups.
- **Geo-Redundancy.** Replicate your SQL backups to an off-site location to protect from catastrophic loss and disaster.
- **Cost-Effective Archival.** Archive your SQL backups to the cloud and store them on lower-cost storage tiers for long-term retention.

Use a Protection Group to schedule regular SQL backups and assign a Protection Policy to include archiving and replicating those backups for long-term retention and disaster recovery.

Figure 3: SQL Backups in Cohesity are Available to Replicate and Archive



Take Local Snapshots

Protect your SQL backups over time by maintaining a series of local Cohesity snapshots.

Use Cohesity Protection Groups to schedule and automate SQL backup management.

Replicate Backups Off-Site

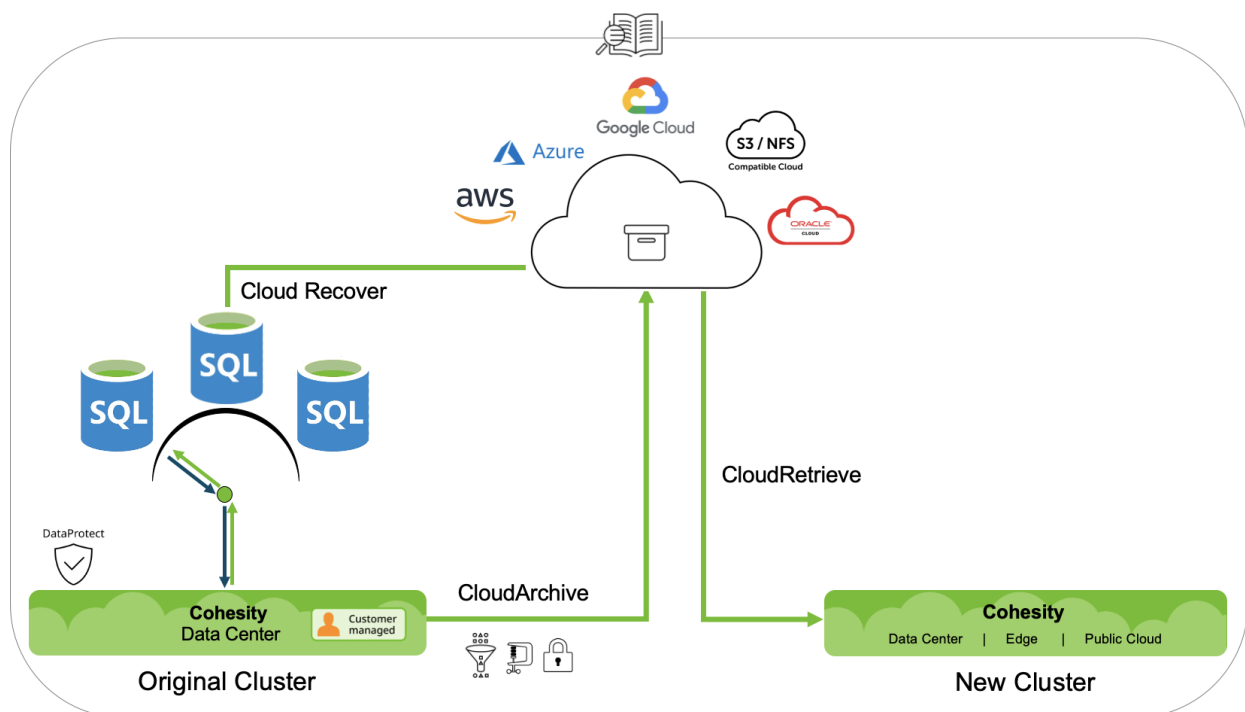
Protect your entire set of SQL backups from catastrophic loss by replicating your SQL backups to an off-site location. Choose a Protection Policy for your Protection Group that automatically copies the SQL backups to a second, off-site Cohesity cluster. By default, deduplication and compression are enabled for replication, and Cohesity sends *only the changed data* over the network, producing a significant reduction in network traffic and cost.

Archive Backups to the Cloud

Archive your SQL backups to the cloud as a way to address long-term data retention requirements and simultaneously lower the cost of storage. Cohesity provides a policy-based method to archive to public clouds (AWS, Azure, and GCP), as well as to any S3-compatible storage.

With Cohesity CloudArchive, Cloud Recover, and CloudRetrieve, your SQL backups are available for recovery to their original Cohesity cluster or onto a different Cohesity cluster, for geo-redundancy and disaster recovery.

Figure 4: Cohesity CloudArchive, Cloud Recover, and CloudRetrieve Provide Disaster Recovery



Best Practices for Cohesity SQL Server Protection

Configuring the right Cohesity settings dramatically improves the performance of your backups, and the efficiency of your storage and archives. Manage your backups by choosing the optimal settings for deduplication, compression, and encryption.

- **Use inline deduplication.** Deduplication (enabled by default) prevents duplicate blocks of repeated data from being stored, dramatically reducing your storage consumption.

With inline deduplication, the process occurs as Cohesity is saving the blocks, instead of waiting until *after* Cohesity has written the data to its Storage Domain. Cohesity recommends you use deduplication and wherever possible, enable inline deduplication.

- **Use inline compression.** Compressing your data significantly reduces the space needed to store your backups and frees up space for more backups and other important data.

With inline compression, the process occurs as Cohesity is saving the blocks, instead of waiting until after Cohesity has written the data to its Storage Domain. Both compression and inline compression are enabled by default, and Cohesity recommends you take advantage of them. For details, see [Create or Edit Storage Domains](#) in the online Help.

- **Use encryption.** When a platform governs access to data across the systems in your environment, it is crucial to protect the data it manages. Cohesity recommends you enable encryption — at rest, in flight, and in the cloud — for all your SQL Server backups. For more, see [Cohesity Security Features](#) in the online Help.
- **Keep multiple snapshots to guard against corruption.** Cohesity recommends you maintain five to seven local snapshots of your backups.

When you capture and store your backups like this, you protect your data from corruption over time. By taking and maintaining several snapshots, you are in a position to recover data from its state *prior* to being corrupted. Snapshots are efficient because they capture just the changed blocks of data, and then use deduplication and compression.

Cohesity recommends protecting all your SQL Servers with regularly scheduled backups, and then in turn moving some of those backups off-site and archiving them under a long-term retention plan for disaster recovery.

- **Validate the backups.** Cohesity recommends a periodic restore of a SQL Server database in a test environment from its backup. This is an important step in the overall backup strategy because it tests, verifies, and validates the integrity of the backup. Restore a sample from the snapshot set to a non-production server and evaluate it. In addition to confirming that the right data is backed up properly, this practice also ensures that you have already validated your method of restoring objects *before* a critical event necessitates it.
- **Verify logging and auditing operations.** It is very important to log and audit changes on a SQL Server. Cohesity logs its recovery process. In Cohesity, navigate to **Data Protection > Recoveries** and click into your SQL database recovery task to view detailed logs.

- **Shelter in the cloud.** Cohesity recommends archiving to the cloud for low-cost, long-term storage, and protection from regional disasters.

Cohesity provides a policy-based method to archive to public clouds (AWS, Azure, Google Cloud Platform) or any S3-compatible storage. This makes it easy to change policies, meet regulatory requirements, and retrieve your data to different geographical locations.

- **Use replication to defend against site disaster loss.** Protect your entire set of SQL Server backups from catastrophic loss by replicating them to a different geographical site. Cohesity can automatically replicate the SQL database backups stored in the Cohesity cluster to a second, off-site Cohesity cluster.

Cohesity replication always performs source-side deduplication and compression first and sends only the changed data over the network for cost-effective disaster recovery. As such, Cohesity replication is an essential part of every disaster recovery (DR) plan.

- **Be prepared before disaster hits with a DR plan.** One of the best things you can do to protect your SQL Server is to include it when you create your DR plans.

Appendix A: Terminology

There are several concepts and terms that are important to understand as you learn how to take advantage of all of Cohesity's features for SQL database protection.

- **Protection Group.** A collection of objects from your registered sources that share a recurring backup schedule of Protection Runs. Use a Protection Group to identify which SQL databases to protect. When you create a Protection Group, you associate it with a Cohesity Protection Policy.
- **Protection Policy.** A reusable collection of settings that define how and when objects are backed up, replicated, and archived.
- **Cohesity Replication.** Replication automatically makes copies of snapshots captured by Protection Runs on one Cohesity cluster and puts the copies on a second Cohesity cluster.

Appendix B: Product Documentation

Review our SQL Server product documentation for in-depth details:

- [MS SQL Requirements](#)
- [Cohesity MS SQL Best Practices](#)
- [Key Concepts](#)
- [Protect MS SQL](#)

Your Feedback

Was this document helpful? [Send us your feedback!](#)

About the Authors

Scott Lorenz is a SQL 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:

- Bart Abicht, Senior Technology Writer and Editor

Document Version History

VERSION	DATE	DOCUMENT HISTORY
1.1	July 2024	Republishing
1.0	Jan 2022	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](#).

© 2024. Cohesity, Inc. All Rights Reserved. The information supplied herein is the confidential and proprietary information of Cohesity and may only be used (a) by the intended recipients and (b) in conjunction with validly licensed Cohesity software and services. Find the terms of Cohesity licenses at www.cohesity.com/agreements.

Cohesity, the Cohesity logo, SnapTree, SpanFS, DataPlatform, DataProtect, Helios, the Helios logo, DataGovern, SiteContinuity, DataHawk, and other Cohesity marks are trademarks or registered trademarks of Cohesity, Inc. in the US and/or internationally. Other company and product names may be trademarks of the respective companies with which they are associated. This material (a) is intended to provide you information about Cohesity and our business and products; (b) was believed to be true and accurate at the time it was written, but is subject to change without notice; and (c) is provided on an "AS IS" basis. Cohesity disclaims all express or implied conditions, representations, warranties of any kind.