

TECHNICAL WHITE PAPER

Configuring Commvault with FlashBlade//S: Best Practices

Enhance Commvault and FlashBlade//S performance with the simplest setup.



Contents

How to Use This Guide3

General Best Practices3

Details4

Media Agents9

 Detailed Best Practices 10

Object Storage 12

 Planning for Object SafeMode 13

 Detailed Best Practices15

NFS 38

 Detailed Best Practices39

CommServe DR Backup..... 49

 Detailed Best Practices 50

Common Failure Scenarios 53

Terms and Concepts 56

About the Author 58



How to Use This Guide

This best practices guide is intended for use by Pure Storage® systems engineers, solution architects, backup administrators, and others to assist with the design and implementation of Pure Storage FlashBlade//S® into Commvault environments. The guide is arranged in sections, each focused on specific elements. Each section contains a summary, which covers the design and tuning concepts at a high level, and a details section, which explains how to implement them.

This guide is also applicable for first-generation FlashBlade® systems. Any configuration differences between models will be called out in that section.

General Best Practices

The best write performance on FlashBlade//S storage is achieved by spreading the load across as many blades as possible. Configuring Commvault for maximum writers and distributed streams will give the best outcome with the simplest setup. Commvault supports FlashBlade//S systems as a back-end storage target using both object (Amazon S3) and file (NFS) protocols. S3 is simpler to configure and scales more easily, while NFS is more familiar to many admins. This guide contains best practices for both approaches. Specific best practices for all FlashBlade//S configurations are to:

- Choose the appropriate protocol for your environment
- Enable SafeMode™ ransomware mitigation on FlashBlade//S
- Use Commvault client-side compression by default
- Use Commvault client-side deduplication
- Ensure deduplication databases (DDBs) perform well, ideally using partitioned DDB
- Upgrade DDBs and enable garbage collection
- Run multiple client readers
- Set maximum writers on the library and MediaAgents
- Share mount paths between MediaAgents
- Configure multiple data paths and round-robin in storage policies
- Match VMware disk format and transport mode
- Store CommServe DR backups on a FlashBlade//S file system using SMB

Some best practices include using [Commvault additional settings](#).



Details

Choose the Appropriate Protocol for Your Environment

In most cases, the NFS and object storage models are interchangeable in terms of functionality and raw performance. Each has pros and cons, shown in Table 1.

Storage Model	Pros	Cons
Object (Amazon S3 Protocol)	<ul style="list-style-type: none"> Simple configuration Excellent load distribution across blades Object SafeMode™ support Commvault Storage Accelerator support 	<ul style="list-style-type: none"> Less familiar to administrators
File (NFS)	<ul style="list-style-type: none"> More familiar protocol to admins FlashBlade SafeMode snapshot support 	<ul style="list-style-type: none"> More complex to configure Less efficient load distribution Not supported with Windows MediaAgents

Table 1. Storage model pros and cons

Object storage is generally the best practice approach due to its simplicity. There are use cases where [Storage Accelerator](#) is preferred. Table 2 shows the recommended protocols for each use case.

Use Case	Protocol
VMware Backup, Hotadd Transport Mode with Separate MediaAgents	Amazon S3 with Storage Accelerator
VMware Backup, SAN Transport Mode with VSA on MA	Amazon S3
Centralized Remote Site Backup	Amazon S3 with Storage Accelerator
Live VM Recovery	Amazon S3
VMware Live Mount	Amazon S3
Windows MediaAgents	Amazon S3
Database Instant Recovery	Amazon S3
Ransomware Mitigation with Immediate Immutability	Amazon S3 with Object SafeMode
Ransomware Mitigation with Smallest Storage Footprint¹	NFS with SafeMode snapshots

Table 2. Protocols for use cases

¹ Storage consumption varies based on data change and growth rates. Object SafeMode may have a smaller footprint in some scenarios.





Enable SafeMode Ransomware Mitigation

FlashBlade//S includes ransomware mitigation solutions for both object and file storage that can be enabled at no extra charge. Both are array-wide solutions that provide an immutability window for backup data to protect against malicious or accidental destruction or encryption. Both are enabled and modified through Pure Support, and only with an authorized company designee. However, they work very differently, and as a result, they provide different levels of protection and different recovery experiences. When you've decided on your approach and are ready to proceed with SafeMode protection, contact Pure Support to designate an authorized contact and configure the feature.

Object SafeMode works with Amazon S3 protocol. When it is enabled, each object written to FlashBlade//S is locked until it reaches a retention age you define with Pure Support. This provides immediate protection for all backups as soon as they are written. It also ensures that recovery is not disrupted in the event of a ransomware attack or other production outage. It maintains the simplicity of the object model. Because individual objects are locked until different times, Commvault must maintain data completeness; this is accomplished using data vaults, which are self-contained copies of deduplicated data stores that automatically seal periodically. Object SafeMode requires more storage than file-based SafeMode snapshots in most, but not all, environments. This guide contains a subset of the details and best practices from the paper [“Ransomware Mitigation with Pure Storage and Commvault”](#).

SafeMode snapshots create immutable, point-in-time views of file systems on the FlashBlade//S. In addition to protecting any snapshot you create, a schedule you define with Pure Support also creates separate, periodic snapshots of all file systems. Enabling SafeMode snapshots also disables manual eradication of all snapshots and file systems, preventing malicious or accidental destruction of data as it existed at that point in time. With the snapshot rollback capability on FlashBlade//S, it's easy to return the backup storage to the time of the snapshot, and from there it's straightforward to start recovering data. Rolling back to a snapshot inherently involves a level of data loss, and after an attack you may have to recover your backup storage before you can restore your production systems. You may also have to recover the Commvault CommCell from a DR backup if you roll back more than a few days. SafeMode snapshots require additional FlashBlade//S storage, but less than Object SafeMode in most cases. We recommend enabling SafeMode snapshots to protect CommServe DR backups, even if you implement object storage as your primary backup target. Table 3 compares the pros and cons of each SafeMode option. The Object and NFS sections contain more detail and best practices for their respective SafeMode solutions.

Solution	Pros	Cons
Object SafeMode	<ul style="list-style-type: none"> • No loss or outage for backup data • Immediate immutability • Simple to implement • Operationally transparent • Immediate availability for recovery 	<ul style="list-style-type: none"> • Additional configuration in Commvault • Higher storage than SafeMode snapshots for most environments
SafeMode Snapshots	<ul style="list-style-type: none"> • Invisible to Commvault • Protects all snapshots and file systems, not just scheduled ones • Complete point-in-time view of data 	<ul style="list-style-type: none"> • Potential for data loss between snapshots • Additional recovery required after the event

Table 3. Pros and cons of SafeMode solutions

Use Commvault Client-Side Compression by Default

While FlashBlade//S has effective hardware compression, using it requires sending uncompressed data over the network. In most environments, data reduction from client-side compression can yield effective network throughput higher than the actual





available bandwidth. Commvault's deduplication algorithm also reduces the effectiveness of FlashBlade//S compression. Ignoring deduplication, when Commvault compression is enabled, backups are usually faster overall and consume around the same amount of physical storage. If backups are underperforming or consuming too much CPU time on clients, you can disable compression in Commvault; generally, however, it should be left in the default enabled state.

Use Commvault Client-Side Deduplication

Commvault deduplication provides global data reduction across large data sets, for improved storage efficiency. Deduplication at the client-side will reduce the amount of data sent over the network to MediaAgents. Note that because most data is removed at the client, only initial full backups will send large amounts of data to FlashBlade//S.

Ensure Performant Deduplication Databases

Deduplication database (DDB) performance is critical to fast backups. Deduplication processing, including DDB ingestion, will in many cases be slower than the throughput FlashBlade//S is capable of. Configuring multiple DDB partitions and segregating different data types between storage policies will give you better bandwidth utilization with FlashBlade//S. A high-throughput, very low latency SSD or NVMe device optimized for random access will maximize performance.

Pure Storage FlashArray//X™ may be suitable for DDB storage depending on your environment. However, placing DDBs on shared storage can affect performance if the other workloads change significantly. You should discuss the configuration with your Commvault account team and test it thoroughly before implementing it for production backups.

Run Multiple Client Data Readers

As parallel streams are important to getting the best throughput, configuring multiple data readers can improve backups. With multiple data readers, Commvault can run parallel processes on the client system to pull data from primary storage. When you use server backup plans in Command Center, Commvault will manage the data readers automatically. While you can override the behavior, this is not recommended since you will miss any future automation improvements Commvault adds.

If you choose not to use plans, data readers are usually configured on subclients. Specific GUI option names and locations vary by agent type, and optimal values will vary by agent type, client hardware, deduplication database performance, and data profile. Tuning will typically be required for the specific data set. Data readers are configured in the CommCell Console interface. The following are recommended as starting points when the source data is on FlashArray™.

- **SQL server:** Four backup streams, configured in subclient properties, using the Number of Data Backup Streams field on the Storage Device tab.
- **Oracle:** Four backup streams; for database backups, configured in subclient properties, using the Number of Data Backup Streams field on the Storage Device tab. For archive log backups, configured in the instance properties, use the Number of Archive Log Backup Streams field on the Log Backup tab under the Storage Device tab.
- **VMware:** Three data readers x number of Virtual Server Agent proxies, which are configured in the subclient properties, using the Number of Data Readers field on the Advanced tab.
- **File system:** Four data readers, enable Allow multiple readers within a drive or mount point option, configured in the subclient advanced properties, using the Number of Data Readers field on the Performance tab.





Set Maximum Writers on the Library and MediaAgents

Commvault allows limiting the number of writers for a given library, mount path, or MediaAgent, which limits the number of concurrent streams that object will allow. Unless there is something in the environment that dictates limiting the number of writers, the best practice is to leave the defaults for all of these.

Share Mount Paths Across MediaAgents

You can share mount paths to enable Commvault load distribution between MediaAgents without having to configure multiple storage pools. The process for NFS and Amazon S3 paths is slightly different, so it will be detailed in the later sections.

CAUTION: You should not use the DataServer-IP sharing option for mount paths on FlashBlade//S. This option routes all access to the mount point through a single MediaAgent, using more resources and network bandwidth on that MediaAgent. It also reduces the efficacy of the solution's scaling since the connections to FlashBlade//S will all come from the same MediaAgent and won't distribute across blades as effectively

Configure Multiple Data Paths and Round Robin in Storage Policies

Multiple data paths enable access through the shared mount paths and let clients send data to FlashBlade//S through different MediaAgents. Round robin distributes backup jobs across MediaAgents. When you use server plans, Commvault will add all MediaAgents for a library as data paths. In most cases, sharing a mount path to a new MediaAgent will automatically update existing storage policies that use that library, but new storage policies you create after sharing the mount path may default to only one data path. Data paths are configured within storage policy copies. To configure the storage policy:

1. In the CommCell Console, expand Policies, then Storage Policies, then the target storage policy. As shown in Figure 1, expand the desired policy, then open the Properties dialog for the policy copy configured to write to FlashBlade//S.

NOTE: If the storage policy uses global deduplication, data paths are set from the global deduplication policy instead.

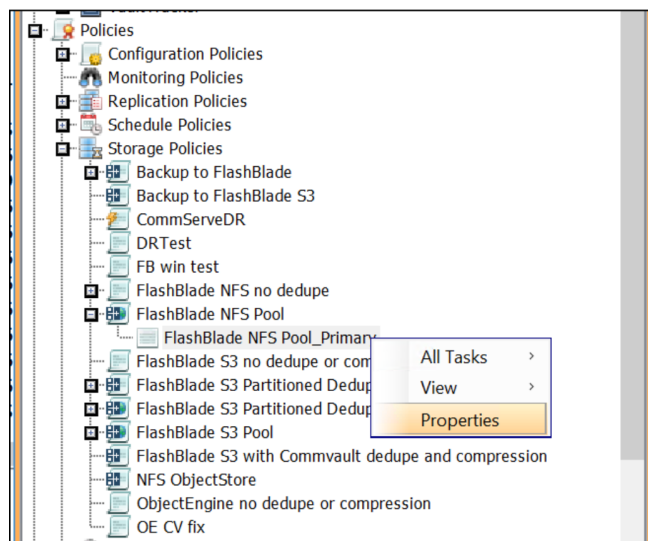


Figure 1. Accessing storage policy copy properties

2. As shown in Figure 2, select the **Data Paths** tab. Click the **Add** button.



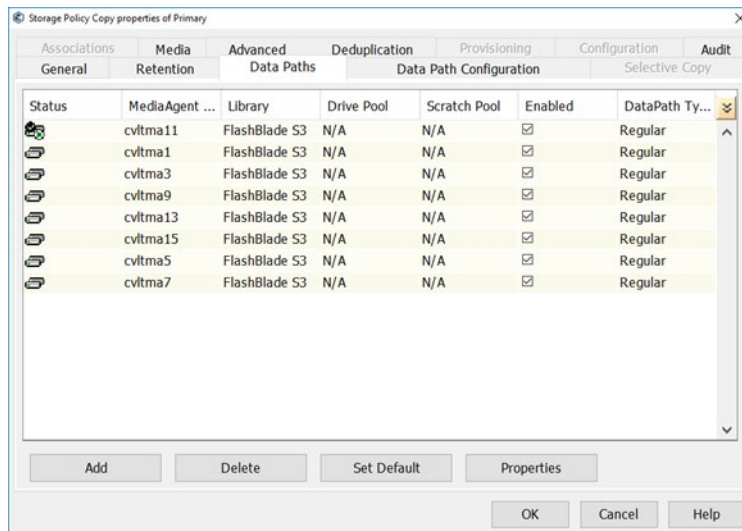


Figure 2. Adding data paths

3. As shown in Figure 3, select the MediaAgents that will use the FlashBlade//S, then click OK. Make sure you select the correct library if multiple FlashBlade//S libraries are available.

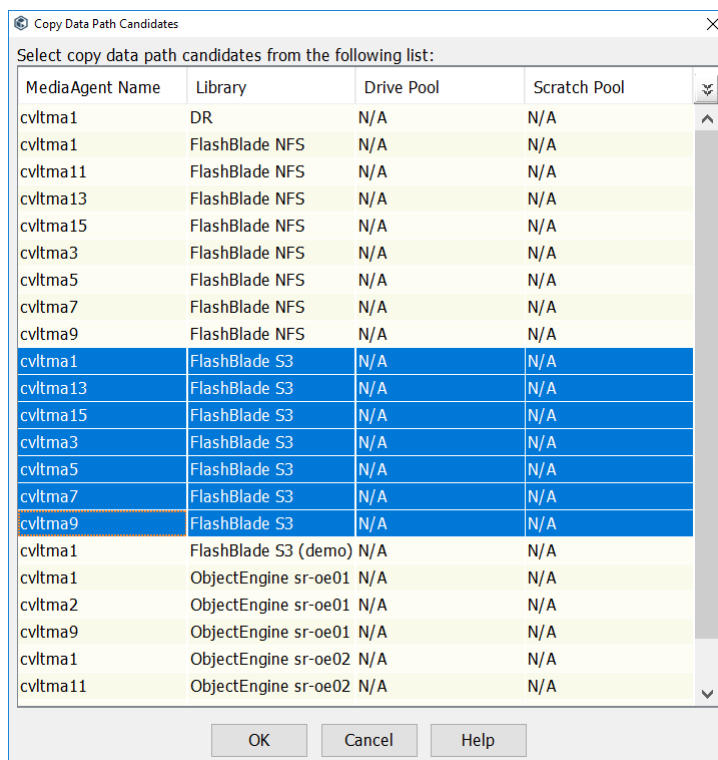


Figure 3. Selecting MediaAgents

4. As shown in Figure 4, select the Data Path Configuration tab. Ensure that the **Automatically add new data path** box is checked and the **Round-Robin between Data Paths** option is selected.



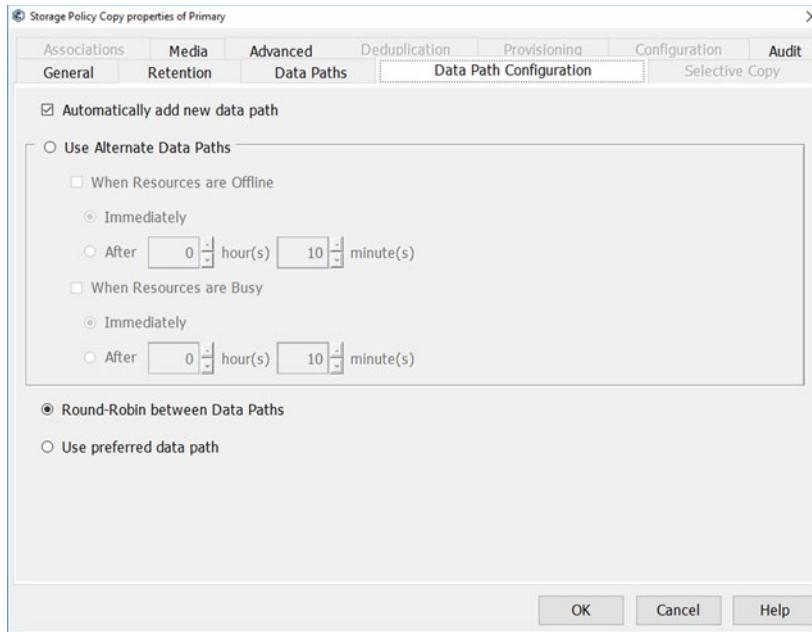


Figure 4. Data path configuration

5. Click OK to apply the changes to the storage policy.

Match VMware Disk Format and Transport Mode

VMware virtual disks allow several formats. When using SAN transport mode, restore performance is best with the thick provisioned eager zero format. Lab testing has shown restore throughput with eager zero formats at more than double that with thick provisioned lazy zero formats. The thin-provisioned format is the slowest with SAN mode, according to both VMware and Commvault. For thin-provisioned or thick-provisioned lazy zero virtual disks, use HotAdd or NBD transport for best throughput. While this best practice is not specific to FlashBlade//S, a mismatch may limit throughput below what you would otherwise see.

Store CommServe DR backups on a FlashBlade//S File System Using SMB

Storing CommServe DR backups on a FlashBlade//S file system lets you centralize protection for CommServe databases and removes dependencies on specific servers. Optional SafeMode snapshots and FlashBlade//S replication add layers of protection against ransomware and accidental deletion. The SMB protocol adds support for NTFS ACLs to limit the attack surface for the database backups. See best practices for [CommServe DR backups](#).

Media Agents

MediaAgents are an important factor in optimal backup and recovery performance, as they are the primary communicators with FlashBlade//S. Important factors to consider are:

- Operating system
- MediaAgent hardware specifications
- MediaAgent count
- Stream count
- Network bandwidth





- Segregate client and storage networks
- Deduplication database storage
- Index cache storage

Detailed Best Practices

Operating system: Commvault supports Windows, Linux, and Unix as MediaAgents in a data mover role. Windows and Linux can also host DDBs. Performance and configuration are similar across operating systems, so you should choose the option that works best for your environment. MediaAgent and client operating systems do not have to match. For example, you can back up a Linux client through a Windows MediaAgent.

MediaAgent hardware specifications: When using Commvault deduplication, hardware sizing for MediaAgents is based on data under management and deduplication topology. Commvault provides a series of [building block specifications](#) for different deduplication configurations.

A typical design scales out building blocks, consolidating backup on a small number of high-specification servers. For example, as of this writing, the extra-large MediaAgent building block with one DDB disk supports roughly 100TiB of mixed front-end data, on 250TiB of FlashBlade//S storage. Generally, for every 100TiB of front-end data, you would add another extra-large MediaAgent.

Partitioned deduplication will provide better performance and resilience by distributing deduplication load across 2- to 4-node DDB grids, with a shared FlashBlade//S back end of up to 250TiB per node. In this model, you scale to more MediaAgents by adding new grids.

MediaAgent count: Calculate MediaAgent requirements primarily on front-end data size, following Commvault's building block guidance. Size FlashBlade//S capacity according to the building blocks.

When planning for immutability, review the sizing guidance for [Object SafeMode](#) or file-based [SafeMode snapshots](#), in this document.

Stream count: Follow Commvault's building block guidance on the number of streams per MediaAgent. By default, MediaAgents are set to allow up to 100 streams. MediaAgents with sufficient resources can have the maximum stream count increased.

To increase the maximum streams:

1. In Command Center, navigate to the MediaAgents list. Navigate to **Manage** and then **Infrastructure**.
2. Click the MediaAgents tile. Click the appropriate MediaAgent name in the MediaAgents list.
3. As shown in Figure 5, on the MediaAgent properties page, locate the Control tile, and click the Edit icon.



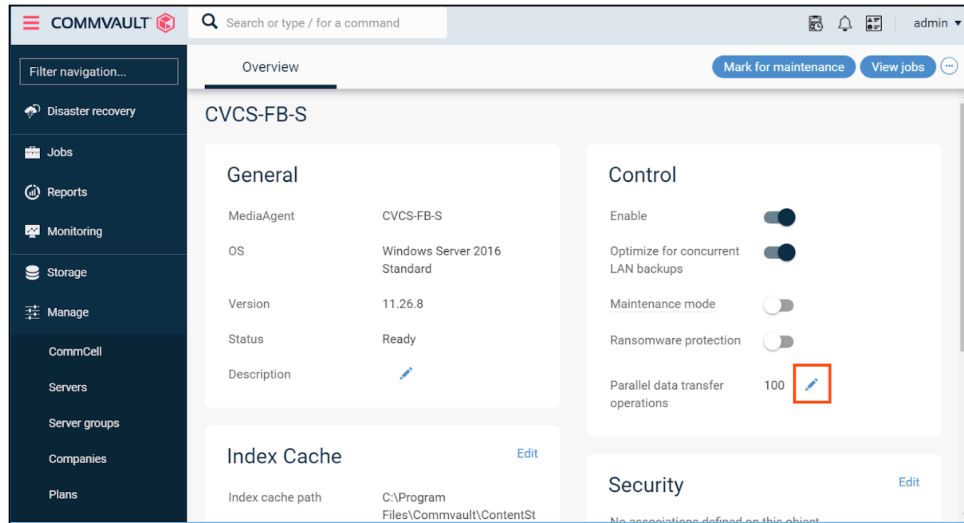


Figure 5. MediaAgent properties

- As shown in Figure 6, in the **Edit Parallel Data Transfer Operations** form, set the **Parallel data transfer operations** field to the appropriate number based on the Commvault building blocks. Click the **Save** button to commit the change.

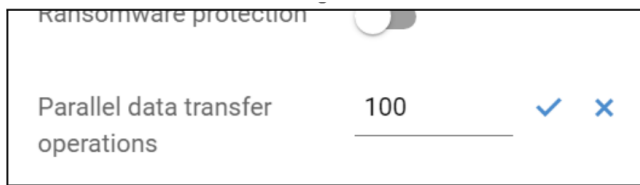


Figure 6. Setting parallel data transfer operations

Network bandwidth: MediaAgents must have sufficient network bandwidth to handle the backup and recovery traffic to meet your SLAs. When assessing available networking, you must consider how much total bandwidth you need to recover your key systems and how much bandwidth you can allocate to a single MediaAgent. You also have to consider how fast you can send data to your primary storage; 100Gbps networking is wasted if your storage can only accept data at a third that rate. In general, MediaAgents need at least 10Gbps links with preferably two or more in a teamed or bonded configuration. Links of 25Gbps or greater are strongly recommended if available.

Segregate client and storage networks: We recommend using separate networks on MediaAgents with separate network adapters for communication to clients and FlashBlade//S. While not necessary, segregation serves two purposes: It allows you to restrict which systems can access the FlashBlade//S VIP, and it ensures storage and client traffic will not collide or affect each other's bandwidth.

Deduplication database storage: As mentioned previously, DDB performance is critical to high-performance backups. DDB storage must meet the specifications in the Commvault building block guide. NVMe is recommended for its performance characteristics.

Index cache storage: The index cache is used for catalog access during recovery, and temporary storage in Live Mount and Live Recovery cases. Using a fast storage device will improve performance and the overall experience. For synthetic full backups, Live Mount, and Live Recovery specifically, a fast index cache is critical to achieving optimal performance. In environments using large or extra-large MediaAgents, you should place the index cache on low-latency, high IOPS internal flash storage such as SSD drives. FlashBlade//S is not a suitable platform for index cache storage. FlashArray may be an





acceptable platform if it is already present; it may not be cost-effective solely for this use case, and most IT organizations will prefer to use internal SSD or NVMe storage.

Object Storage

Commvault can use FlashBlade//S as a cloud storage pool through the Amazon S3 object storage protocol. Object storage has the advantages of simplicity and scale compared with NFS, plus the higher level of protection Object SafeMode offers. With enough available network bandwidth, a single MediaAgent writing to a single object bucket can reach nearly the maximum FlashBlade//S write performance. NFS requires multiple mount paths and a more complex Commvault configuration to achieve a comparable result. For best object storage results:

- Use Commvault 11.26 or later
- Use Purity//FB 3.3.2 or later
- Configure a single FlashBlade//S bucket and mount path
- Disable Transport Layer Security (TLS), if appropriate
- Share the bucket across MediaAgents
- Use Storage Accelerator for additional horizontal scaling
- Use separate credentials for Storage Accelerator
- Set deduplication block size to 128KB
- Run space reclamation regularly
- Optional: Increase cloud thread pool size
- Optional: Increase pipeline buffer count
- Optional: Increase look-ahead size
- Enable Object SafeMode for ransomware mitigation

When to use object storage: Object storage is the preferred model for all Commvault operations. The same FlashBlade//S can be configured as both an object and NFS target in Commvault, if needed, using separate storage pools for each protocol.

How Commvault uses object storage: Commvault uses a very different process to read and write to object storage than it uses with file storage. With file storage, each backup or restore stream is broken into chunks, which are written sequentially into large data files, with a 1:1 relationship between active files and streams.

With object storage, the data files are broken into smaller binary large objects (BLOBs) before they are written. The system creates a thread pool that is shared across all streams. As each thread is activated, it opens a TCP connection to storage, and the threads write BLOBs in a highly parallel manner. Commvault automatically expands the thread pool as needed, up to a tunable maximum, to improve throughput, and it can reuse the same thread for multiple objects. Because the threads each have their own TCP connections, Commvault's model results in excellent load distribution across blades. In most cases, Commvault does not benefit from increasing the maximum thread count, but in certain resource-limited environments, it can increase backup throughput.





For object reads, Commvault uses a separate, smaller thread pool to perform ranged reads, or [byte-range fetches](#), of object data. The size of this pool is not tunable.

Planning for Object SafeMode

Global deduplication saves a lot of storage, but it causes challenges with object-level immutability. Since backups reference existing data (Figure 7), the oldest shared objects must be preserved long enough to ensure the newest backups are recoverable—otherwise, an attacker could destroy last night's backup by deleting last week's data. However, that would mean locking all objects forever, which would undo the efficiency gains of deduplication.

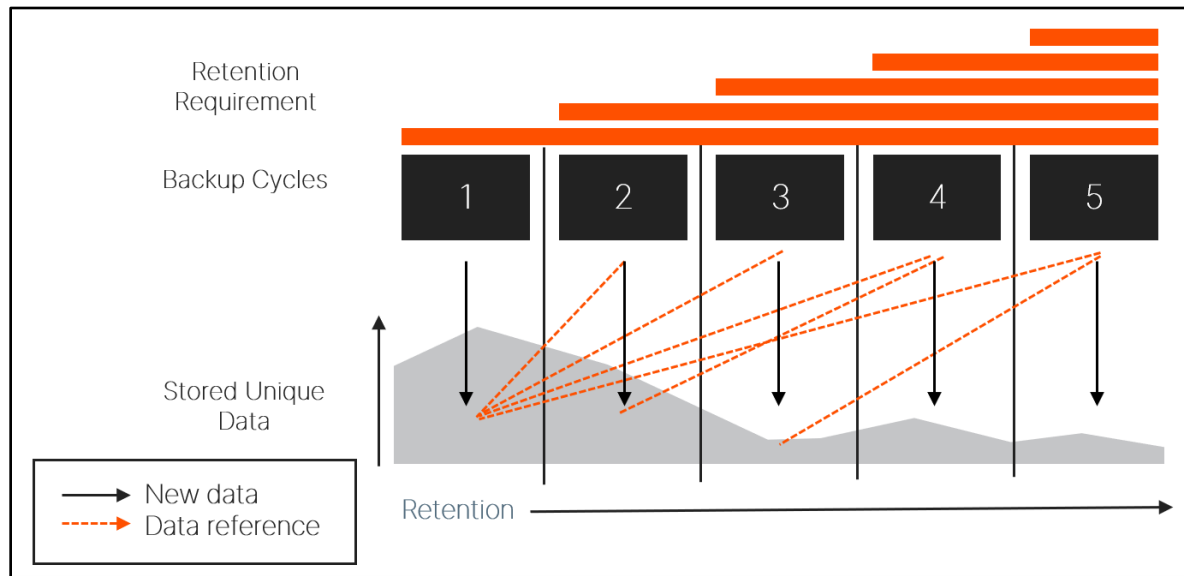


Figure 7. Deduplication dependencies

The solution is to use layered data vaulting where the deduplicated data is periodically frozen and a new baseline created. In practice, Commvault seals its deduplication database on a frequency that matches your required immutability period. Object SafeMode protects the data for double the immutability period to ensure that the last backup in the vault has the necessary protection. This solution maintains a significant level of data reduction but also provides the necessary data availability and operational transparency. Figure 8 illustrates the relationship between the vault layers.

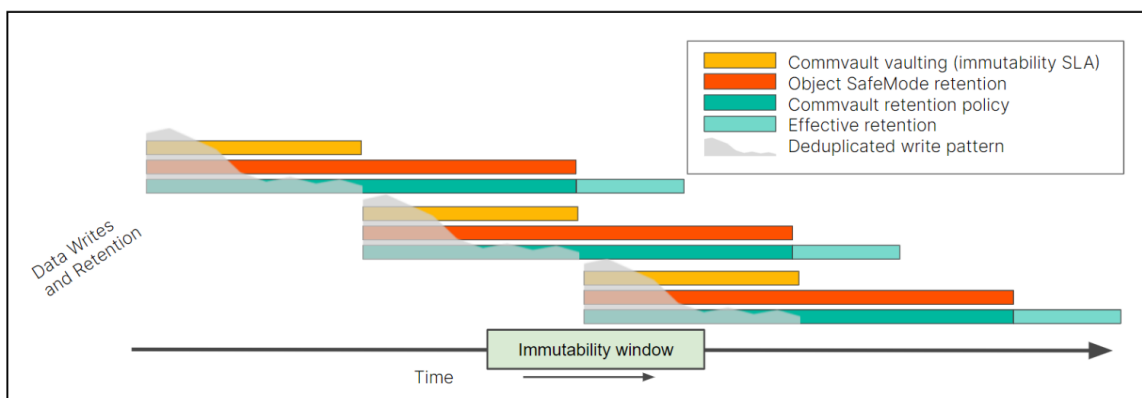


Figure 8. Layered data vaulting





Designing Retention Policies

When you are designing your layered vaulting approach, you should start with the immutability SLA, the length of time you need to guarantee your data is protected. While it is possible to work backward starting with your backup retention policies, it is easier to build layers up from the SLA.

NOTE: As you plan for vaulting, remember that Object SafeMode is a system-level policy and will apply to all object data stored on the FlashBlade//S, not just Commvault data. If you need guidance or discussion on determining the right settings for your environment, please contact your Pure account team.

The Commvault vaulting interval should match your SLA. Aligning to a multiple of seven days will give the easiest calculations for the other layers.

The Object SafeMode retention period should be double the Commvault vaulting interval to ensure each entire vault meets the SLA. For example, for a seven-day vaulting interval, Object SafeMode needs to have a 14-day retention period to protect the data from day seven for seven days. An authorized company contact will work with Pure Storage Support to configure the Object SafeMode retention period.

The Commvault retention policy should be set to match Object SafeMode. For example, if Object SafeMode retention is set for 14 days, Commvault retention should also be set to 14 days. While it can be longer or shorter depending on your specific needs, the storage calculations are simpler when you align to backup cycles. Setting the Commvault retention policy to less than the Object SafeMode retention period will not save any storage since data will still be locked after it ages within Commvault software. A longer retention policy will require more storage.

Figure 9 shows the relationship between the vault layers and the formulas for each layer.

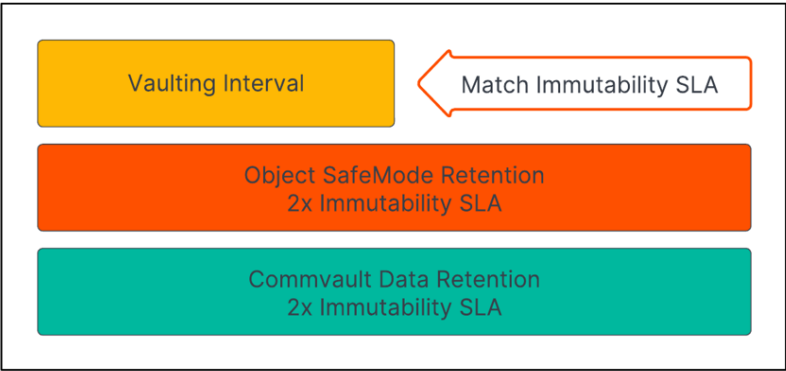


Figure 9. Vault layer relationships and formulas.

Table 4 shows the recommended settings for several example scenarios. You should contact your Commvault or Pure account team if you have questions or want to discuss more complex retention needs.





Immutability SLA	Commvault data vault	Object SafeMode retention	Commvault retention policy
7 days	7 days	14 days	14 days
14 days	14 days	28 days	28 days
21 days	21 days	42 days	42 days

Table 4. Layered data vaulting formulas and examples

Estimating Capacity Requirements

In a Commvault environment, implementing Object SafeMode will require the capacity to store the data vaults until they expire. For most environments, you will need about twice as much space as the data you plan to protect **at the end of the projection period**.

For example, if you have 100TiB of data and grow at 20TiB per year, you will have 160TiB at the end of year three. You would therefore expect to need around 320TiB peak storage on FlashBlade//S.

Be aware that this is a simplistic estimate. There are several factors that affect consumption, such as change rate, data reduction ratios, and full backup scheduling, so the exact amount of storage you will need can vary widely. For the example above, peak three-year consumption could be as low as 180TiB or as high as 470TiB as those factors change. You should consult with your Pure Storage and Commvault sales teams to get an accurate estimate for your environment.

Estimating Deduplication Database Requirements

The vaulting process will retain deduplication databases (DDBs) until all their references are stale. You will need enough storage to accommodate at least three DDBs to ensure you don't run into issues. If you use partitioned DDBs, each partition will need enough storage available.

Detailed Best Practices

Use Commvault 11.26 or later: As of this writing, Commvault 11.24 is the latest long-term support (LTS) release. It includes all performance and management enhancements related to object storage. Release 11.26 includes features to optimize capacity usage with object storage.

Use Purity//FB 4.0 or later: Purity//FB 4.0 is the minimum supported release for FlashBlade//S.

NOTE: On first-generation FlashBlade, 3.3.2 is the recommended minimum release, although previous releases are compatible and supported.

Create restrictive object access policies: Purity//FB allows you to create object access policies, custom collections of S3 access rights that you can use to limit access to your object buckets. You should create a policy that is only applicable to your Commvault object account and grants only the access rights Commvault needs. If you use Storage Accelerator on Commvault release 11.23 or later, you can create a separate access policy and user to further restrict the bucket access rights for Commvault clients.





NOTE: To limit an object access policy to a specific object account, you must first create the object account.

Following Purity//FB documentation, create an object access policy that is associated with the Commvault object account on FlashBlade//S. Grant the access rights shown in Table 5. When you later create the object user account, select this policy to restrict its access.

s3:DeleteObject
s3:DeleteObjectVersion
s3:GetBucketVersioning
s3:GetObject
s3:GetObjectAcl
s3:GetObjectTagging
s3:GetObjectVersion
s3:GetObjectVersionTagging
s3:ListBucket
s3:PutObject

Table 5. Required object access rights for Commvault

Your policy rule should also restrict the source IP addresses so that only the MediaAgents have access to the object data. You can enter specific addresses or subnets. Using the specific MediaAgent addresses will force an attacker to gain access to the MediaAgents to reach the object buckets, but you will have to update the policy when you add or remove any MediaAgent from the environment. Using subnets is simpler to manage, as it does not require updates when you change MediaAgents—provided new MediaAgents are on a defined subnet—but it means an attacker could potentially use any system on that subnet to access the object data. Figure 10 shows a fully configured rule that uses the MediaAgent IP addresses.





Edit Rule

Name cvaccess
Rule name consisting of letters and numbers, with no hyphens or underscores.

Effect Allow
Allow S3 requests that match all of the criteria below. 'Allow' rules are additive.

Actions
 s3:DeleteObject X s3:DeleteObjectVersion X
 s3:GetBucketVersioning X s3:GetObject X
 s3:GetObjectAcl X s3:GetObjectTagging X
 s3:GetObjectVersion X s3:GetObjectVersionTagging X
 List of permissions to grant, in addition to any from other rules and policies.

☐ Ignore Action Restriction Enforcement

Resources * X
List of bucket names and object paths, with a wildcard (*) to specify objects in a bucket; e.g., bucket1, bucket1/*, bucket2, bucket2/*.

Source IPs 10.21.242.6 X 10.21.242.7 X
List of IPs and subnets from which this rule should allow requests; e.g., 10.20.30.40, 10.20.30.0/24, 2001:DB8:1234:5678::/64.

S3 Prefixes
List of 'folders' (object key prefixes) for which object listings may be requested.

S3 Delimiters
List of delimiter characters allowed in object list requests. Grants permissions to list 'folder names' (prefixes ending in a delimiter) instead of object keys.

Cancel Save

Figure 10. Object access rule with IP address restriction

Disable TLS, if appropriate: Disabling TLS can improve maximum FlashBlade//S throughput for a MediaAgent by up to 30%. However, using TLS protects access keys from being intercepted by a network sniffer. We recommend disabling TLS only when the MediaAgent and FlashBlade//S communicate on a restricted network. The bucket configuration process below explains how to disable TLS.

Configure a single object bucket: Commvault requires only a single bucket. Any cloud storage pool can create access paths under the same bucket, simplifying the configuration.

Unless required for multitenant environments or other business reasons, Commvault should be configured to use a single bucket in a single cloud storage pool.

To create a cloud storage pool using FlashBlade in release 11.26 and later:

1. Follow FlashBlade//S documentation under the create a user account and bucket on the FlashBlade//S for Commvault data.
2. As shown in Figure 11, in Commvault Command Center navigate to Storage, then Cloud. Click the **Add** link.



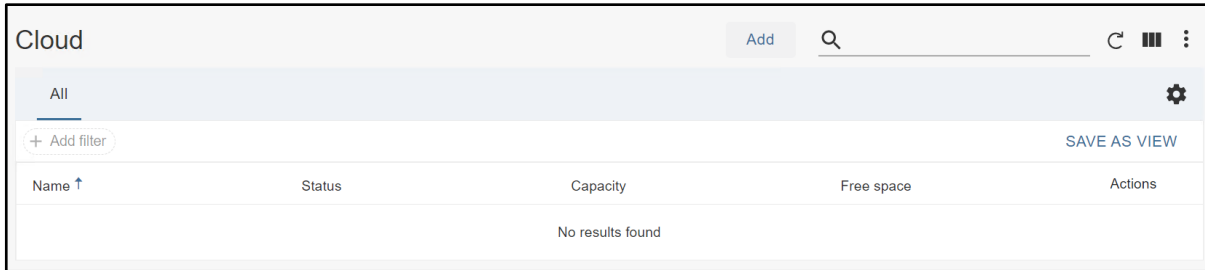


Figure 11. Cloud storage management

3. Click the **Cloud storage** tile to select the storage type (Figure 12).

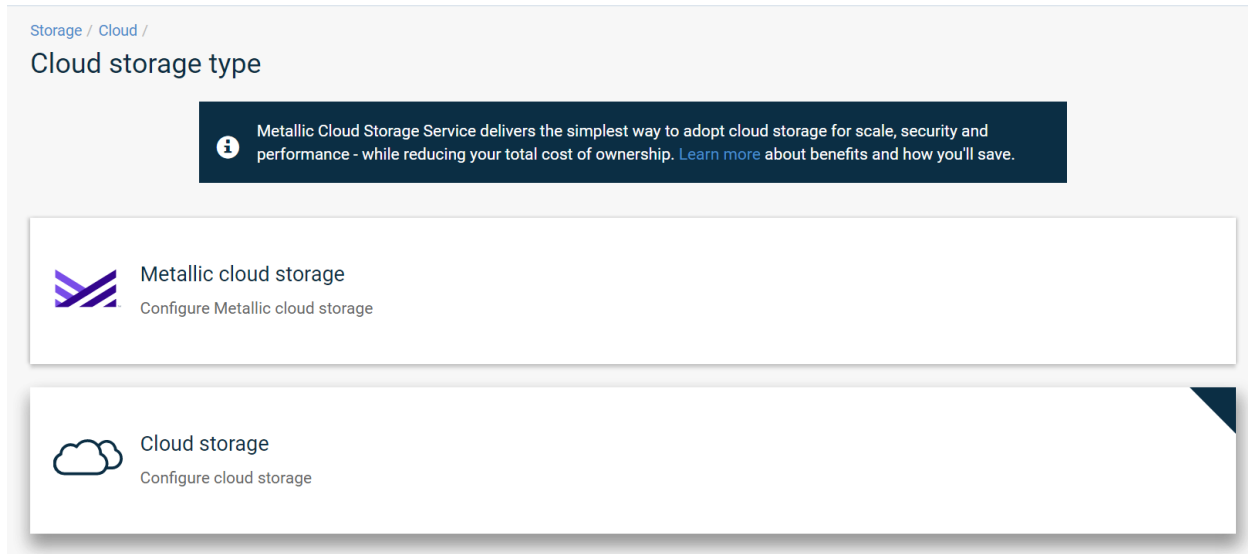


Figure 12. Selecting the cloud storage type

4. Complete the Add cloud storage form (Figure 13) as follows:
 - a. In the **Name** field, enter a display name to help identify the FlashBlade.
 - b. From the **Type** dropdown, select "S3 Compatible Storage."
 - c. From the **MediaAgent** dropdown, select the MediaAgent that will act as primary controller for the bucket.
 - d. In the **Service host** field, enter the DNS name or IP address for the FlashBlade data VIP. If you wish to disable TLS, include "http://".





Cloud / Cloud storage type

Add cloud storage

Configure cloud

Name *

FlashBlade//S

Storage

Type

S3 Compatible Storage

MediaAgent *

sn1-r720-g08-07

Service host *

http://10.21.237.25

Credentials *

FlashBlade//S S3

Bucket *

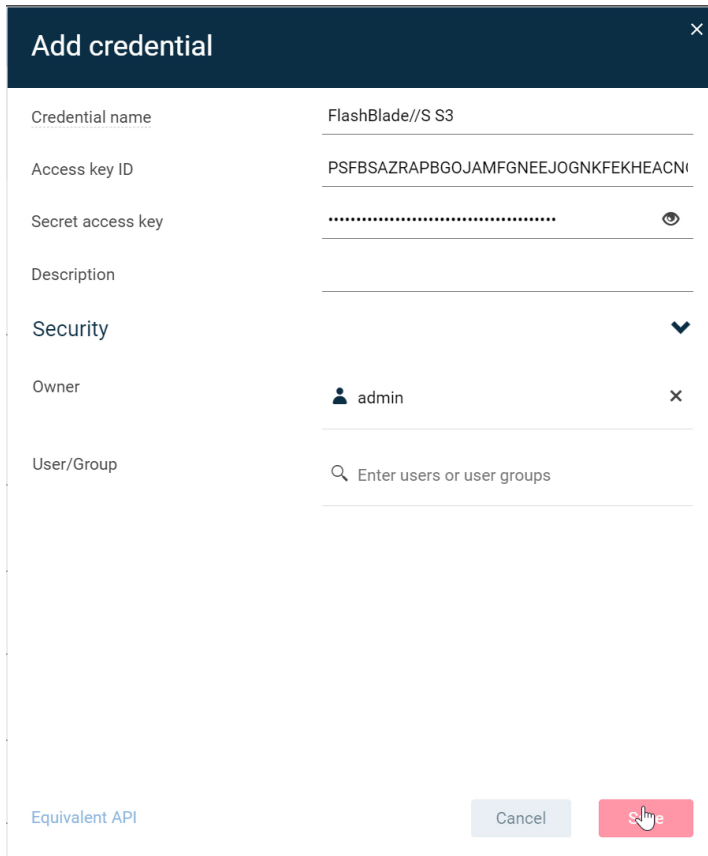
cvbucket

Figure 13. Add cloud storage form

- e. Next to the **Credentials** dropdown, click the **Create new** (+) button to add the FlashBlade keys into a stored credential. If you already have a stored credential you wish to use, select it instead and skip to step 5.
5. In the **Add credential** form (Figure 14), enter a name for the credential. Switching between the FlashBlade and Commvault interfaces, use the **Copy** button in the FlashBlade interface (Figure 15) to copy the access key ID and secret key and paste them into the appropriate Command Center fields. You may enter descriptive text in the **Description** field. Click the **Save** button to create the stored credential and return to the **Add cloud storage** form.

NOTE: You should always clear the clipboard after copying sensitive data such as object storage access keys.



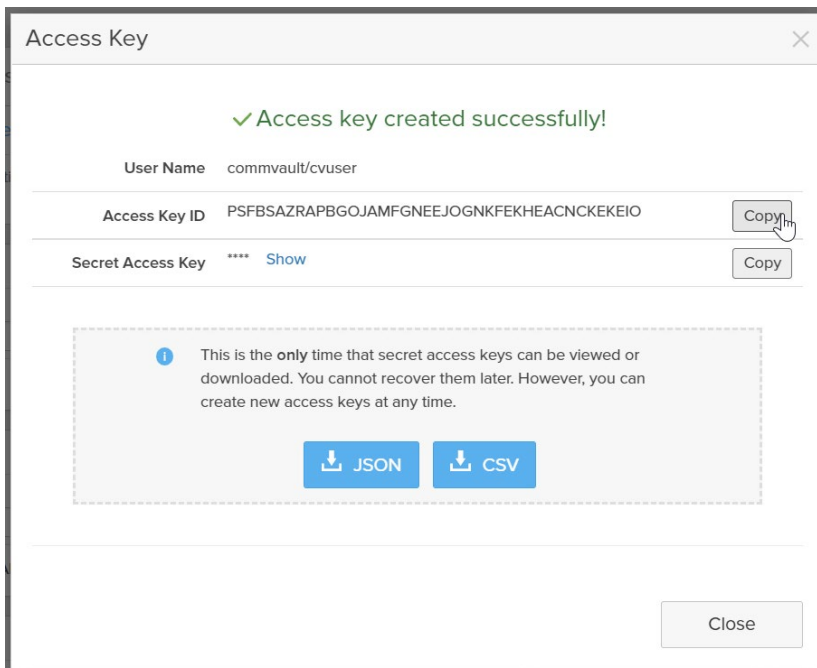


Add credential [X]

Credential name	FlashBlade//S S3
Access key ID	PSFBSAZRAPBGOJAMFGNEEJOGNKFEKHEACN
Secret access key [Eye icon]
Description	
Security	▼
Owner	[User icon] admin [X]
User/Group	🔍 Enter users or user groups

[Equivalent API](#) Cancel Save

Figure 14. Add credential form



Access Key [X]

✓ Access key created successfully!

User Name	commvault/cvuser
Access Key ID	PSFBSAZRAPBGOJAMFGNEEJOGNKFEKHEACNCKEKEIO [Copy]
Secret Access Key	**** [Show] [Copy]

i This is the **only** time that secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

[JSON] [CSV]

[Close]

Figure 15. Copying access keys

6. Complete the **Add cloud storage** form.





- a. In the **Bucket** field, enter the name of the object bucket you created on the FlashBlade (Figure 16).

Figure 16. Add cloud storage page

- b. For each DDB partition you want to create, click the **Add** link next to the **Deduplication DB location** section. In the dialog that appears, select the appropriate MediaAgent, and enter or browse to the path in the file system where you want the DDB to reside (Figure 17). We recommend creating partitions on at least two MAs to provide performance and resilience. You can create up to four partitions per storage pool.

Figure 17. Adding a DDB

- c. Click the **Save** button to create the cloud storage pool.

To create a cloud storage pool using FlashBlade in release 11.22 through 11.25:





1. Following FlashBlade//S documentation, under the create a user account and bucket on the FlashBlade//S for Commvault data.
2. As shown in Figure 18, in Commvault Command Center, navigate to Storage, then Cloud. Click the **Add** link.

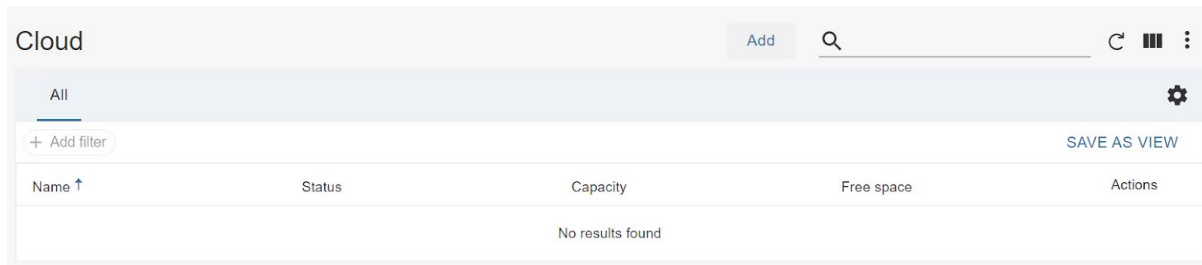


Figure 18. Cloud storage management

3. As shown in Figure 19, click the Cloud storage tile.

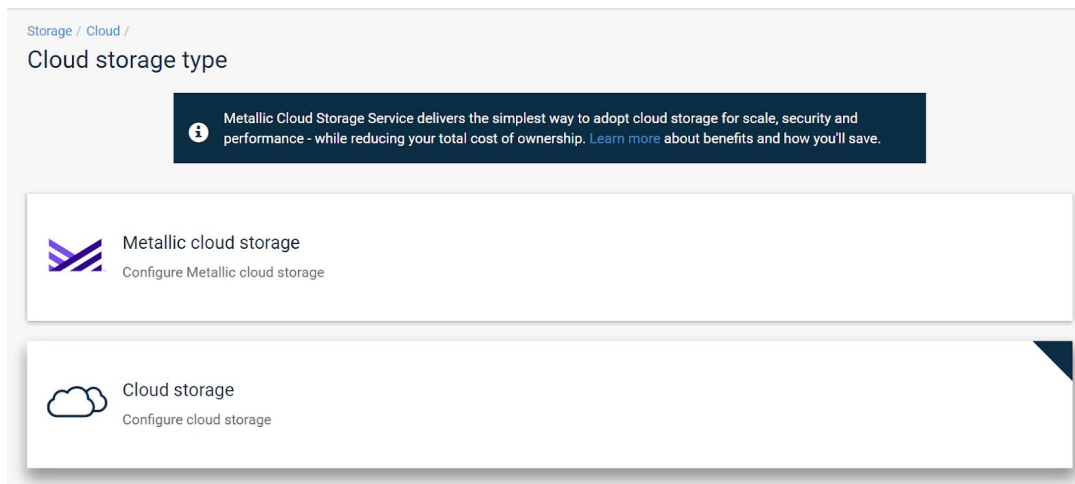


Figure 19. Cloud storage type selection

4. As shown in Figure 20, in the **Add cloud storage page**, enter the information as follows:
 - a. In the **Name** field, enter a display name for the cloud storage pool.
 - b. In the **Type** field, select "S3 Compatible Storage."
 - c. In the **MediaAgent** field, select the MediaAgent that should manage the pruning and be the preferred data path. If you need to add a MediaAgent, click the **Create New (+)** button and complete the deployment process before proceeding.
 - d. In the **Service host** field, enter the DNS name or data virtual IP address (VIP) for Commvault to use to access FlashBlade//S. Note that by default, Commvault will use TLS to secure the Amazon S3 connection. If the FlashBlade does not have a trusted certificate applied that matches the provided name, TLS will fail. There are three ways you can address this:
 - i. If you wish to use TLS, following FlashBlade//S documentation, obtain and apply a certificate signed by a trusted certification authority (CA). For TLS to work, you must enter a DNS name in Commvault that is associated with the FlashBlade//S data VIP, and the certificate must include that exact name in its Subject Alternative Name field.





- ii. On Commvault 11.26 or later, set the `nCloudServerCertificateNameCheck` additional setting on the MediaAgent, with a value of 2. This will leave TLS enabled, but it skips certificate validation for self-signed certificates for that MediaAgent. CA-signed certificates will still be validated.
 - iii. Include “http://” in the host value. This will prevent Commvault from attempting TLS.
 - iv. On Commvault 11.25 and earlier, set the `nCloudServerCertificateNameCheck` additional setting on the MediaAgent, with a value of 0. This will leave TLS enabled, but it skips certificate validation for all cloud libraries that MediaAgent connects to. This includes the public cloud. If TLS is required, a trusted certificate is a better method than bypassing certificate validation. For more details, see [Commvault's related documentation](#).
- e. In the **Credentials** field, click the **Create new (+)** button.
- i. In the **Credential name** field, enter a descriptive display name for the newly stored credential.
 - ii. In the **Access Key ID** field, enter the S3 access key ID for the FlashBlade object user account.
 - iii. In the **Secret Access Key** field, enter the secret key for the FlashBlade object user account.
 - iv. Click the **Save** button when all fields are configured.
- NOTE:** To reduce the risk of key compromise, copy the key values directly from the FlashBlade GUI and avoid saving them in a file.
- f. In the **Bucket** field, enter the name of the target bucket under the configured user account.
- g. In the **Deduplication DB location** field, click the **Add** link to open the **Add Deduplication DB location** form. Select the MediaAgent that will host the DDB, then enter or browse to a path to house the deduplication database for the storage pool. Click the **Save** button to add the DDB location and return to the **Add cloud storage** form. You can add up to four total locations to create a partitioned DDB. If you do not wish to use Commvault deduplication, instead disable the **Use deduplication** setting.
- h. Click the **Save** button when all fields are configured. The designated MediaAgent will connect to FlashBlade and validate the keys and bucket name.





Configure cloud

Name

Storage

Type

MediaAgent +

Service host

Credentials + ✎

Bucket

☒ Use deduplication

Deduplication DB location Add

MediaAgent ↑	DDB Location	
ac-cvcs	c:\ddb	✕
ac-cvma	c:\ddb	✕

[Equivalent API](#)

Figure 20. Add cloud storage page

Share the bucket across MediaAgents: The FlashBlade//S bucket should be shared to the MediaAgents with access to it. This lets Commvault manage a single FlashBlade//S bucket, with automatic load balancing across multiple data paths. To share a bucket:

1. In Commvault Command Center, navigate to the cloud storage pool.
2. In the **Buckets** tile, click the **Actions** button for the bucket, then click **Add MediaAgent**, as shown in Figure 21.

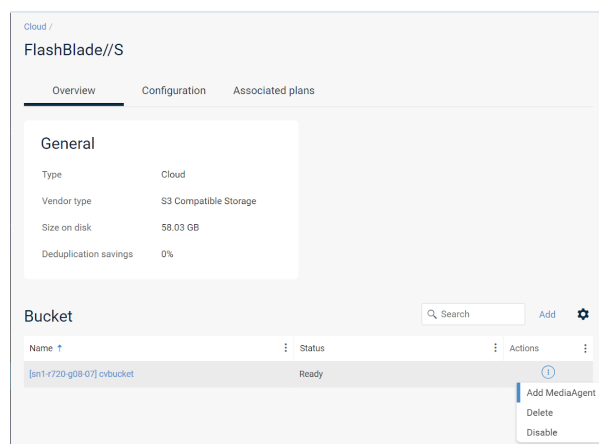


Figure 21. Cloud buckets

3. As shown in Figure 22, select the MediaAgents that should access the mount path, then click the **Save** button. MediaAgents will be added with read/write access.





Figure 22. Add MediaAgent form

Use Storage Accelerator for additional horizontal scaling: The Storage Accelerator feature lets clients write directly to object storage without having the MediaAgent package installed, and without having to share mount paths to them in the storage target configuration. The MediaAgent controlling jobs (referred to as a Data Server in this configuration) can manage more streams with fewer resources. Figure 23 illustrates the architectural differences with and without Storage Accelerator. Because clients are bypassing the MediaAgent to access the FlashBlade, network restrictions are critically important. If a security policy does not allow endpoints to directly access backup storage, they must use a traditional consolidated data mover architecture.

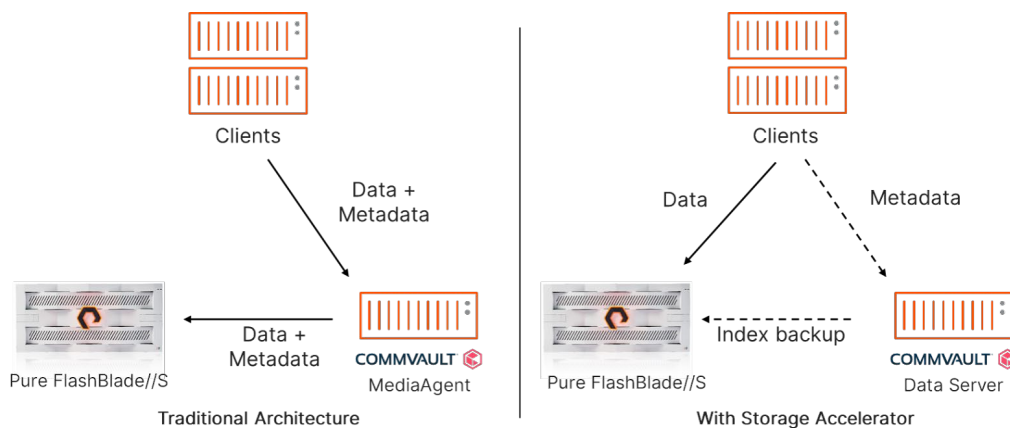


Figure 23. Traditional and Storage Accelerator architectures

Storage Accelerator improves overall backup performance across an environment, but it does increase the CPU load on an individual client. Using Storage Accelerator may even degrade backup and restore speed for clients whose CPUs are heavily loaded during backups. In these cases, you should consider using the traditional architecture for these systems and deploying Storage Accelerator on other, less-loaded clients. Using the same plans and policies for clients with and without Storage





Accelerator is fully supported and does not require any extra configuration. Simply deploy Storage Accelerator on the desired clients, and Commvault will automatically use it during backup and recovery.

Optionally, on Commvault release 11.20 or later, you can configure the CommServe to deploy the Storage Accelerator package automatically. Once automatic deployment is enabled, Commvault will automatically install the package at the end of the first subsequent backup job if it is not already installed.

To automatically install Storage Accelerator, you must use the CommCell Console interface.

1. On the **Home** ribbon menu, click the **Control Panel** button. Click the **Media Management** icon to open the **Media Management Configuration** dialog.
2. Locate the **Config parameter to enable the storage accelerator feature** item and change the value (Figure 24). Setting a value of 1 will enable automatic installation, while a value of 0 will disable it. Click **OK** to commit the change.

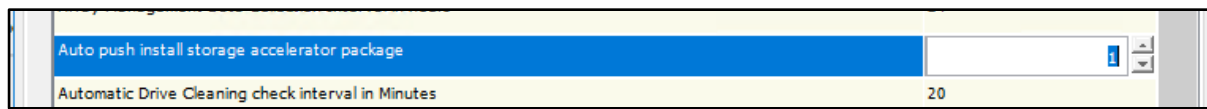


Figure 24. Enabling or disabling automatic push of Storage Accelerator

IMPORTANT: New CommCells deployed on 11.26 or later have this option enabled by default. If you do not wish to use Storage Accelerator, be sure to disable the option before running any backups to FlashBlade//S.

To enable Storage Accelerator on 11.22 and later:

3. Install the Storage Accelerator package.
 - a. In Command Center, navigate to Manage, then Servers. Select All from the Type dropdown to show all clients. Click the Actions (...) button for the client where you want to install Storage Accelerator, then select **Add software** from the popup menu.
 - b. As shown in Figure 25, select the Storage Accelerator package from the dropdown menu. You can type in the search box to filter the list. Click the **Install** button to start the installation.

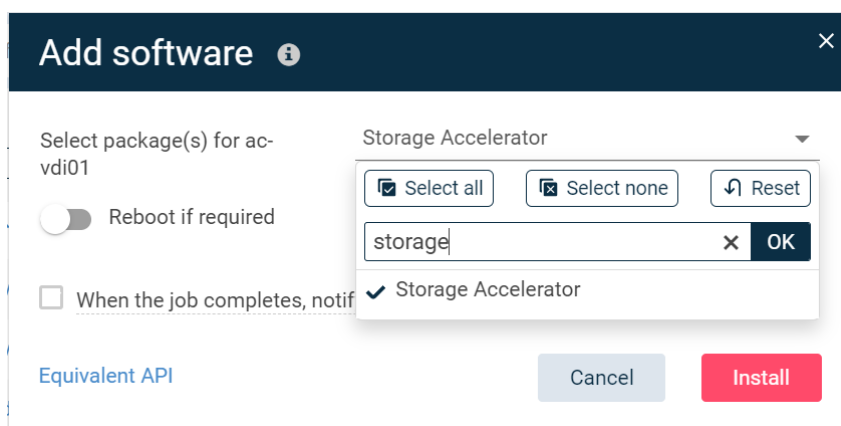


Figure 25. Installing the Storage Accelerator package

- c. Repeat the installation process for all clients where you want to enable Storage Accelerator.





4. For releases 11.22 and 11.23, you must globally enable Storage Accelerator functionality in the CommCell Console interface.
 - a. On the **Home** ribbon menu, click the **Control Panel** button. Click the **Media Management** icon to open the **Media Management Configuration** dialog.
 - b. Locate the **Config parameter to enable the storage accelerator feature** item and change the value to 1 (Figure 26). Click OK to commit the change.

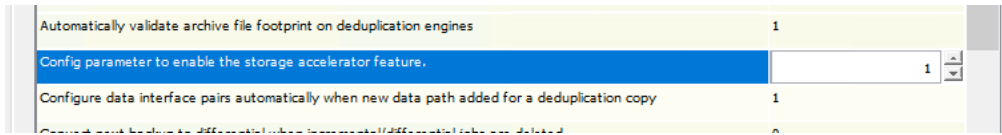


Figure 26. Globally enabling Storage Accelerator

Use separate credentials for Storage Accelerator: Beginning in Commvault release 11.23, you can assign a secondary credential to a cloud storage pool that will only apply for Storage Accelerator. This lets you create a restricted user that can read and create objects but not delete them. To assign a Storage Accelerator credential:

1. On the FlashBlade//S, create a new object access policy. Follow the same procedure as you did to create the first policy and select only the access rights listed in Table 6. The policy should not restrict the source IP addresses. This would limit the effectiveness of Storage Accelerator and could cause backup and recovery failures.

```
s3:GetBucketVersioning
s3:GetObject
s3:GetObjectAcl
s3:GetObjectTagging
s3:GetObjectVersion
s3:GetObjectVersionTagging
s3:ListBucket
s3:PutObject
```

Table 6. Object access rights for Storage Accelerator secondary credentials

2. Create a new object user in the object account and assign the new object access policy. Create keys, but do not close the key creation page.
3. In Commvault Command Center, navigate to the FlashBlade//S cloud storage pool. In the Storage Accelerator credential field, click the **Create (+)** button to open the New credential form.
4. Give the credential a clear name. Copy the access key ID and secret key from the FlashBlade//S interface and paste them in the appropriate fields in Command Center (Figure 27). Optionally, enter a description. Click the **Save** button to finish creating the credential.





Add credential

Credential name: FlashBlade//S Storage Accelerator

Access key ID: PSFBSAZRKDAGMKCLPKMEGKMBOAJFKFHCFD

Secret access key: [Masked] [Eye icon]

Description: Restricted access for Storage Accelerator

Security: [Dropdown arrow]

Owner: [User icon] admin [X icon]

User/Group: [Search icon] Enter users or user groups

[Equivalent API](#) [Cancel] [Save]

Figure 27. Creating a saved credential for Storage Accelerator

5. The new credential will appear in the field. Click the check mark button to apply the credential to the cloud storage pool (Figure 28).

Configuration

Enable: [Toggle On]

Prevent new data writes to backup location: [Toggle Off]

Storage accelerator credentials: FlashBlade//S Storage Accelerator [Checkmark] [X]

Figure 28. Applying Storage Accelerator credentials

Set deduplication block size to 128KB: Deduplication block size affects performance and storage consumption. Larger block sizes yield better performance but consume more storage. Commvault's recommendation for FlashBlade object storage is to set the block size to 128KB to maximize data reduction. Storage policies tied to cloud storage targets default to 512KB in most cases, so you will need to change the setting manually. Deduplication block size is configured at the storage pool, or at the storage policy if it is not tied to a pool, using the CommCell Console interface.





To set deduplication block size for a storage pool:

1. In the CommCell Browser pane on the left, navigate to **Storage Pools**, then right-click the desired pool. Select **Properties**, then **Storage Pool** to open the **Properties** dialog.
2. Click the **Advanced** tab. As shown in Figure 29, set the **Block level Deduplication factor** dropdown to 128. Click **OK** to commit the change.

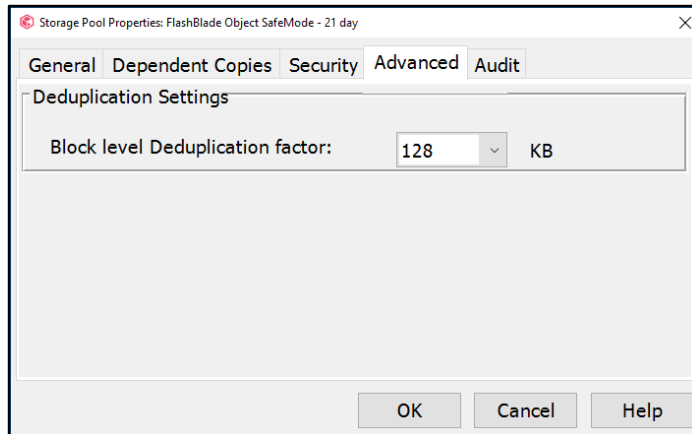


Figure 29. Storage pool deduplication factor

To set deduplication block size for a storage policy that is not tied to a storage pool:

1. Navigate to **Storage Policies**, then open the **Properties** dialog for the desired policy.
2. Click the **Advanced** tab. As shown in Figure 30, set the **Block level Deduplication factor** dropdown to 128. Click **OK** to commit the change.

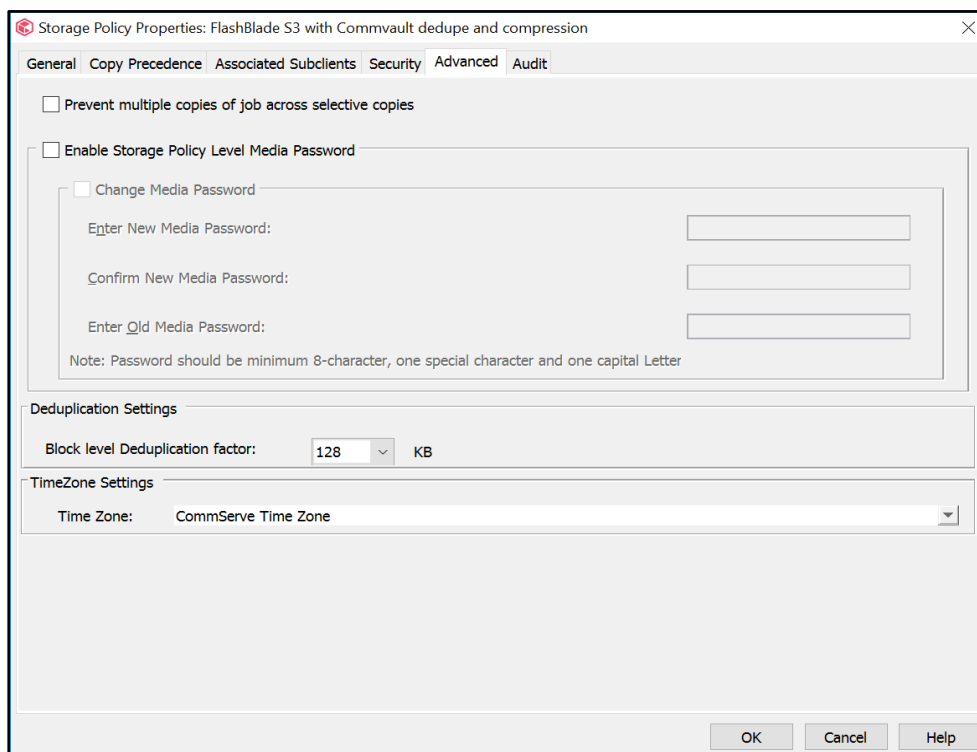


Figure 30. Storage policy deduplication factor





Run space reclamation regularly: Due to the way deduplicated blocks are stored and aged in object storage, stale data can build up over time and consume additional storage. Beginning with release 11.26, Commvault can periodically reclaim some or all of this extra space by rewriting objects using only the valid blocks. You can control when the process runs and how aggressively it reclaims space.

We recommend using either the default aggressiveness of 3, which recreates objects that have 40% stale data, or 2, which waits until objects are 60% stale. A higher aggressiveness level will reclaim more capacity, but it will also put more load on the MediaAgents and FlashBlade//S. Likewise, a lower aggressiveness level will introduce less load but will reclaim less capacity.

For new CommCell deployments on 11.26 and later, Commvault will create a system schedule that runs daily. We recommend changing this schedule to run weekly to align with how data aging works in most environments.

IMPORTANT: You should not run space reclamation when using Object SafeMode, as this may consume significantly more storage that will be difficult to clean up.

To change the schedule frequency:

1. In the CommCell Console, expand **Policies**, then click **Schedule Policies**. Right-click “**System Created DDB Space Reclamation schedule policy**,” then click **Edit**.
2. In the Tasks table, select the schedule, then click the **Edit** button (Figure 31).

System Created DDB Space Reclamation schedule policy

General Associations Alert Security

Name: System Created DDB Space Reclamation schedule policy

Type: DDB Space Reclamation

Description: System created schedule policy for DDB Space Reclamation

Schedule Name	Job Type	Pattern
		Every day at 11:00 AM

Add Edit Delete

OK Cancel Help

Figure 31. Editing the space reclamation schedule





3. You can change the frequency options on the **Schedule Pattern** tab (Figure 32).

Figure 32. Reclamation schedule pattern

4. Click the **OK** button to set the new schedule, then click the **OK** button on the main dialog box to commit the change.

(Optional) Increase cloud thread pool size: Commvault uses a pool of threads to connect to object storage. By default, a process on the MediaAgent or client with Storage Accelerator can create up to 50 connections. All backup jobs using that MediaAgent or client will share the connections. The number of active backup streams does not directly correlate to the number of threads in use.

You should only consider increasing the thread pool size if connection counts from a MediaAgent to the FlashBlade regularly exceed 40 and the MediaAgent network interface is not saturated. First-generation FlashBlade can often benefit from pool sizes of 100 or more, while FlashBlade//S usually does not benefit from a larger pool. We do not recommend setting a pool size greater than 80 with FlashBlade//S.

To set a different thread pool size limit for a client or MediaAgent, you need to apply the `nCloudGlobalUploadThreadPoolMaxCount` additional setting. This does not force the system to use a certain number of threads. Instead, it simply allows it to go beyond the default 50. (Get [more information on the setting.](#))

The best way to apply the setting is through a server group, a logical grouping of Commvault systems. To create a group to control the thread count:

1. In Command Center, expand Manage in the left-hand navigation, then click **Server Groups**. Click the **Add server group** link, as shown in Figure 33.

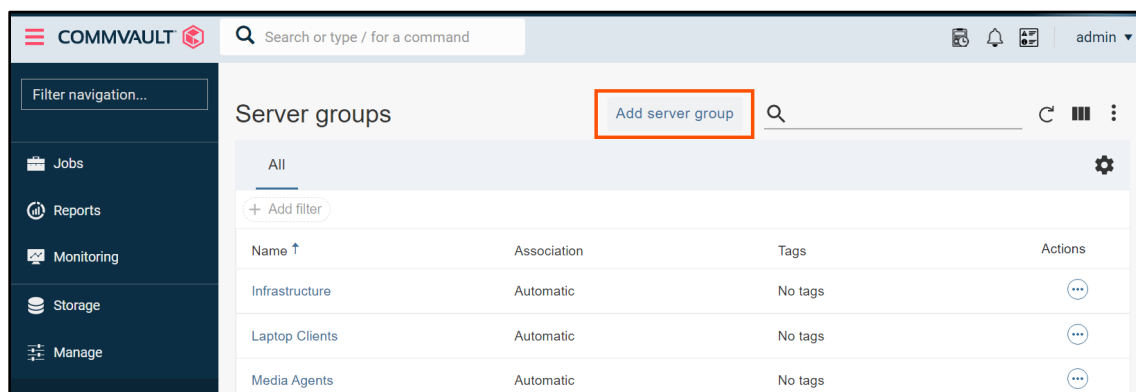


Figure 33. Creating a server group





- As shown in Figure 34, give the group a descriptive name. Select the **Manual association** option. For each client or MediaAgent that will send data to FlashBlade, type part of the client name, then click the **Add** button. When all the appropriate clients are selected, click the **Save** button.

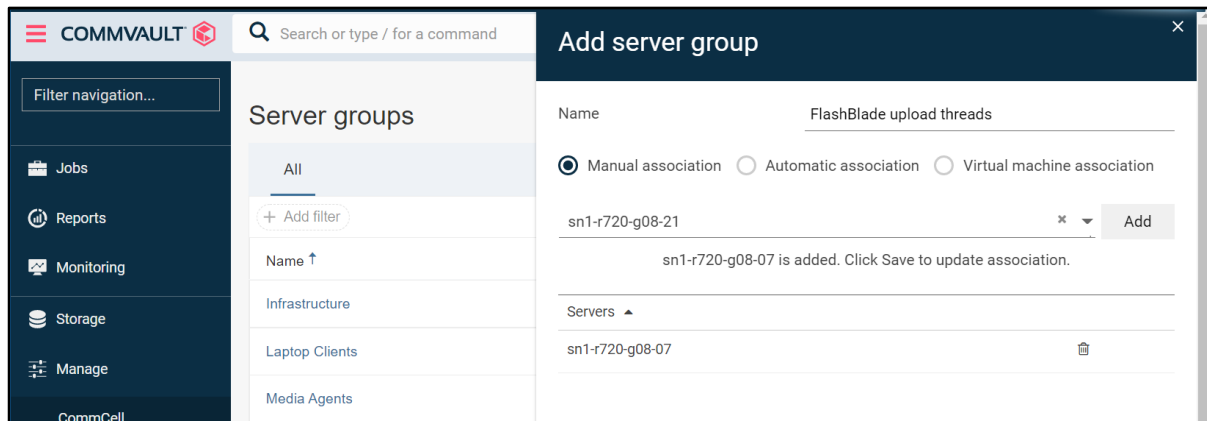


Figure 34. Setting server group members

- In the left-hand navigation, expand **Manage**, then click **System**. In the right pane, click the **Additional settings** tile. As Figure 35 shows, on the Additional settings page, click the **Add** link, then select **Entity Settings**.

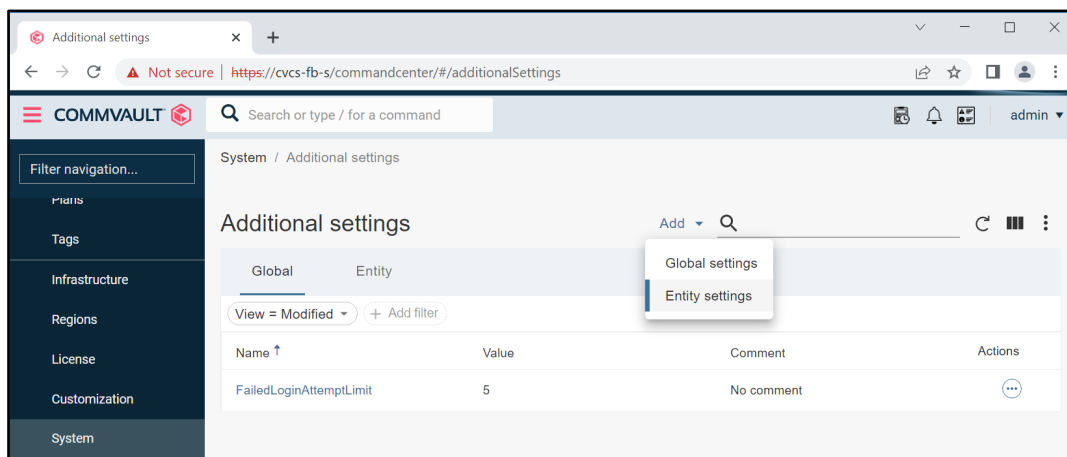
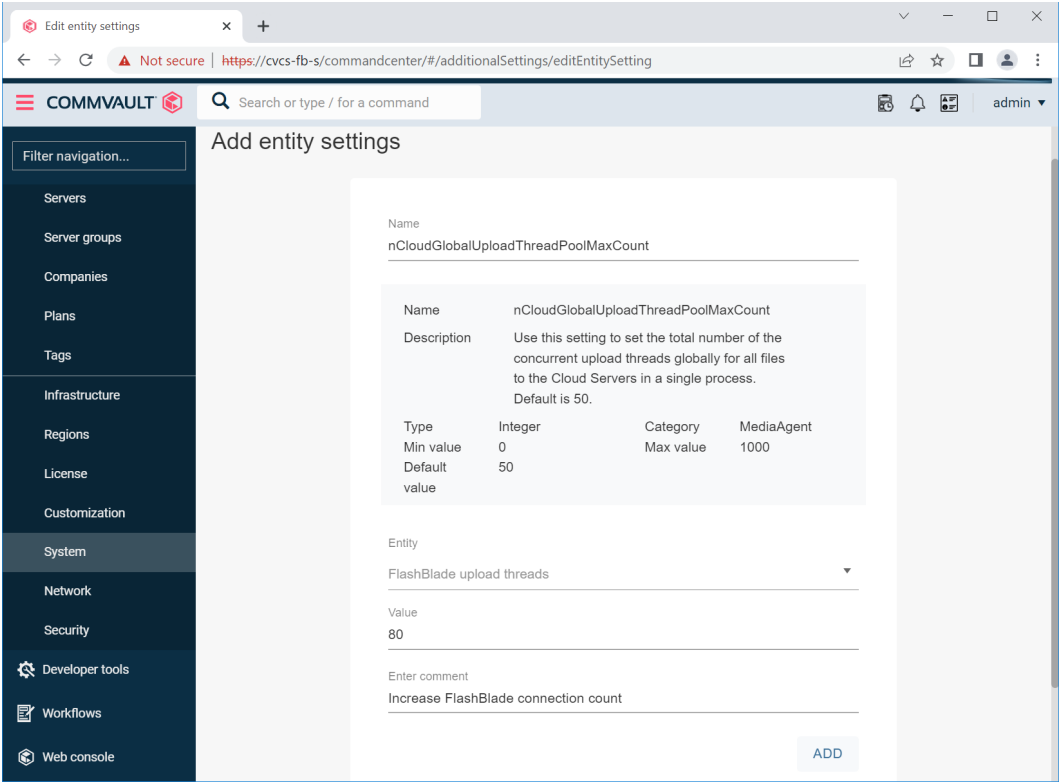


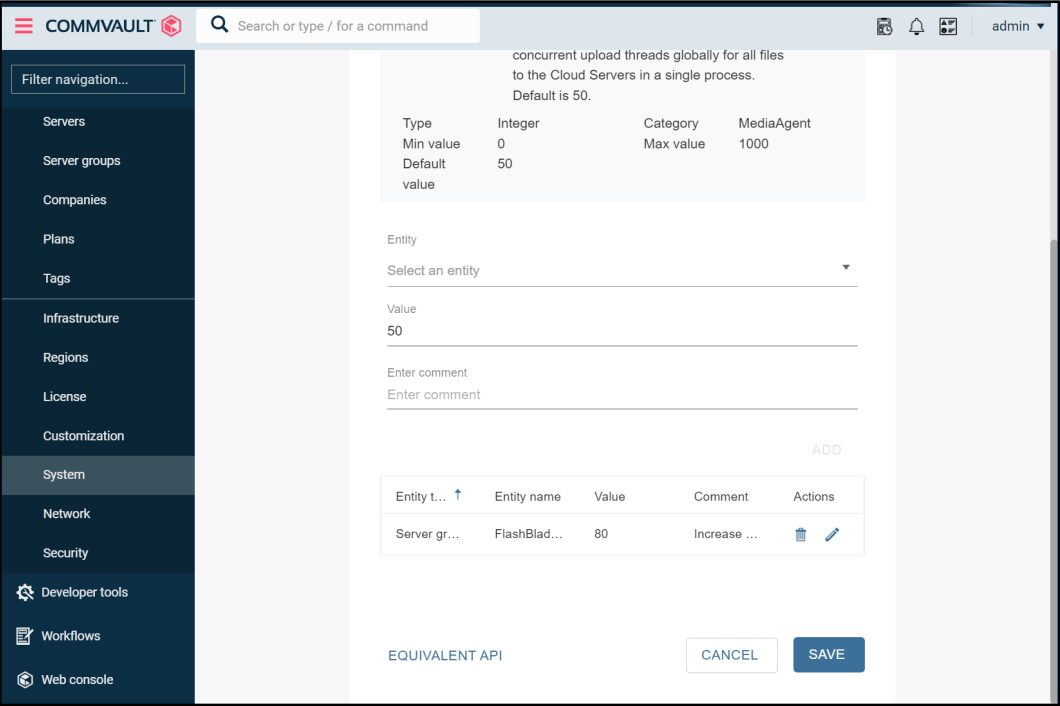
Figure 35. Creating additional settings

- As Figure 36 shows, on the **Add entity settings** page, enter `nCloudGlobalUploadThreadPoolMaxCount` in the **Name** field. When the setting appears in the search list, click it to automatically prepopulate all the fields. From the **Entity** dropdown, expand **Server Groups**, then select the group you just created. In the **Value** field, enter the desired maximum number of threads per system. Enter a text value in the **Description** field. Click the **Add** button to create the association. Repeat the process if you want to set the value for other clients or groups.





5. When all the entity associations are added, click the **Save** button to apply the setting (Figure 37). The change will be honored in the next job writing to FlashBlade for any of the associated clients and group members.





(Optional) Increase pipeline buffer count: In some environments, there may be a network bottleneck between the backup agent and MA components, even if they are installed on the same servers. [Increasing the number of pipeline buffers](#) may improve performance. Figure 38 shows the setting.

nNumPipelineBuffers				
Entity type ↑	Entity name	Value	Comment	Acti...
Server group	Media Agents	330	FB	

Figure 38. Increasing pipeline buffers

This setting increases the RAM usage on both the agent and MA systems, and the optimal setting is highly dependent on the specific environment. Use caution when changing it; increase it by small increments, and test thoroughly before changing production settings.

(Optional) Increase look-ahead size: During restore, Commvault will [read additional deduplicated blocks](#) under the assumption that it will need them, saving time having to fetch them from storage later. By default, with object storage it will fetch two times the deduplicated block size. Increasing the look-ahead size can improve restore performance with FlashBlade//S. The impact may vary between deployments and data sets, so test the setting thoroughly in your own environment before deploying in production. Figure 39 shows the setting.

SILookAheadAsyncIOBlockSizeKB				
Entity type ↑	Entity name	Value	Comment	Actions
Server group	Media Agents	768	Faster restore	

Figure 39. Increasing look-ahead block size

Enabling Object SafeMode Ransomware Mitigation

The following best practices apply only when using Object SafeMode ransomware mitigation. Review [Planning for Object SafeMode](#) before enabling the feature to ensure your FlashBlade//S is sized correctly for immutability.

Configure Commvault vaulting interval with Object SafeMode: You can set Commvault to regularly seal the vault by creating a new DDB on a regular interval. DDB settings are managed in the CommCell Console interface. In the **CommCell Browser** pane, expand **Storage Resources**, then expand **Deduplication Engines**. Right-click the appropriate DDB, then select **Properties**. In the dialog that opens, select the **Deduplication** tab, then the **Settings** tab on that properties page. Enable the first **Create new DDB every** checkbox, then set the desired number of days for the vaulting interval (Figure 40). Click the OK button to commit the change.



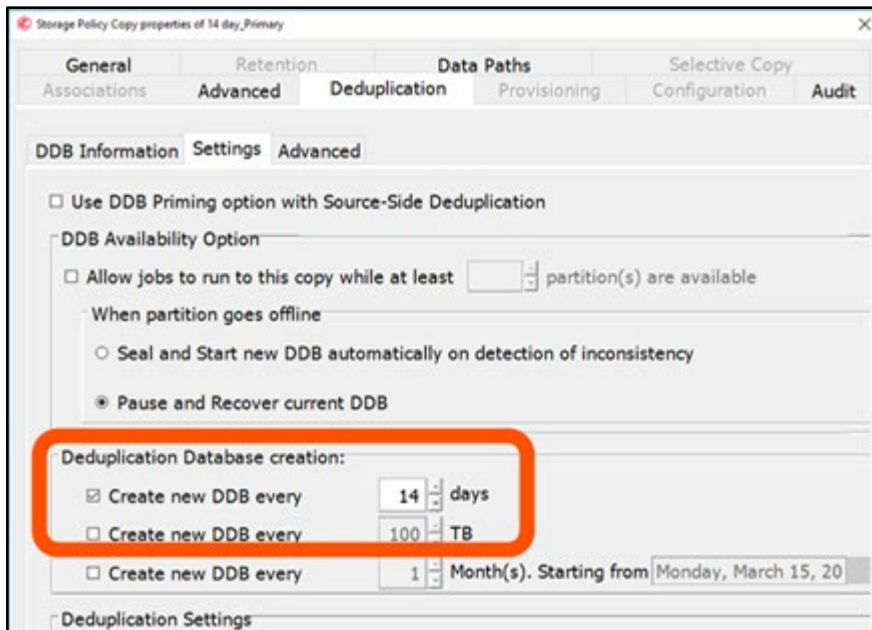


Figure 40. Configuring vaulting interval

Disable micro pruning with Object SafeMode: When using Object SafeMode, it is important to disable the micro pruning feature for the FlashBlade bucket. Micro pruning will periodically refresh certain non-data objects, effectively resetting the lock timeout for these objects and causing pruning failures. While data objects are not affected, some jobs will not age, DDBs will not be deleted, and the non-data objects can consume a significant amount of storage. Micro pruning is managed using the CommCell Console interface. In the CommCell Browser pane, expand **Storage Resources**, then expand **Libraries**. Expand the library associated with the FlashBlade//S storage pool. Right-click the mount path, then select **Properties**. In the dialog that opens, select the **Allocation Policy** tab. Deselect the **Enable Micro Pruning** checkbox (Figure 41), then click the **OK** button.

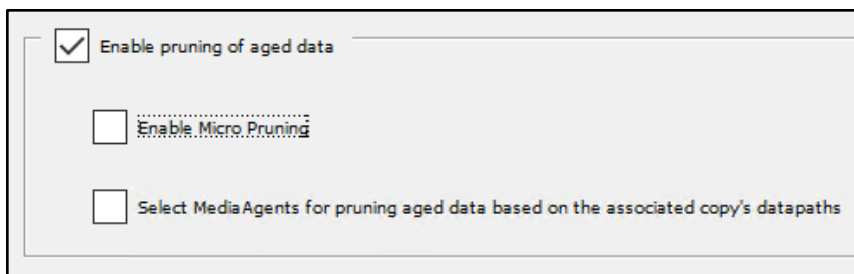


Figure 41. Disabling micro pruning

IMPORTANT: Do not clear the **Enable pruning of aged data** checkbox. This would prevent Commvault from deleting expired data from FlashBlade//S.

Configure non-deduplicated policy for Index and DDB backups: Index and DDB backups related to FlashBlade//S will default to using the same server backup plan as the client data they relate to. However, Commvault manages retention for these backups independently from the backup plans. When combined with periodic DDB sealing and Object SafeMode immutability, these backups can place dependencies on sealed DDBs that will make Commvault keep client backups longer than expected. To avoid these issues, create a separate non-deduplicated storage pool and a storage policy specifically for index and DDB backups, then associate the index and DDB backup subclients to the new policy.





To create a non-deduplicated storage pool:

Create a cloud storage pool following the same procedure as in [Configure a Single Object Bucket](#), but instead of adding DDB partitions, disable the **Use deduplication** option (Figure 42). Use the same bucket name and credentials as the deduplicated storage pool.

Configure cloud

Name *

FlashBlade//S Non-deduplicated

Storage

Type

S3 Compatible Storage

MediaAgent *

sn1-r720-g08-07

Service host *

http://10.21.237.25

Credentials *

FlashBlade//S S3

Bucket *

cvbucket

Use deduplication ☐

EQUIVALENT API CANCEL SAVE

Figure 42. Adding a non-deduplicated cloud storage pool

Share the new storage pool to the same MediaAgents as the deduplicated pool, following the same procedure as in [Share the Bucket Across MediaAgents](#).

To create a storage policy from the pool:

In the CommCell Console, expand **Policies**. Right-click **Storage Policies**, then click **New Storage Policy**. Complete the wizard as follows:

1. Select the **Data Protection and Archiving** option, then click the **Next** button.
2. Enter a name for the storage policy in the **Storage Policy Name** field, then click the **Next** button. Do not enable any of the checkboxes.





3. From the **Storage Pool** dropdown, select the non-deduplicated pool you created (Figure 43), then click the **Next** button.

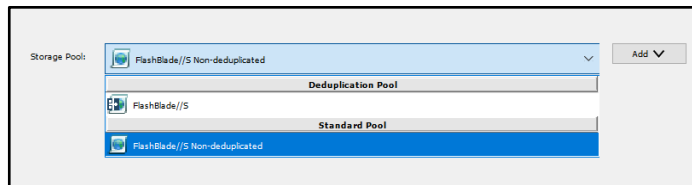


Figure 43. Storage pool selection

4. Set the storage policy retention to be longer than your Object SafeMode retention. If your Object SafeMode retention period is shorter than 30 days, accept the default primary retention setting of 30 days and 1 cycle (Figure 44). Click the **Next** button to continue.

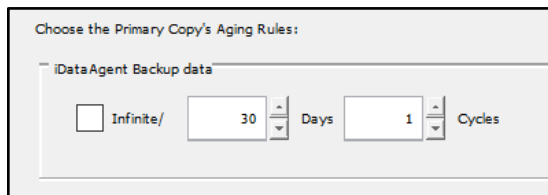


Figure 44. Default data retention

5. Click the **Finish** button to create the storage policy.

To reassociate the index and DDB backup subclients for each FlashBlade//S server backup plan or storage policy:

1. In the CommCell Console, expand **Policies**, then click **Storage Policies**. In the right-hand pane, locate the storage policy associated with the server backup plan. Right-click the policy, then click **Properties**.
2. Select the **Associated Subclients** tab.
3. In the client list, select the index and DDB backup subclients, then click the **Re-Associate** button. You may need to resize the window to see the subclient names. In the Re-Associate Subclient(s) dialog box, select the new storage policy, then click the **OK** button (Figure 45).

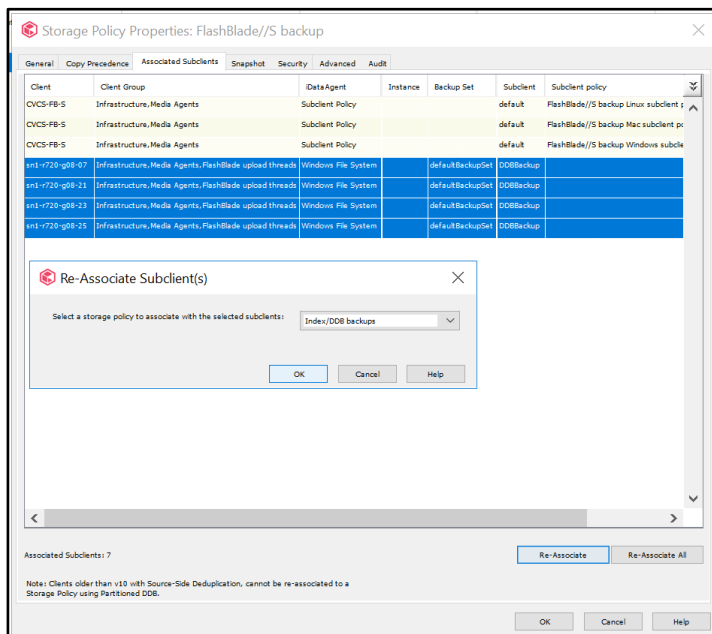


Figure 45. Reassociating subclients





4. Click the **OK** button to complete the association change. Subsequent backups will not be tied to the DDB and will not affect pruning of client data.

IMPORTANT: If you create new server backup plans, storage policies, or storage pools using FlashBlade//S as a target, you must associate any new index or DDB subclients to the non-deduplicated storage policy.

NFS

NFS is a proven protocol for FlashBlade//S with Commvault. It has the advantages of transparency at the OS level and familiarity for IT administrators. Due to TCP connection behavior, NFS requires more configuration elements with Commvault than object storage to achieve comparable scale and performance.

For best NFS results:

- Enable SafeMode snapshots for ransomware mitigation
- Configure four NFS mounts per MediaAgent
- Distribute capacity evenly across mounts
- Do not mount the same file system on different VIPs
- Use 512KB NFS buffers
- Use consistent mount paths between MediaAgents
- Create Commvault mount paths for each NFS mount
- Share mount paths across MediaAgents
- Configure spill and fill across mount paths

When to use NFS: You might prefer to use NFS instead of object storage because it is a more familiar protocol. This is fully supported, it just means you will have more configuration work to do.

Required MediaAgent operating system: Linux is the required operating system for MediaAgents when using NFS. While Windows has an available NFS client that is compatible with FlashBlade, Commvault does not support new deployments using NFS with Windows MediaAgents.

Enabling SafeMode for ransomware mitigation: FlashBlade snapshots operate at the file level. If a file does not change between snapshots, it consumes no extra storage. If the file is deleted or modified, it consumes capacity in the size of the file, minus any savings from compression. With backup software, most files are written once, retained for a period based on policies, then deleted. Backup data, therefore, does not consume significant snapshot space until it is pruned. With backup software, the storage used by snapshots comes primarily from deleted data. Follow these guidelines to ensure you have adequate storage available on FlashBlade.

To estimate the required capacity in an established CommCell, run a **Data Retention Forecast and Compliance** report in the CommCell Console. Set the **Forecasted** option to the planned retention period. If you want to have different retention periods between Commvault backups and SafeMode snapshots, run two reports, one for each value. The data copy will consume the amount in the **Space to Keep (GB)** field in the **Disk Media Summary** section of the report, while SafeMode snapshots will use the amount in the **Space to Free (GB)** field. Add these amounts together to get the total additional storage needed.





To estimate for a new CommCell, you need the baseline size, daily change rate, and expected data reduction rate. Apply the reduction rate to the daily change rate, multiply by the number of days snapshots will be kept, then double the result. Add the baseline to determine the total expected capacity required to implement SafeMode snapshots.

For example, in an environment with 300TiB of data, the baseline after initial data reduction could be 180TiB. If the daily change rate is 10TiB and data reduction is 2:1, the overall backup change rate is 5TiB per day. Across a seven-day retention period, there would be 35TiB of data change, plus another 35TiB kept in snapshots. The total additional capacity would be 250TiB. Your Pure Storage and Commvault sales teams can assist with estimating your data sizes.

Detailed Best Practices

Configure four NFS mounts per MediaAgent: To achieve maximum write throughput, FlashBlade needs TCP connections spread across as many blades as possible. NFS clients consolidate all mounts that use the same IP address into the same one or two TCP connections. More mounts to different IP addresses are needed to distribute connections to more blades. Using four mounts balances between performance and complexity. Increasing the number of mounts may increase backup performance, but you have more file systems and IP addresses to manage. To facilitate sharing file systems across MediaAgents, the best practice is to mount the same file system to each MediaAgent using the same local path and FlashBlade data VIP, as shown in Figure 46.

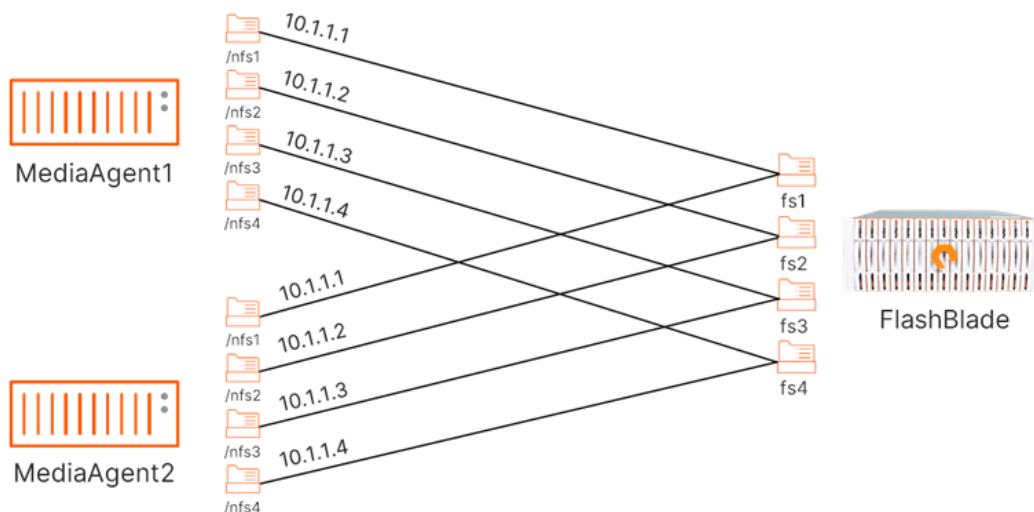


Figure 46. Consistent mounting of file systems across MediaAgents

Configure a restrictive export policy on file systems: Minimizing the attack surface of an environment is critical to preventing data loss due to a ransomware attack, and export policies are an important part of that. Properly configured policies prevent direct access to stored data outside the systems that need it. In this case, only the secondary MediaAgents should have access to the FlashBlade file systems.

The export policy should include the following options:

- NFSv4.1
- NFSv3 (optional)





- Rules to restrict access to only the IP address(es) for the secondary MediaAgent(s). Enter a separate rule for each MediaAgent rather than using a subnet filter.
- rw
- root_squash

For example, as Figure 47 shows, a rule to grant access for a MediaAgent with the address 10.1.1.1 would be written as 10.1.1.1/32(rw,root_squash).

Figure 47. Mount path export policy

Don't mount the same file system using different VIPs: Multiple NFS mounts on the same MediaAgent must use different file systems. Mounting the same file system using multiple IP addresses results in inaccurate library capacity reporting and creates a risk of backup corruption in specific scenarios.

Use 512KB NFS buffers: The read and write buffers for each mount should be set to 512KB for best performance. On Linux, the buffers should be set per mount.

On Linux/Unix, run:

```
mount -t nfs4 -o vers=4.1, rsize=524288, wsize=524288, hard <IP address>:/<file system> <mount path>
```

Optionally, use NFSv3:

```
mount -t nfs -o rsize=524288, wsize=524288, hard <IP address>:/<file system> <mount path>.
```

Use consistent mount paths between MediaAgents: To simplify administration, use a consistent mount scheme across all MediaAgents. For each file system on FlashBlade, use the same data VIP and mount path on each MediaAgent. This provides a consistent list for populating the fstab file on Linux and makes scripting mounts easy. It also simplifies setting up mount path sharing in Commvault.





Create Commvault mount paths for each NFS mount: Each NFS mount must be configured in Commvault as a mount path in a disk storage pool. To create a new storage pool:

1. In Commvault Command center, navigate to **Storage**, then **Disk**. Click the **Add** link (Figure 48).

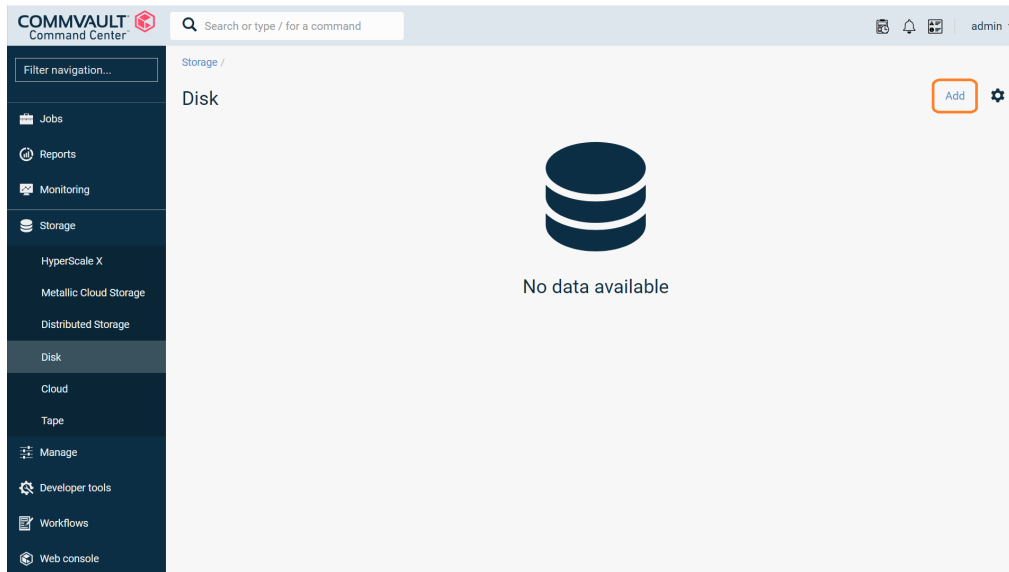


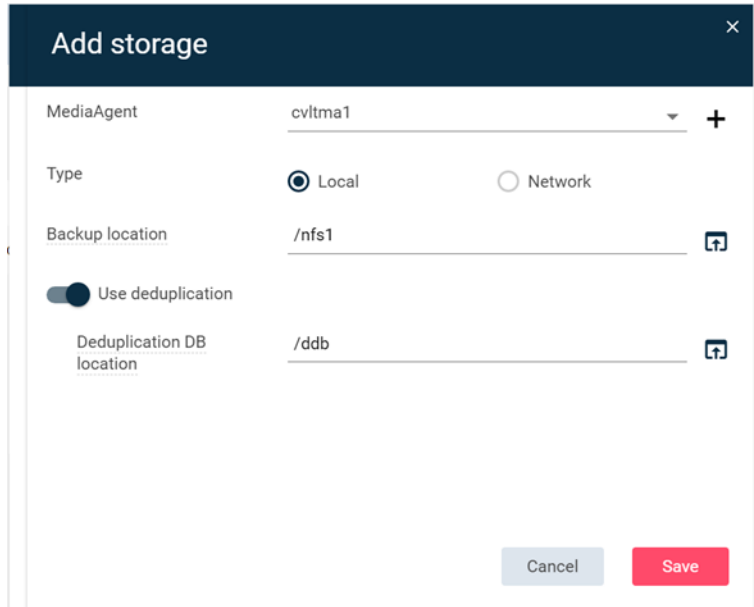
Figure 48. Adding a disk storage pool

2. In the **Name** field, enter a descriptive name for the storage pool. Click the Add link (Figure 49).

Figure 49. Add disk form

3. As shown in Figure 50, select a MediaAgent to act as the primary data path and perform pruning. Enter or browse to the first NFS mount directory. Enter or browse to the path to use for the new deduplication database. Click **Save** when complete.



Add storage [X]

MediaAgent: cvltma1 [v] +

Type: ☒ Local ☐ Network

Backup location: /nfs1 [icon]

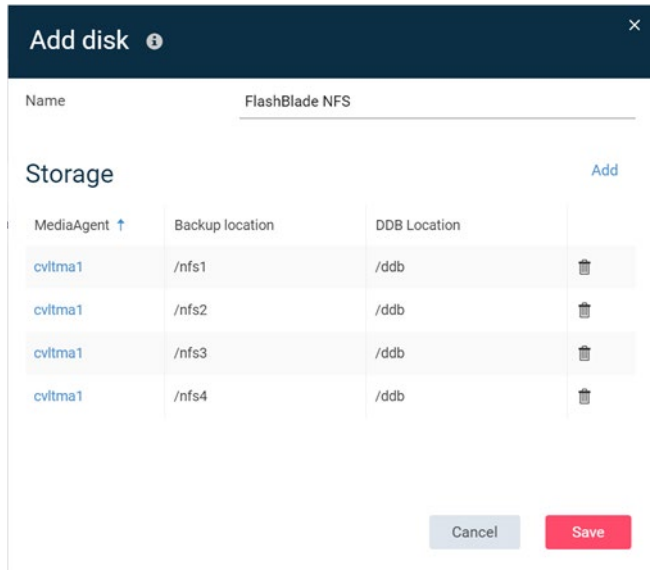
☒ Use deduplication

Deduplication DB location: /ddb [icon]

[Cancel] [Save]

Figure 50. Disk storage pool options

- Repeat Step 3 to add the remaining mounts. Use the same DDB location for all NFS mount paths. Click **Save** to create the disk storage pool. Figure 51 shows the expected result with four mount paths.



Add disk [i] [X]

Name: FlashBlade NFS

Storage [Add]

MediaAgent ↑	Backup location	DDB Location	
cvltma1	/nfs1	/ddb	[trash]
cvltma1	/nfs2	/ddb	[trash]
cvltma1	/nfs3	/ddb	[trash]
cvltma1	/nfs4	/ddb	[trash]

[Cancel] [Save]

Figure 51. Add disk form with four mount paths

Share mount paths across MediaAgents: Each NFS mount will be configured as a mount path in Commvault and shared to the MediaAgents with access to it. This lets Commvault manage a single mount path for each FlashBlade file system, with automatic load balancing across multiple data paths and file systems.

Before you begin, make sure the NFS mounts have been set up on all MediaAgents that will share the mount paths. NFS mount paths are shared in the CommCell Console interface.

To share a mount path:





1. In the CommCell Console, expand Storage Resources, then Libraries. Click the library matching the storage pool you created (Figure 52). The mount paths will appear in the right pane.

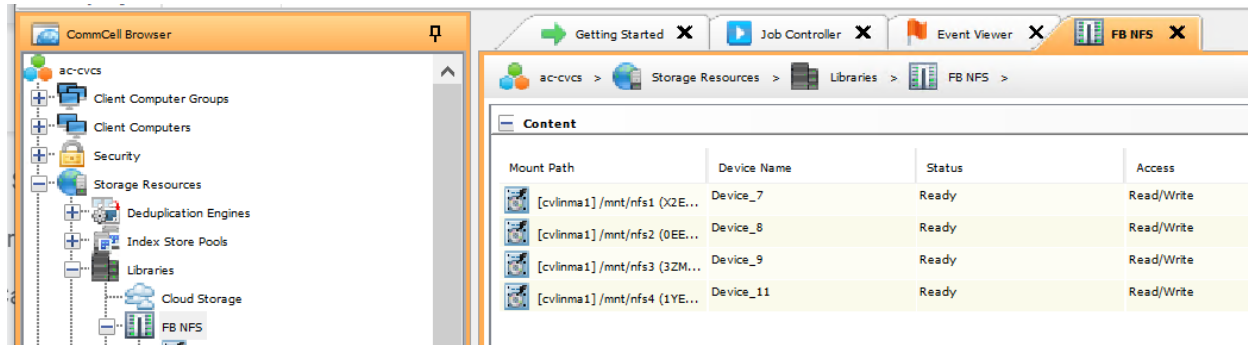


Figure 52. Library mount paths

2. Right-click the first mount path, then select **Share Mount Path** from the context menu to open the **Sharing** tab of the **Mount Path Properties** dialog (Figure 53). Click the **Share** button.

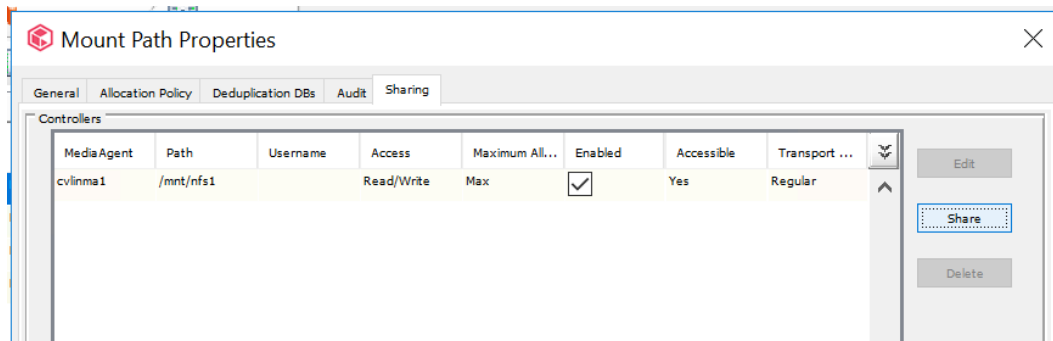


Figure 53. Mount path sharing

3. In the **Share Mount Path** dialog, set the **Transport Type** dropdown to "Regular." Select the checkbox for each MediaAgent that will share the mount path and set its access type to "Read/Write." Select the **Local Path** option, then enter or browse to the path where the file system is mounted. Click **OK** to commit the change (Figure 54).

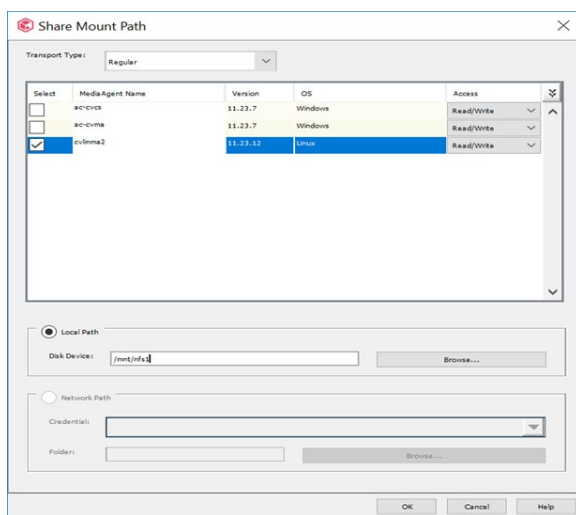


Figure 54. Share Mount Path dialog

4. The additional MediaAgents will appear in the list (Figure 55). Click **OK** to apply the sharing changes for the mount path.





NOTE: Commvault will add the MediaAgents to all the other mount paths, but with the IP sharing model. You will need to remove the MediaAgents from these paths and share them using the Regular model, as with the first path.

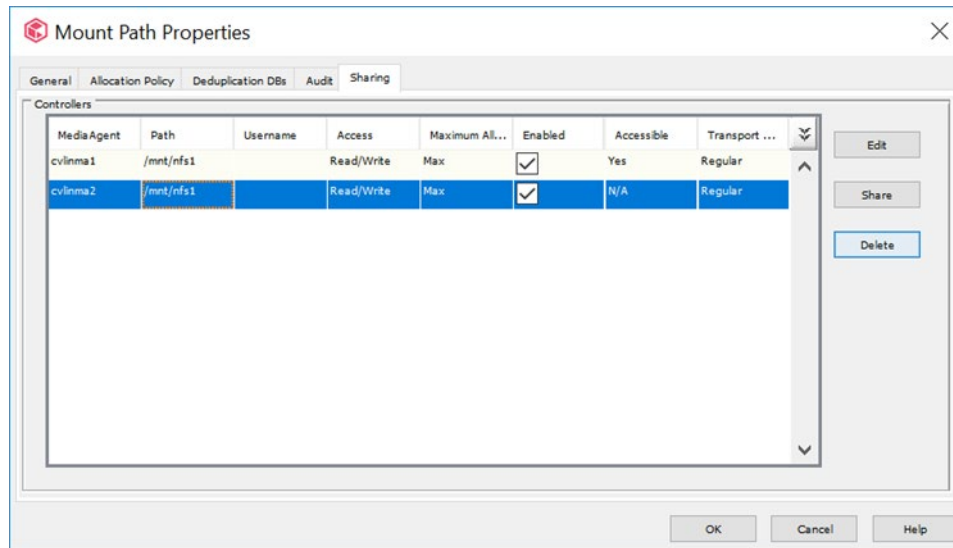


Figure 55. Updated mount path sharing list

For each additional mount path, right-click the path, then select **Share Mount Path** to open the mount path sharing settings. For each MediaAgent set with the IP sharing model, select the MediaAgent and click the **Delete** button (Figure 56). Once all IP sharing has been removed, click the **Share** button. Follow the instructions in step 3 to add the MediaAgent with the Regular sharing model.

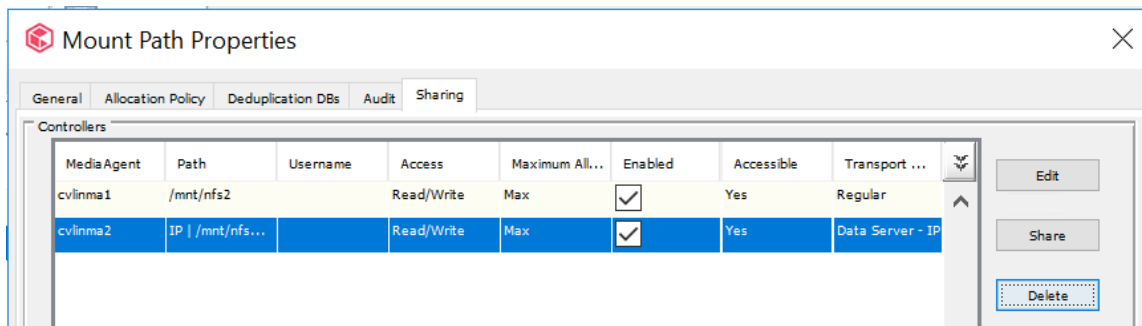


Figure 56. Delete MediaAgent from mount path sharing

When the mount paths have been shared correctly, the mount path properties in Command Center will show all the MediaAgents and their associated local mounts (Figure 57).



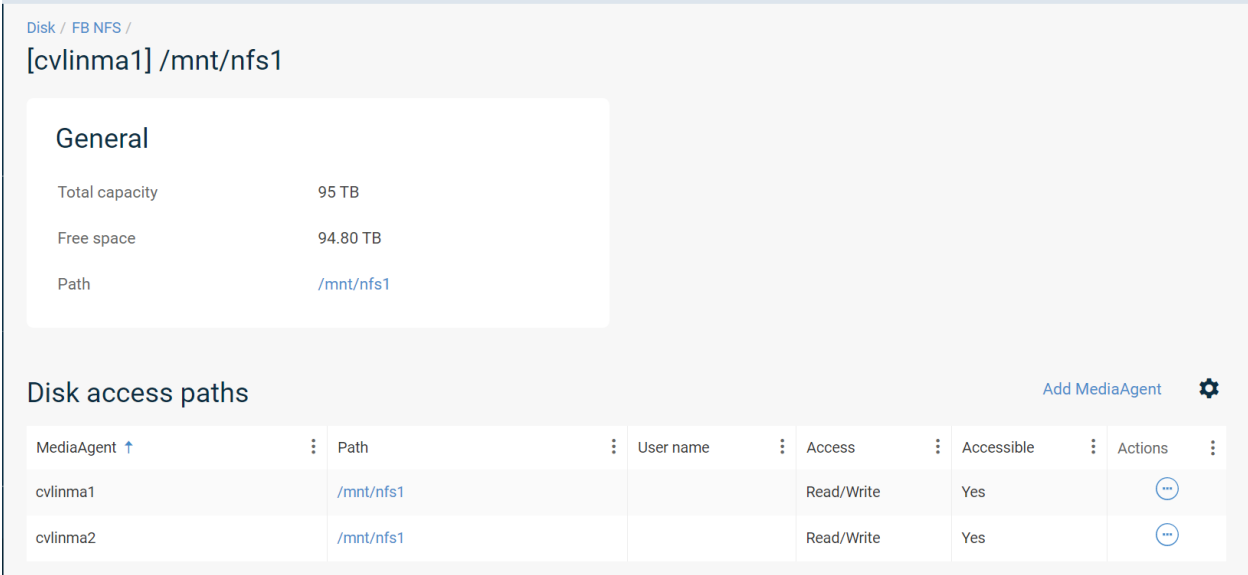


Figure 57. Mount Paths in Command Center

Run space reclamation: Commvault has a space reclamation procedure that removes expired blocks when hole drilling is not supported. The process rewrites the valid blocks into new data files and then deletes the ones that contain expired blocks.

Do not run the cleanup process during the backup window. While it reclaims full capacity, it places additional load on the MediaAgents and FlashBlade that can negatively impact backup performance.

The process should be run at least once a month. As of release 11.26, the process runs on an automatic schedule, so in most cases, no further action is required.

By default, the schedule runs daily at 11:00 AM, with the reclamation level at 3. This will rewrite files that have 40% expired blocks. If the reclamation process is causing too much read activity, you can change the frequency in the policy, or you can create a separate policy with a different reclamation level.





To change the schedule frequency:

1. In the CommCell Console, expand **Policies**, then click **Schedule Policies**. Right-click "System Created DDB Space Reclamation schedule policy," then click **Edit**.
2. In the Tasks table, select the schedule, then click the **Edit** button (Figure 58).

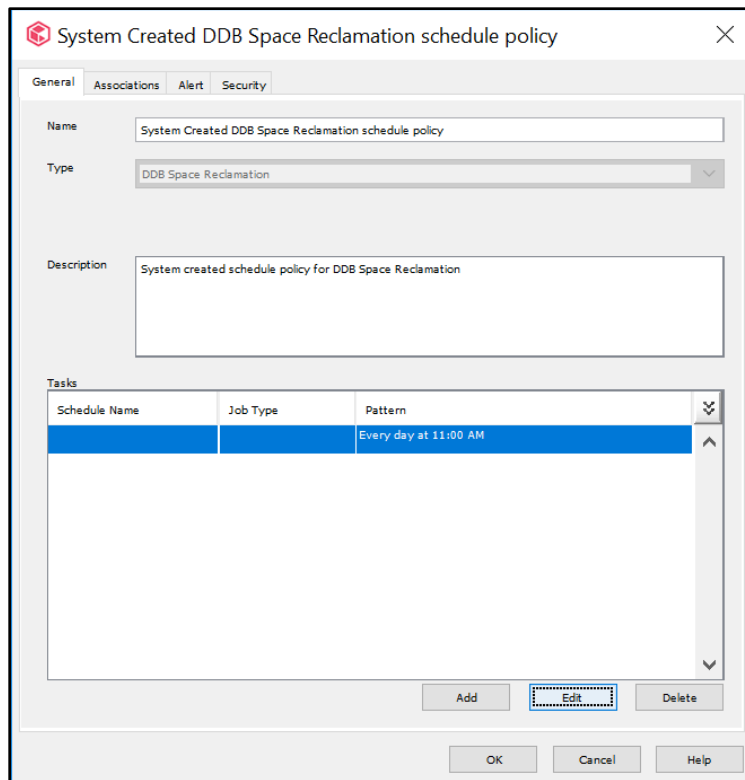


Figure 58. Editing the space reclamation schedule

3. You can change the frequency options on the **Schedule Pattern** tab (Figure 59). We recommend a monthly schedule if you are using SafeMode snapshots, and at least weekly if you are not.

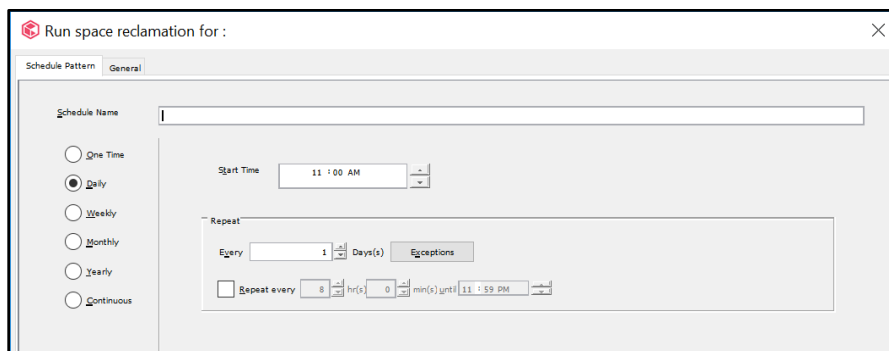


Figure 59. Reclamation schedule pattern

4. Click the OK button to set the new schedule, then click the OK button again to commit the change.

NOTE: The policy will let you change the reclamation level on the General tab; however, the setting will revert to default. You must create a separate schedule to run reclamation more or less aggressively.





To create a schedule with a different reclamation level:

1. In the CommCell Console, expand **Policies**. Right-click **Schedule Policies**, then select **New Schedule Policy**.
2. On the **General** tab (Figure 60), enter a display name for the policy in the **Name** field. Select **DDB Space Reclamation** from the **Type** dropdown. You can also add a description in the **Description** field. Click the **Add** button to create a schedule in the policy.

The image shows a 'New Schedule Policy' dialog box with a 'General' tab selected. The 'Name' field is 'Space reclamation level 2'. The 'Type' dropdown is 'DDB Space Reclamation'. The 'Description' field contains 'Twice weekly Level 2 aggressiveness'.

Figure 60. Schedule policy general settings

3. In the dialog that opens, on the **Schedule Pattern** tab (Figure 61), set the scheduling options you want. We recommend a monthly schedule if you are using SafeMode snapshots, and at least weekly if you are not.

The image shows a 'Schedule Pattern' dialog box. The 'Schedule Name' is 'Monday, Thursday 11:00 AM'. The frequency is set to 'Weekly'. The start time is '11:00 AM'. The days selected are Monday and Thursday. The repeat interval is 'Every 01 Week(s)'. There are also checkboxes for 'On these days' (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday) and a 'Repeat every' section with fields for hours, minutes, and seconds.

Figure 61. Schedule pattern

4. On the **General** tab (Figure 62), adjust the **Reclamation Level** slider to the desired aggressiveness. Moving the slider to the left will make reclamation less aggressive, where more blocks must expire from a file before it will be rewritten, lowering I/O but using more space. Moving it to the right will cause files to be rewritten with fewer expired blocks, using less space but causing more I/O. Click the **OK** button to add the schedule to the policy.



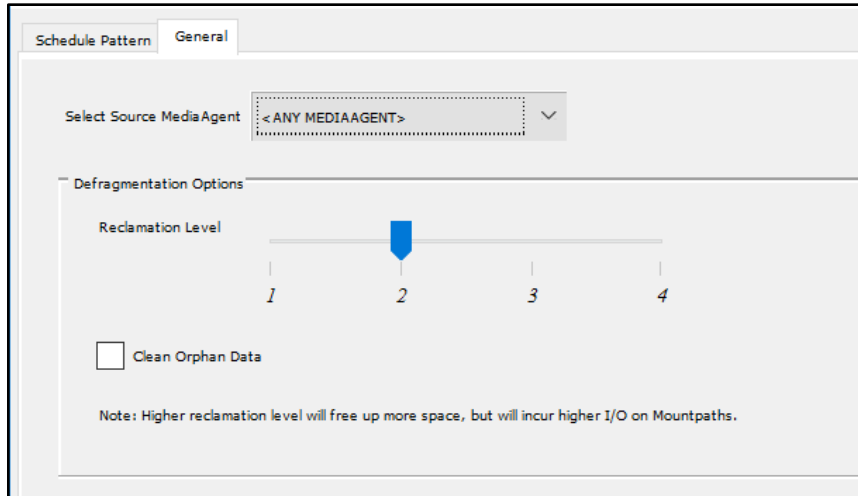


Figure 62. Adjusting reclamation level

5. On the Associations tab (Figure 63), enable the checkbox next to **Deduplication Engines** to apply the schedule to all DDBs. You can also select specific DDBs to apply only to storage pools for FlashBlade//S. Click the **OK** button to finish creating the schedule policy.

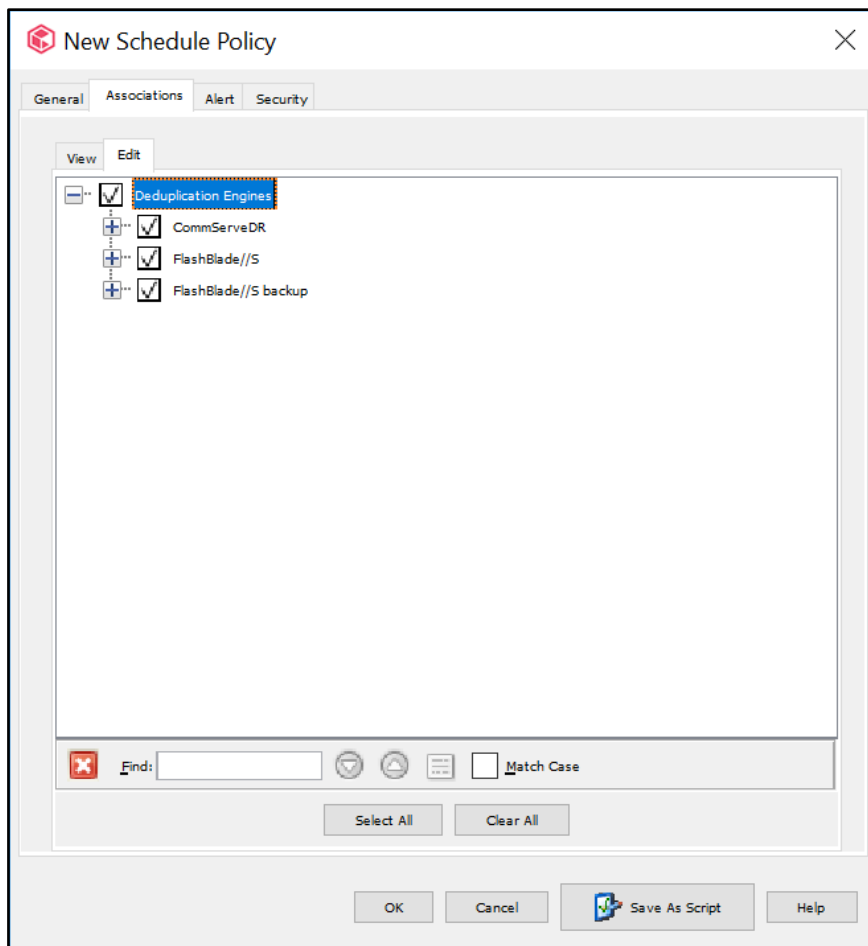


Figure 63. Setting schedule policy associations





Get more detail on the [space reclamation process](#).

Configure spill and fill across mount paths: Spill and fill is a load-balancing algorithm similar to round-robin. Commvault prioritizes distributing data streams across mount paths over reusing mount paths that already have streams. This ensures data is distributed evenly across the FlashBlade file systems.

Spill and fill is usually the default, but you should confirm this and correct it if it is not set. To configure spill and fill:

1. In the CommCell Console, navigate to Libraries. Right-click the FlashBlade library and select Properties.
2. As shown in Figure 64, click the Mount Paths tab. Set the Mount Path Usage option to Spill and fill mount paths if not already selected. Enable the Prefer mount paths with more free space option to further balance storage consumption.

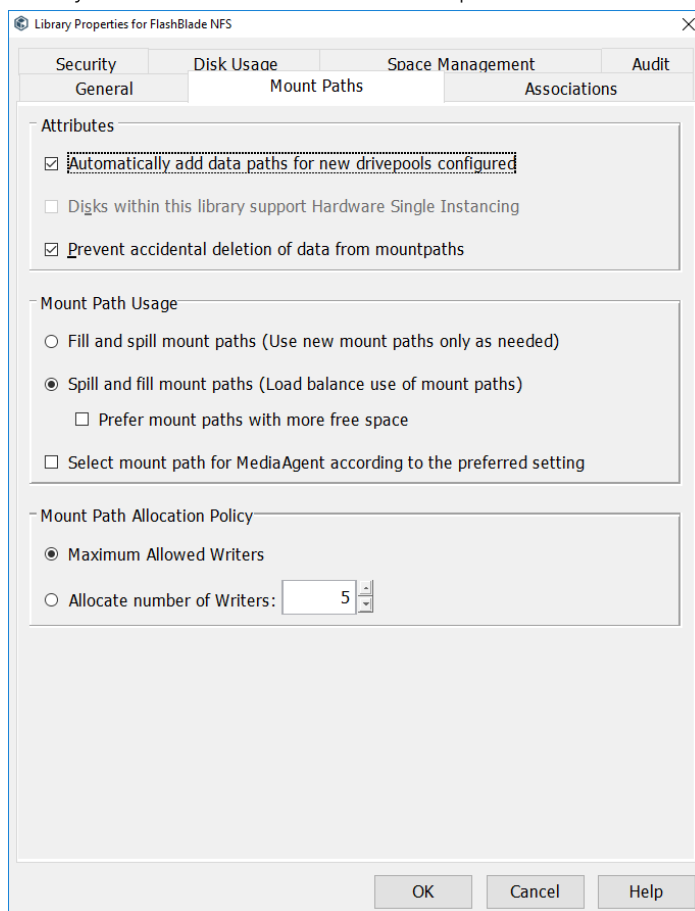


Figure 64. Mount path options

CommServe DR Backup

CommServe DR backups also benefit from SafeMode snapshot protection when using an SMB share on FlashBlade//S. Remember that SafeMode snapshots will apply to all file systems on the FlashBlade//S. For instructions on enabling SMB support on FlashBlade and connecting to Active Directory, please see FlashBlade//S documentation.





CommServe DR backup best practices are:

- Use a dedicated service account
- Configure SMB export policy
- Restrict access using ACLs
- Consolidate DR backups for multiple CommServe systems
- Schedule DR backups close to the SafeMode snapshot schedule
- Use FlashBlade replication to provide offsite availability
- Upload backups to Commvault cloud
- Upload backups to the cloud library

Detailed Best Practices

Use a Dedicated Service Account: Using a dedicated service account ensures that the DR backups can't be accessed and therefore altered or deleted by any other account. The service account should not be used for any other purpose or allowed local login to any systems. If you have a password vault product, use it to store the password.

IMPORTANT: If you are using SMB in AD RFC2307 mode, the service account must have values set for the uidNumber and gidNumber attributes in Active Directory for authentication and ACLs to work properly.

Configure SMB Export Policy: As shown in Figure 65, set up the DR file system with only SMB enabled. On Purity//FB release 3.0 and earlier, set the Native SMB ACLs option, which was removed in release 3.1.

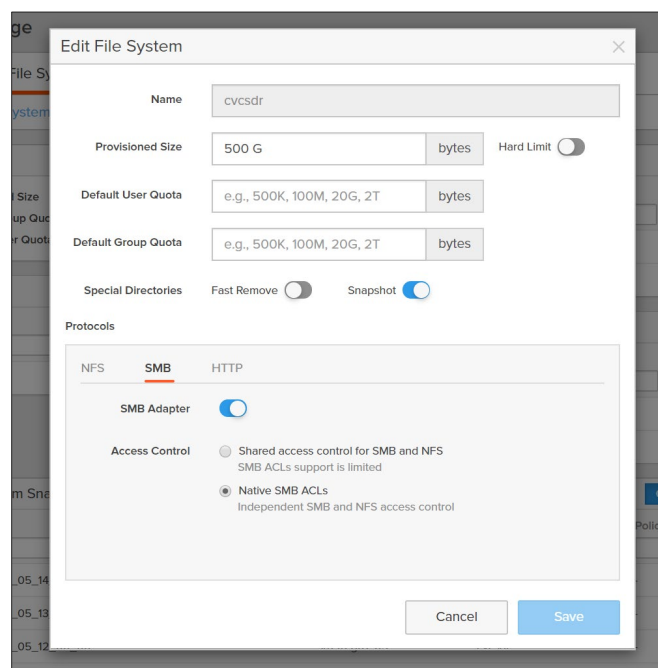


Figure 65. SMB export policy

Restrict access using ACLs: The DR backup share ACL needs to restrict access so that only the service account can write to and manage the file system. Grant full control access for the service account to all files and directories in the share.





The CommServe DR recovery uses a restore within Microsoft SQL Server that runs as the SQL Server service account. Using the Commvault recommended configuration, this process will access the SMB share as the CommServe computer account. For DR recovery to work, the standby CommServe computer account also needs access. The ACL should grant only read access. For DR recovery on the production CommServe, the production CommServe computer account will also need read access.

For easier permissioning, create a group in Active Directory and add all the CommServe computer accounts to the group. Grant the group read access to the DR backup SMB share. Figure 66 shows the full ACL.

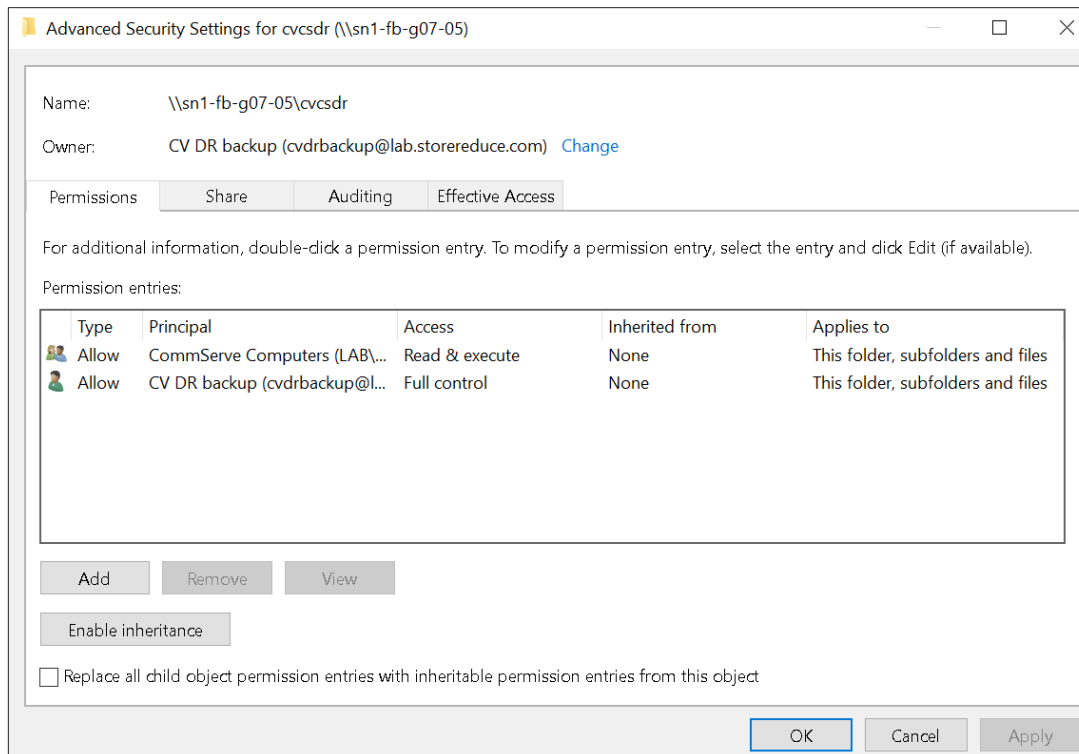


Figure 66. DR Backup SMB share ACL

Consolidate DR backups for multiple CommServe systems: Every CommServe in an environment needs to run DR backups, including standby systems. Every CommServe can benefit from SafeMode snapshots if you consolidate the DR backups onto a single file system. Create a separate directory per CommServe to avoid conflicts between CommServe systems. All DR backups can use the same service account.

The ACL on the DR backup share needs to grant read access to every CommServe computer account in Active Directory to ensure recoverability.

Schedule DR backups close to the SafeMode schedule: To minimize the period where CommServe DR backups are not protected by SafeMode snapshots, schedule the DR backups to occur just before the snapshot policy schedule, making sure to allow enough time for the backup to complete. For example, if the SafeMode snapshot schedule runs at 10:00 am, and the DR backup completes in one minute, schedule DR backups for 9:55 am.

Use FlashBlade//S replication to provide offsite availability: While not detailed as part of this document, native replication between FlashBlades coupled with SafeMode will provide an extra layer of defense for CommServe DR backups. Enabling





replication with SafeMode can have broader implications, which you should discuss with your Pure account team before implementing. Refer to FlashBlade//S documentation for more detail on enabling replication.

Upload backups to FlashBlade//S cloud library: DR backups can be uploaded automatically to a configured cloud library, with longer retention than the first stage network share backup. Enabling this option is an easy way to get a longer-term copy of DR backups on FlashBlade//S Object storage.

To enable cloud library upload using Commvault Command Center, navigate to the **Manage/System** view, then select **Maintenance**. Click the **DR backup (Daily)** tile to fetch the settings, then click the **Edit button** (gear icon) to open the properties. As shown in Figure 67, enable the **Upload backup metadata to cloud library** option, then select the FlashBlade storage pool in the **Cloud library** dropdown. Click the **Save** button to commit any changes.

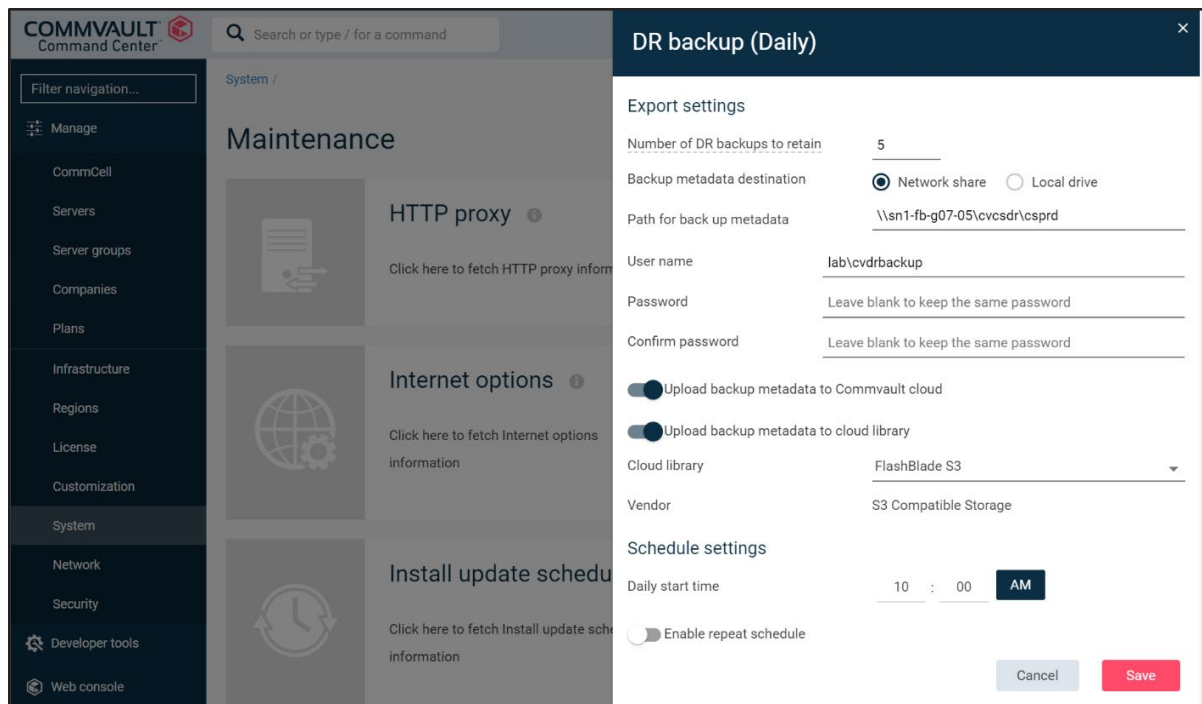


Figure 67. DR Backup configuration in Commvault Command Center

Upload backups to Commvault cloud: Commvault provides cloud storage for DR backups as part of a paid support agreement. This ensures an offsite copy is available in case of site loss or other situation that prevents using the local copies. This option should be enabled if allowed by your company policies.

Set appropriate DR backup retention: By default, Commvault will keep 5 daily DR backups on the FlashBlade SMB share. SafeMode Snapshots will extend that period based on the retention policy you define. For example, if the DR backup retention in Commvault is set to 5, and the SafeMode snapshot retention policy keeps 7 days, 12 days of DR backups will be available for recovery.





Common Failure Scenarios

This section describes potential causes for common implementation failures.

Cloud Target Creation Fails

During setup for the cloud storage pool, Commvault connects to the bucket and creates a folder. There are several reasons this may fail. If the GUI reports an error and no folder was created in the FlashBlade bucket, it may be due to one of these.

Cause	Error Messages	Explanation	Resolution
Unsuccessful TLS certificate validation with self-signed certificate	Could not connect Failed to check cloud server status, error = [[Cloud] The server failed to do the verification. Error = 44037]	By default, FlashBlade//S uses a self-signed certificate, and Commvault will fail the TLS handshake	Use a trusted certificate Disable TLS or certificate validation check for the MediaAgent On 11.26 and later, disable validation of self-signed certificates See Disable TLS, If Allowed and Configure a Single Object Bucket
Unsuccessful TLS certificate validation with CA-signed certificate	Could not connect Failed to check cloud server status, error = [[Cloud] The server failed to do the verification. Error = 44037]	A certificate signed by a trusted CA must include a Subject Alternate Name (SAN) that resolves in DNS to the FlashBlade//S data VIP. Commvault must be configured to use the same name as the SAN in the certificate.	Reissue and re-sign the certificate on FlashBlade//S, including the Subject Alternate Name (SAN) Ensure the MediaAgent(s) can use DNS to resolve the name in the SAN field resolves Ensure the bucket in Commvault is configured to use the same name as the SAN field
Incorrect keys provided	Access denied	The wrong access key ID or secret key was provided	Confirm Commvault is configured with the correct keys See Configure a Single Object Bucket
Incorrect bucket name	Bucket not found	The bucket name was entered incorrectly	Confirm and retype the bucket name
Incorrect object access policies applied on object user account	Bucket not found	The access rights in the object access policy do not grant the required access	Grant the required access rights to the object user account See Create Restrictive Object Access Policies

Table 7. Cloud target creation failure reasons

Backups or Restores Fail to Send Data

Backup or restore jobs may fail or change to Pending state without writing any data to FlashBlade. This may be due to one of the following issues.





Cause	Error Messages	Explanation	Resolution
Unsuccessful TLS certificate validation with self-signed certificate	Could not access mount path	Disabling certificate validation must be performed on each MediaAgent. Backups will fail for a shared mount path if a MediaAgent is selected that has validation enabled and the default self-signed certificate is still in use.	Use a trusted certificate On 11.26 and later, disable validation of self-signed certificates Disable TLS or certificate validation check for the MediaAgent See Disable TLS, If Allowed and Configure a Single Object Bucket
Unsuccessful TLS certificate validation with CA-signed certificate	Could not access mount path	Name resolution failed on the MediaAgent for the FlashBlade//S data VIP	Correct the name resolution issue
Expired CA-signed certificate	Could not access mount path	Expired certificates will cause TLS validation to fail and prevent connection	Renew the certificate
Network error	Could not access mount path	The writer MediaAgent or Storage Accelerator client is unable to reach the FlashBlade.	Check for firewall between writer and FlashBlade

Table 8. Failure reasons during data transmission

Throughput

With Commvault client-side deduplication enabled, many environments do not see network saturation on MediaAgents. At peak load, Commvault should report effective throughput higher than the available network bandwidth.

Restore throughput should reach approximately 80% of available bandwidth, based on the slowest link in the data path. If throughput is lower than expected, the cause may be one of the following issues.

Cause	Error Messages	Explanation	Resolution
Commvault DDBs are unable to process data fast enough	Effective throughput is not faster than raw throughput to FlashBlade Average query and insert (Q&I) times are in the hundreds or thousands of milliseconds	DDBs require storage capable of thousands of IOPS and are sensitive to latency. Slow media can negatively impact overall backup throughput. Partitioned DDBs can perform better than standalone DDBs.	Upgrade the DDB storage or add one or more partitions. See Ensure Performant Deduplication Databases
DDBs are not located on dedicated storage	Effective throughput is not faster than raw throughput to FlashBlade Average query and insert (Q&I) times are in the hundreds or thousands of milliseconds	DDBs share storage with index cache or other data and do not receive sufficient IOPS.	Move the DDB to dedicated SSD or NVMe storage.





MediaAgent cannot open sufficient connections to FlashBlade	TCP connection count is equal to thread pool maximum. This is more likely to occur with first-generation FlashBlade than with FlashBlade//S	Commvault thread pool is not large enough to maximize throughput.	<p>Increase the thread pool maximum size.</p> <p>This also suggests a CPU and memory resource limitation on the MediaAgent. Consider upgrading the MediaAgent hardware.</p> <p>See (Optional) Increase Cloud Thread Pool Size</p>
MediaAgent is dividing bandwidth	The MediaAgent cannot reach over 50% of its available bandwidth	Read and write traffic are using the same network interfaces and dividing available bandwidth.	<p>Separate front-end and backend network traffic on different physical adapters.</p> <p>See Segregate Client and Storage Networks</p>
MediaAgent is network limited	MediaAgents use all available bandwidth but cannot reach optimal FlashBlade throughput	MediaAgents do not have enough bandwidth for all FlashBlade traffic.	<p>Use Storage Accelerator to increase the overall available bandwidth.</p> <p>See Use Storage Accelerator for Additional Horizontal Scaling</p>
Insufficient MediaAgent CPU resources	MediaAgents CPU runs at 100% during backup or recovery	MediaAgents require enough CPU to process data at full throughput. Deduplication and compression processing can have a significant CPU impact and reduce the overall throughput. Undersized MediaAgents may be CPU limited, especially during restore operations.	<p>Increase CPU resources and/or reduce CPU load</p> <p>Enable source-side deduplication and compression or disable compression. Using Storage Accelerator can reduce the MediaAgent CPU load.</p>

Table 9. Reasons for lower throughput than expected





Terms and Concepts

Below you will find useful terms and concepts.

Terms	Explanation
Agent (also iDataAgent)	Commvault client software that interfaces to a specific data type for backup and recovery. Represented in Commvault GUIs as a configuration object.
Amazon S3	Industry-standard object-based protocol for writing data over HTTP. Generally considered slower than file-based protocols.
Client Data Reader	A component of the Commvault agent processes that reads data from the client's storage for backup. By default, agents use 1-2 readers per subclient (data set) and a single reader per local volume.
Client-side Deduplication	A Commvault feature to reduce network traffic and improve backup performance. The backup agent communicates with the MediaAgent to determine which data is already known and discards duplicates.
Cloud Storage Accelerator	A Commvault feature that lets clients write data directly to object storage, bypassing the MediaAgent.
CommCell Console	Commvault's Java-based console. More capable but less simple than Command Center. Procedures in this guide that use Command Center can also be performed in the CommCell Console.
Commvault Command Center	Commvault's HTML console. Simple web interface with some advanced configuration capability.
Data Vaulting	A practice to ensure immutability for deduplicated data. Commvault periodically seals its deduplicated data store to ensure the completeness and integrity of data. Object SafeMode prevents changes to the vault.
FlashBlade SafeMode™ Snapshots	A FlashBlade feature that mitigates against ransomware attacks on file system data. When SafeMode snapshots are enabled, the FlashBlade periodically creates immutable snapshots of all file systems and prevents manual eradication of file systems and snapshots.
HotAdd Transport Mode	VMware mechanism for backing up virtual machines. Virtual disks are attached to a VM running backup software and read within the VM as local disks.
Live VM Recovery	A Commvault feature for instant VM recovery. A VM is powered on from a virtual datastore Commvault presents, then migrated to a permanent datastore.
MediaAgent	Commvault data mover and distributed index store. MediaAgents are the primary communicators with back-end storage such as FlashBlade//S.
Mount Path	A mount path is a unique back-end storage target, accessible by one or more MediaAgents. Multiple mount paths can be grouped within a storage pool, or library.





<u>Network File System (NFS)</u>	Industry Industry-standard file-based protocol for writing unstructured data to network storage. Common on Linux, not widely used on Windows.
Object SafeMode	A FlashBlade feature that mitigates against ransomware attacks on object data. When Object SafeMode is enabled, objects cannot be deleted or overwritten for a minimum retention period.
SAN Transport Mode	VMware mechanism for backing up virtual machines. Datastore disks on SAN storage are attached to a physical server running backup software, and virtual disks are read from the datastore.
Software Compression	Commvault can compress data being backed up to reduce network and/or storage consumption. Files are analyzed to look for repeated patterns that can be consolidated. Compression can be performed at the backup client or the MediaAgent. Client-side compression reduces network utilization but increases CPU load on the client. Compression at the MediaAgent reduces client load but increases network consumption and CPU load on the MediaAgent.
Subclient	A Commvault configuration object within the agent configuration that defines a data set and the options and policies used to protect it. An agent can have multiple subclients.
Target-side Deduplication	Similar to client-side deduplication, but performed on the MediaAgent. The agent sends the full data without removing duplicates, resulting in more network utilization but lower CPU and memory load on the client. CPU and memory utilization increases on the MediaAgent.
VM Live Mount	A Commvault feature for DevOps and similar use cases. Similar to Live VM Recovery, a VM is powered on from a virtual datastore Commvault presents; however, it is not migrated, and its configuration and lifecycle are managed by policy.

Table 10. Terms





About the Author



Roy Child is a Senior Solution Architect with Pure Storage, responsible for defining data recovery solutions and reference architectures for primary workloads such as Oracle, SQL, and VMware. Roy has worked in and with the data protection industry for more than 20 years, from end-user to IT architect with multiple backup and recovery products, followed by product management with Commvault. Roy joined Pure Storage in April 2019.

The Pure Storage products and programs described in this documentation are distributed under a license agreement restricting the use, copying, distribution, and decompilation/reverse engineering of the products. No part of this documentation may be reproduced in any form by any means without prior written authorization from Pure Storage, Inc. and its licensors, if any. Pure Storage may make improvements and/or changes in the Pure Storage products and/or the programs described in this documentation at any time without notice.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. PURE STORAGE SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Pure Storage, Inc.
650 Castro Street, #400
Mountain View, CA 94041

[purestorage.com](https://www.purestorage.com)

800.379.PURE

