COHESITY

Version 1.2

July 2024

Use Cohesity DB Migration to Move MS SQL Databases

Use Cohesity's SQL Database Migration Solution to Move a Database

ABSTRACT

Go beyond data management. Use your Cohesity backups to move a SQL database. In this guide, we use Cohesity's SQL DB Migration feature to move a database from one data center to another.

Table of Contents

Introduction to Cohesity MS SQL Database Migration 3
Think About Other Use Cases 3
Know All the Details 3
Terminology 3
Real-World Example: Migrating a Live Database 4
Visualize Database Migration 5
Migrate a SQL Database 7
Map the Workflow to Success 7
Check the Prerequisites 7
Migrate the Database to a Secondary Host 7
Finalize the Database Migration 12
Your Feedback 14
About the Authors 14
Document Version History 14

Figures

Figure 1: Manual Database Migration Steps 5
Figure 2: Cohesity Database Migration Process 6

Tables

Table 1: Migration Concepts 4

Introduction to Cohesity MS SQL Database Migration

Cohesity DB Migration makes it easy to move a database.

Partnering Cohesity with Microsoft SQL Server® has three major phases:

- Discovery
- Deployment Considerations
- Solution Procedures

You are at the Solution Procedures phase of pairing Cohesity with Microsoft SQL Server.

This guide focuses on the process of moving a SQL database to an existing SQL instance, and is aimed at IT and database administrators who manage data protection for Microsoft SQL Servers.

NOTE: The Cohesity DB Migration feature is enabled for version 6.5.0 and above.

Think About Other Use Cases

The use case in this guide is to introduce a stand-alone database to an existing SQL Instance.

There are other implications for this process. Migrating a database to a second SQL server. They are:

- Moving a large database from one instance to another for load balancing or upgrading a hardware upgrade.
- Pre-staging databases at a different regional DR site.

Know All the Details

There is a lot to know about Microsoft SQL technology and how it is applied in the field. Cohesity is uniquely suited to work with MS SQL Server.

Find out more by reading these articles:

- Protect SQL Server with Cohesity — Deployment Configurations Guide
- SQL Server Instance Prerequisites and Restrictions

Terminology

There are several concepts and terms that are important to understand as you learn about the Microsoft SQL Server Backup features in Cohesity.

Table 1: Migration Concepts

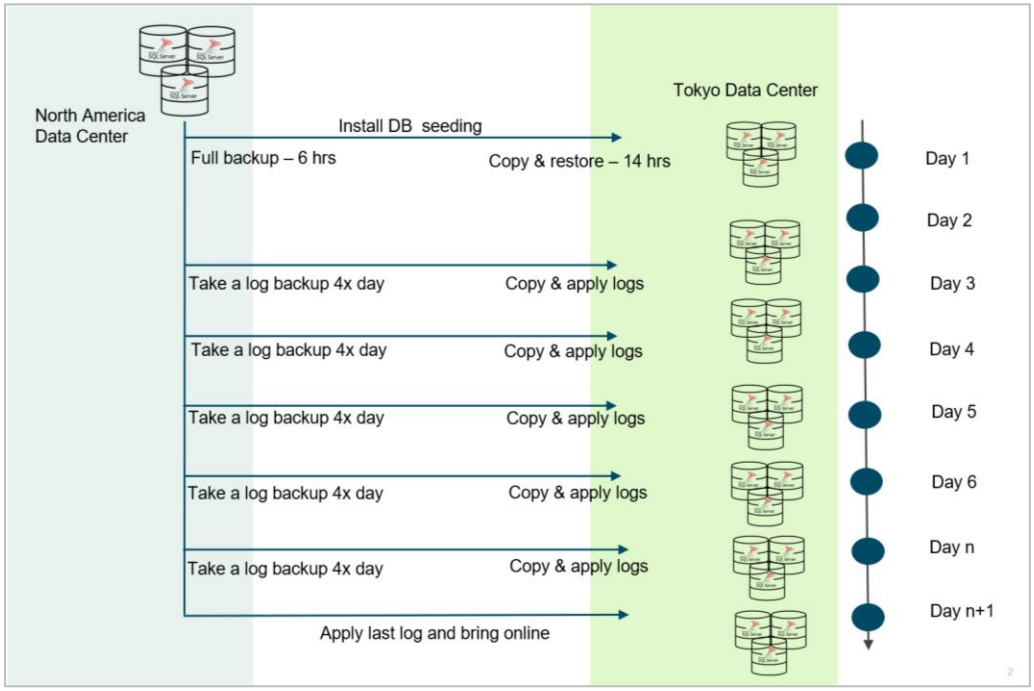
TERM	DEFINITION
Protection Group	A Cohesity Protection Group is a backup job that runs repeatedly, based on an associated Protection Policy, to back up data from a source and store it on the Cohesity cluster. A Protection Group can also store the data on External Targets that can be other clusters, cloud object storage, or tape.
Microsoft DB Migration (Technique)	The process of moving an MS SQL Server database from one SQL Server instance to a different SQL Server instance without data loss. This can be performed manually or in some automated form.
Cohesity DB Migration	The Cohesity implementation of the DB Migration technique. Specifically, moving an initial copy of the database and then sequentially applying incremental backups.

Real-World Example: Migrating a Live Database

While there are several ways to migrate a SQL database, doing so has traditionally been a very manual process.

In most cases, database administrators choose their migration method based on the size of the database and the amount of down time the customer can accept. If the database is small or the customer can absorb significant down time, you can take a simple backup and copy it over. In most enterprise data centers, however, the databases are large and even small amounts of down time are costly. In these cases, as Figure 1 below illustrates, migrating a database involves taking an initial backup, copying it to the target server, and then applying log files until the cut-over date, when the customer starts using the migrated database again.

Figure 1: Manual Database Migration Steps



Moving a live database requires careful planning, scheduling, and timing. And when copying large files from *Data Center A* over *to Data Center B*, there is always a possibility of file corruption.

Visualize Database Migration

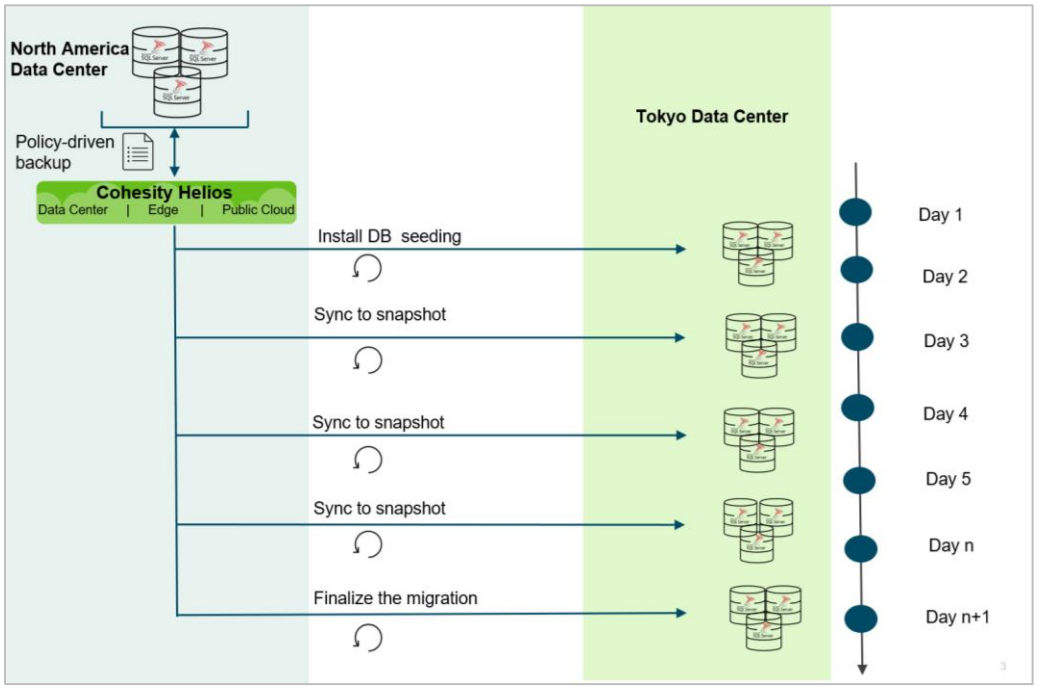
To continue with our example, the DB Migration feature is designed to migrate a copy of the database from *Data Center A* (DC-A) over to *Data Center B* (DC-B) efficiently and reliably.

Cohesity's DB Migration accomplishes this in one workflow:

1. Initial seeding (copy) of the database files from a snapshot on Cohesity over to the SQL instance in DC-B.
2. A series of snapshot synchronizations keep the copy in DC-B up-to-date with the most recent incremental snapshot.
3. Finalize the migration.
   - Synchronize any outstanding changes with the target.
   - Attach the database to the SQL instance.

With Cohesity, the entire process is defined and automated in a Protection Group and the Protection Policy that is applied to that Group, as illustrated by the example in Figure 2 below.

Figure 2: Cohesity Database Migration Process



Cohesity DB Migration does all the heavy lifting. It decreases the work needed in a traditional migration to a simple and reliable process.

Migrate a SQL Database

In the following sections, we will seed a database (which we'll call "NewBraunfels") over to a second SQL instance using Cohesity's DB Migration feature.

Map the Workflow to Success

Before jumping into the process, it's important to know what all the stages will be.

To migrate the database:

1. Check the prerequisites.
2. Migrate the database to a host replica.

IMPORTANT: Cohesity *strongly* recommends you thoroughly test this process in a test environment before applying it in a production environment.

Check the Prerequisites

Most of these prerequisites are already established, but it's always good to check them anyway.

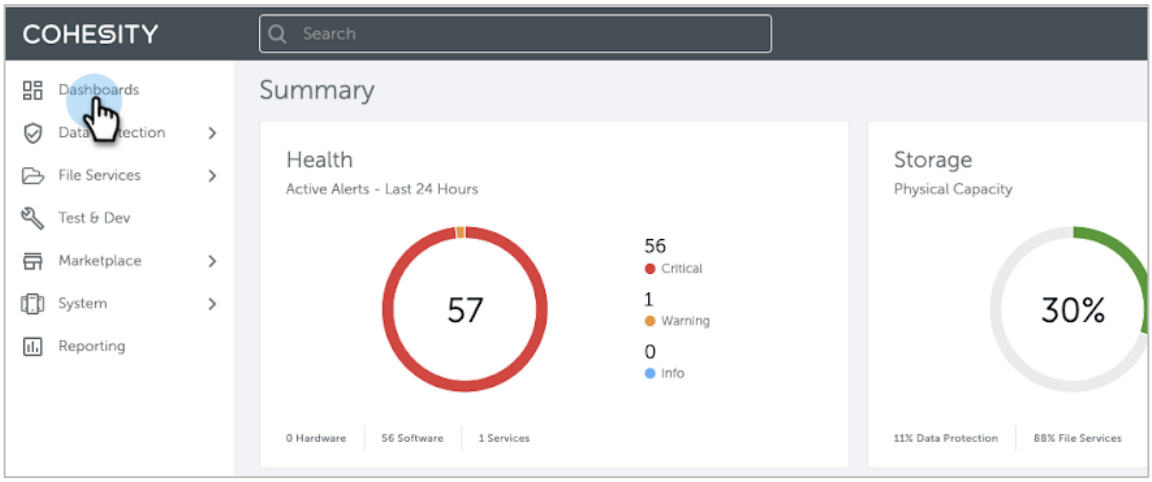
- All the SQL instances and the Cohesity cluster are joined in an Active Directory Domain.
- All the versions of SQL instances are compatible.
- The database you are working with is a qualified database.
- The database has a recent file-based incremental snapshot under a Cohesity Protection Group.

Migrate the Database to a Secondary Host

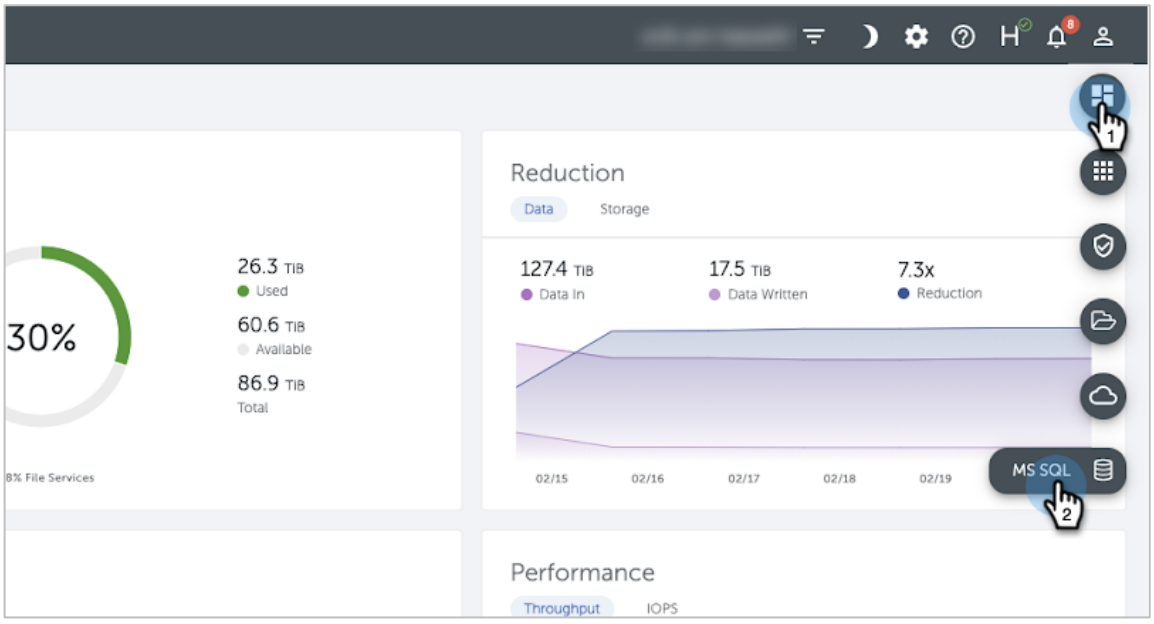
The next step is to migrate the database to get an initial copy onto the secondary host.

To migrate the database:

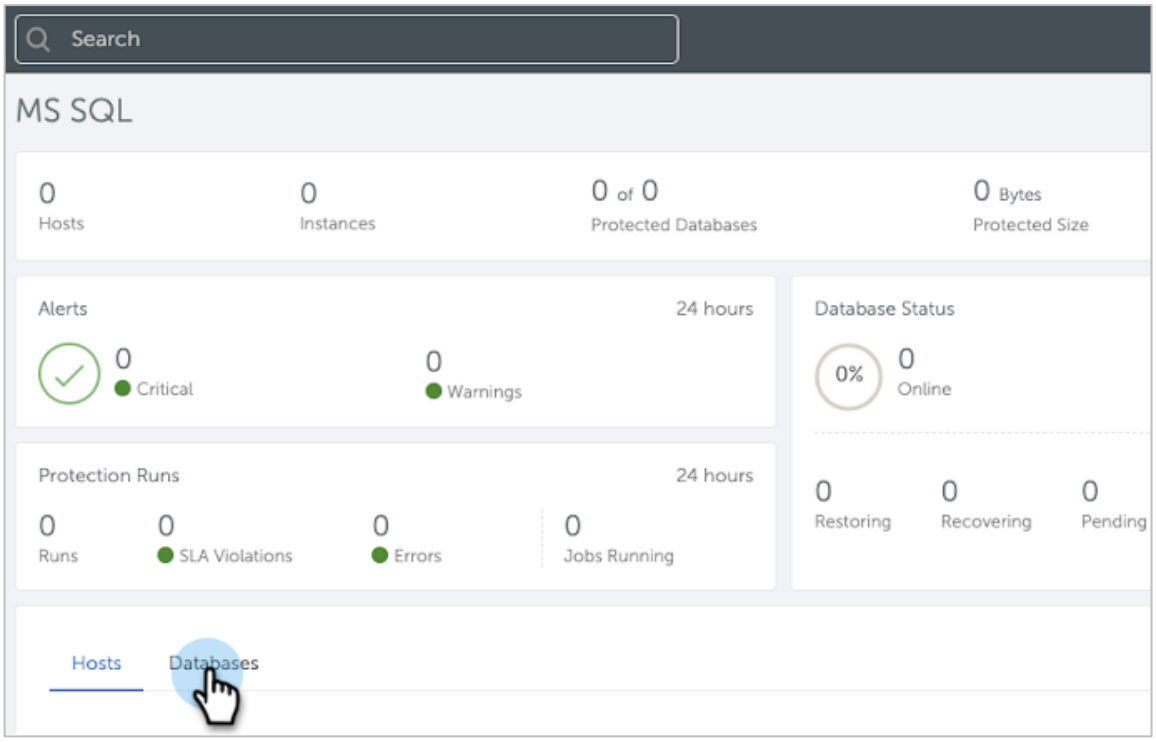
1. Log in to Cohesity.
2. Click **Dashboards**.



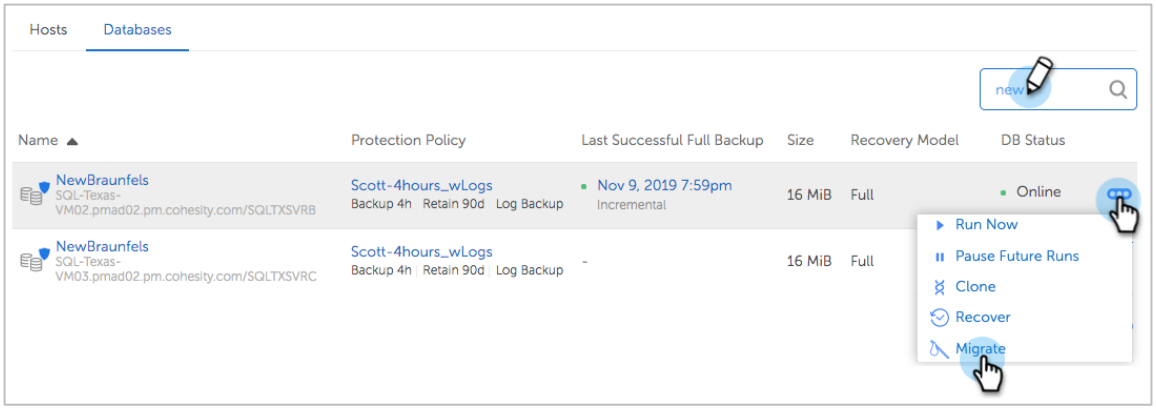
3. Click the **Dashboards** button and select **MS SQL**.



4. Click the **Databases** tab.

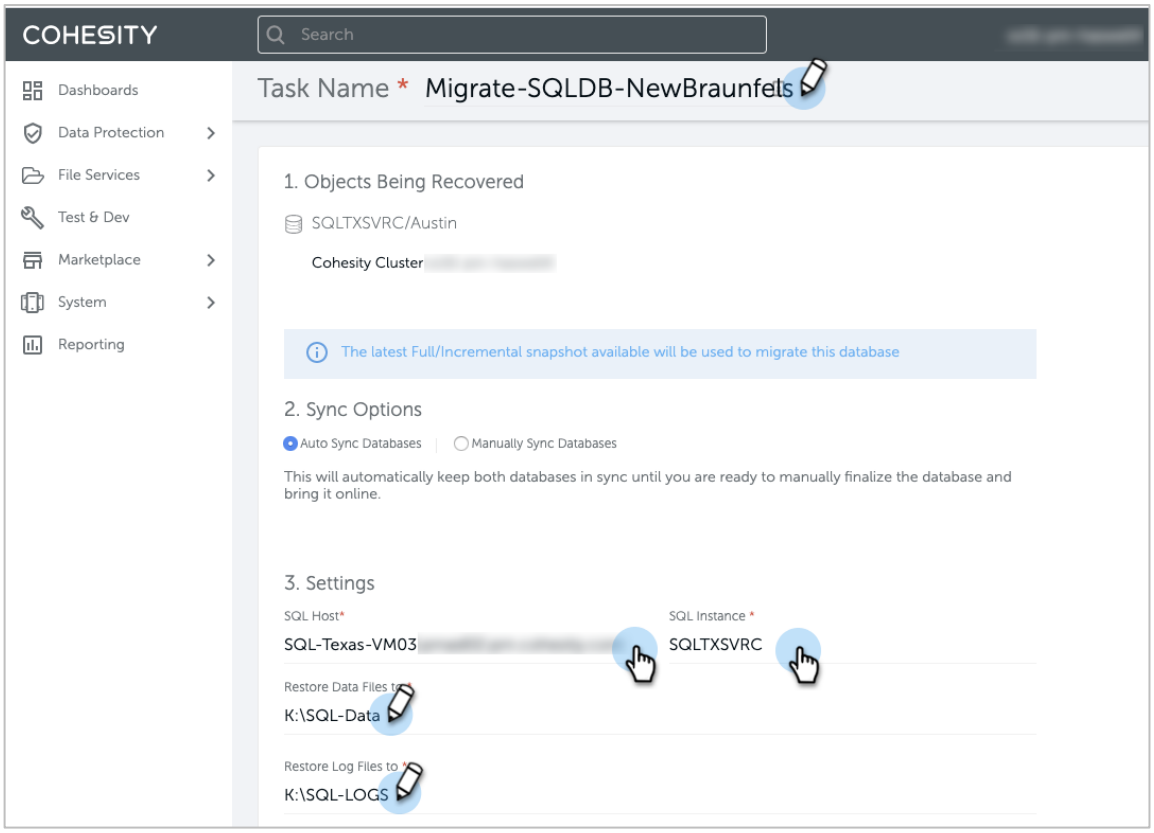


5. In the **Search** box, enter the database name, then click the action menu on the right, and select **Migrate**.



NOTE: Cohesity recommends using the most recent file-based incremental snapshot. Point-in-time (PIT) migration is part of a future enhancement for this feature. Currently, only incremental and full backups are used to migrate a database.

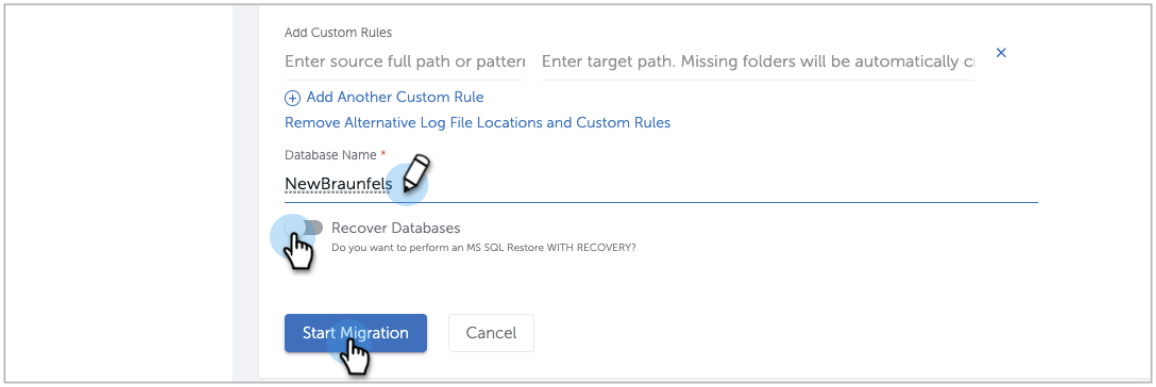
6. Enter the migration **Task Name**, select a replica node as the **SQL Host** (SQL-TEXAS-VM03 in our example), and enter the locations for **Restore Data Files to** and **Restore Log Files to**.



NOTE: The **Auto Sync Databases** feature automatically synchronizes to the latest incremental or full snapshot. You can enable or disable the feature even after creating the migration task.

Auto sync checks for a new snapshot every two minutes and disables itself after three synchronization failures or a Finalize failure. Auto sync is retryable on failure.

7. In the same form, enter a **Database Name**, enable **Recover Databases**, and click **Start Migration**.



CAUTION: Be careful to *enable* **Recover Databases** in this form so that the database will be in an "online" state when the migration is complete.

If the rate of change is high in your database, Cohesity recommends that you synchronize the migrated database frequently.

Cohesity's DB Migration feature uses a robust communication mechanism that ensures that interruptions in connectivity during the migration process do not necessitate restarting the migration. If something should happen, the migration will pick up where it left off.

Finalize the Database Migration

The goal is to have a copy of NewBraunfels on the target SQL server.

To finalize the database migration:

1. **Take an incremental backup**.

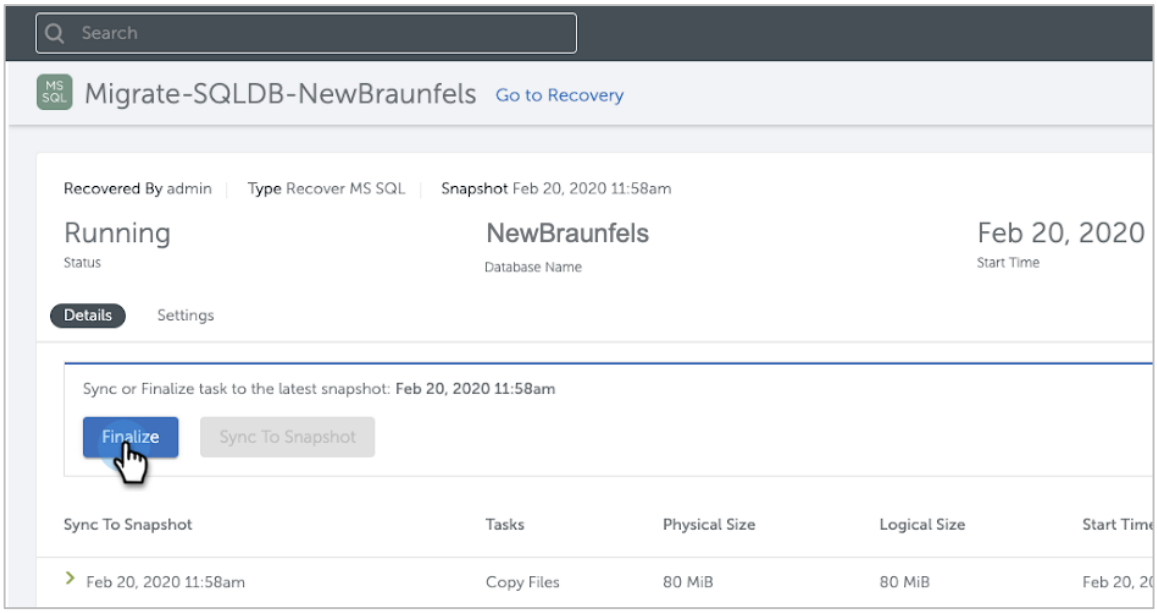
From the Cohesity Protection Group, take an incremental backup of the NewBraunfels database. This is to ensure that the secondary database will now be as similar as a copy to the primary database as possible.

2. **Synchronize the snapshot**.

Synchronizing the snapshot will copy over any changes captured by the primary database's snapshot and apply them to the secondary database.

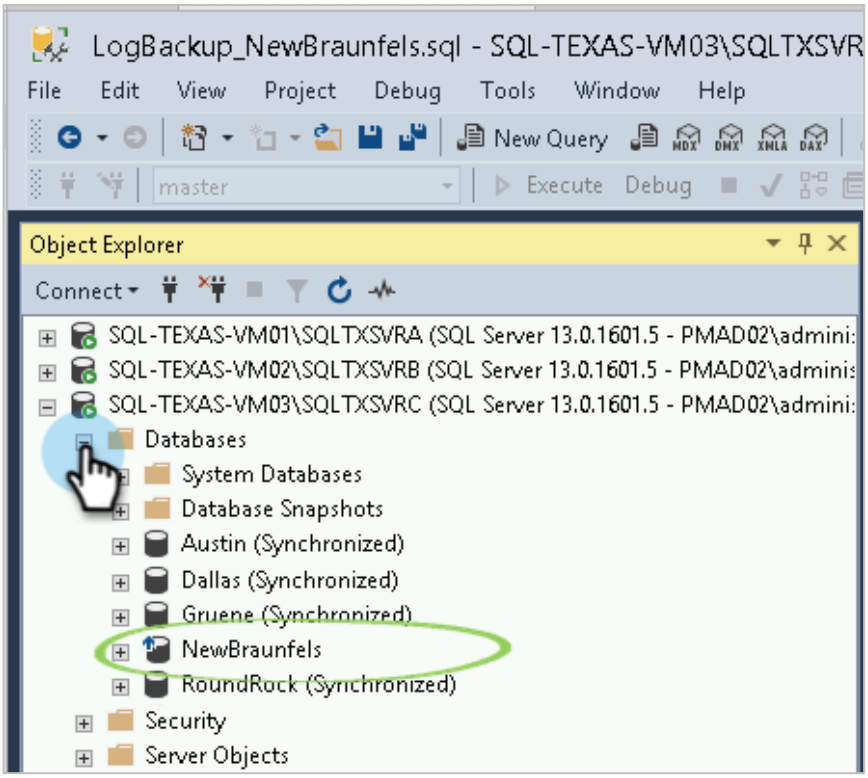
NOTE: Application transactions might still be occurring on the primary database (NewBraunfels, in our example). Consider briefly pausing the application before finalizing the migration.

3. **Finish the migration**. Navigate to the migration task and click **Finalize**.



This copies the latest changes, applies them to the migrated database, and attaches the database to the secondary SQL Instance.

In your SQL Server Management Studio (SSMS), the migrated database should now appear on the TEXAS 03 secondary replica.


LogBackup_NewBraunfels.sql - SQL-TEXAS-VM03\SQLTXSVR
File Edit View Project Debug Tools Window Help
New Query
master
Execute Debug
Object Explorer
Connect
SQL-TEXAS-VM01\SQLTXSVRA (SQL Server 13.0.1601.5 - PMAD02\admini
SQL-TEXAS-VM02\SQLTXSVRB (SQL Server 13.0.1601.5 - PMAD02\admini
SQL-TEXAS-VM03\SQLTXSVRC (SQL Server 13.0.1601.5 - PMAD02\admini
Databases
System Databases
Database Snapshots
Austin (Synchronized)
Dallas (Synchronized)
Gruene (Synchronized)
NewBraunfels
RoundRock (Synchronized)
Security
Server Objects

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, and data protection with Enterprise and Cloud Storage. Scott has over twenty-six years as an enterprise level SQL DBA in a SaaS production environment.

Other essential contributors included:

- Bart Abicht, Senior Technology Writer and Editor at Cohesity
- Freddy Grahn, Senior Technical Field Enablement Engineer

Document Version History

VERSION	DATE	DOCUMENT HISTORY
1.2	July 2024	Republishing
1.1	September 2021	Rebranding updates
1.0	April 2020	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.