NetBackup™ for PostgreSQL Administrator's Guide

Release 11.1

VERITAS
now part of Cohesity

Last updated: 2025-11-24

Legal Notice

Cohesity, Inc.
2625 Augustine Drive
Santa Clara, CA 95054

http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Cohesity account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Cohesity website.

Cohesity Services and Operations Readiness Tools (SORT)

Cohesity Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1

Overview

This chapter includes the following topics:

Overview of configuring and protecting PostgreSQL assets in the NetBackup web UI

Table 1-1 Steps to configure and protect PostgreSQL assets

Step	Action	Description
Step 1	■ Open a web browser and go to the URL ■ Enter your credentials and click **Sign in**. ■ On the left, click **Security** > **RBAC** > **Add**. ■ Select **Default PostgreSQL Administrator** and provide a **Role name**, **Role description**, required permissions and assign a webUI user to this role.	For more information on **Sign In** see the *Sign into the NetBackup web UI* in *NetBackup Web UI Administrator's Guide*. **Note:** To perform the PostgreSQL administrator tasks, the **Default PostgreSQL Administrator** role should have the minimum required RBAC permissions. See "Default PostgreSQL Administrator" on page 16.
Step 2	Configure and manage PostgreSQL workload.	See "Configure PostgreSQL instance " on page 10.
Step 3	Add and manage credentials.	See "Manage credentials for an instance" on page 12.
Step 4	Configure a PostgreSQL protection plan.	See "Protect PostgreSQL instances and databases" on page 20.
Step 5	Protect PostgreSQL instances and databases.	See "Protect PostgreSQL instances and databases" on page 20.

Table 1-1 Steps to configure and protect PostgreSQL assets *(continued)*

Step	Action	Description
Step 6	Restore PostgreSQL instances and databases.	See "Restore PostgreSQL instance and database " on page 25.

Chapter 2

Managing PostgreSQL instances and databases

This chapter includes the following topics:

Quick configuration checklist to protect PostgreSQL instances and databases

Use NetBackup web UI to protect and restore the instances and databases that are created on the PostgreSQL platform. You can also use APIs for the same.

See “Using APIs to manage, protect or restore PostgreSQL” on page 42.

The following table describes the high-level steps to protect the PostgreSQL environment:

Table 2-1 Configure and protect PostgreSQL using NetBackup

Step overview	Description and reference
Deploy NetBackup to protect PostgreSQL instances and databases.	On a very high level to protect PostgreSQL instances and databases you need: ■ NetBackup primary server ■ NetBackup media server (Recommended) ■ NetBackup client that can act as a backup machine
PostgreSQL installed bin directory path should be added to path environment variable.	Verify if PostgreSQL installation bin path is set in environment variable. For Example: ■ For Windows : `PATH = C:\Program Files\PostgreSQL\14\bin` ■ For Linux : `export PATH=$PATH:/usr/pgsql-13/bin`
Protecting PostgreSQL instances and databases.	See "Protect PostgreSQL instances and databases" on page 20.

Configure PostgreSQL instance

You can configure PostgreSQL backup for user to perform backup and recovery by configuring the following environment variables:

Note: Add these environment variables to the user for which NetBackup backup is to be run.

- (optional) path - Add PostgreSQL bin path to this environment variable for running queries and connecting to databases.
- (optional) LIB_PQ_PATH - For Windows set this environment variable to provide the location of `libpq.dll` library. For Linux set this environment variable to provide the location of `libpq.so` library.
- (optional) PG_PRO_BACKUP_DUMP_DIRECTORY - Set this environment variable as temporary backup dump directory for non streaming backup. For example, On Linux computer, user can set this environment variable to required location using the command:

  ```
  echo "export
  PG_PRO_BACKUP_DUMP_DIRECTORY=/home/custom_dump_dir_location/" > >
  ~/.bashrc
  ```

  For Windows, user can create new environment variable and add path of folder location as:`PG_PRO_BACKUP_DUMP_DIRECTORY=C:\custom_dump_dir_location`

- (optional) LVM SNAPSHOT_SIZE - Set this environment variable to provide the snapshot size for LVM backup for Linux operating system only. You can set environment variable of LVM Snapshot size to 500 MB using the command:`echo "export LVM_SNAPSHOT_SIZE=500MB" >> ~/.bashrc`

  Note: The default snapshot size is set to 500MB.

- (optional) DELETE_WAL_LOGS - Set this environment variable to delete `Wal` logs after backup is done. The value can be set to 0 or 1.
- PGSQL_COMPRESSION_VALUE - Level of compression which is given to the compression algorithm. Value can be between 0 and 9. Zero being lowest and nine being highest compression ratio.
- (Optional) PG_BACKREST_MAX_PROCESSES_<port> - Set this environment variable to specify the number of threads to use for copying multiple data files concurrently while performing backup and recovery with pgbackrest utility

Add a PostgreSQL instance

You can add a PostgreSQL instance and its credentials.

To add PostgreSQL instance and its credentials

1. On the left, click **PostgreSQL** then click the **Instances** tab.
2. Click **Add** to add a PostgreSQL instance and enter the following:
   - **Host**
   - **Instance name**
3. Enter or use the up, down arrow keys to add details of **Port number**.
4. Click **Next**.

   Note: You will be redirected to the **Permissions** page and you can also manage credentials of the created instance.

5. Click **Finish**.

   Note: If you click **Previous**, instance created will not be saved.

Assign permissions to PostgreSQL instance

You can assign permissions to an instance added.

To assign permissions to the PostgreSQL instance

1. Click **Add** to add permissions to this instance.
2. Select role and permissions.
3. Click **Save** > **Finish**.

Inline actions on PostgreSQL instance

You can run the following inline actions on a PostgreSQL instance:

- **Recover**: Recovers the PostgreSQL instance.
- **Manage credentials**: Manages the instance credentials.
- **Deactivate**: Deactivates the PostgreSQL instance.
- **Remove**: Removes the PostgreSQL instance.

Actions on multiple PostgreSQL instances

You can select one or more PostgreSQL instances and perform the following actions:

- **Deactivate**: Deactivates the PostgreSQL instances.
- **Manage credentials**: Manages the credentials of the PostgreSQL instances.
- **Remove**: Removes the selected PostgreSQL instances.

Auto-discovered cluster asset:

- PostgreSQL primary node instance is discovered and added in web UI asset automatically.
- PostgreSQL standby node instance is discovered and added in web UI asset automatically.

Manage credentials for an instance

You can add or update credentials for instances. When you add an instance, you can choose not to include the credentials at the time of its entry.

To add credentials for an instance at the time of its entry into the repository follow the below steps:

1. Select **Manage credentials**.
2. In the **Manage credentials** screen, select one of the appropriate methods:
   - **Select from existing credentials**.

- **Add credentials**.

See “Add new PostgreSQL credentials” on page 15.

3 Click **Next**.

Discover PostgreSQL databases

You can discover PostgreSQL databases.

To discover PostgreSQL databases:

1 On the left, click **PostgreSQL** then click the **Database** tab.

2 Click **Discover** to discover a PostgreSQL database.

3 Select the required instance from the list of instances for which you need to discover the databases.

4 Click **Discover**.

Remove PostgreSQL instances

Use this procedure to remove PostgreSQL instances.

To remove PostgreSQL instances

1 On the left, click **PostgreSQL**, then click the **Instances** tab.

Note: The tab lists the names of instances that you have access to.

2 Select one or more PostgreSQL instances.

3 Select **Actions > Remove** or select **Remove** from top bar.

Note: If you delete an instance, all databases that are associated with the removed PostgreSQL instance will also get deleted.

4 If you are sure that you want to delete the PostgreSQL Instance, click **Remove**.

Change the autodiscovery frequency of PostgreSQL assets

Automatic discovery of PostgreSQL assets occurs at regular intervals. The default frequency is every 8 hours. Use this procedure to change the autodiscovery frequency.

To change the frequency of autodiscovery of PostgreSQL assets:

1. On the left, click **Workloads > PostgreSQL**.
2. On the right, click **PostgreSQL settings > Autodiscovery**.
3. Select **Frequency > Edit**.
4. Enter the number of hours or use the up or down arrows to choose how often you want NetBackup to perform autodiscovery of PostgreSQL assets. Then click **Save**.

Note: The range from which you may choose is 1 hour to 24 hours. To set the autodiscovery frequency in minutes or seconds or to disable autodiscovery, you must use the PostgreSQL autodiscovery API.

Chapter 3

Managing PostgreSQL environment credentials

This chapter includes the following topics:

Add new PostgreSQL credentials

You can add a new credential to an instance at the time of its creation. See "Manage credentials for an instance" on page 12.

To add new PostgreSQL credentials

1. On left, click **Credential management**.
2. On the **Named credentials** tab, click **Add**.
3. Provide **Credential name**, Tag, and **Description**.

   Note: Credential name should not contain a **%** character.

4. Click **Next**.
5. Select **PostgreSQL server** from the **Category** drop-down.
6. Enter **Instance username** and **Instance user password** and click **Next**.

7 On the **Permissions** page, click **Add**.

8 Select role and permissions.

9 Click **Save** and click **Next**.

10 Review and click **Finish**.

Note: You can **Edit** or **Delete** the added credentials.

Default PostgreSQL Administrator

This role has all the permissions that are necessary to manage PostgreSQL and to back up those assets with protection plans.

Table 3-1 RBAC permissions for Default PostgreSQL Administrator role

Type	Permissions
Global permissions > NetBackup management	
Access hosts	View, Create, Delete
Agentless hosts	View
Host Properties	View
Media Server	View
External Credential Management System (External CMS)	View, Create, Update, Delete, External CMS-Import
NetBackup hosts	View, Create, Update
NetBackup backup images	View, View Contents
Jobs	View
Resource limits	View, Create, Update, Delete
Trusted primary servers	View
Global permissions > Storage	
Storage servers	View, Create, Update, Delete
Disk volumes	View, Create, Update, Delete
Storage units	View, Create, Update, Delete

Table 3-1 RBAC permissions for Default PostgreSQL Administrator role *(continued)*

Type	Permissions
Assets	
PostgreSQL assets	Full permissions
Protection plans	Full permissions
Credentials	Full permissions

Validate credentials of PostgreSQL instance

To validate PotgreSQL instance credentials

You can validate a specific or multiple instance's credentials.

1. On the left, click **Workloads** > **PostgreSQL**, then click the **Instances** tab.
2. Locate and select one or more PostgreSQL instances.
3. Click **Manage Credentials** > **Select from existing credentials**.
4. Click **Next** and select the credentials that you want to use for this instance.
5. Click **Next** > **Close**.

Note: NetBackup verifies the current credentials for the selected PostgreSQL instance.

If the credentials are not valid, NetBackup indicates **Invalid** under **Credentials**.

For auto-discovered cluster instances, assign credentials for PostgreSQL primary or standby node instance.

View the credential name that is applied to an asset

You can view the named credential that is configured for an asset type. If the credentials are not configured for a particular asset, this field is blank.

To view credentials for PostgreSQL

1. On the left, select **Workloads** > **PostgreSQL**.
2. On the PostgreSQL **Instances** tab, scroll right to locate the **Credential name** column.

Edit or delete a named credential

You can edit the properties for a named credential or delete a named credential from the Credential management.

Edit a named credential

You can edit a named credential when you want to change the credential **Tag**, **Description**, **Category**, authentication details, or permissions. You cannot change the credential name.

To edit a named credential

1. On the left, click **Credential management**.
2. Click **Edit** and update the credential as needed.

   Note: When you update PostgreSQL instances, this action automatically starts the discovery of the PostgreSQL instance.

3. Review the changes and click **Finish**.

Delete a named credential

You can delete a named credential that you no longer need to use.

Warning: Apply another credential to any asset that uses the credential you want to delete or else backup and restore may fail for those assets.

To delete a named credential

1. On the left, click **Credential management**.
2. On the **Named credentials** tab, locate and click on the credential that you want to delete.
3. Click **Delete**.
4. If you are sure that you want to delete, click **Delete**.

Chapter 4

Protecting PostgreSQL instances and databases

This chapter includes the following topics:

Things to know before you protect PostgreSQL instances and databases

Protection plans can be used to predefine backup policies which are then used by others to protect their data. The following table describes the permissions with which PostgreSQL non-root database user should be created:

Table 4-1 User Privileges

User	Privileges
Protection and Recovery	Superuser

To set the database user privileges, run the following command at PostgreSQL command line:

```
ALTER USER "username" WITH SUPERUSER;
```

Protect PostgreSQL instances and databases

Use the following procedure for subscribing a protection plan to PostgreSQL instance and database. When you subscribe an asset to a protection plan, you assign predefined backup settings to the asset.

Note: The RBAC role assigned to a user must have access to the assets that you want to manage and also to the protection plans you want to use.

To protect a PostgreSQL instance or database:

1. On the left pane, click **PostgreSQL**.
2. On the Instances tab or Databases tab, click the box for the instance or the database and click **Add protection**.
3. Select a protection plan and click **Next**.
4. You can edit one or more of the following settings:
   - **Schedules and retention**
     Change when backups occur and the backup start window.

     Schedules:
     - **Full**: Complete instance backup using **snapshot, pg_basebackup, pg_dumpall, or pgbackrest** and completes database-backup using **pg_dump** utility.
     - **Differential Incremental**: Based on the previous backup timestamp, NetBackup identify the changed set of transaction logs (WAL files) and perform its backup.
   - **Backup options**
     Adjust the **Database options** like **Job limit** and **Backup method**.
     - **Snapshot**: This option is used to take the snapshot of an instance. For Windows - VSS snapshot method is used. For Linux - LVM snapshot method is used.
     - **pg_basebackup**: This utility of PostgreSQL performs the backup of an instance. It is recommended in the case of non-LVM deployment.
     - **pg_dumpall**: This utility of PostgreSQL performs logical backup of instance. It is recommended in the case of non-LVM deployment.
     - **pg_dump**: This utility of PostgreSQL performs logical backup of an individual database.

- **pgbackrest**: It is an utility which performs physical backup of an instance using an NFS share of existing NetBackup deduplication pool (MSDP) on Linux platform only.
  Ensure the following:
  - **pgbackrest** utility is compatible with the PostgreSQL installed.
  - MSDP storage is configured. For more information, see *NetBackup Deduplication guide* > *Configuring and managing universal shares* > *Prerequisites to configure universal shares* section.
  - Binary path of **pgbackrest** utility must be exported in the path environment variable.
    For Linux, use: `echo "export PATH=$PATH:<pgbackrest_installation_path>" >> ~/.bashrc`
  - You can configure **PG_BACKREST_MAX_PROCESSES** environmental variable. See "Configure PostgreSQL instance " on page 10.
  - Maximum Jobs per client should be set to greater than 1 in primary server.
  - Media server and storage server should be on Linux Platform only.
  - SpanFS storage unit is NOT supported.

Note: In the case of snapshot-based backups, it is recommended to keep the archive directory and the data directory in separate locations.

Note: If a PostgreSQL instance has tablespaces configured, then use `pg_dumpall` backup method. In the case of `pg_dumpall` and `pg_dump` backup method, incremental backups are not supported.

5 Click **Protect**.

Note: If the PostgreSQL instance is deployed on the root LVM, then Snapshot backup method is not recommended.

In case of cluster deployment of PostgreSQL, instances can be protected of primary or standby node.

Customize protection settings for the PostgreSQL assets

To customize protection settings for the PostgreSQL assets

You can customize certain settings for a protection plan, including schedules.

1. On the left, select **Workloads > PostgreSQL**.
2. Click on the instance whose protection is to be customized.

   Note: This action allows custom protection for the asset and removes it from the original protection plan. Any future changes to the original plan are not applied to the asset. The customization operation cannot be reversed.

3. Click **Customize protection** > **Continue**.
4. You can edit one or more of the following settings:
   - **Schedules and retention**
   - **Backup options**
5. Click **Protect**.

Remove protection from PostgreSQL instances

You can unsubscribe PostgreSQL instances from a protection plan. When the asset is unsubscribed, backups are no longer performed.

Note: When you unsubscribe an asset from a protection plan, there is a possibility that the asset displays **Classic policy** in the **Protected By** column on the web UI. This situation can happen when an asset is subscribed to a protection plan and a backup runs for that asset. Such assets get unsubscribed from the protection plan. The web UI then displays **Classic policy**, that may or may not have an active policy protecting the asset.

To remove protection from a PostgreSQL instance

1. On the left, click **PostgreSQL**.
2. On the **Instances** tab, select the instance.

3 Click the instance name.

4 Click **Remove protection** > **Yes**.

Under **PostgreSQL**, the asset is now listed as **Not protected**.

View the protection status of PostgreSQL instance

You can view the protections plans that are used to protect PostgreSQL instance.

To view the protection status of PostgreSQL instance

1 On the left, click **PostgreSQL**.

2 On the **Instances** tab, select the instance. The **Protection** tab shows the details of the asset subscription plans.

Note: If the asset has been backed up, but status indicates that it has not, See "Error during the PostgreSQL instances and databases discovery phase" on page 38.

---

3 If the asset is not protected, click **Add protection** to select a protection plan.

Chapter 5

Restoring PostgreSQL Instances and Databases

This chapter includes the following topics:

Things to know before you restore the PostgreSQL instances and databases

Ensure that the restore server that is added to Netbackup environment should have PostgresSQL footprint on it.

About the pre-restore check

The pre-restore check verifies the following:

- Availability of the PostgreSQL environment.

- Available space with the storage.
- (For Windows) For pg_basebackup protection, OpenSource TAR utility must be installed on the windows deployment.
  ICACLS windows command-line utility packages must be installed and installed path must be a part of environment path variable.

Restore PostgreSQL instance and database

You can restore a PostgreSQL instance or database either to an original backup location or to an alternate location. You can choose to recover from the default copy of the instance or database. The default copy is also known as the primary copy.

To restore a PostgreSQL instance

1. On the left, click **Workloads > PostgreSQL**.
2. On the **Instances** tab, select the instance that you want to recover.
3. Click **Recover** from the top bar.
4. On the **Recovery points** tab, select the date with available backup.

   Note: In the calendar view, dates with available backups are indicated with a green dot.

5. From the listed **Backup images/ Recovery points**, select the desired image or recovery point.

   Note: The backup images or recovery points are listed in rows with the respective backup timestamp.

6. Click **Actions** and then select either **Perform complete instance recovery** or **Perform point in time recovery.**

   Note: Point in time recovery option is only available for incremental backup.

   Perform point in time recovery:

   - On the **Recovery option** window, select one of the following:
     - **Recovery point selected** for the default recovery point.
     - **Point in time** and then select exact date and time for recovery.

Note: Select the point in time which is post the previous full backup and includes a time through selected incremental recovery point.

7 Click the search icon in the **Host** field, select the desired host and click **Save**.

- If the recovery is to an alternate host, then select the corresponding valid credentials from the displayed list.

For more information, See "Restore target options" on page 29.

8 Select the appropriate instance directory path from one of the following options:

- **Restore everything to original location**: Files are restored to the location where they were originally backed up from.
- **Restore everything to a different location**: Files are restored to the specified alternate location. The folder structure of the restored data within the alternate location is the same as the original data that is same folder and subfolder setup.
  - **Directory for Restore** – PostgreSQL full backup data is restored to the specified path.
  - **Write Ahead logging Directory** – PostgreSQL WAL files are restored in this directory. PostgreSQL incremental backup data is restored to the specified path.

Note: If the instance is added manually and not discovered automatically, then only **Restore everything to a different location** option can be selected.

Note: WAL files will be restored in the data directory when **Restore** option is selected, whereas, in case of **Restore and Recovery**, WAL files will be restored to path mentioned in **WAL directory for restore** option.

9 Click **Next** and follow the instructions prompted.

10 On the **Recovery source** tab, review the storage details.

11 Click **Next**.

12 On the **Recovery points** tab, select the **Restore** or **Restore and recovery** option to perform instances and database restore and recovery:

Note: For point in time recovery, Restore and recovery option is triggered automatically.

- **Restore**: Restores the instances.
- **Restore and recovery** – Recover the instances.

Note: For LVM and VSS if the **Restore and recovery** option is selected, then contents of the target data directory is deleted by recovery operation.

Note: In the **Restore and recovery** option, the PostgreSQL service must be up and running. If the service is stopped, the restore will fail.

The data folder must be empty to run the **Restore** option on the same path. If the folder is not empty, then, the data will not restored.

---

13 Click **Next**.

14 On the **Review** tab, review the details and click **Start recovery**.

To restore a PostgreSQL database

1 On the left, click **Workloads > PostgreSQL**.

2 On the **Databases** tab, select the database that you want to recover.

3 Click **Recover** from the top bar.

4 On the **Recovery points** tab, select the date with available backup.

Note: In the calendar view, dates with available backups are indicated with a green dot.

---

5 From the listed **Backup images/ Recovery points**, select the desired image or recovery point.

Note: The backup images or recovery points are listed in rows with the respective backup timestamp.

---

6 Click **Actions > Perform complete database recovery**.

7 Click the search icon in the **Host** field, select the desired host and click **Save**.

- If the recovery is to alternate a host, then select the corresponding valid credentials from the displayed list.

For more information, See “Restore target options” on page 29.

8 Select the appropriate **Database directory paths** from one of the following options:

- **Restore everything to original location**: Files are restored to the location where they were originally backed up from.
- **Restore everything to a different location**: Files are restored to the alternative location which you can specify. The folder structure of the restored data within the alternate location is the same as the original data that is same folder and subfolder setup.
  - **Directory for Restore** – The PostgreSQL data directory. PostgreSQL full backup data is restored to the specified path.
  - **Write Ahead logging Directory** – PostgreSQL WAL files are restored in this directory. PostgreSQL incremental backup data is restored to the specified path.

For more information, See “Restore target options” on page 29.

9 Click **Next**.

10 On the **Recovery source** tab, review the storage details.

11 Click **Next**.

12 On the **Recovery points** tab, select the **Restore** or **Restore and recovery** option to perform instances and database restore and recovery:

- **Restore** – Restores the database.
- **Restore and recovery** – Recovers the database.

13 Click **Next**.

14 On the **Review** tab, review the details. For any changes you can edit the recovery target, recovery source and recovery options or click **Start recovery**.

Restore target options

Table 5-1 Restore target options

Step overview	Description and reference
Host	■ Host field is pre-populated with the source PostgreSQL client stored during last successful discovery for respective instance. ■ If you want to perform a restore on another NetBackup client, click search and select the required client from the list. **Note:** Ensure that you select clients with homogenous platforms. ■ If search option is unavailable, manually enter **Host**.
Instance directory paths	■ **Change staging location on client**: If you want to provide a different staging location other than the default staging location, enter the desired path. Staging location path must have only ASCII characters. ■ **Instance directory paths** : Based on your requirement, select one of the following appropriate Instance directory paths between: ■ **Restore everything to original directory** ■ **Restore everything to different directory** Provide different directory path to restore.

Pre-restore checks for PostgreSQL

Table 5-2 Pre-restore checks

Validation	Description and reference	Input Source
Restore client space	Checks for the required space on restore location.	Restore client
Target client connectivity	Checks if target client is accessible from restore client.	Target client and Target client name
Target client alternate location on a local disk	Checks if target client alternate location is not a network path.	Target client alternate location

Table 5-2 Pre-restore checks *(continued)*

Validation	Description and reference	Input Source
Target client location space	Checks if the required space is available on target client alternate location. **Note:** Required space is total size of selected file with space required for restore and space needed for logs and other files.	Target client alternate location
Target client alternate location permissions	Checks if provided user is an owner and has RBAC permissions on target client alternate location.	Target client alternate location
Target client default alternate location path	Checks if provided target client alternate location path contains valid characters. Non-ASCII characters are not supported in target client alternate location path.	Target client alternate location
Target client operating system	Checks if target client has a supported OS.	General

Table 5-3 Permissions for all PostgreSQL assets

Operation	Description	Additional required operations	Additional optional operations
Restore and recovery	Restore backup images of PostgreSQL asset. This permission is required on PostgreSQL.	**Global > NetBackup management > NetBackup backup images > View** **Global > NetBackup management > NetBackup backup images > View contents** **Global > NetBackup management > NetBackup hosts > View** **Assets > PostgreSQL assets > Restore**	**Assets > PostgreSQL Assets > Restore to alternate location**

Steps to perform recovery after restore operation

The procedure to perform post-recovery is as follows for various platforms:

For Windows (VSS):

1. Go to **Control Panel > System and Security > Administrative Tools > Services**.
2. Select PostgreSQL service and stop it.
3. Delete or move everything from the PostgreSQL data directory.

   Note: Post restores, change the attributes of the restored data directory and files by using the following command:

   ```
   attrib -S restore_path/*.* /S /D
   ```

4. Copy all the contents of the restored data directory to PostgreSQL data directory.

5 Edit the `postgresql.conf` file from the PostgreSQL data directory and edit the `restore_command` parameter as `restore_command = 'copy "restored_WAL_directory\\%f" "%p"'`.

   For precise point-in-time recovery, specify the timestamp up to which recovery is performed. Update the `recovery_target_time` parameter as `recovery_target_time = 'yyyy-mm-dd hh:mm:ss'`

6 Create an empty file in the data directory and name it `recovery.signal`.

7 Start PostgreSQL service.

For Linux (LVM):

1 Stop PostgreSQL services.

2 Delete or move everything from the PostgreSQL data directory.

3 Extract and copy the data directory and WAL directory contents to respective location.

4 Create an empty file with name `recovery.signal` in data directory.

5 Edit the `postgresql.conf` file from the PostgreSQL data directory and edit the `restore_command` parameter as `restore_command = 'copy "restored_WAL_directory\\%f" "%p"'`.

   For precise point-in-time recovery, specify the timestamp up to which recovery is performed. Update the `recovery_target_time` parameter as `recovery_target_time = 'yyyy-mm-dd hh:mm:ss'`

6 Change ownership of PostgreSQL data directory and permission to 700.

   For example:

   `chown -R postgres:postgres /full/path/of/PostgreSQL/Data/Dir`

   `chmod - R 700 /full/path/of/PostgreSQL/Data/Dir`

7 Start PostgreSQL service.

Recovery steps for backup done by `pg_basebackup` utility

1 Stop PostgreSQL services.

2 Delete or move everything from the PostgreSQL data directory.

3 Extract and copy the data directory and WAL directory contents to respective location.

4 Create an empty file with name `recovery.signal` in data directory.

5 Edit the `postgresql.conf` file from the PostgreSQL data directory and edit the `restore_command` parameter as `restore_command = 'copy "restored_WAL_directory\\%f" "%p"'`.

For precise point-in-time recovery, specify the timestamp up to which recovery is performed. Update the `recovery_target_time` parameter as `recovery_target_time = 'yyyy-mm-dd hh:mm:ss'`

6 (For Windows) Provide access to data directory for network service.

7 (For Linux) Change ownership of PostgreSQL data directory and permission to 700.

For example:

```
chown -R postgres:postgres /full/path/of/PostgreSQL/Data/Dir
chmod -R 700 /full/path/of/PostgreSQL/Data/Dir
```

8 Start PostgreSQL service.

Note: Remove the restore data from the restored path `/full/path/of/restore/directory` after successful recovery, else the next backup job may fail.

Recovery steps for backup using pgbackrest utility

1 Stop the PostgreSQL service.

2 Delete or move everything from PostgreSQL data directory.

3 Perform the restore operation using pgbackrest utility using the following command:

```
pgbackrest --stanza=<stanza_name> --pg1-path=<data_directory_path>
--repo1-path=<restore_path>
```

Note: The stanza name can be identified by examining the backup directory within the restore path. This directory contains a subdirectory named after the stanza, which holds the associated backup sets.

For precise point-in-time recovery, specify the timestamp up to which recovery is performed with the help of `--target` option.

For example:

```
pgbackrest --stanza=<stanza_name> --pg1-path=<data_directory_path>
 --repo1-path=<restore_path> --type=time --target="yyyy-mm-dd
hh:mm:ss+00"
```

Note: Ensure that the time is GMT format.

To optimize restore performance, use the `--process-max` option to define the number of parallel processes (1–999) for copying data files concurrently.

For example:

```
pgbackrest --stanza=<stanza_name> --pg1-path=<data_directory_path>
 --repo1-path=<restore_path> --process-max=5
```

4 Change the ownership of PostgreSQL data directory and set the permission to 700.

For example:

```
chown -R postgres:postgres /full/path/of/PostgreSQL/Data/Dir
chmod -R 700 /full/path/of/PostgreSQL/Data/Dir
```

5 Start PostgreSQL service.

Database recovery steps for backup done by `pg_dumpall` utility

- For Windows: `psql.exe -h localhost -p` *`port_num`* `-U` *`username`* `-f` *`full\path\to\dumpall\file`*`\filename.out`

- For Linux: `psql -h localhost -p port_num -U username -f full/path/to/dumpall/file/filename.out`

Database recovery steps for backup done by `pgdump` utility

- For Windows: `pg_restore -U username -d dbname full\path\to\dump\file\filename.dump`
- For Linux: `pg_restore -U username -d dbname full/path/of/dump/file/filename.dump`

Steps to perform after Restore and Recovery for PostgreSQL cluster deployment

The following procedure is applicable for `snapshot, pgbackrest,` and `pg_basebackup` backup methods, and not applicable for the `pg_dumpall` and `pg_dump` backup methods.

If the recovery done from backup of PostgreSQL primary node to same or alternate primary node, do the following on the PostgreSQL standby node:

- Stop PostgreSQL services.
- Clean the PostgreSQL data directory path.
- Run the `$ pg_basebackup -h primary_node_ip -U db_replication_user --checkpoint=fast -D data_directory_path -R --slot=unique_slot_name -C` command with database user.
- Start the PostgreSQL services.

If the recovery done from backup of PostgreSQL standby node to same or alternate primary node, do the following on the PostgreSQL primary node:

- Stop the PostgreSQL services.
- Delete the `standby.signal` file from the data directory.
- Fix the archive command in `postgresql.conf` file from data directory.
  For example: For Linux: `archive_command='cp %p /path/to/archive/location/%f'` and for Windows: `archive_command = 'copy "%p" "path\\to\\archive\\location\\%f"'`
- Start the PostgreSQL services.

PostgreSQL standby node:

- Stop PostgreSQL services.

- Clean the PostgreSQL data directory path.
- Run the below command with database user:`$ pg_basebackup -h master_node_ip -U db_replication_user --checkpoint=fast -D data_directory_path -R --slot=unique_slot_name -C`
- Start the PostgreSQL services.

Note: When the recovery is done from backup of PostgreSQL primary node to PostgreSQL standby node, then the standby node becomes an independent primary node.

Limitations

- Cross-platform recovery of individual files is not supported. The restore client must be the same platform as the instances that you want to restore. Windows instances can be restored using Windows operating systems and Linux instances can be restored only using Linux operating systems.
- For client platform and file system support and limitations, see https://www.veritas.com/content/support/en_US/doc/NB_70_80_VE .
- If a backup and a restore occur simultaneously on the same database, one or both jobs can have unexpected results.

  Note: If a backup or a restore exits with a non-zero NetBackup status code, one possible cause is simultaneous jobs occurring on the same instance.

- Restore job fails, if NetBackup does not have sufficient privileges or if there is insufficient space in the client memory.
- NetBackup does not support non-ASCII characters in target client location path.

Chapter 6

Troubleshooting PostgreSQL operations

This chapter includes the following topics:

Troubleshooting tips for NetBackup for PostgreSQL

For more information about PostgreSQL troubleshooting, check the following details:

- For discovery failures:
  - Check the `ncfnbcs` log.
- For backup job failures:
  - Check the `bprd`, `bprm`, `bphdb` and `nbpgsql` logs.
- For restore job failures:

- Check the `bprd`, `bprm` and `tar` logs.

Error during PostgreSQL credential addition

Table 6-1 Error during PostgreSQL credential addition

Error message or cause	Explanation and recommended action
Credential validation failed. Provide correct host name.	The host name is not valid NetBackup client. Ensure that the hostname is registered client of NetBackup and it is white listed.

Error during the PostgreSQL instances and databases discovery phase

The following table describes the problem that might occur when you try to discover PostgreSQL database.

Table 6-2 Error run into during the PostgreSQL instance and database discovery phase

Error message or cause	Explanation and recommended action
The PostgreSQL assets are not discovered after the correct PostgreSQL instance credentials are added.	Run discover database and retry the database discovery manually. - Ensure that the update permission assigned to the logged in web UI user. - Contact Cohesity Technical Support and share `nbwebservice` logs from NetBackup primary server and `ncfnbcs` logs from NetBackup client.

Error during the PostgreSQL Protection Plan Creation

The following table describes the problem that might occur while creating protection plan for PostgreSQL workload.

Table 6-3 Error during the PostgreSQL Protection Plan Creation

Error message or cause	Explanation and recommended action
A plan with this name already exists.	Protection plan with same name is already present. ■ Please create protection plan with another name.
Storage disk pool is not present.	Before adding protection, we need to add storage unit. ■ Please add Storage Unit from **Storage Configuration** >**Add**.

Error while subscribing protection plan to PostgreSQL asset

The following table describes the problem that might occur during subscribing protection plan to a PostgreSQL asset.

Table 6-4 Error while subscribing protection plan to a PostgreSQL asset

Error message or cause	Explanation and recommended action
This subscription must be reset to protection plan defaults before it can be customized.	If subscription has been already modified, the below warning message will be displayed. ■ User can reset subscription using 'Restore original settings' button and then try to customize subscription again.
Storage disk pool is not present	Before adding protection, we need to add storage unit. ■ Please add Storage Unit from **Storage Configuration** >**Add**.

Error while removing PostgreSQL asset

Table 6-5 Error while removing PostgreSQL Asset

Error message or cause	Explanation and recommended action
Removed 0 of 1 instance.	If protection plan is attached to PostgreSQL asset, then we cannot delete such an asset. ■ First unsubscribe protection plan from asset and then delete the asset.

Error while backup of PostgreSQL asset

The following table describes the problem that might occur when you back up PostgreSQL asset. Backup jobs fail with error code 6.

Table 6-6 Error while backing up PostgreSQL assets

Error message or cause	Explanation and recommended action
The backup failed to back up the requested files	Verify the PostgreSQL service is up and running on client. ■ Contact Cohesity Technical Support and share `bphdb` and `nbpgsql` logs from backup client.

Error while restoring PostgreSQL asset image

The following table describes the problem that might occur when you restore PostgreSQL asset.

Table 6-7 Error while restore of PostgreSQL asset image

Error message or cause	Explanation and recommended action
Unable to change the Host while modifying the restore target or destination.	If you cannot see the list of the host, you might not have access to NetBackup Host in RBAC. ■ Contact the NetBackup security administrator to resolve this issue.

Table 6-7 Error while restore of PostgreSQL asset image *(continued)*

Error message or cause	Explanation and recommended action
Restore failed with below error: Restore initiated from XBSA Failed to query the object... 17	If the database user provided for restore operation is different from the backup operation database user. The permission of file differs in the NetBackup file system and hence restore fails. ■ Use the same database user for restore which was used while taking backup of asset, so that file system permissions will be available to the restore user as well.
Restore Image not found at alternate location on recovery host.	No image was found on recovery host alternate location. ■ Contact Cohesity Technical Support and share `tar` log from the recovery host.

Chapter 7

API for PostgreSQL instances and databases

This chapter includes the following topics:

- Using APIs to manage, protect or restore PostgreSQL

Using APIs to manage, protect or restore PostgreSQL

This topic lists the APIs to manage, protect or restore the PostgreSQL instances and databases. Only the important variables and options are mentioned in this topic.

Following sections are part of this topic:

- See the section called “Add a PostgreSQL instance” on page 43.
- See the section called “PostgreSQL Discovery API” on page 43.
- See the section called “Create a PostgreSQL Protection Plan” on page 44.
- See the section called “PostgreSQL Recovery point Service API ” on page 44.
- See the section called “Restore the PostgreSQL instance and database at the original location ” on page 45.
- See the section called “Restore the PostgreSQL instance and database to an alternate location ” on page 45.

For detailed information on the APIs, use these references:

- All the NetBackup APIs are listed at the following location:
- Services and Operations Readiness Tools (SORT) > Knowledge Base > Documents

Add a PostgreSQL instance

Table 7-1 Add a PostgreSQL instance

API	Important variables and options

`POST /netbackup/asset-service/queries`
 	
`GET /netbackup/asset-service/queries/{aqcId}`
 	
`GET /netbackup/asset-service/workloads /postgresql/assets`
	■ `clientName` is the name of the PostgreSQL instance.
	■ `sqlHostName` is hostname of a NetBackup client.
	■ `credentialName` are credentials associated with PostgreSQL instance.
	Note: The credential must exist with `credentialName` mentioned.
	■ `port` is port number of PostgreSQL instance.

PostgreSQL Discovery API

Table 7-2 Discover the PostgreSQL asset for given client

API	Important variables and options

`POST /netbackup/admin/discovery /workloads/postgresql/start`
 	
`POST /netbackup/admin/discovery/workloads /postgresql/stop`
 	
`GET /netbackup/admin/discovery/workloads /postgresql/status`
 	
`POST /netbackup/admin/discovery/workloads /postgresql/allclientsdiscovery`
	■ `serverName` is used to identify instance or database
	■ `discoveryHost` is hostname where discovery needs to be triggered
	■ `allclientsdiscovery` triggers discovery for all the clients host associated with the primary.

Create a PostgreSQL Protection Plan

Table 7-3 Create a PostgreSQL Protection Plan

API	Important variables and options

`POST /netbackup/servicecatalog/slos`
 	
`POST /netbackup/servicecatalog/slos/{sloId} /subscriptions`
 	
`POST /netbackup/servicecatalog/slos/{sloId} /backup-now`
	■ `policyType` is `DataStore`.
	■ `Add scheduleName` can have values like FULL_AUTO or INCR_AUTO for adding PostgreSQL instance.
	■ `keyword` can have the following values to back up an instance or database using different backup options:
	  ■ pg_dump
	  ■ pg_basebackup
	  ■ Snapshot
	  ■ pg_dumpall
	  ■ pgbackrest
	■ `sloId` is the identifier to protection plan
	■ `selectionId` is the AssetId which needs to be subscribed with given sloId

After you create a protection plan, other processes like creating the schedule for the policy or triggering the policy backup remain the same.

PostgreSQL Recovery point Service API

Table 7-4 PostgreSQL asset backup instances available for recovery

API	Important variables and options

`GET /netbackup/recovery-point-service /workloads/postgresql/recovery-points`
 	
`GET /netbackup/recovery-point-service /workloads/postgresql/recovery-points /{backupId}`
 	
`GET /netbackup/wui/workloads/postgresql /recovery-point-calendar-summary`
	■ `backupId` is identifier that was used at the time of backup.
	■ `assetId` is identifier that was used to identify instance or database.
	■ `client hostname` is name of backup client.

Restore the PostgreSQL instance and database at the original location

Table 7-5 Restore the PostgreSQL instance and database at the original location

API	Important variables and options

`POST /netbackup/recovery/workloads/postgresql/scenarios/instance-complete-recovery/recover`
 	
`POST /netbackup/recovery/workloads/postgresql/scenarios/database-complete-recovery/recover`
	■ `backupId` is identifier that was used at the time of backup.
	■ `assetId` is identifier that was used to identify instance or database.
	■ `Client` is server that is to be used as the PostgreSQL recovery host to perform this recovery. Set the following value:
	
	`renameAllFilesToSameLocation`

Restore the PostgreSQL instance and database to an alternate location

Table 7-6 Restore the PostgreSQL instance and database to an alternate location

API	Important variables and options

`POST /netbackup/recovery/workloads/postgresql/scenarios/instance-complete-recovery/recover`
 	
`POST /netbackup/recovery/workloads/postgresql/scenarios/database-complete-recovery/recover`
	■ `backupId` is identifier that was used at the time of backup.
	■ `assetId` is identifier that was used to identify instance or database.
	■ `Client` is server that is to be used as the PostgreSQL recovery host to perform this recovery. Set the following value:
	
	`renameEachFileToDifferentLocation`