COHESITY

Version 1.2

July 2024

Oracle Data Protection Deployment Using Cohesity Oracle SBT Plug-in

Options and Instructions to Deploy Cohesity Data Protection for Oracle Servers

ABSTRACT

This document provides a step-by-step workflow to protect your oracle databases using Cohesity Oracle Data Protection. We will step through the installation of Cohesity Oracle SBT Plug-in, the creation of a View, scheduling the protection job, and the restore of the Oracle database .

Table of Contents

Tables

Figures

Introduction

This document provides a step-by-step guide for customers to protect Oracle databases using the Oracle Data Protection with Cohesity Oracle SBT Plug-in.

Oracle Data Protection Methods

The Cohesity Oracle solution extends the scalability of RMAN and provides the features and tools needed for automating backups, recoveries, and managing your data within a single pane of glass.

Cohesity offers three backup methods:

- Cohesity Oracle Adapter
- Cohesity Oracle SBT Plug-in
- RMAN NFS Target

This guide is focused on the beneficial use of Cohesity Oracle SBT Plug-in.

Cohesity Oracle RMAN SBT Plug-in: The Cohesity Oracle SBT Plug-in allows you to use Cohesity multi-cloud data platform as target storage for backing up Oracle databases.

Cohesity Oracle Adapter: Cohesity Oracle Adapter integrates with RMAN to provide application-consistent backup and recovery for Oracle databases. The adapter provides a unified user interface allowing users to manage and provision all the Oracle database protection services, supporting Oracle single instance, RAC, or Oracle Multitenant databases. Additionally, Cohesity Oracle Adapter provides an incremental forever backup approach eliminating the need for regular expensive full backups. For more details, see Oracle Data Protection Deployment Using Cohesity Oracle Adapter.

Cohesity Oracle RMAN NFS Target: Cohesity can be presented as an NFS volume to be used as inline-deduplication Oracle databases backup target. Oracle RMAN can use the Cohesity on-premise or multi-cloud data platform as a deduplication target and automatically leverage Cohesity's QoS policy for optimal data tiering as well as global deduplication and compression to reduce secondary storage consumption. For more details, see Oracle Data Protection Deployment Using RMAN NFS Target.

Deciding an Oracle Data Protection Method

The decision on which method to use for data protection of Oracle on Cohesity depends on the operating system, Oracle configuration, and type of database. For self-managed Cohesity deployments either on-prem or in the cloud, use the following decision process to choose an Oracle Data Protection Method.

You might choose to use the **SBT Plug-in** if:

1. You want to use source-side dedupe to reduce network load.
2. You don’t want to create or manage NFS mounts for data protection.
3. You cannot use the NFS protocol in your environment.
4. The SBT Plug-in is supported for Linux and AIX operating systems.

Select a Deployment Option

Use the table below to choose a Deployment Option for Oracle Data Protection based on your environment.

Table 1: Oracle Data Protection Selection Based on Operating System

OPERATING SYSTEM	SINGLE INSTANCE	RAC	STANDBY / DATAGUARD	MULTI-TENANCY (CDB/PDB)	ORACLE TDE
Linux (RHEL, Centos, OEL 6/7/8), (SuSE/SLES 11, 12, 15)	Oracle Adapter*, **RMAN SBT**, or RMAN NFS Target				
Windows (2012, 2012 R2, 2016, 2019)	Oracle Adapter **RMAN SBT**, or RMAN SMB Target	**RMAN SBT**, or RMAN SMB Target			
AIX 7.1, 7.2	Oracle Adapter, **RMAN SBT**, or RMAN NFS Target		**RMAN SBT**, or RMAN NFS Target		
HPUX (All)	RMAN NFS Target				
Solaris (All)					
Other DB and OS (Ex - Oracle 10, RHEL 5)					

* CDB/PDB supports only 6.6.0x and later.

* Oracle supported by version (Oracle Adapter(11gR2 and above), SBT(11gR2 and above), RMAN-NFS(all version)

* See Cohesity Software Support Matrix to validate your specific configuration details.

-

Using Cohesity Oracle SBT Plug-in

This section focuses on the use of Cohesity Oracle SBT Plug-in for data protection of Oracle.

NOTE: Instructions in this document are for version 6.6, however, some screenshots have taken using version 6.5.1.

High-level Workflow

1. **Install the Cohesity Oracle SBT Plug-in** on the Linux, AIX, or Windows Host.
   a) Install Cohesity Oracle SBT Plug-in (For Linux).
   b) Install Cohesity Oracle SBT Plug-in (For AIX).
   c) Install Cohesity Oracle SBT Plug-in (For Windows).
2. Download the bash shell scripts from Github to generate the customized RMAN SBT script and run the RMAN backup, duplication, and restore. This step is optional, but highly recommended.
3. **Create a Cohesity View** as the target where the databases need to be backed up.
4. **Create a Cohesity Policy**.
5. **Create a Cohesity View Protection Job**.
6. **Create and Test the Backup Script** with the proper parameters (the script can be the customer's own script or the downloaded script from Github).
7. **Schedule the Backup with Cohesity Remote Adapter** on Cohesity or external scheduler by running the script from the previous step.

   Cohesity Remote Adapter: Cohesity Oracle Remote Adapter is a built-in component of Cohesity multi-cloud data platform that offers flexible management of scripts running on remote hosts. This allows the Cohesity cluster to manage data protection processes and schedules as well as provide consolidated logs of all related activities. For more information, see Remote Adapter.

   Bash shell scripts have also been developed for use in conjunction with either Cohesity Remote Adapter, other cron jobs or schedulers. These scripts provide customers with quick and easy methods to start Oracle RMAN backup, duplication, and recovery; and are available via GitHub here: https://github.com/diana-hui-yang/rman-cohesity/tree/master/sbt.

   NOTE: RMAN scripts are generated and executed by the bash shell script based on the Cohesity cluster configuration (IPs and View) in the customer environment.

Preparations

Open Firewall Ports

1. Verify that the following ports are open on the Cohesity cluster:
   - 11113 for gRPC (default SBT protocol)
   - 11111 for SunRPC (alternative SBT protocol)
2. To test whether the ports are open or not, depending on what packages are installed on the server, use one of the following commands.

> **NOTE**:
> The examples shown assume that one of the Cohesity node VIPs is 10.99.1.65.
>
> nc -zv 10.99.1.65 11113
>
> curl -XGET 10.99.1.65:11113/flagz
>
> telnet 10.99.1.65 11111
>
> nmap -Pn -p 11111 10.99.1.65

Download Scripts from GitHub

1. On the database server or on a centralized server, create a directory for the scripts.
2. Download scripts from GitHub and place them in the desired directory (https://github.com/diana-hui-yang/rman-cohesity/tree/master/sbt) as an Oracle user.
   a) If your Oracle server has access to GitHub, you can download the Linux scripts from the following link: https://github.com/diana-hui-yang/rman-cohesity/blob/master/sbt/linux-sbt-download.
   b) Download the AIX scripts from the following link: https://github.com/diana-hui-yang/rman-cohesity/blob/master/sbt/aix-sbt-download.

Download SBT Library and Tools

1. See SBT Library download link: https://downloads.cohesity.com/oracle_sbt/RPC-Library/6.5.1.

   NOTE: You may have to login first. Once you login, you need to copy and paste the link again on the browser.

2. Copy the SBT Library to the following folder: **<script directory>/rman/lib**.

3. See tools download link: Linux Tools or AIX Tools.

   NOTE: Please reach out to Cohesity Support or your Account SE if you are unable to access this link.

4. Download the **sbt_list** and **sbt_perf_test** tools to the **<script directory>/rman/tools** folder.

   NOTE: These tools are very useful and highly recommended for managing retention, measuring performance, and troubleshooting.

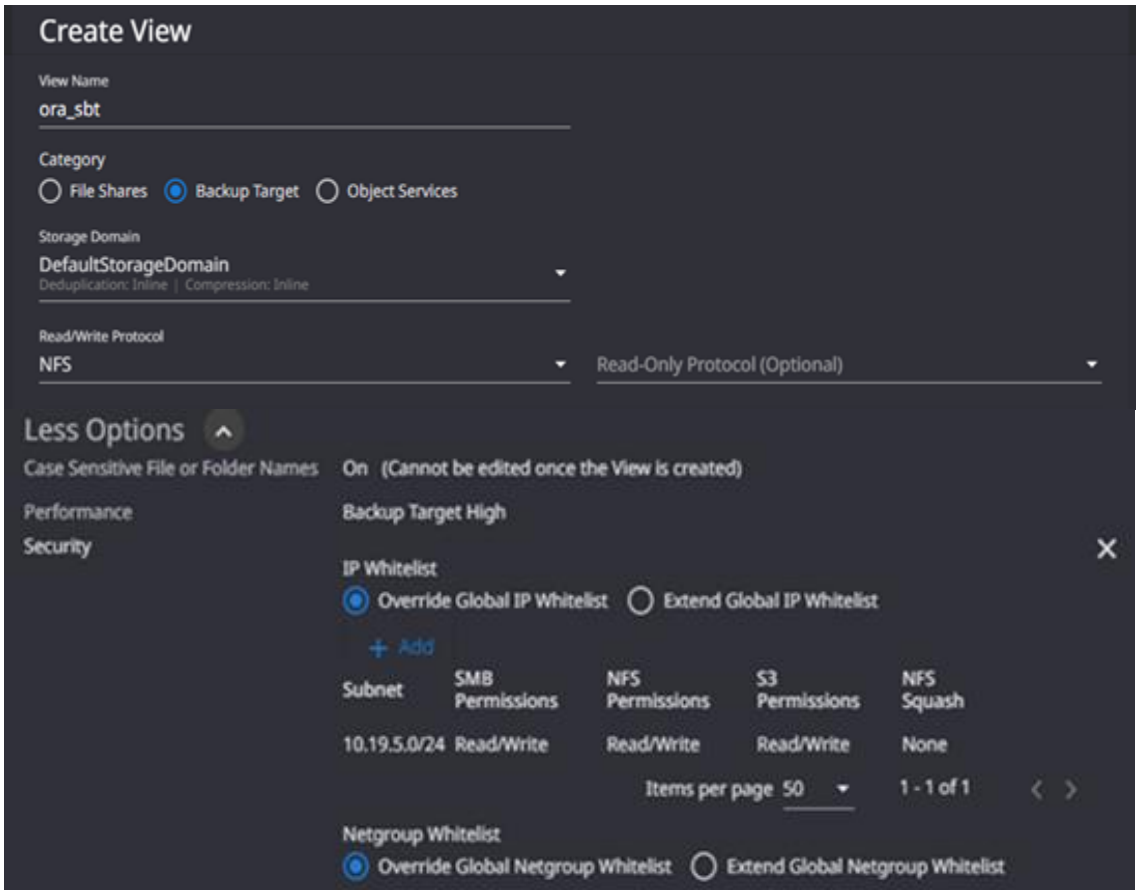
Create a Cohesity View

To create a Cohesity View follow the steps given below.

To create a Cohesity View

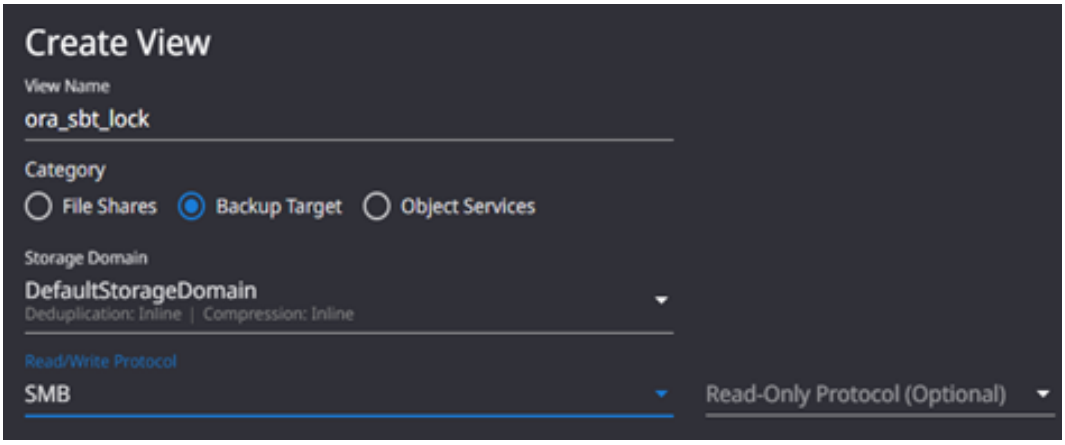
1. On the Cohesity Dashboard, select **File Services > Views > All Views**.
2. In the **All Views** page, click **Create > View**.

3. Populate the following fields on the **Create View** screen with the following settings:
   a) **View Name**: ora_sbt
   b) **Category**: Backup Target
   c) **Storage Domain**: DefaultStorageDomain or (customer specified)
   d) **Protocol**: NFS
   e) **Case Sensitive File or Folder Names**: On
   f) Performance: **Backup Target High**
   g) **IP Whitelist (IP Allowlist)**: Override Global IP Whitelist
   h) Add the IP Address of the Oracle Host Server to be able to access this View (e.g. 10.19.5.0 subnet) and provide Read/Write permissions for SMB, NFS, and S3.
   i) **NFS Squash**: None
   j) **Netgroup Whitelist**: Override Global Netgroup Whitelist

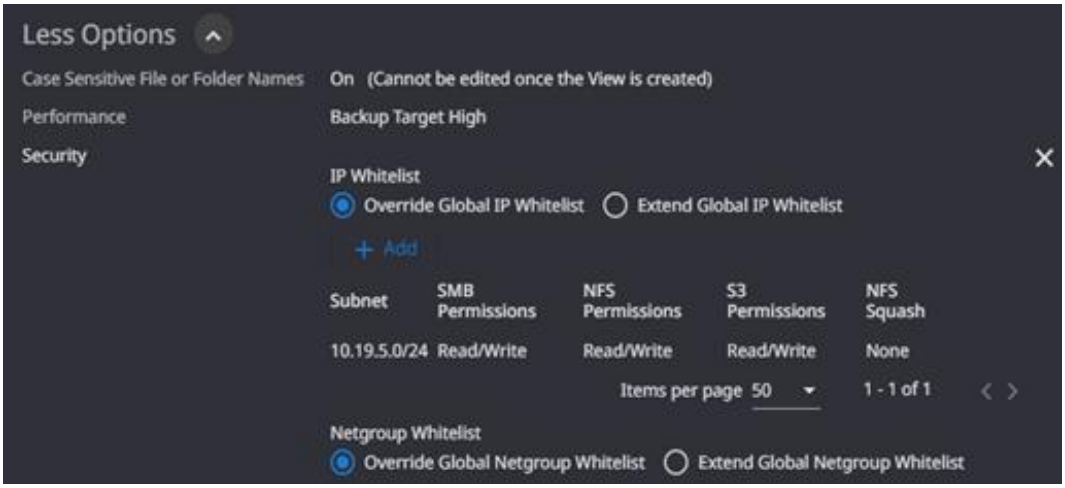


To create a Cohesity View with DataLock Enabled

1. On the Cohesity Dashboard, select **File Services** > **Views** > **All Views**.
2. In the **All Views** page, click **Create** > **View**.
3. Populate the following fields on the **Create View** screen with the following settings:
   a) **View Name**: ora_sbt_lock
   b) **Category**: Backup Target
   c) **Storage Domain**: DefaultStorageDomain or (customer specified)
   d) **Read/Write Protocol**: NFS



   e) **Case-sensitive File or Folder Names**: On
   f) Performance: **Backup Target High**
   g) **IP Whitelist** (IP Allowlist): **Override Global IP Whitelist**
   h) Add the IP Address of the Oracle Host Server to be able to access this View (e.g. 10.19.5.0 subnet) and provide Read/Write permissions for SMB, NFS, and S3.
   i) **NFS Squash**: None
   j) **Netgroup Whitelist**: **Override Global Netgroup Whitelist**



k) **DataLock Mode**: **Enterprise**

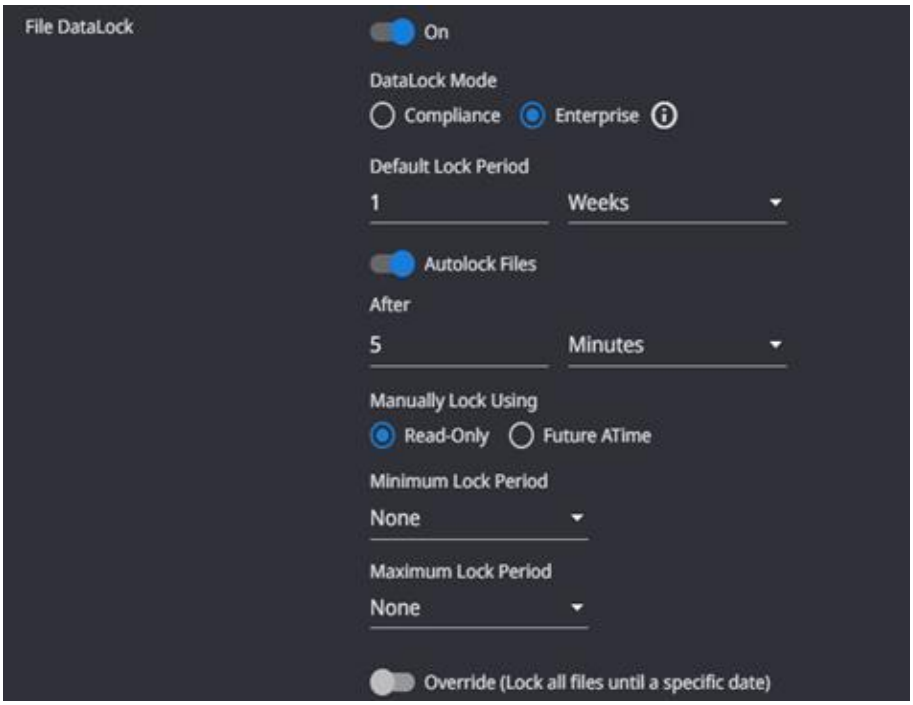
l) **Default Lock Period**: 1 week

m) **Autolock Files**: On

n) **After**: 5 Minutes

o) **Manually Lock Using**: **Read-Only**

p) **Minimum and Maximum Lock Period**: **None**



NOTE: Cohesity highly recommends creating a view with Datalock enabled in order to enhance security. The Datalock period can be as long as the retention or at minimum one week. For more information, see Create a View.

Protect the View

As the next step, you must set up a View Protection Job on Cohesity. The protection job will make regular copies of the backups contained in the view. The protection job can be configured to replicate the view or archive it to the cloud.

Create a View Protection Job

1. Complete the steps as outlined in the following topic: Add or Edit a Protection Group for a View.

NOTE: This must be setup by a Cohesity Administrator or an Oracle DBA user who has a Cohesity login account and is assigned to a “self service” role.

Create a Protection Policy for the Oracle View

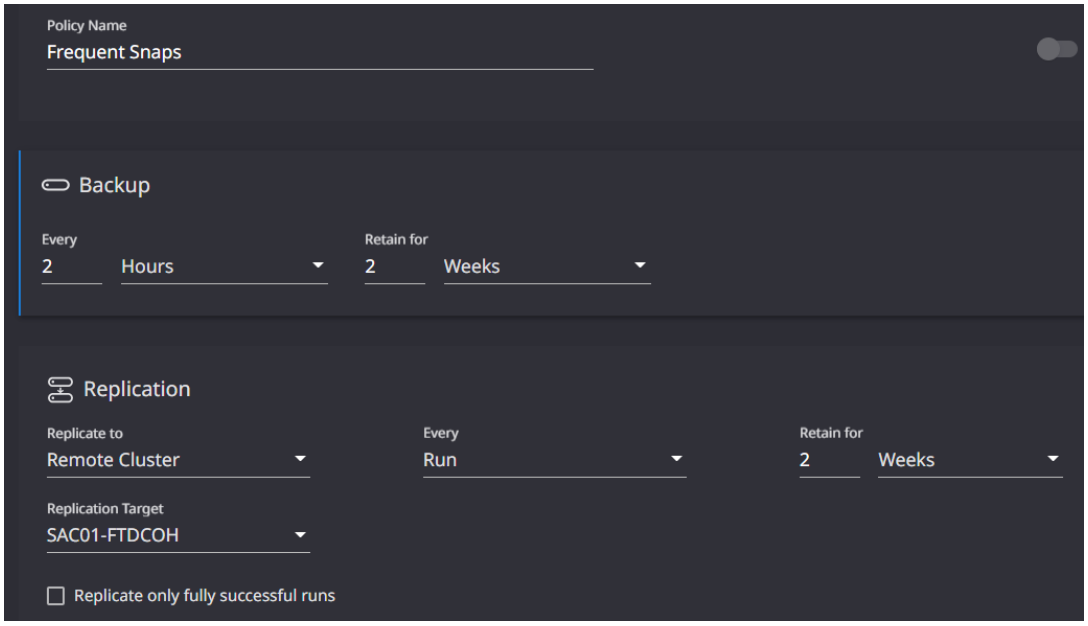
The steps for creating a Protection Policy are included in our online documentation. See Create or Edit a Standard Policy for more information. The details provided below offers additional guidance for creating the right policy for an Oracle Data Protection using SBT Plug-in.

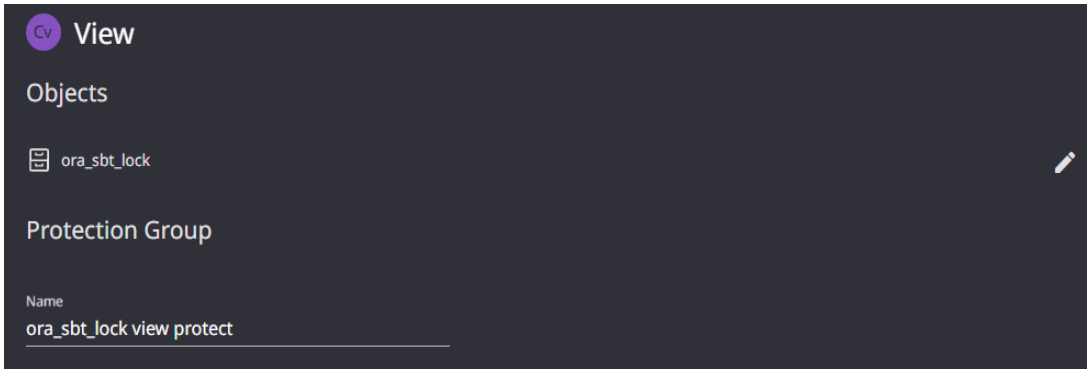
1. In the Cohesity Dashboard, select **Data Protection** > **Policies**.
2. Click **Create Policy** located at the top right of the page.
3. Specify a protection **Policy Name**.
4. Optional: Set View DataLock as an additional data lock option. It is different from Cohesity FileDataLock. It prevents the snapshots of the view to be unintentionally deleted. Most organizations picked FileDataLock option for SBT backup. If View DataLock is required, toggle on **DataLock** to lock the policy and any snapshots that belong to Protection Groups associated with the policy.

NOTE:

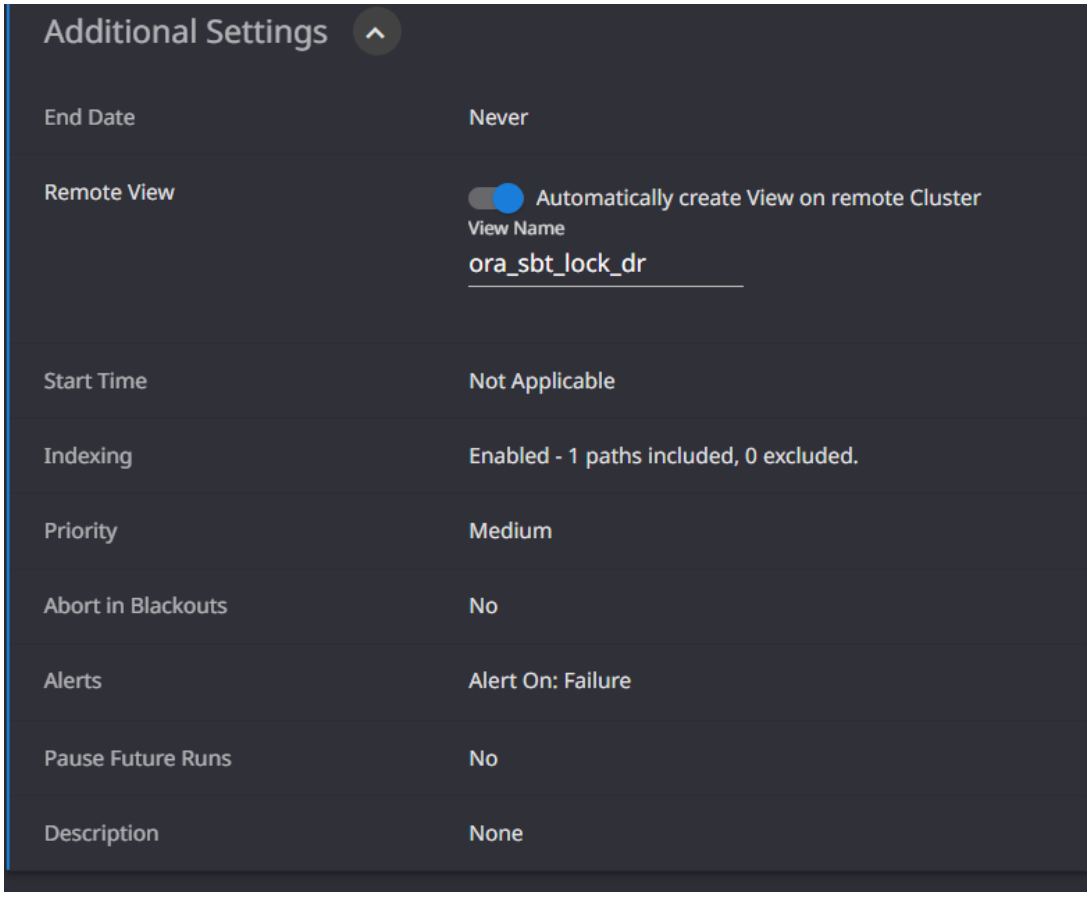
The Create a Protection Policy for the Oracle View that is shown in this example takes a snapshot of the view every 15 minutes and backups are retained for 2 weeks with replication setup.

This Protection Policy will be used to take a snapshot of the NFS View where the Oracle SBT library data is written day-over-day and stored using Oracle backup retention. This protection policy retention has no bearing on the Oracle backup retention. Please set retention for this Protection job to minimum days. This is different from Oracle backup retention policy. That policy is managed either by RMAN script or a separate Unix script

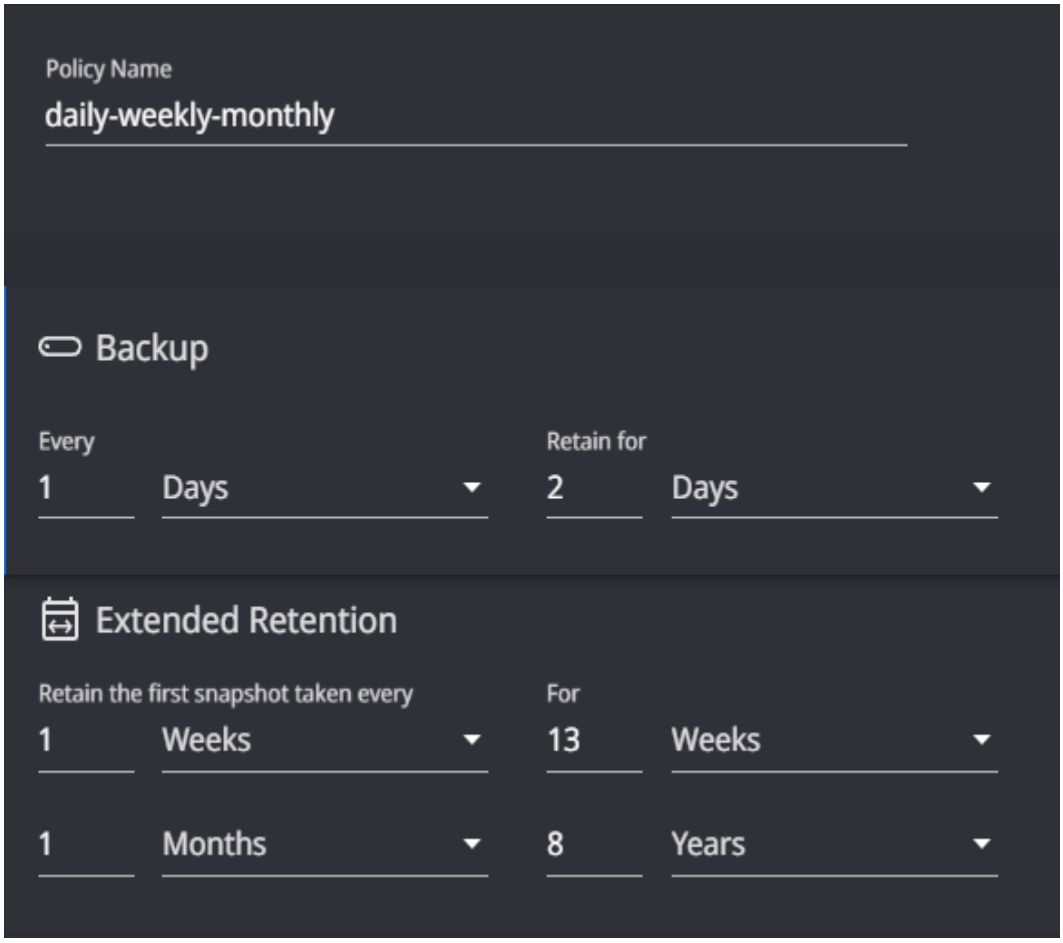




5. Click **Additional Settings** to create the following additional settings.
   - **Remote View** will have the data that is updated automatically on remote cluster with the latest backup data.
   - **Alerts** allow emails to be set when backup fails.



6. Create a policy with the desired attributes. For example:
   - Daily backup - Retained for 2 Days
   - Extended Retention - Weekly Backup - Retained for 13 weeks
   - Monthly Backup - Retained for 8 Years



Download Backup Script

You can download backup scripts from GitHub. These can be modified to fit your environment, or you may create your own.

If you have not done so already, please complete steps as outlined in the Preparation section:

- Download Scripts from GitHub.
- Download SBT Library and Tools.

Back up Oracle Using Cohesity Oracle SBT Plug-in

Setup the backup scripts

To back up Oracle using Cohesity Oracle SBT Plug-in:

1. Launch the backup script (`backup-ora-coh-sbt.bash`).
2. The syntax will display on the screen when it is run without any input.

> **NOTE**: For customers who prefer to use their own backup script, this command has an option "-w yes" to generate RMAN command syntax based on the customer's environment with options they can choose. These customers can then modify their own script based on the RMAN syntax generated by this script. The option explanation and example are listed on GitHub link. There are also detailed backup examples with parameters in the **backup-ora-coh-sbt/backup-example** folder on the GitHub site.

> **NOTE**: The default RMAN channels the script generates is four. The number of channels chosen is determined by the desired performance and infrastructure. Four channels in most cases meet the performance requirements.

> **NOTE:** RMAN "delete obsolete" command is used in this script to delete expired backups. By default, it is commented out. Please check the Oracle Bug report "Oracle Bug 29633753" and apply the necessary fixes before you uncomment that line

3. Choose one of the following options to Manage **Oracle Backup Retention**.
   a) RMAN "delete obsolete"
   b) **Sbt_list Tool** can delete the backup files older than the retention period. Sbt_list is a tool created by Cohesity Engineering. The backup script calls sbt_list to delete the expired files.

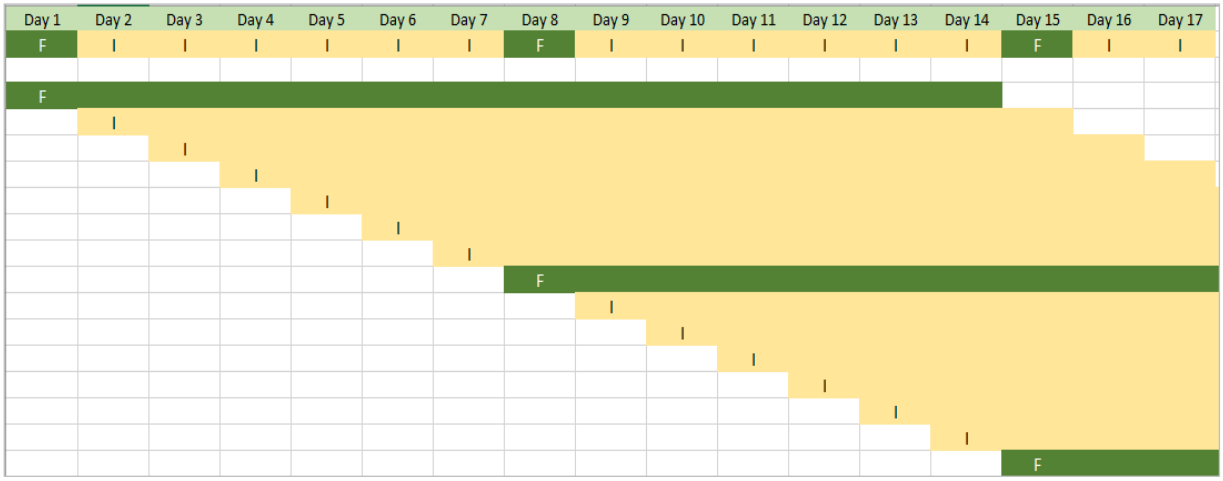
> **NOTE**: The second option is highly recommended as it can guarantee that expired files are deleted.

Considerations When Using Sbt_list Tool to Delete Expired Backup Files

The **sbt_list** tool deletes expired backup files by the unit of days specified only.

For example, if a DBA is doing weekly full and daily incremental backups, the retention time used in this script should be Oracle retention plus 8 days. The reason is that when a full backup is deleted, the subsequent incremental backup before the next full backup becomes invalid. Therefore, if a DBA wants to restore an Oracle database within 7 days, the retention input for the script should be 15 days (as the following diagram illustrates). On day 15, the full is deleted. DBA can only restore the data within 7 days.

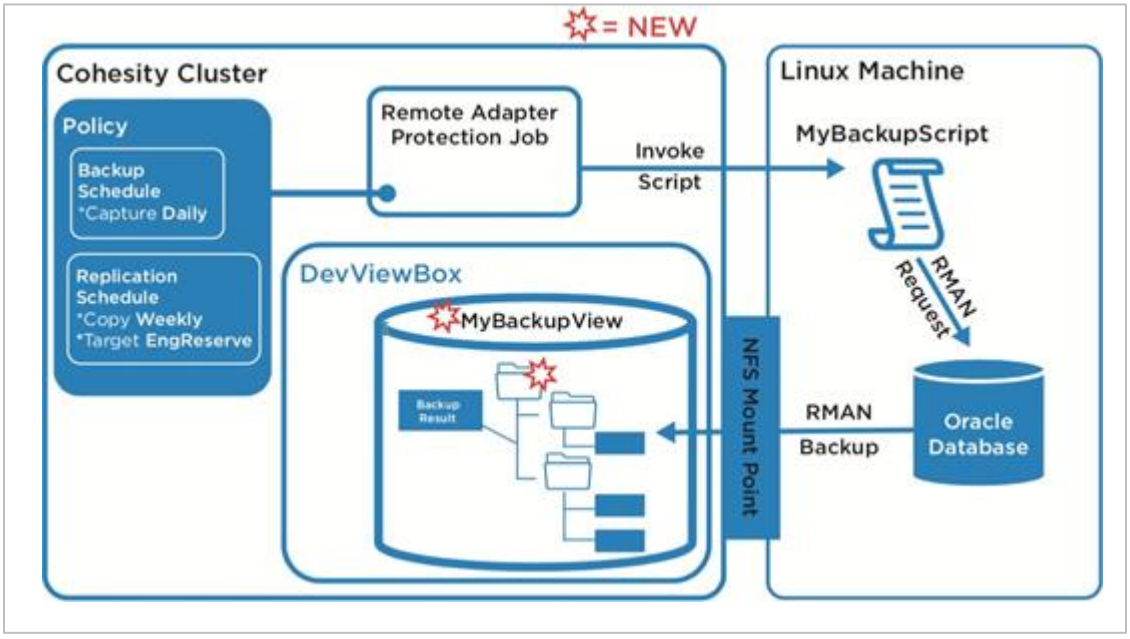
NOTE: Most customers store the backup script on the individual Oracle servers. However, it can also be stored and run from a centralized location.



Schedule the Oracle Backup

1. Cohesity recommends scheduling the backup script with the Cohesity Remote Adapter. However you may use another scheduling tool.

   Figure 1: Schedule the Oracle Backup

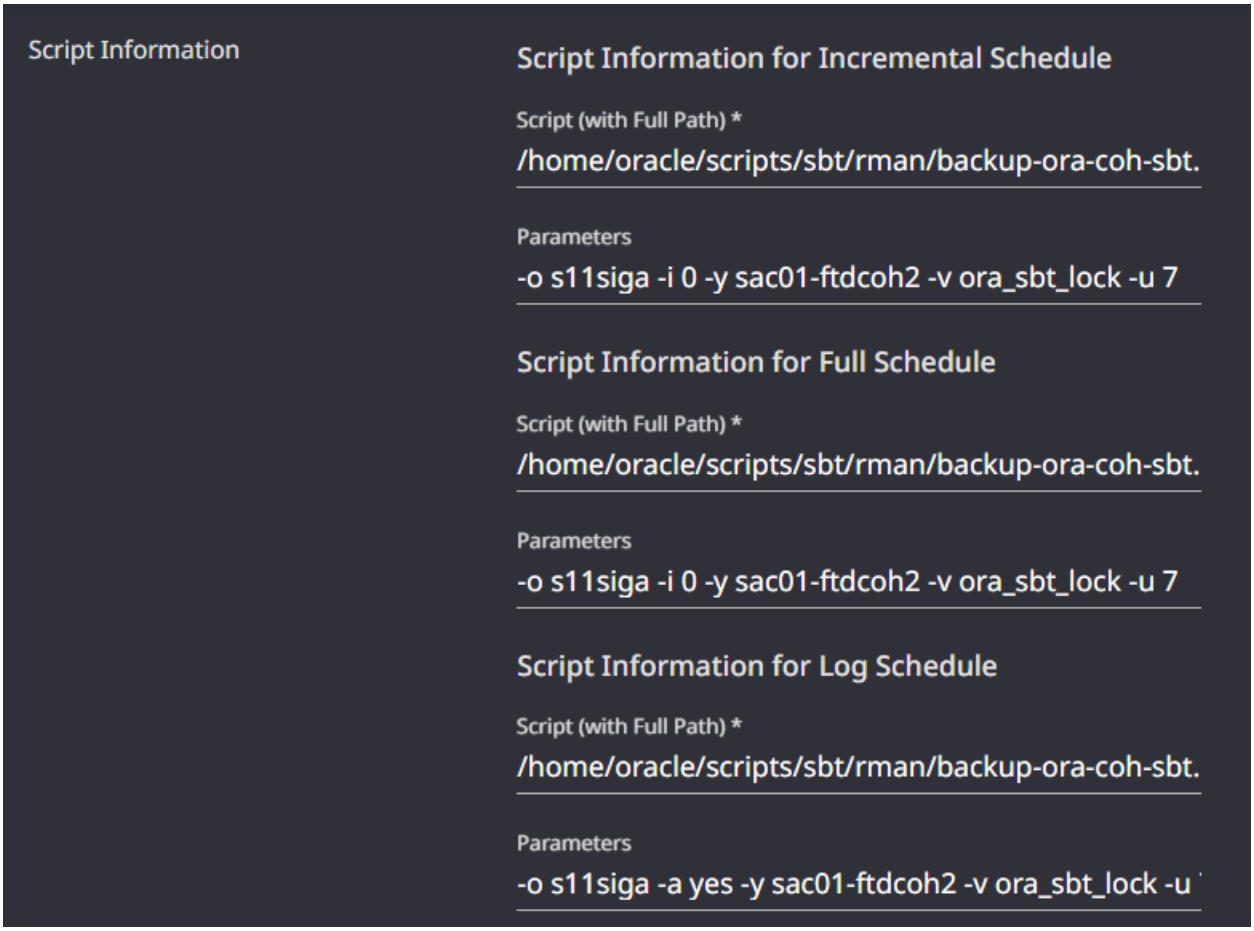


   For instructions on how to Create a Remote Adapter Job, see Add or Edit a Remote Adapter Protection Group.

For more information on scheduling Cohesity Remote Adapter job to run Oracle backup, see About the Remote Adapter. This can be setup by a Cohesity Administrator or an Oracle DBA user who has a Cohesity login account and is assigned to the “Self Service Data Protection” role.

Using Cohesity Remote Adapter

When using Cohesity Remote Adapter, only one script will run in one remote adapter protection job. For most cases, one job per database can meet the requirement. Customers can schedule incremental, full, and archive logs backup in one remote adapter protection job.



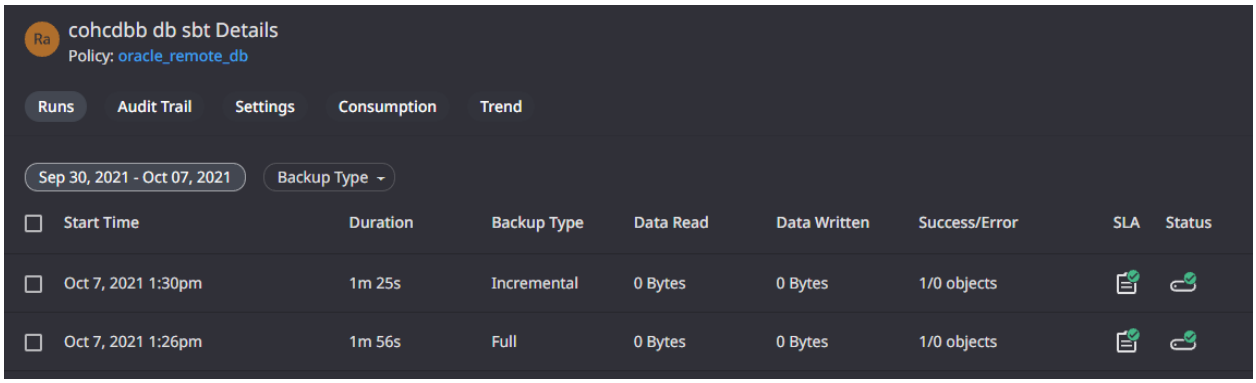
If archive logs backup should be allowed to run concurrently with full or incremental backup, two Remote Adapter Jobs are created per Oracle Database. The following is an example to create two remote adapter protection jobs

1. The first Remote Adapter Job is for the Oracle Database backup.
2. The second Remote Adapter job is for the Oracle Archive Log backup.

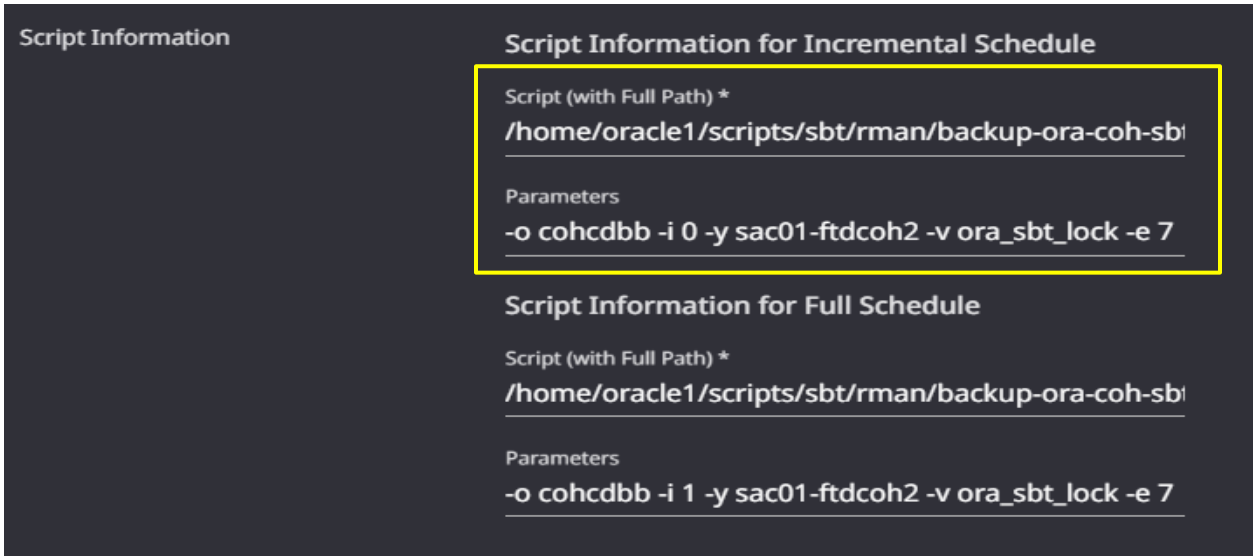
> **NOTE:** Both these backup jobs can run in parallel.

The following examples illustrate the job details, script, and parameter settings used for this pair of Remote Adapter jobs needed to protect an Oracle Database (EXAMPLE db name= **cohcdbb)**.

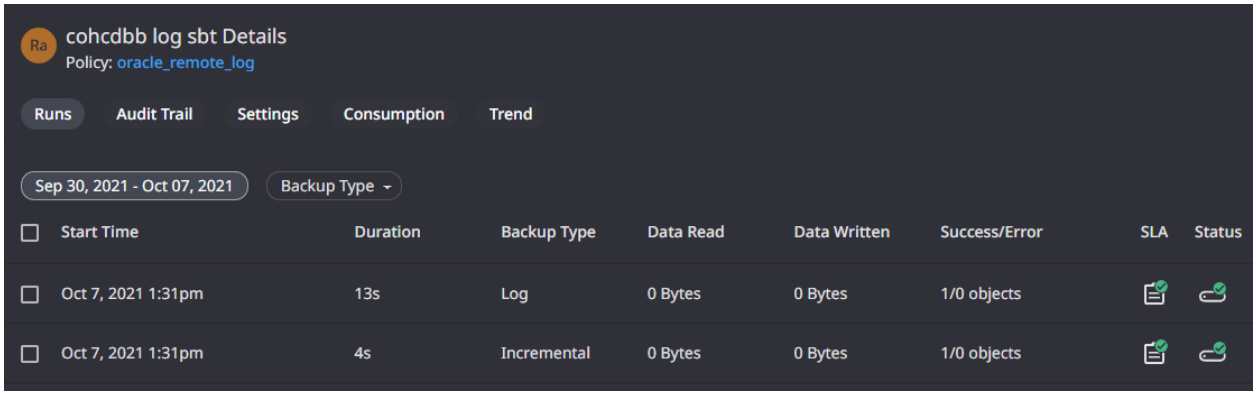
- **Example 1:** shows the details of both the Full and Incremental database backups.



When creating the Remote Adapter job, all the information shown under **Script (with Full Path)** and **Parameters** must be completed based on the customer environment. The following displays the Script and Parameter Information for both the Full and Incremental **cohcdbb** Remote Adapter backup jobs.

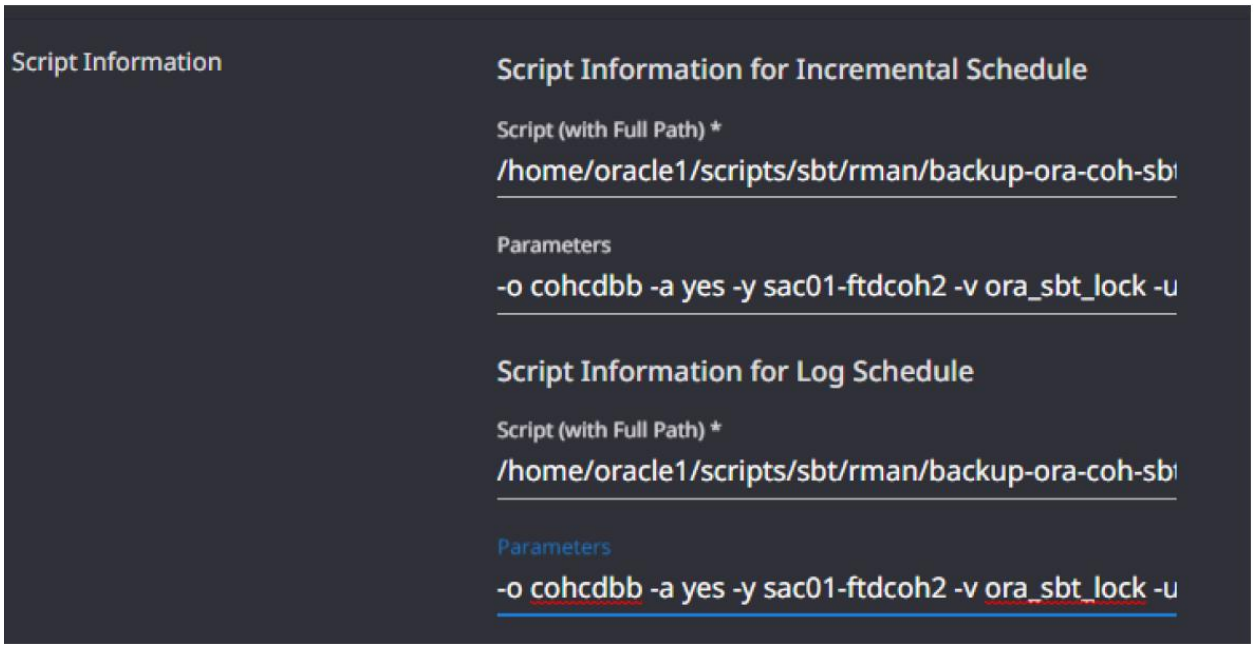


- **Example 2**: shows the details of the Archive Log Backup job details, script, and parameters.



In the example above, the Cohesity schedule requires an incremental backup to run. The incremental job entered here is an Oracle Archive Log backup job. Alternatively, an echo.bash job can be created. Here is an example of the script:

```
echo "This schedule job is for Oracle log backup"
```



Additionally, the following gflag needs to be enabled to allow both jobs to run in parallel. This needs to be handled by the Cohesity Administrator.

```
iris_cli cluster update-gflag gflag-
name="magneto_master_enable_puppeteer_backup_parallelism" gflag-value=true
service-name=magneto effective-now=true reason="enable parallel backup and
archive"
```

The following gflag allows Remote Adapter to check the jobs for as long as 24 hours.

```
iris_cli cluster update-gflag gflag-
name="magneto_puppeteer_poll_interval_secs" gflag-value=300 service-
name=magneto effective-now=true reason="Increase polling interval to 5
minutes"

iris_cli cluster update-gflag gflag-name="magneto_puppeteer_poll_retries"
gflag-value=288 service-name=magneto effective-now=true reason="Increase
polling frequency from 3 to 288"
```

For parameter examples, see https://github.com/diana-hui-yang/rman-cohesity/tree/master/sbt/backup-ora-coh-sbt/backup-example

Creating a Copy of the Oracle Database Using Cohesity SBT Backup

To create a copy of an Oracle database using the SBT Backup method, complete the following steps on the Oracle Server.

Prepare the Duplication/Copy

1. Create the initialization parameter file for the duplicated database. The init file example can be found in the Appendix section Sample init file.
2. Test Oracle connection string to source database (database that backup was taken).

NOTE: When using the sbt_tape option to perform duplication, RMAN needs to be able to connect to the source database (RMAN considers this database as target database). Use the following command to test the connection:

```
tnsping <oracle connection string>
rman target sys/<password>@<oracle connection string>
```

NOTE: Oracle database files can be on ASM or file directories. RMAN uses several parameters to direct the files to new location for the duplicated database if it is ASM: DB_CREATE_FILE_DEST in pfile, SET NEWNAME FOR DATABASE and SET NEWNAME FOR DATAFILE in RMAN duplicate command. “SET NEWNAME FOR DATAFILE” precedes “SET NEWNAME FOR DATABASE” which precedes “DB_CREATE_FILE_DEST”.

If the database file is using file directories, two parameters in the initialization parameter file can be used to change datafiles and logs to new directories. These two parameters are DB_FILE_NAME_CONVERT and LOG_FILE_NAME_CONVERT.

Run Duplicate Scripts

Same as the backup script, the duplicate script (`duplicate-ora-coh-sbt.bash`) syntax will display on the screen when it is run without any input. The syntax explanation and example are listed on GitHub link. The script will create rman commands first in **log/<host>/<database>.duplicate…rcv** file. It can be previewed when it is run with `-w yes` option.

Creating a Duplicate Standalone or CDB Oracle Database

On the Oracle server, create an oracle duplication script with all the proper parameters. The parameter examples are located in: https://github.com/diana-hui-yang/rman-cohesity/blob/master/sbt/duplicate-ora-coh-sbt/duplicate-example/dup-cohcdbr1-cohcdbr2.bash.

The source database is cohcdbr1 and the target database is cohcdbr2.

Duplicate PDB Oracle Database

On the Oracle server, create an oracle duplication script with all the proper parameters. The parameter examples are: https://github.com/diana-hui-yang/rman-cohesity/blob/master/sbt/duplicate-ora-coh-sbt/duplicate-example/dup-cohcdbr1-cohcdbc-cohpdbr1.bash.

- The **source CDB** database is **cohcdbr1**.
- The **target CDB** database is **cohcdbc**, and the PDB database is cohpdbr1.
- The script dup-cohcdbr1-cohcdbc-cohpdbr1.bash will duplicate PDB database cohpdbr1 from CDB database cohpdbr1 to CDB database cohpdbc.

Restoring an Oracle Database Using Cohesity Oracle SBT

Run the Restore Scripts

Same as the backup script, the restore script (restore-ora-coh-sbt.bash) syntax will display on the screen when it is run without any input. The syntax explanation and example are listed on GitHub link.

The script will create rman commands first in log/<host>/<database>.restore…rcv file. It can be previewed when it is run with `-w yes` option

Validate the Restored Database

When this script runs without "`-f yes`", it will just run RMAN "restore validate." It will not overwrite the existing database. It is useful when testing restore performance. RMAN reads the backup data from Cohesity Data Cloud and writes to /dell/null.

NOTE: This feature is used when customers want to know what performance to expect WITHOUT actually restoring and overwriting the existing database.

Restore Database Datafile

The "`-l no`" or no "`-l`" option of the script will restore the database assuming the control file is still intact.

NOTE: This will OVERWRITE the existing data file.

Complete Restore

In order to do a complete database restore, both "-f yes" and "-l yes" should be specified. The script will restore spfile and controlfile from the backups first, then restore/recover database. A simple init file for this database should be created first.

NOTE: You should use complete restore only if the customer has lost everything and are restoring the Control File and Data File.

Performance and Troubleshooting

This section gives you an overview of the performance factors and troubleshooting tips that you need to consider when you use the Oracle SBT Plug-in for Oracle data protection.

Performance Considerations

Correct Cohesity Sizing

Sizing the Cohesity solution should consider both storage requirements and performance requirements. Periodically, most likely weekly, a full backup is required for SBT type Oracle backups. If sizing only includes an incremental backup method (such as through Oracle adapter use), it will lead to undersizing the number of Cohesity nodes and the customer will experience slow performance.

Number of Channels

In most cases, a higher number of channels will lead to higher performance. However, this is not always true for all situations. A single channel should allow for an optimal performance throughput range (no bottleneck anywhere, no other workload). If higher performance is desired, more Cohesity nodes can easily be added to the Cohesity cluster. However, more channels require more CPU on Oracle server as well, as there is a limit on the number of RMAN channels before it starts to impact Oracle database performance for their users.

Troubleshooting Tips

RMAN Backup Failure

The most common reasons for RMAN backup failure using Cohesity SBT are the following:

- **The IPs are not in Cohesity Allowlist**. Often there are several IPs on one Oracle server. They can also be in different subnets. When backing up Oracle RAC using multiple nodes, the IPs on all nodes should be in Cohesity Allowlist
- **The view name is incorrect**. On a Linux server for a View created as NFS 3, use the "`showmount -e <cohesity cluster name>`" command to find the correct View Name.

To troubleshoot RMAN Backup Failure:

1. **Run the Oracle "sbttest" command** — as shown in the example below:

```
ora_sbt_lock
Cohesity VIP: 10.19.0.90 (can use cohesity cluster name too)

mount_path=10.19.0.90:/ora_sbt_lock vips=10.19.0.90 sbttest test -libname
/home/oracle/scripts/sbt/rman/lib/libsbt_linux_x86_64.so
```

2. **Review the RMAN backup logs** — this can provide a lot of information about what went wrong. The RMAN logs are located in the **<script directory>/log/<server name>** directory when using Cohesity scripts.

Slow Backup

If the RMAN backup performance is below the expected number based on Cohesity node type and number of nodes, the following factors should be investigated further:

1. **Network bandwidth between the Oracle server and Cohesity**. The best tool to measure this is the **iPerf3** tool.

2. **Number of CPUs on Oracle server**. 4 CPUs is considered to be low.

3. **The primary storage Oracle database is using**. Flash storage can provide very high throughput.

Cohesity engineers have also created an **sbt_perf_test** tool to test SBT performance between Oracle to the Cohesity node or even from Cohesity node to Cohesity node. RMAN backup performance can be predicted based on these numbers. This tool is described further in Appendix - sbt_perf_test usage.

Troubleshooting Tools

This section describes the available troubleshooting tools used on Oracle Linux and AIX servers to test performance with Cohesity.

The sbt_perf_test usage Tool

This tool writes output to **/tmp/trace.txt** file:

```
# Test 10 GB data write/read performance with 4 concurrent processes

./sbt_perf_test --sbt_test_mount_path sac01-ftdcoh2:/ora_sbt_lock --
sbt_test_vips 10.19.2.90 --sbt_test_block_count 40000 --
sbt_test_num_channels 4

# Test 2 GB data write/read performance
```

```
./sbt_perf_test --sbt_test_mount_path sac01-ftdcoh2:/ora_sbt_lock --
sbt_test_vips 10.19.2.90 --sbt_test_block_count 8000

# Test 2 GB and with source side dedup off

./sbt_perf_test --sbt_test_mount_path sac01-ftdcoh2:/ora_sbt_lock --
sbt_test_vips 10.19.2.90 --sbt_disable_source_side_dedup true --
sbt_test_block_count 8000

# Read throughput
```

The tool does write first, then read. If the tool is run with just 1 channel, you can view the throughput result using the following command. The first line is write throughput. The second line is read throughput. If there are four channels, you need to change “tail -2” to “tail -8”.

```
grep Throughput /tmp/trace.txt |tail -2
```

EXAMPLE: Following is the output example of 4 channels and with source-side depute on. The first four lines are write throughput per channel. The last four lines are read throughput per channel:

```
Throughput (Logical/Physical): 187.87 / 23.84 MB/s Elapsed: 53.00 seconds.
Throughput (Logical/Physical): 181.20 / 20.98 MB/s Elapsed: 55.11 seconds.
Throughput (Logical/Physical): 175.48 / 18.12 MB/s Elapsed: 56.78 seconds.
Throughput (Logical/Physical): 166.89 / 6.68 MB/s Elapsed: 59.69 seconds.
Throughput (Logical/Physical): 224.11 / 224.11 MB/s Elapsed: 44.57 seconds.
Throughput (Logical/Physical): 221.25 / 221.25 MB/s Elapsed: 45.11 seconds.
Throughput (Logical/Physical): 225.07 / 225.07 MB/s Elapsed: 44.30 seconds.
Throughput (Logical/Physical): 227.93 / 227.93 MB/s Elapsed: 43.87 seconds.
```

Appendix A: Oracle parameters

Sample init file for database w2cdbbr1 (initw2cdbbr1.ora)

```
w2cdbbr1.__data_transfer_cache_size=0
w2cdbbr1.__db_cache_size=7918845952
w2cdbbr1.__inmemory_ext_roarea=0
w2cdbbr1.__inmemory_ext_rwarea=0
w2cdbbr1.__java_pool_size=33554432
w2cdbbr1.__large_pool_size=67108864
w2cdbbr1.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
w2cdbbr1.__pga_aggregate_target=3355443200
w2cdbbr1.__sga_target=10066329600
w2cdbbr1.__shared_io_pool_size=536870912
w2cdbbr1.__shared_pool_size=1476395008
w2cdbbr1.__streams_pool_size=0
*.audit_file_dest='/u01/app/oracle/admin/w2cdbbr1/adump'
*.audit_trail='db'
*.compatible='12.2.0'
*.control_files='/oradata/w2cdbbr1/controlfile/control1.ctl','/oralog/fra/w2
cdbbr1/controlfile/control2.ctl'#Restore Controlfile
*.db_block_size=8192
*.db_create_file_dest='/oradata/w2cdbbr1'
*.DB_CREATE_ONLINE_LOG_DEST_1='/oralog/fra/w2cdbbr1'
*.db_name='w2cdbbr1'
*.db_recovery_file_dest='/oralog/fra/w2cdbbr1'
*.db_recovery_file_dest_size=15162m
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=w2cdbbr1XDB)'
*.enable_pluggable_database=true
*.log_archive_format='%t_%s_%r.dbf'
*.nls_language='AMERICAN'
*.nls_territory='AMERICA'
*.open_cursors=300
```

```
*.pga_aggregate_target=3192m
*.processes=320
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=9574m
*.undo_tablespace='UNDOTBS1'
```

Duplicate set example if source database is built using OMF

```
# Setting
set newname for database to "'/oradata/w2cdbbr1/%b';"
```

Appendix B: Product Documentation

For in-depth details see the below in the online Help:

- Oracle Adapter Requirements
- Cohesity Oracle best Practices
- Oracle Adapter Troubleshooting

Your Feedback

Was this document helpful? Send us your feedback!

About the Authors

This document was authored by the Database Solution Architecture team of: Diana Yang, Dave Porco, Sanjay Dhruv, and Matthew Ellis.

Other significant contributors include:

- Scott Lorenz, Solutions Engineer
- Aditya Tandon, Director, Product Management
- Janell Filkin, Implementation Specialist
- Kevin Hill, Mgr, Solutions Architects
- Saad Jafri, Product Solutions Architect
- Sunil Moolchandani, VP, Product Solutions
- Steve Lee, Technical Staff
- Karthick Radhakrishnan, Director, Technical Solutions Engineering
- Subash Babu, Staff Technology Editor

Document Version History

VERSION	DATE	DOCUMENT VERSION HISTORY
1.2	July 2024	Republishing
1.1	May 2023	Minor updates
1.0	June 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.