COHESITY

Version 1.0

April 2024

Protect PostgreSQL with Cohesity

Cohesity Solution for Backup and Restore of PostgreSQL Databases with Best Practices

ABSTRACT

This document provides an overview of Cohesity Database Protection features for PostgreSQL, general workflows, and options, along with best practices.

Table of Contents

Postgres Protection Using Cohesity Database Plugin 4
Guide Overview 4
Use Cases 4
Technical Considerations 5
How Cohesity Works with Postgres Databases 5
Cohesity Differentiators 6
Cohesity Features 6
Backup and Restore Features for Postgres 6
Deployment Steps 7
Install Adapter and Plugin 8
Download and Install the Linux Physical Adapter 8
Field Notes 9
Download and Install the Postgres Plugin 9
Cohesity Database Source Registration 10
Register Database as Source 10
Postgres Database Protection 14
Create a Protection Group 14
Retention for Backups 16
Postgres Database Restore 18
Restore a Specific Backup 18
Postgres Troubleshooting 21
Your Feedback 22
About the Authors 22
Document Version History 22

Figures

Figure 1: How Cohesity Works with Postgres Databases 5
Figure 2: Adapter and Plugin Installation..... 7

Tables

Table 1: Postgres Backup and Restore Features..... 7
Table 2: Steps to install the Cohesity Physical Adapter for Linux..... 8
Table 3: Install the Postgres Plugin..... 9
Table 4: Components and Paths to Log File 21

Postgres Protection Using Cohesity Database Plugin

Postgres administrators contend with the challenges of Increasing backup duration, rapid data growth, increased storage costs, and the lack of flexibility and storage management tools in Postgres Studio.

In addition to these challenges, today's database protection must encompass more than getting a clean copy. It must address factors such as security, storage efficiency, minimized impact on production systems, automation, and scaling.

Cohesity protection for Postgres provides a solution to these challenges. It reduces the complexity of database backups and restores secure, streamlined workflows. You can protect and manage your workloads and execute available protection and recovery workflows with a single pane of glass with a few steps. Cohesity Postgres Adapter provides flexible deployment options to make Postgres backup and restore simple and secure.

Guide Overview

This guide focuses on Postgres database protection using the Cohesity Postgres Plugin. This guide is specific to Postgres x86-64. Its objectives are to provide an overview of features and options and their related recommendations and best practices.

Use Cases

You can use this workflow for Postgres protection if you want the following:

- Automated backup protection.
- Protection for your Postgres environment using an on-prem Cohesity cluster.
- Simple Full or Point-In-Time restore.
- Simple UI-based, restored to an alternate host.
- Meet your backup SLA.
- Faster and more secure backups and restore performance using RPC.
- Move away from script creation and management.
- Automated storage configuration and management.
- Reduce storage space and cost for your backups.

Technical Considerations

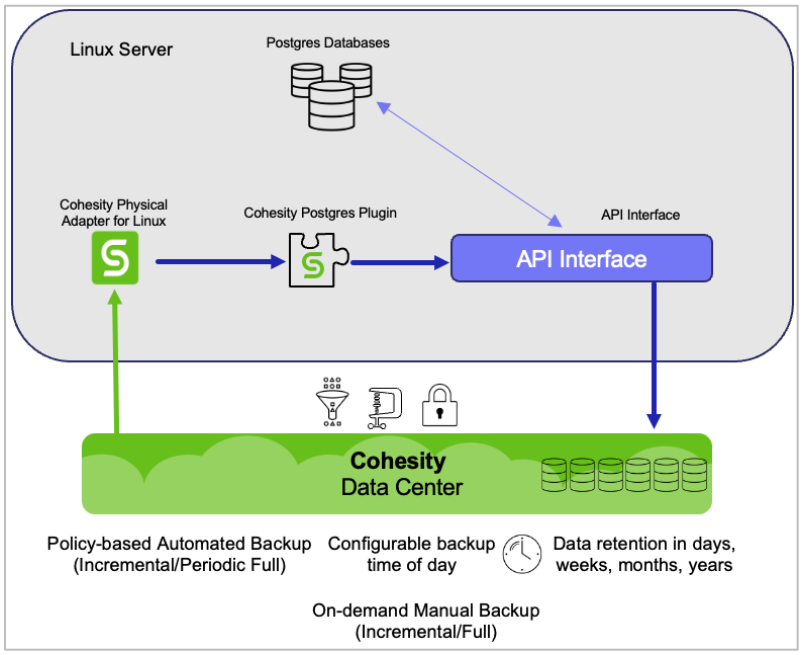
Consider the following technical aspects before you make major decisions about your solution.

- No SSH and NFS mounts are needed.
- The gRPC and secure gRPCprotocols are used for faster backups and restores.
- Backup and restores are balanced based on mounts and concurrency, as defined in the job. (no scripts to change).
- You can perform point-in-time restores.
- Only cluster-level restore is supported.
- The first backup is always a FULL backup.
- Ensure third-party agents are removed or uninstalled.
- You can protect multiple databases using a single Protection Group.
- You must install the Cohesity Adapter and the Postgres Plugin on each database host you want to protect.
- For other considerations, please refer to Plan and Prepare for Postgres Protection Considerations. Please ensure that you are familiar with the different options available.

How Cohesity Works with Postgres Databases

Cohesity's software-defined platform natively integrates with databases to provide a simple, fast, cost-effective backup and recovery solution for deployments.

Figure 1: How Cohesity Works with Postgres Databases



Cohesity uses the Postgres API to connect to the Postgres databases. Backups are transferred from the Postgres database to the Cohesity Postgres Plugin, which runs on the Postgres database server and then sends the backups to the Cohesity cluster.

Backup types for Postgres databases are Full and Differential. Log backups which Postgres automatically generate, are captured to provide point-in-time restores. Cohesity provides the option to perform a full and incremental backup. The first backup is always full, and successive backups can either be full or incremental, depending on business requirements.

Cohesity Differentiators

The following factors differentiate Cohesity from the competition:

1. Global space efficiency for your backups.
2. Postgres backups and restores are optimized for Cohesity's performance.
3. Scale-out architecture: Postgres backup and restores are load-balanced across Cohesity cluster nodes, providing redundancy, and protecting against backup failure.
4. Simplification—Cohesity automates Postgres backups.
5. Flexibility - Cohesity UI offers flexibility with options in the Postgres backup and restore workflows.
6. Ease of Use - All backup and restore workflows can be used out-of-box.
7. Optimization - Cohesity database solution uses optimized backup and restore parameters in workflows.
8. Immutability - Cohesity backups cannot be modified or tampered with and are secured from accidental loss or deletion.
9. Security - We support key-based or password-based authentication methods. gRPC and RPC are secure transport types.

Cohesity Features

The Database Connector Adapter supports Postgres databases with a range of features.

Backup and Restore Features for Postgres

Postgres Backup and Restore features can be found here: Postgres Protection.

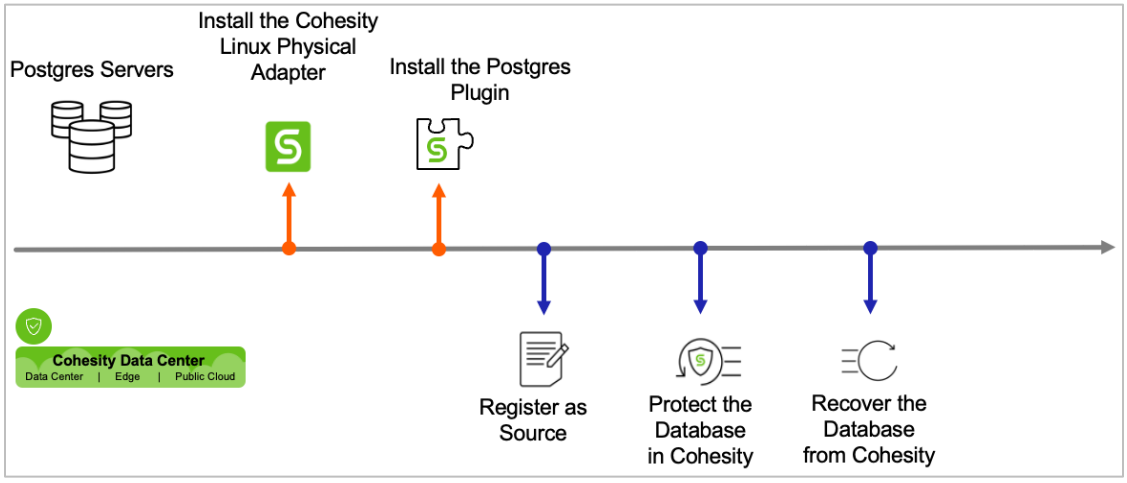
Table 1: Postgres Backup and Restore Features

Feature	Linux
Full Backup	Yes
Incremental Backup	
Log Backup	
Restore Backup to Same Host	
Restore Point In Time Same Host	
Restore Point in Time Alternate Host	
Policy or On-Demand Backups	

Deployment Steps

Install the Cohesity Physical Adapter for Linux and the Postgres Plugin on each database host you want to protect.

Figure 2: Adapter and Plugin Installation



Once you install both components on the database host, you can register the database as a source.

Install Adapter and Plugin

Downloading and installing the Linux Physical Adapter and the Postgres Plugin on a server allows you to register Postgres as a source with the Cohesity cluster.

Before you register your Postgres deployment as a source with Cohesity and protect Postgres databases, ensure the following prerequisites:

- Supported Postgres Versions
- Port Requirements
- Considerations

Download and Install the Linux Physical Adapter

Use the RPM Installer for Postgres applications.

Table 2: Steps to install the Cohesity Physical Adapter for Linux

Action	How To
Download the Agent Installer	From the download agent window, select the **RPM** and download it to the server you want to protect.
Navigate to the downloaded directory	As the root user (required) with local system privileges on that server, change the directory to the location of the installer package.
Make the installer executable	Make the installer executable, for example: chmod +x cohesity_agent_X.X_linux_x64_installer
Install the agent	rpm -ivh el-cohesity-agent-6.8.1_u6-1.x86_64.rpm
Location	• Installation directory:/home/<username>/cohesityagent or /root/cohesityagent • Log file: /home/cohesityagent/cohesityagent/logs

More details about the Cohesity Physical Adapter installation can be found in the Download and Install the Linux Agent and Linux Agent installer options.

Field Notes

Like all packages, dependencies need to exist on the Linux server. For a complete list of dependencies, see Install and Manage the Agent on Linux Servers.

Download and Install the Postgres Plugin

Download the Postgres Plugin from the Cohesity Download portal.

Table 3: Install the Postgres Plugin

Operation	Install User	Install Command
Postgres Database Connector Install	sid<adm>	`rpm -ivh cohesity-postgres-connector-7.1-1.x86_64.rpm`

Cohesity Database Source Registration

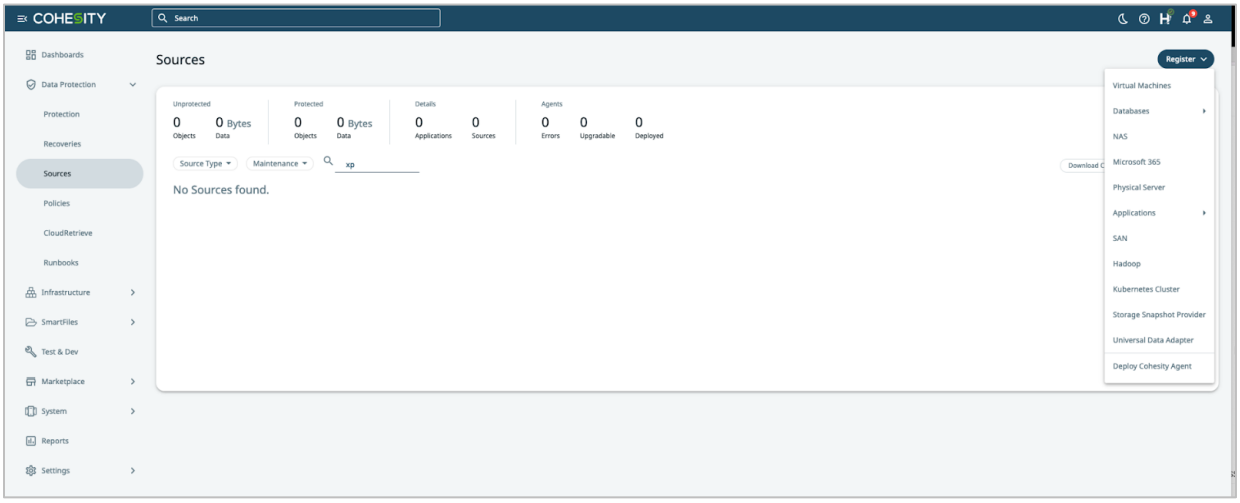
After you install the Cohesity Adapter and the Postgres Plugin, you need first to register your database as a source on Cohesity, as shown in the screenshots below.

Register Database as Source

To protect your databases with Cohesity, register it as a Cohesity source. Once it's registered in Cohesity, you can add it to a Protection Group and configure the settings for your databases.

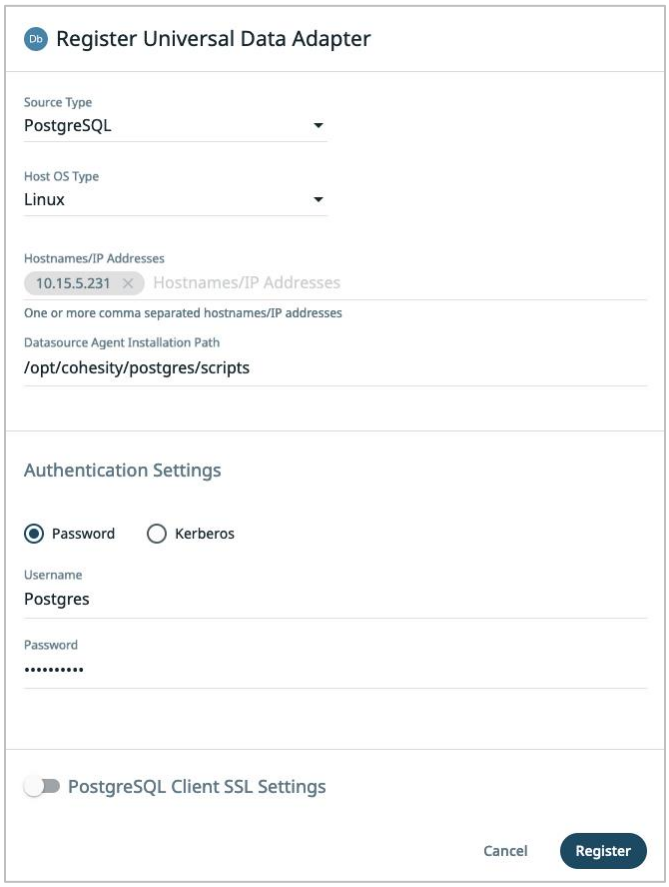
To register your database as a Register Source in Cohesity:

1. Navigate to **Sources** > **Register** > **Universal Data Adapter**.



The **Register Universal Data Adapter** form will guide you through the types of databases and their host operating systems it supports.

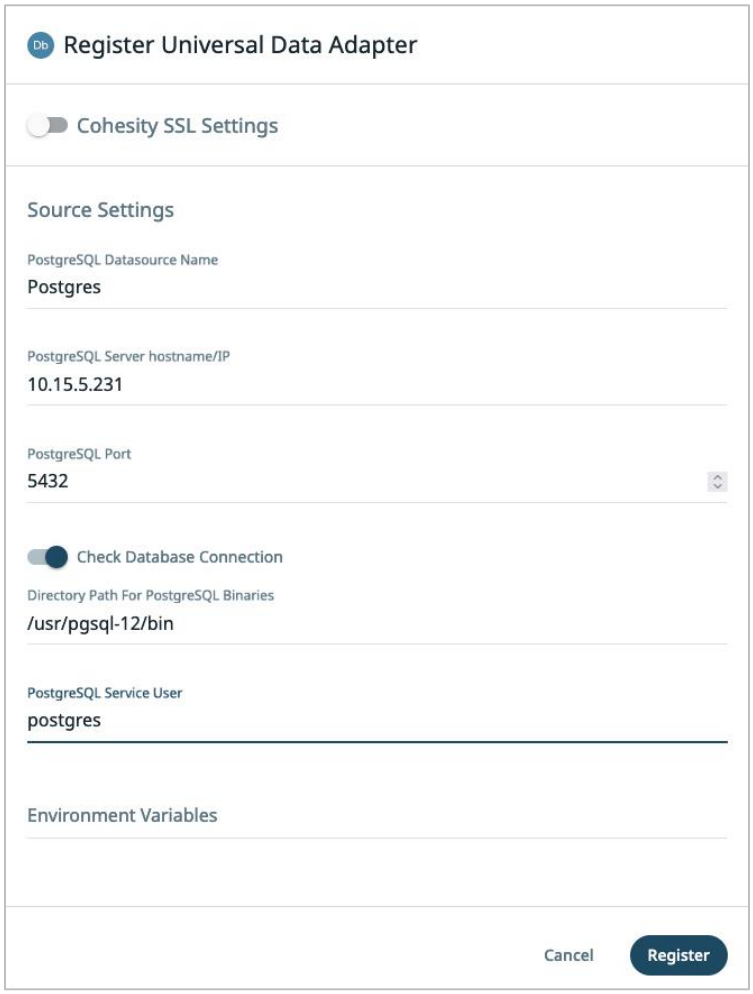
2. In the **Register Universal Data Adapter** form, choose Postgres from the **Source Type** drop-down list. Then, choose Linux from the **Host OS Type** drop-down list, which is the type of OS the host is running.



3. In the **Hostname/IP Addresses** field, enter the hostname or IP address of the node you have identified to run the PostgreSQL connector. In the case of an HA cluster, enter the hostnames or IP addresses of all the PostgreSQL nodes.
4. In the **Datasource Agent Installation Path**, enter the directory path on the PostgreSQL Linux machine on which you have installed the PostgreSQL connector scripts. Example: `/opt/cohesity/agent/uda_scripts`.
5. In the **Username** and **Password** fields, enter the Postgres username and password. The user must have super admin privileges.

> **NOTE**: Authentication is at the database level. The user must have super admin privileges.

Scroll down to continue the Source registration.

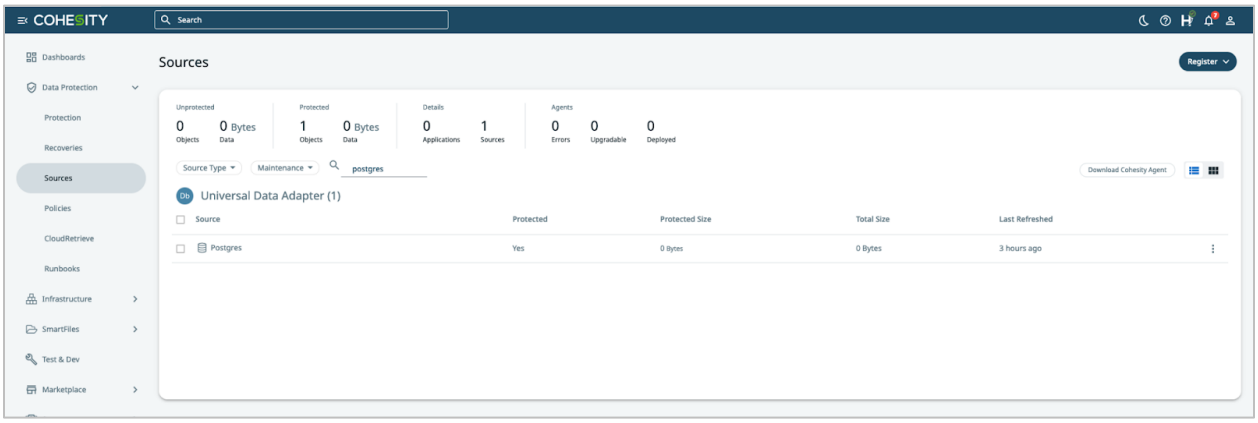


6. In **PostgreSQL Data Source Name**, enter a unique name to identify your Postgres cluster in Cohesity.
7. In the **Server hostname/IP address**, enter the address of the controlling node or the IP where PostgreSQL is listening.
8. In **PostgreSQL Port,** enter the port on which the PostgreSQL server is listening. The default is 5432.
9. **Host OS Type:** A dynamic list of OS types based on the **Source Type**.
10. In the **Directory Path for PostgreSQL Binaries** field, enter the directory path to the PostgreSQL binaries, such as/usr/pgsql-9.6/bin. By default, the value from the $PATH environment variable is taken.
11. The PostgreSQL Service User defaults to `postgres`.

    For more information about registering your Postgres host, see Register and Manage the Postgres Source.

12. Complete your source registration by clicking **Register.**

    Successful Registration



You can update, refresh, and unregister your database source from the **Sources** page.

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

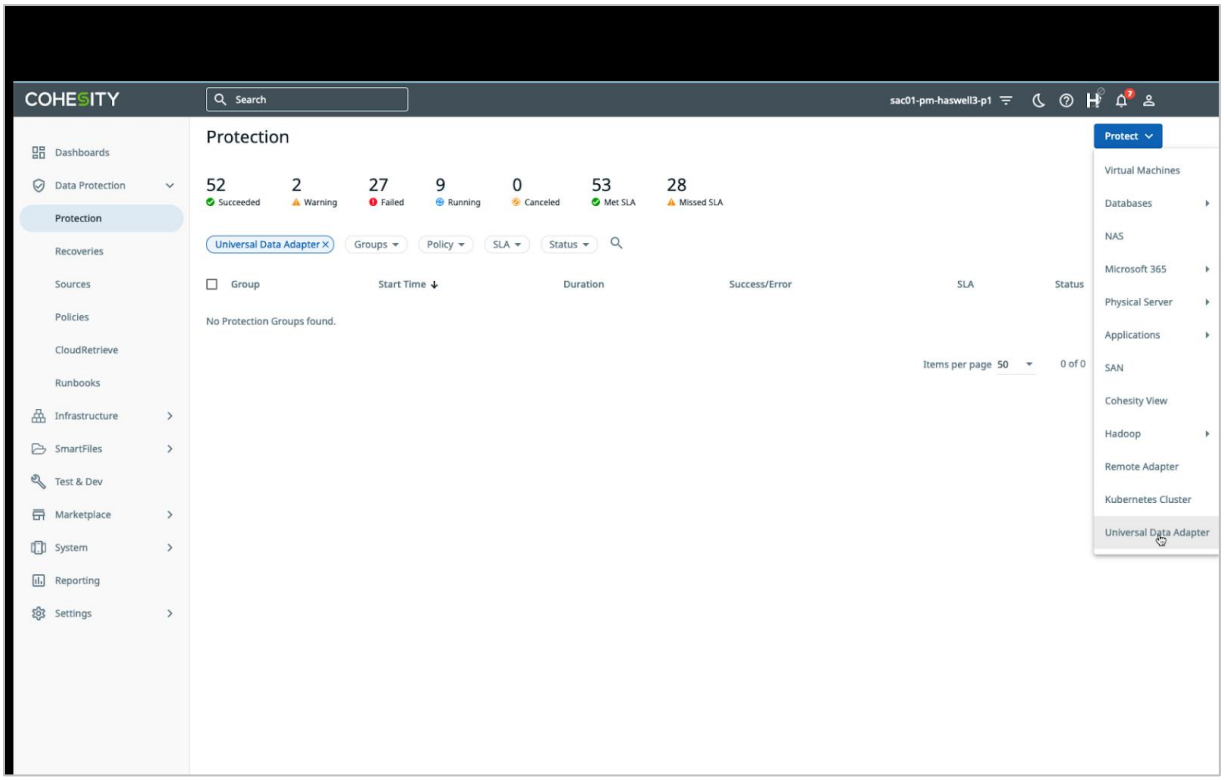
Postgres Database Protection

After registering the Postgres host as a source on Cohesity, you can start configuring your backups.

Create a Protection Group

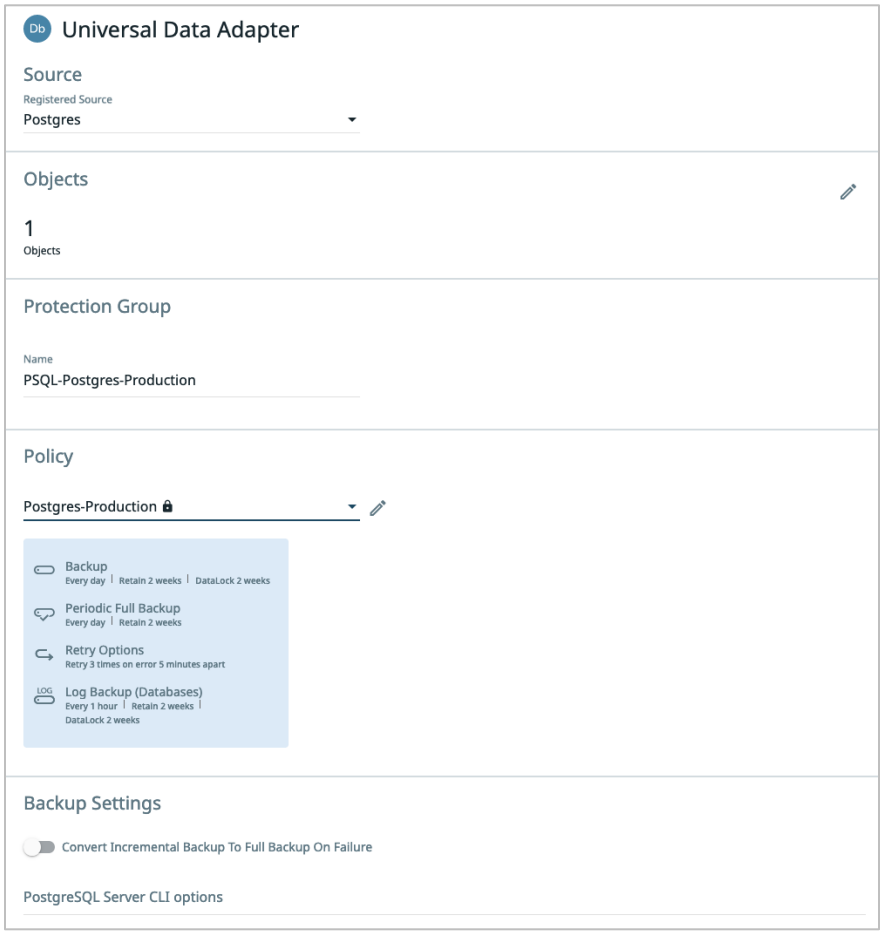
To create a Protection Group:

1. Log in to Cohesity and navigate to **Data Protection** > **Protection**. Then click **Protect** and select **Databases** > **Universal Data Adapter**.



2. In the **New Protection** form, under **Source**, select the Source you registered earlier.

New Protection Group Form

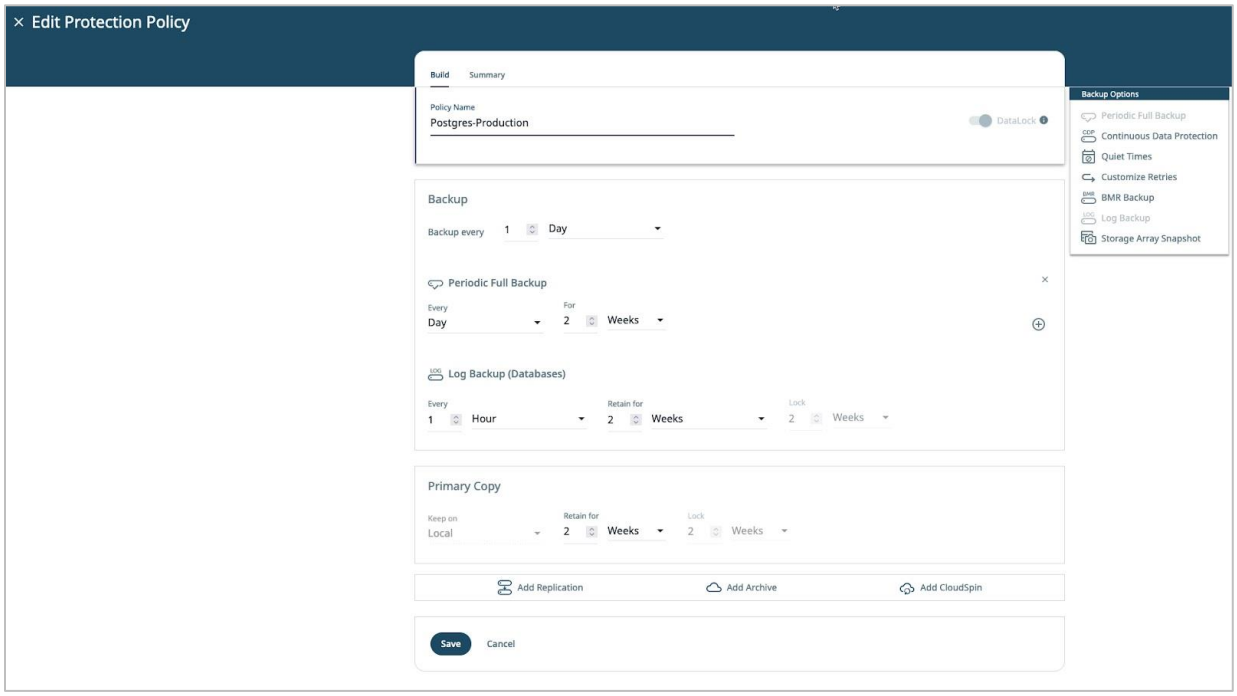


3. In this form, enter a unique Protection Group **Name.**
4. Continue by selecting a **policy.**

   You can use the default standard policies or create your custom policies. Policies save time because you do not need to enter settings repetitively.

   A policy is a reusable set of settings that define how and when objects are protected, replicated, or archived. When configuring a protection group, you select which policy to use.

Protection Policy Page



Retention for Backups

The DBA maintains a combination of backups to *restore* the database to any point in time. A good combination of backups consists of FULL, DIFFERENTIAL (in Cohesity, *incremental*), and LOG backups.

NOTE: Cohesity recommends you take a periodic FULL Backup.

Database restores must begin with a FULL database backup. Then, a DIFFERENTIAL can be applied to FULL, and finally, LOG backups can be applied in sequence to complete the database restore.

IMPORTANT: Postgres manages its log backups outside of Cohesity. It is optional to schedule Log backups for Postgres in Cohesity.

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: Database backups, differentials, and logs depend on a FULL backup to perform a database restore. Postgres databases require that you start with a FULL backup before applying its transaction logs. This means your backup retention policy must keep a FULL backup along with its LOG backups to successfully restore a database.

Simply put, a database restore requires a FULL backup to seed the database, then DIFFERENTIAL and/or LOG backups are applied to roll the database forward to the specified point in time.

We recommend retaining two sets of FULL backups with their DIFFERENTIAL.

Once you set the Policy for a Protection Group, all the databases assigned to that Protection Group will be conveniently managed in the same way.

For more information about Policy features, see Create or Edit a Standard Policy in the online Help.

Continue defining the **New Protection Group** by selecting the remaining settings.

5. **Storage Domain:** 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.
6. **Start Time**: Take the default.
7. **Custom Options**:
   - Do not enter any backup script arguments in the Full Backup and Incremental Backup fields if you plan to perform full backup or incremental backup, respectively.
8. **Mounts:** Ignore the Mounts field. It does not apply to Postgres data protection.
9. **Concurrency:** Enter two times the number of VIPS. The recommendation is to use a multi-stream approach. This dramatically shortens the backup time compared to a single stream. You should experiment with the number of streams in your backup to determine the optimal performance gain for a multi-stream approach.

After you have completed the settings, if you need to change any additional settings on the New Universal Data Adapter Protection Group page, scroll down and click Edit on the right.

Your new Protection Group is active and running and appears on the **Protection** page.

Now that you have created a Protection Group for your databases, you may change the Protection Policy and settings. This way, all your databases in this Protection Group will be managed alike.

For example, keep your backups for a longer period and increase the retention setting in the Policy assigned to this Protection Group.

Postgres Database Restore

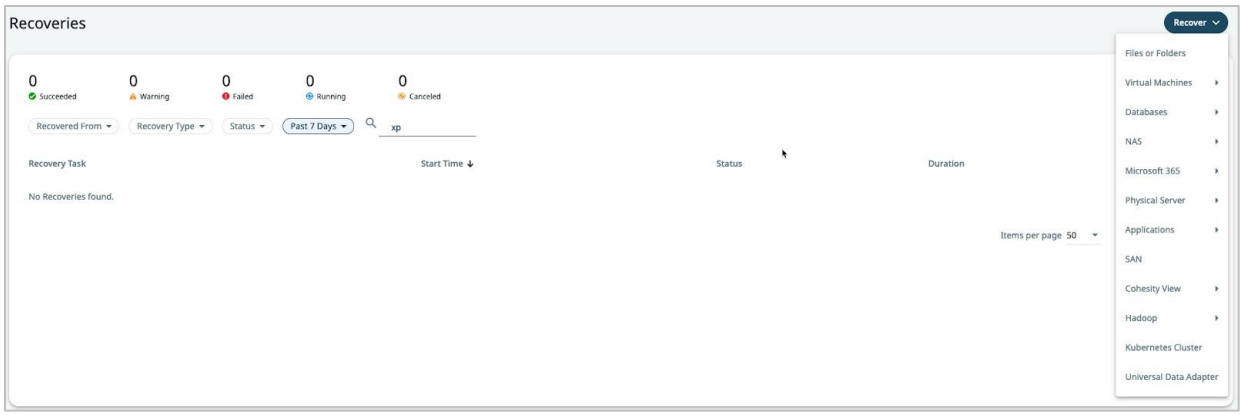
Cohesity allows you to restore individual databases to their original or alternate locations.

IMPORTANT: When restoring a database an empty target database must be created first.

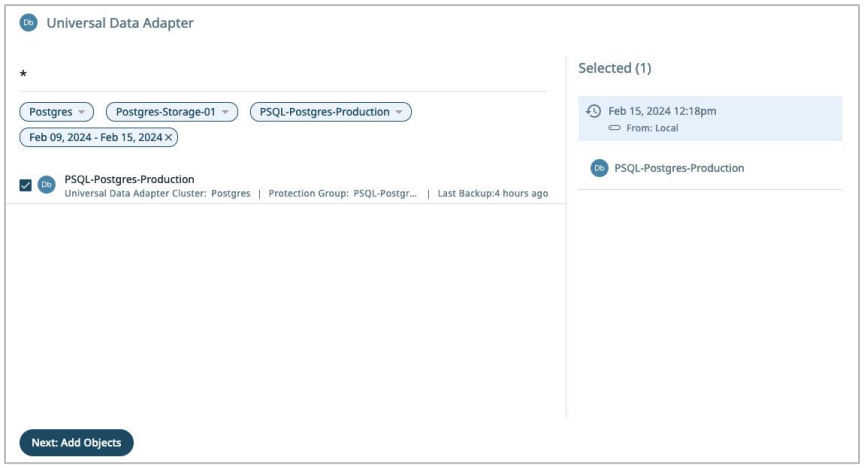
Restore a Specific Backup

To restore the database:

1. Log in to Cohesity and navigate to **Data Protection** > **Recoveries**. Click **Recover** and select **Universal Data Adapter**.

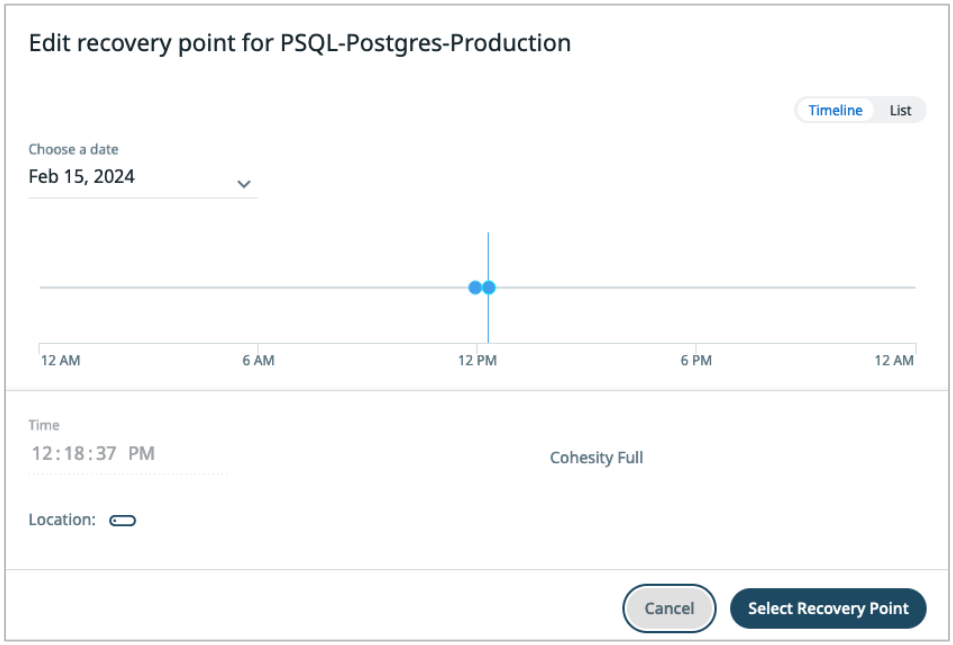


2. Search for the backup. You can start with the wildcard **"*"** to get a general listing.



3. Select the backup.

4. Use the **Edit Recovery Point** form for the backup, then **Select Recovery Point**. FULL and DIFFERENTIAL snapshots are shown as blue dots.

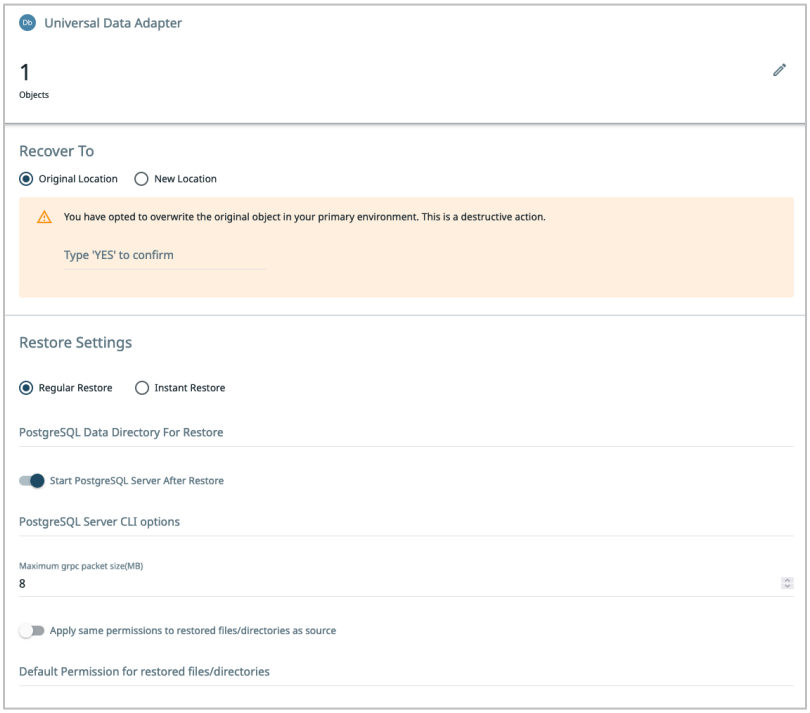


Choose a Date: Choose a valid date.

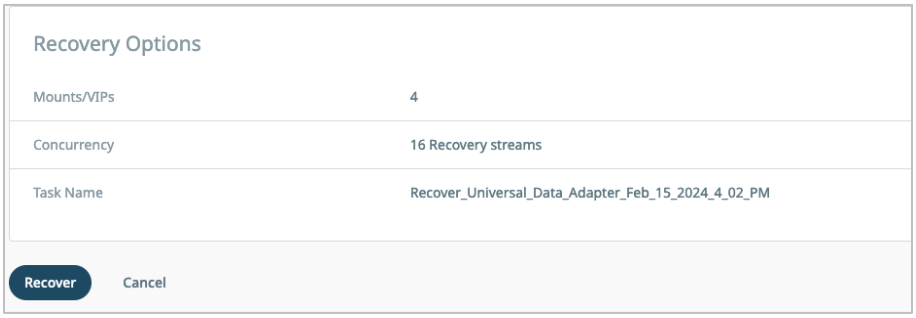
- The timeline shows 24 hours with valid snapshots.
- Each blue dot on the timeline represents FULL or DIFFERENTIAL snapshot points.
- Blue dots can sometimes be clumped together if the snapshots are taken frequently.
- The green bar represents valid log ranges.
- Gaps in the green bar represent breaks in the log chain.
- The slider will snap back to the latest valid time when positioned in an invalid range.

5. Select the recovery point and complete the recovery options.
   Note: Only cluster-level restore is supported, hence the given object name will be ignored by PostgreSQL restore workflow.
6. **Original Location:** To restore to the same Postgres node.
7. **New Location:** To restore to a different Postgres node.

8. Enable **Overwrite** existing object with the same name as type **Yes** if you want to overwrite the existing object with the same name.



9. **Regular Restore:** Restore the Postgres database files to the specified location.
10. **Instant Restore:** Create a clone of the Postgres database.



11. **Mount/VIPs:** Enter the number of VIPs you want to use while reading data from the Cohesity cluster. usually equal to the number of VIPs on the Cohesity Cluster.
12. **Concurrency:** Enter two times the number of VIPS.
13. Select **Recover.**

    Once your files are restored you will need to start Postgres.

Postgres Troubleshooting

Table 4: Components and Paths to Log File

Component	Path to the Log File (Linux)
Cohesity Postgres Plugin logs	`/var/log/cohesity/uda`
Cohesity Physical Adapter logs	`/var/log/cohesity`

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 databases, applications, cloud storage, and enterprise data protection. Scott has over 26 years of experience as an enterprise DBA.

Other essential contributors included:

- Dave Porco is a Principal Solutions Architect
- Adai Arumugan, Sr Director, Product Solutions

Document Version History

VERSION	DATE	DOCUMENT HISTORY
1.0	April 2024	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 42 of the Fortune 100.

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