COHESITY

Version 1.1

July 2024

Oracle Data Protection Deployment Using RMAN NFS Target

Options and Instructions to Deploy Cohesity Data Protection for Oracle Servers

ABSTRACT

The document provides a workflow to help deploy the Oracle Data Protection using RMAN NFS Target for a customer's environment.

Table of Contents

Tables

Introduction

This document provides a step-by-step guide for customers to deploy the Oracle Data Protection using RMAN NFS for their environment.

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 RMAN NFS Target.

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 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.

Cohesity Oracle Adapter: Cohesity Oracle Adapter integrates with Oracle Recovery Manager (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 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. For more details, see Oracle Data Protection Deployment Using SBT Plug-in.

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 Cohesity as your **Backup Target** if:

1. You want to use or manage your own RMAN Scripts.
2. RMAN scripts are very tightly integrated with your current tools and workflows (rare scenario).
3. You use a version of Oracle or OS vendor/version that the Oracle Adapter does not support. For example, Ex - Oracle 9i or RHEL 5.x or HP-UX. See Support Matrix for more details.

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

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 RMAN NFS Target to Protect Oracle Database

Introduction

This section focuses on using Cohesity Oracle NFS mount target integration option. Bash shell scripts have been developed and are available, via GitHub, to provide customers quick and easy methods to start Oracle RMAN backup, duplication, and recovery. RMAN scripts are generated and executed by the bash shell script based on the Cohesity cluster configuration (ips and view) in the customer environment.

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

NOTE: In order to access the example bash scripts on GitHub, you will first need to register and create an account.

TIP: For a growing library of example scripts, see Example-RMAN-Scripts in Cohesity's Oracle-Scripts GitHub repository, a collection of sample scripts that you can use to write and customize the scripts you need for your environment.

WARNING: GitHub examples are provided on a best effort basis and are not in any way officially supported or sanctioned by Cohesity. The examples in this repository are provided as-is and the author accepts no liability for damages resulting from its use.

Preparation

Open Firewall Ports

1. Verify that the following ports are open on the Cohesity cluster:
   - 111, 2049 for NFS Mount.

   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.

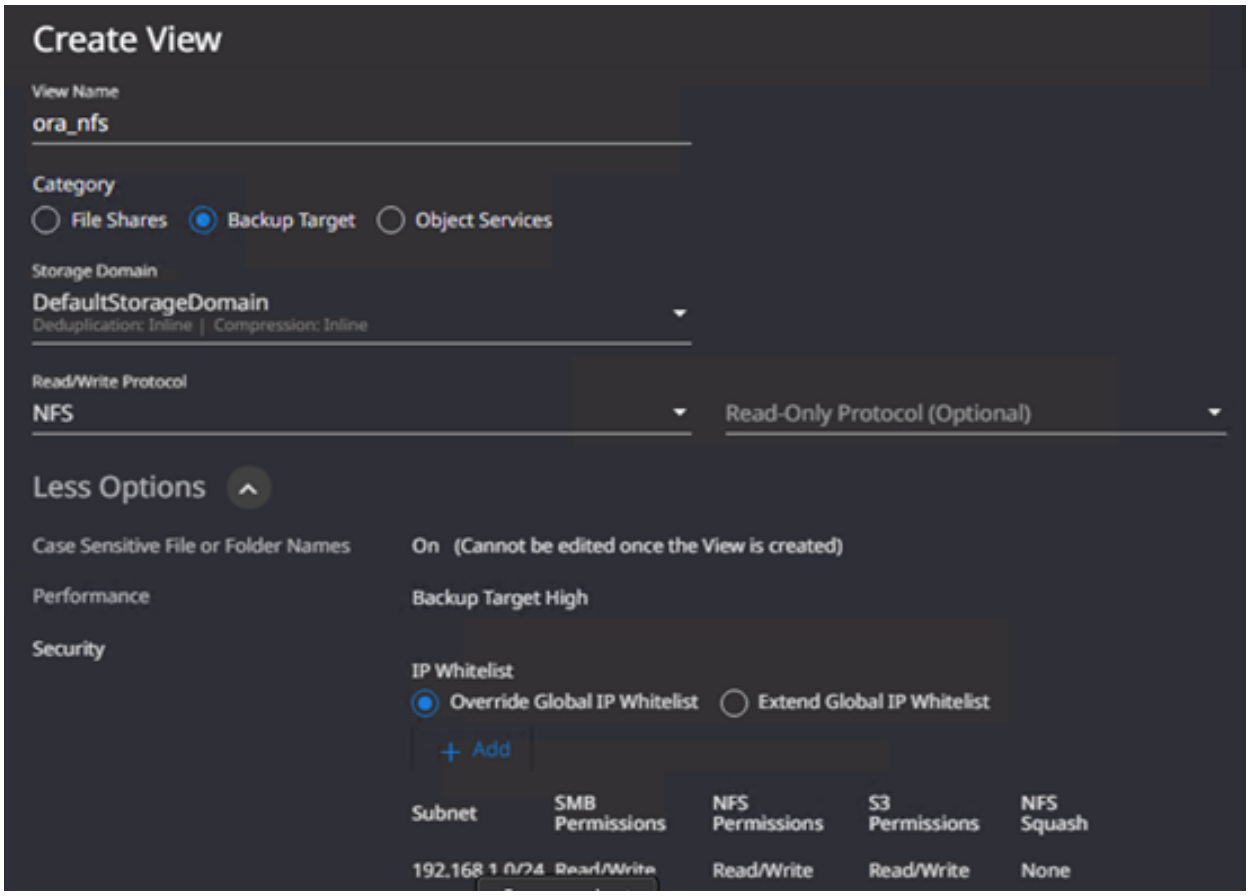
   ```
   nc -zv 10.99.1.65 2049
   telnet 10.99.1.65 2049
   nmap -Pn -p 2049 10.99.1.65
   ```

Create a Cohesity View

To create a Cohesity View, follow the steps given below.

To create a Cohesity View

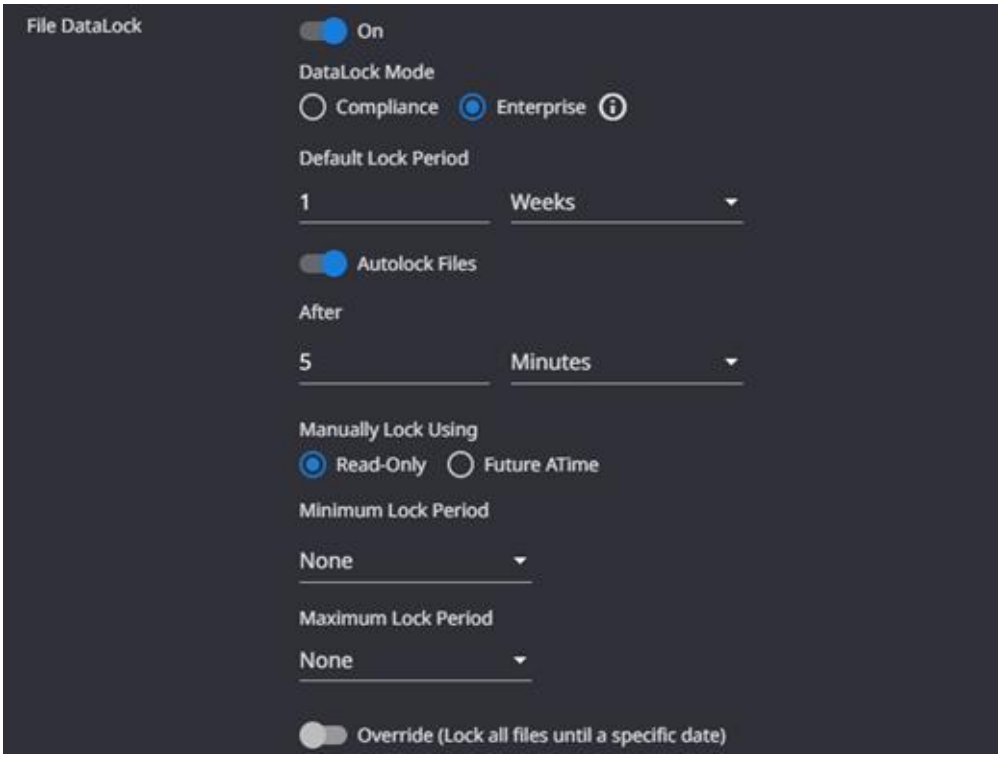
1. On the Cohesity Dashboard, select **File Services** > **Views** > **All Views**.
2. In the **All Views** page, click **Create** > **View**.
3. Populate the following fields on the **Create View** screen with the following settings:
   a) **View Name**: ora_nfs
   b) **Category**: Backup Target
   c) **Storage Domain**: DefaultStorageDomain or (customer specified)
   d) **Protocol**: NFS
   e) Click **More Options** and select **Override Global Allowlist**.
   f) Enter the IP Address of the Oracle servers so that it can mount the NFS Share and write to it.



To Create a Cohesity View with DataLock Enabled

On the Cohesity Dashboard, select **File Services** > **Views** > **All Views**.

1. In the **All Views** page, click **Create > View**.
2. 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) **Protocol**: SMB
   e) Enter the Oracle Server IPs that are allowed to access this view: **10.19.5.0** subnet.
   f) Enable **File DataLock** on and select **Enterprise** DataLock Mode.
   g) Specify the **Default Lock Period**: 1 week.
   h) Enable **Autolock Files**, so files automatically locked after **5 Minutes** after backup on Cohesity.
   i) Click **Create** to create the View.

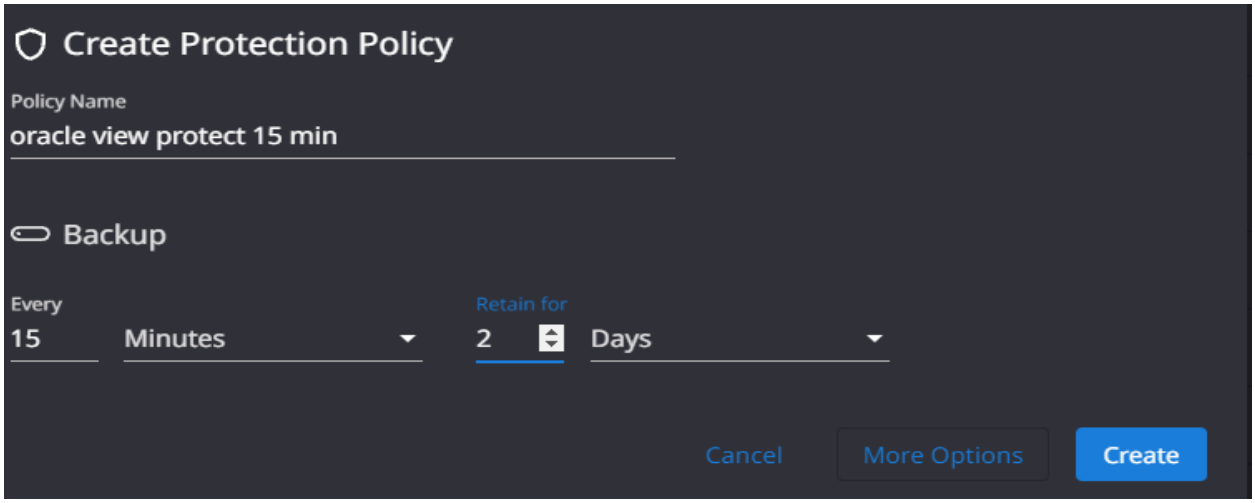


Create View Protection Job

Next, a View Protection Job must be set up on Cohesity.

1. Follow the steps as outlined in the topic: Add or Edit a Remote Adapter Protection Group.

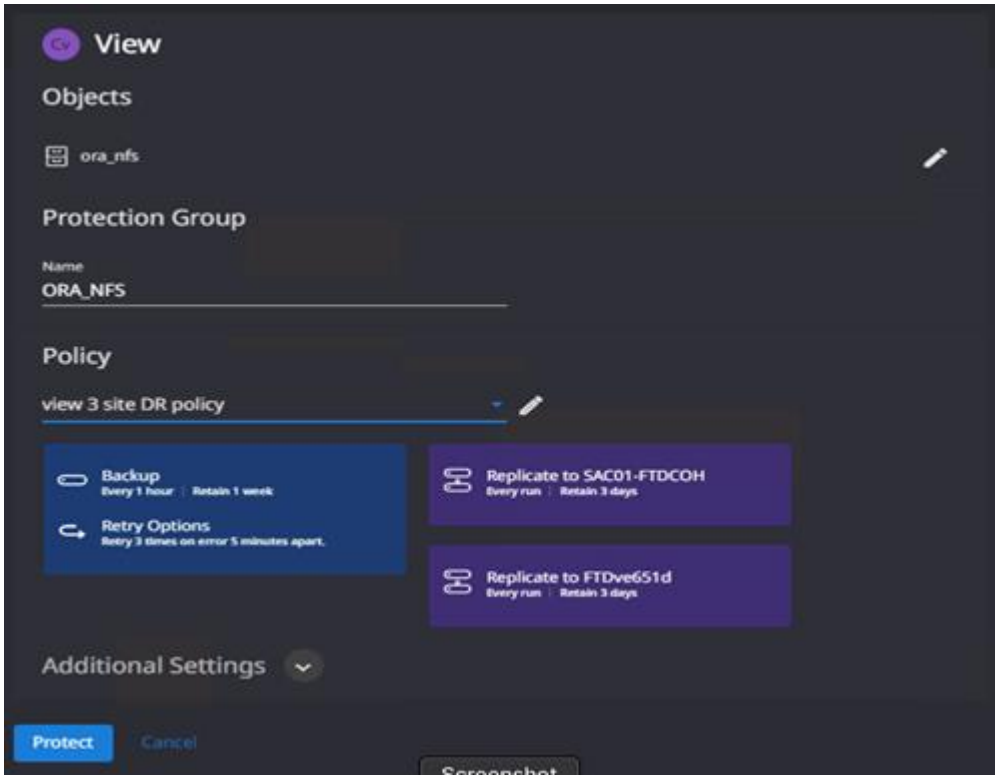
> **NOTE:** This needs to be set up by a Cohesity Administrator or an Oracle DBA user who has a Cohesity login account and is assigned to a “self service” role. The policy shown in this example takes a snapshot of the view every 15 minutes and backups are retained for 2 days.



> **NOTE**: This Protection Policy will be used to take a snapshot of the NFS View where the Oracle SBT library data is written daily and stored using Oracle backup retention and is different from the Oracle backup retention. Set retention for this Protection job to minimum days.

2. In the Cohesity Dashboard, select **Data Protection** > **Protection > Protect**.
3. Select **Cohesity View** and choose the appropriate View Name to be protected.
4. Click **Save**.

5. Provide a name for the **Protection Group** and select the **Policy** for backup frequency, set the start time and click **Protect.**



Manually Mount the NFS view on Unix/Linux Hosts

1. Depending on the number of RMAN channels you are planning on using for the database backup, on the host create the same number of directories (i.e., for 4 RMAN Channel backups create /mnt/coh1, /mnt/coh2, /mnt/coh3, /mnt/coh4 ….).
2. Mount the Cohesity View using different Cohesity VIP Addresses for each of the directories above using the NFS parameters as described below:

```
sudo mount -o
intr,hard,rsize=1048576,wsize=1048576,proto=tcp,vers=3,nolock <<VIP
Address of Cohesity>>:/ora_nfs /mnt/coh1
```

```
<<VIP Address of Cohesity>>:/ora_nfs /mnt/coh1
```

NOTE: To make the mount persistent add it to `fstab`, `vfstab`, `/etc/filesystem` file as per OS instructions.

Linux:

```
intr,hard,rsize=1048576,wsize=1048576,proto=tcp,vers=3,nolock
```

AIX:

```
rw,bg,hard,intr,rsize=524288,wsize=524288,vers=3,proto=tcp,sec=sys,llock,
noac
```

Solaris:

```
vers=3,proto=tcp,rsize=1048576,wsize=1048576,timeo=600,hard,intr,noac
```

3. To automate the NFS View mount/unmount on Unix/Linux Hosts via Script, download the script from the location for each OS below and make the script executable by providing the following parameters.

> **NOTE**: This requires a user role with mount and unmount privileges.

- -y: Cohesity Cluster DNS name
- -v: Cohesity View that is configured to be the target for Oracle backup
- -p: Mount prefix (for example: if the mount is /coh/ora1, the prefix is /coh/ora)
- -n: number of mounts
- -m: yes means mount Cohesity view, no means umount Cohesity view

Linux:

```
curl -O https://raw.githubusercontent.com/diana-hui-yang/rman-
cohesity/master/nfs/nfs-coh-mount-umount/linux/nfs-coh-mount-umount.bash
```

AIX:

```
curl -O https://raw.githubusercontent.com/diana-hui-yang/rman-
cohesity/master/nfs/nfs-coh-mount-umount/aix-nfs-coh-mount-umount.bash
```

Cohesity Oracle NFS backup

Option 1: **This script assumes that the NFS mounts already exist on the Oracle server**.

Download the backup script for the OS onto the Oracle server. You will find the Linux, AIX, and Solaris scripts at the following URL:

https://github.com/diana-hui-yang/rman-cohesity/tree/master/nfs/backup-ora-coh-nfs

Option 2 : **Scripts listed below will perform the NFS mount and unmount during backups**.

NOTE: Make sure the user running this has permission to mount and unmount commands.

Download the backup script for the OS onto the Oracle server. You will find the Linux scripts at the following URL:

https://github.com/diana-hui-yang/rman-cohesity/tree/master/nfs/backup-ora-coh-nfs-mount

Setup the Backup Scripts

The backup script (**backup-ora-coh-nfs.bash** or **backup-ora-coh-nfs-mount.bash**) syntax will display on the screen when it is run without any input. Customers who prefer to use their own backup wrapper script can execute the above bash script with "`-w yes`" option. This will generate RMAN command syntax and print it on the console.

Customers can 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-nfs/backup-example** folder on the github site.

Customers can choose to run the above-mentioned scripts for RMAN backups using either of the following methods:

1. cron/Ctrl+M
2. Using Cohesity Remote Adapter.

Using Unix cron Scheduler or a Third-party Scheduler

1. On the Oracle server, create an Oracle Backup Wrapper script with all the required parameters. The parameter examples are listed here at the following URL:

   https://github.com/diana-hui-yang/rman-cohesity/tree/master/nfs/backup-ora-coh-nfs/backup-example

Using Cohesity Remote Adapter

For step-by-step instructions on how to configure and schedule a Cohesity Remote Adapter Job for Oracle backup, see Protecting Oracle Databases using Remote Adapter.

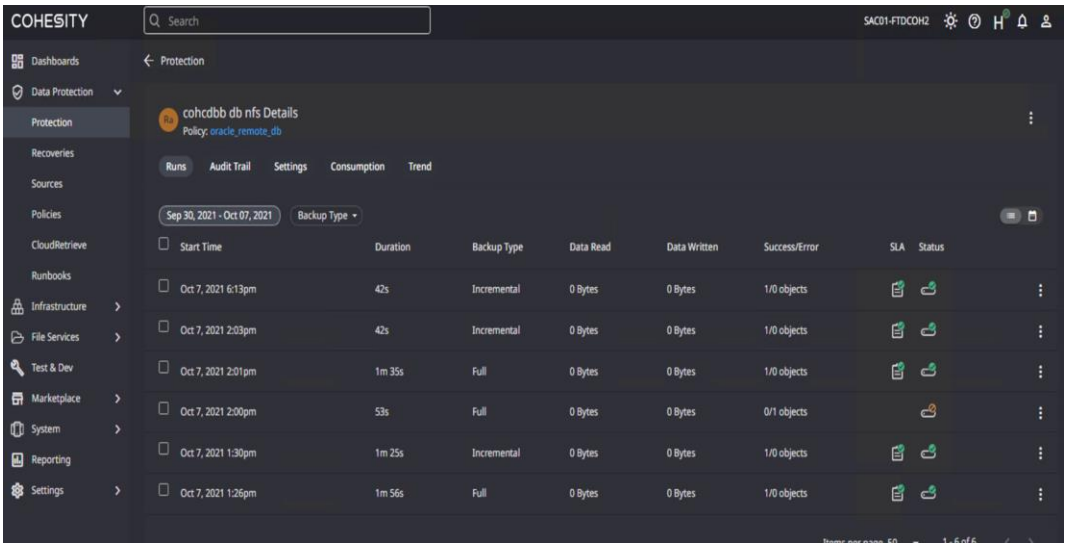
This can be setup by a Cohesity administrator or an Oracle DBA user who has a Cohesity login account and is assigned to the "Self Service Data Protection" role.

When using Cohesity Remote Adapter, two (2) Remote Adapter Jobs are created for one (1) Oracle Database.

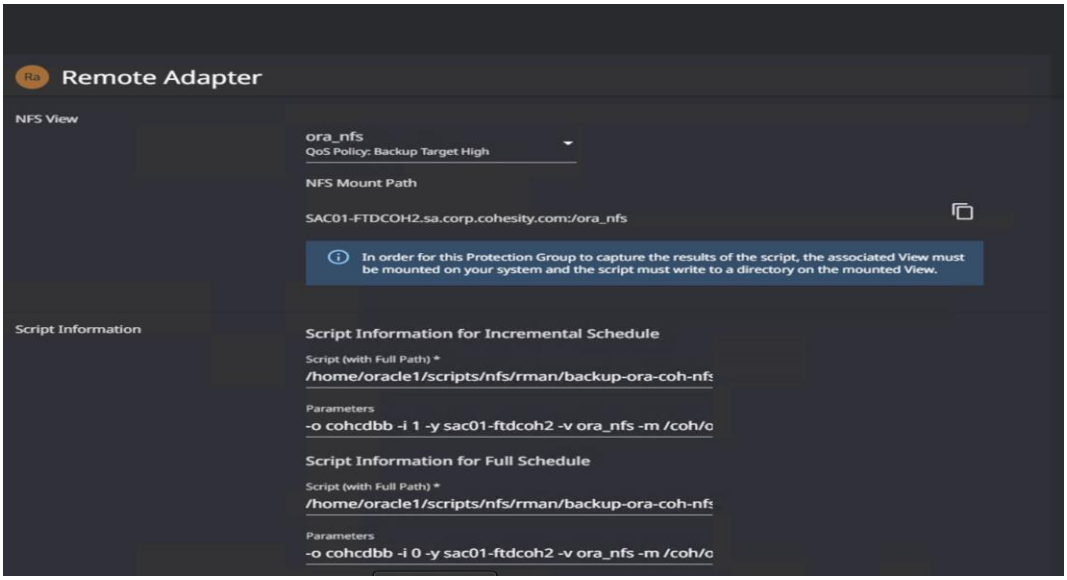
1. The first Remote Adapter Job is for the Oracle Database backup.
2. The second Remote Adapter Job is for the Oracle Archive Log backup.

NOTE: Both backup jobs can run in parallel.

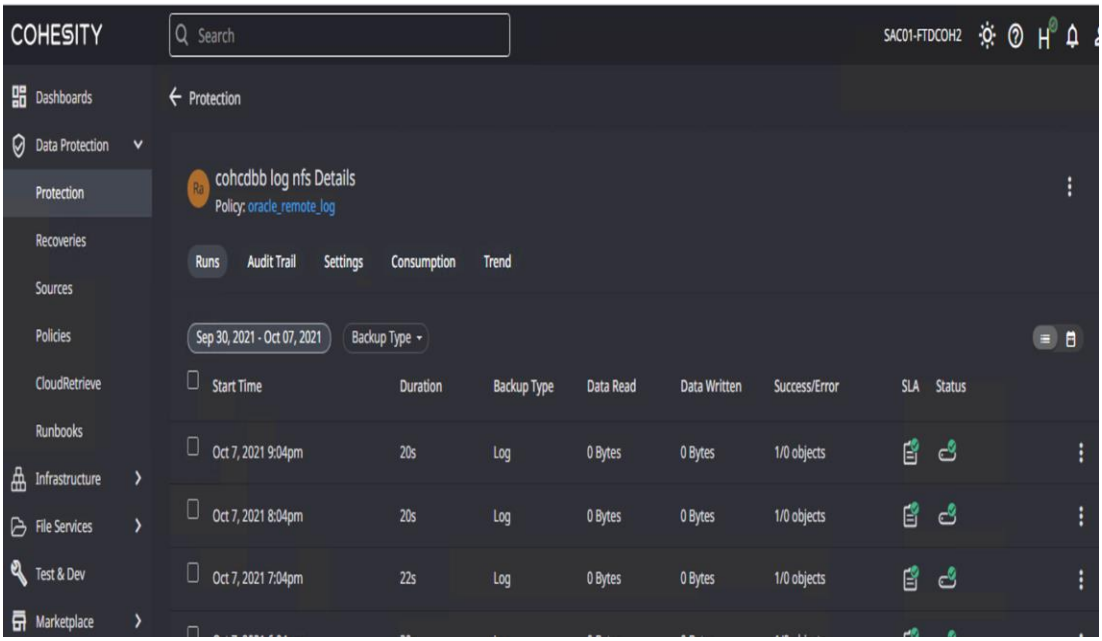
Example 1: Shows database details of both the Full and Incremental database backups.



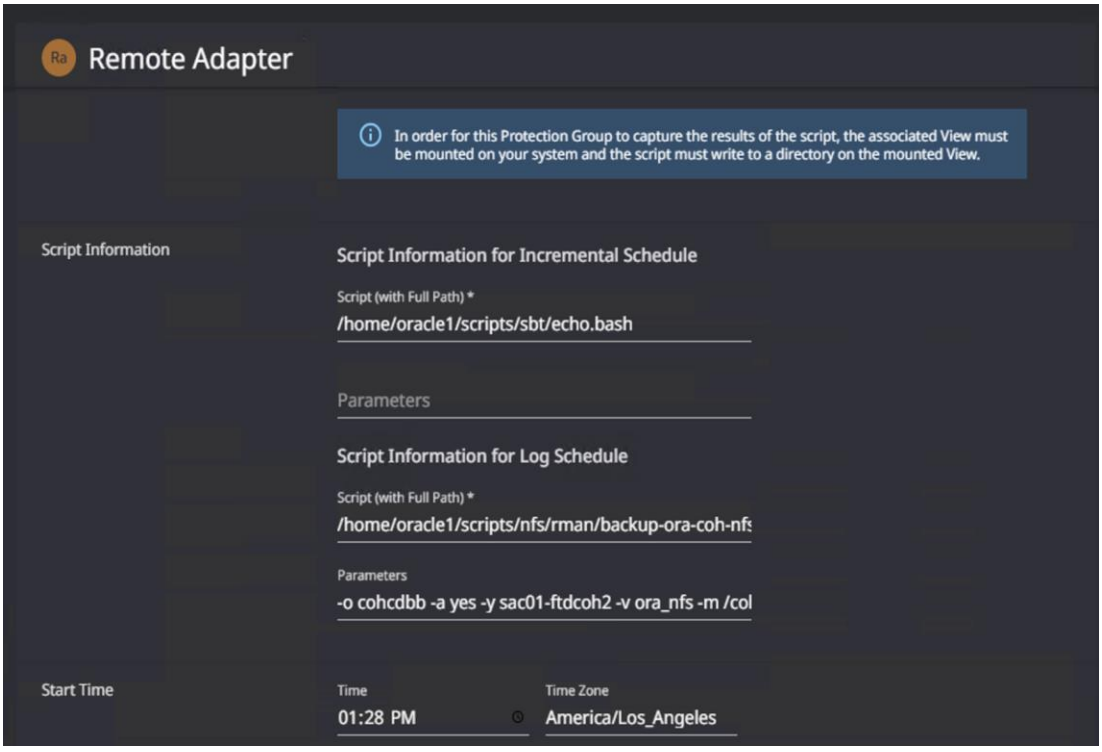
Example 2: Shows details of database **cohcdbb** full and incremental backup job editing.



Example 3: shows database **cohcdbb** Archive Log Backup status.



Example 4: Shows database **cohcdbb** Archive Log Backup editing.



For options to allow both jobs to run in parallel, contact Cohesity Technical Support.

See Cohesity's Oracle RMAN Best Practice Guide for more details and code examples.

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 RMAN NFS Target for Oracle data protection.

Performance Considerations

Correct Cohesity Sizing

Sizing the Cohesity solution should consider both storage requirements and performance requirements. Periodically (For example, weekly), a full backup is required for NFS Target 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 NFS 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.
- **Review RMAN backup logs**. These can provide a lot of information about what went wrong. The RMAN logs are in `<script directory>/log/<server name>` directory when using our 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.

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

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.
```

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';"
```

Duplicate set example if source database is NOT built using OMF

```
# Setting
set newname for database to "'/oradata/w2cdbbr1/%b';"
SET NEWNAME FOR DATAFILE 1 TO "'/oradata/w2cdbbr1/system01.dbf'";
SET NEWNAME FOR DATAFILE 3 TO "'/oradata/w2cdbbr1/sysaux01.dbf'";
SET NEWNAME FOR DATAFILE 4 TO "'/oradata/w2cdbbr1/undotbs01.dbf'";
SET NEWNAME FOR DATAFILE 5 TO "'/oradata/w2cdbbr1/pdbseed/system01.dbf'";
SET NEWNAME FOR DATAFILE 6 TO "'/oradata/w2cdbbr1/pdbseed/sysaux01.dbf'";
SET NEWNAME FOR DATAFILE 7 TO "'/oradata/w2cdbbr1/users01.dbf'";
SET NEWNAME FOR DATAFILE 8 TO "'/oradata/w2cdbbr1/pdbseed/undotbs01.dbf'";
SET NEWNAME FOR DATAFILE 9 TO "'/oradata/w2cdbbr1/w2pdbb/system01.dbf'";
SET NEWNAME FOR DATAFILE 10 TO "'/oradata/w2cdbbr1/w2pdbb/sysaux01.dbf'";
SET NEWNAME FOR DATAFILE 11 TO "'/oradata/w2cdbbr1/w2pdbb/undotbs01.dbf'";
SET NEWNAME FOR DATAFILE 12 TO "'/oradata/w2cdbbr1/w2pdbb/users01.dbf'";
SET NEWNAME FOR TEMPFILE 1 TO "'/oradata/w2cdbbr1/temp01.dbf'";
SET NEWNAME FOR TEMPFILE 2 TO "'/oradata/w2cdbbr1/pdbseed/temp01.dbf'";
SET NEWNAME FOR TEMPFILE 3 TO "'/oradata/w2cdbbr1/w2pdbb/temp01.dbf'";
#set until time \"to_date("'2020-08-23 11:30:00','YYYY/MM/DD
HH24:MI:SS'")\";
#set until sequence 15 thread 1;
```

Appendix: Product Documentation

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

- Oracle Adapter Requirements
- Cohesity Oracle best Practices
- Oracle Adapter Troubleshooting
- Cohesity's Oracle RMAN Best Practice Guide

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, Sr. Manager, Technical Marketing
- Subash Babu, Staff Technology Editor

Document Version History

VERSION	DATE	DOCUMENT HISTORY
1.1	July 2024	Republishing
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.