

TECHNICAL WHITE PAPER

Configuring Commvault with FlashBlade: Best Practices

Enhance Commvault and Pure Storage FlashBlade performance with the simplest setup.

Contents

How to Use This Guide	4
General Best Practices	4
Choose the Appropriate Protocol for Your Environment	4
Enable SafeMode Ransomware Mitigation on FlashBlade	5
Use Commvault Client-side Compression by Default	6
Use Commvault Client-side Deduplication	6
Ensure Performant Deduplication Databases	6
Run Multiple Client Data Readers	7
Set Maximum Writers on the Library and MediaAgents	7
Share Mount Paths Across MediaAgents	7
Match VMware Disk Format and Transport Mode	7
Store CommServe DR Backups on a FlashBlade File System Using SMB	7
Best Practices for MediaAgents	8
Operating System	8
MediaAgent Hardware Specifications	8
MediaAgent Count	8
Stream Count	9
Network Bandwidth	10
Segregate Client and Storage Networks	10
Deduplication Database Storage	10
Index Cache Storage	10
Object Storage	11
When to Use Object Storage	11
How Commvault Uses Object Storage	11
Planning for Object Lock	12
Deduplication and Immutability	12
Layered Vaulting	13
Estimating Capacity Requirements	14
Estimating Deduplication Database Requirements	14
Detailed Best Practices	14
Use Commvault 11.32 or Later	14
Use Purity//FB 4.3.4 or Later	14
Apply a Signed Array Certificate	15
Use Commvault Workflow to Provision FlashBlade Object Storage	15
Create Restrictive Object Access Policies	20
Store Object Access Keys in Commvault Saved Credentials	21



Create Only Required Object Buckets	23
Share the Bucket across MediaAgents	28
Use Storage Accelerator for Additional Horizontal Scaling	29
Use Separate Credentials for Storage Accelerator	31
Reduce Deduplication Block Size	32
Rotate FlashBlade Access Keys Regularly	33
Optional: Configure Cloud Upload Thread Pool Size	34
Optional: Increase Pipeline Buffer Count	35
Optional: Increase Look-ahead Size	35
Optional: Limit Auxiliary Copy Streams with FlashBlade//E	35
Enabling Object Lock for Ransomware Mitigation	35
NFS	40
Use of NFS	40
Required MediaAgent Operating System	40
Enabling SafeMode for Ransomware Mitigation	41
Mount Path Configuration	41
CommServe DR Backup	53
Use a Dedicated Service Account	53
Configure SMB Export Policy	53
Restrict access Using ACLs	54
Consolidate DR Backups for Multiple CommServe Systems	54
Schedule DR Backups Close to the SafeMode Schedule	54
Use FlashBlade Replication to Provide Off-site Availability	54
Upload Backups to FlashBlade Cloud Library	55
Upload Backups to Metallic Cloud	55
Set Appropriate DR Backup Retention	55
Common Failure Scenarios	56
Cloud Target Creation Fails	56
Backups or Restores Fail to Send Data	57
Throughput	58
Terms and Concepts	59



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® into Commvault environments. The guide is arranged in sections, each focused on specific elements. Each section contains a summary that covers the design and tuning concepts at a high level and a details section, which explains how to implement them.

This guide is applicable for all FlashBlade systems, including first-generation FlashBlade, FlashBlade//S™, and FlashBlade//E™. Any configuration differences between models will be called out in that section.

NOTE: Some links to Commvault and Pure Storage documentation require a customer account to access them.

General Best Practices

The best write performance on FlashBlade storage is achieved by spreading the load across as many blades as possible. Configuring Commvault for maximum writers and distributed streams will yield the best outcome with a simple setup. Commvault supports FlashBlade systems as a back-end storage target using both object (Amazon S3) and file (NFS) protocols. While S3 is simpler to configure and scales more easily, NFS is more familiar to many administrators. This guide contains best practices for both approaches, including specific recommendations for all FlashBlade configurations.

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

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)	• Simple configuration • Excellent load distribution across blades • Object lock support • Commvault Storage Accelerator support	• Less familiar to administrators
File (NFS)	• More familiar protocol to admins • FlashBlade SafeMode™ snapshot support	• More complex to configure • Less efficient load distribution • Not supported with Windows Media Agents

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 lock and SafeMode Retention Lock
Ransomware Mitigation with Smallest Storage Footprint ¹	NFS with SafeMode snapshots

TABLE 2 Protocols for use cases

Enable SafeMode Ransomware Mitigation on FlashBlade

FlashBlade 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 Storage support, and only with at least two authorized company designees. 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.

The Object Lock and SafeMode Retention Lock features in Pure Storage FlashBlade provide immediate protection against accidental or malicious deletion of object data. Object Lock applies an expiration time to the object, preventing anyone from deleting or overwriting the object before then. SafeMode Retention Lock prevents anyone from reducing object lock policies and permanently deleting an object bucket that contains backup data. Commvault leverages the object lock APIs to intelligently manage object lifecycles.

For more information on Object Lock, refer to [FlashBlade 4.1: Introducing Object Lock](#) and the [Layered Vaulting](#) section of this guide.

SafeMode snapshots create immutable, indelible, point-in-time views of file systems on the FlashBlade. 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, 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 storage,

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



but less than object lock in most cases. Pure Storage recommends 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 Lock with SafeMode Retention Lock	- No loss or outage for backup data - Immediate immutability - Simple to implement - Operationally transparent - Immediate availability for recovery	- Additional configuration in Commvault - Higher storage than SafeMode snapshots for most environments
SafeMode Snapshots	- Invisible to Commvault - Protects all snapshots and file systems - Complete point-in-time view of data	- Potential for data loss between snapshots - Additional recovery required after the event

TABLE 3 Pros and cons of Object lock and SafeMode solutions

Use Commvault Client-side Compression by Default

While FlashBlade 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 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.

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 is capable of. Configuring multiple DDB partitions and segregating different data types between storage policies will give you better bandwidth utilization with FlashBlade. A high-throughput, very low latency SSD or NVMe device optimized for random access will maximize performance for all FlashBlade models.

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 Commvault 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. 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 will all come from the same MediaAgent and won't distribute across blades as effectively.

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, a mismatch can limit the throughput.

Store CommServe DR Backups on a FlashBlade File System Using SMB

Storing CommServe DR backups on a FlashBlade file system lets you centralize protection for CommServe databases and removes dependencies on specific servers. Optional SafeMode snapshots and FlashBlade 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](#).



Best Practices for MediaAgents

MediaAgents play a crucial role in achieving optimal backup and recovery performance, serving as the primary communicators with FlashBlade. This section includes details on configuration 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.

Commvault may require specific operating systems depending on the storage protocol. With object storage, you can use any operating system, and you can share buckets between different MediaAgent OSes. With file storage, Commvault requires Unix or Linux to use NFS and Windows to use SMB.

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 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 back end of up to 250TiB per node. In this model, you scale to more MediaAgents by adding new grids. You can have larger grids with multiple petabytes when you use FlashBlade//E as a target for selective secondary copies.

MediaAgent Count

Calculate the MediaAgent requirements primarily on front-end data size, following Commvault's building block guidance. Size FlashBlade capacity according to the building blocks. When planning for immutability, see the topics [Object Lock](#) or file-based [SafeMode snapshots](#).



Stream Count

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

To increase the maximum streams for a MediaAgent:

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

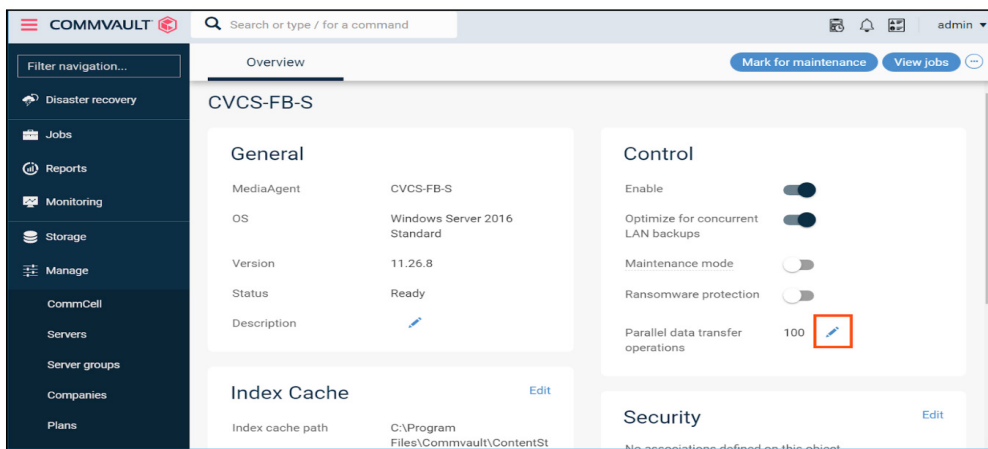


FIGURE 1 MediaAgent properties

5. As shown in Figure 2, 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.

FIGURE 2 Setting parallel data transfer operations

6. Click **Save** to commit the change.



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 requires 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

Pure Storage recommends using separate networks on MediaAgents with separate network adapters for communication to clients and FlashBlade. While not necessary, segregation serves two purposes:

- It allows you to restrict which systems can access the FlashBlade VIP.
- 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 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 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 write performance. NFS requires multiple mount paths and a more complex Commvault configuration to achieve a comparable result.

To achieve the best object storage results, follow the steps outlined below.

When to Use Object Storage

Object storage is the preferred model for all Commvault operations. The same FlashBlade 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 Lock

When planning for a object lock, you should consider the following key points to ensure a smooth implementation:

- Deduplication and immutability
- Layered vaulting
- Estimating capacity requirements
- Estimating deduplication database requirements

Deduplication and Immutability

Global deduplication is a well-established technology for reducing storage consumption. It tracks data patterns that have been written, and when those patterns show up again, the existing data gets referenced rather than written again. While this is indisputably valuable, it has major implications for object-level immutability. It's important to understand that impact to make the right design decisions. Deduplication and immutability are directly at odds; virtually all of your backups will depend on data that was written during earlier backups, and that complicates when it's safe to unlock objects.

As an example, assume you need to keep your backups for 14 days. After two weeks, your first backups have met their service level agreement (SLA) and are ready to delete. However, in that time, you've added more backups with dependencies on the first week, so the first week actually needs to stay immutable until those backups expire. Each week adds more dependencies on the previous weeks, and none of that data can be exposed without risking the more recent backups. The nature of deduplication means that to add immutability you have to lock all your data forever—which defeats the very purpose of deduplication. Figure 3 illustrates these relationships and dependencies.

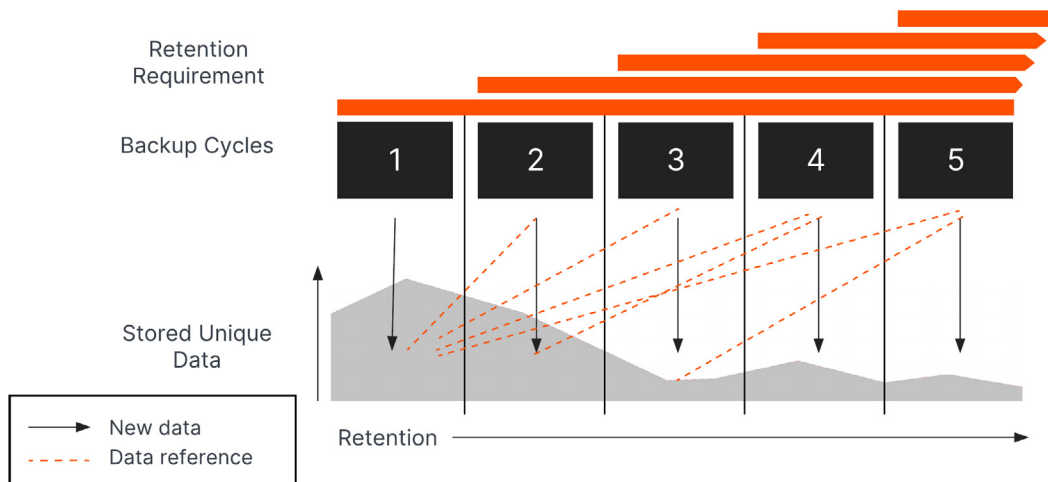


FIGURE 3 Deduplication dependencies

There are two ways to solve this problem. The first is to turn off deduplication. This entirely avoids the dependencies between backups. But it also means you cannot get any of the capacity savings of deduplication, so you'll spend more on storage.



Layered Vaulting

The better solution is to use a layered data vaulting concept. Commvault periodically “closes” the vault against any new data and starts a new one, putting a finite timeline on the dependencies. Within each vault, Commvault leverages its dynamic locking technology and FlashBlade Object Lock to align the object locks for the entire vault to expire at the same time.

For example, consider a vault with a 14-day retention requirement. The last backups, taken day 14, will need to exist until day 28 to meet the SLA. The earliest data in the vault must also be protected until day 28 to ensure that all jobs in the vault have their dependencies covered. To accomplish this, Commvault sets the object locks for the first day’s backups to expire after 28 days. On the second day, locks are set to expire after 27 days. This continues until the 14th day, when locks are set to expire after 14 days—also on day 28. Commvault then closes the vault on the 15th day and starts the cycle over. This approach gives granular control over immutability and backup scheduling while allowing the entire vault to be deleted at the same time, which frees up capacity as quickly as possible after meeting immutability and retention SLAs.

Over time, you will have two vaults at any given time: the active vault, with new data being written to it, and the most recently sealed vault. The previous sealed vault expires when the new open vault is being created. Having two vaults ensures the immutability of SLA is met; there will be dependencies going back to the oldest data in the sealed vault, and both vaults are fully protected. The immutability Figure 4 illustrates the vaulting timeline and the immutability window, which covers the backups that must be immutable to meet your SLA.

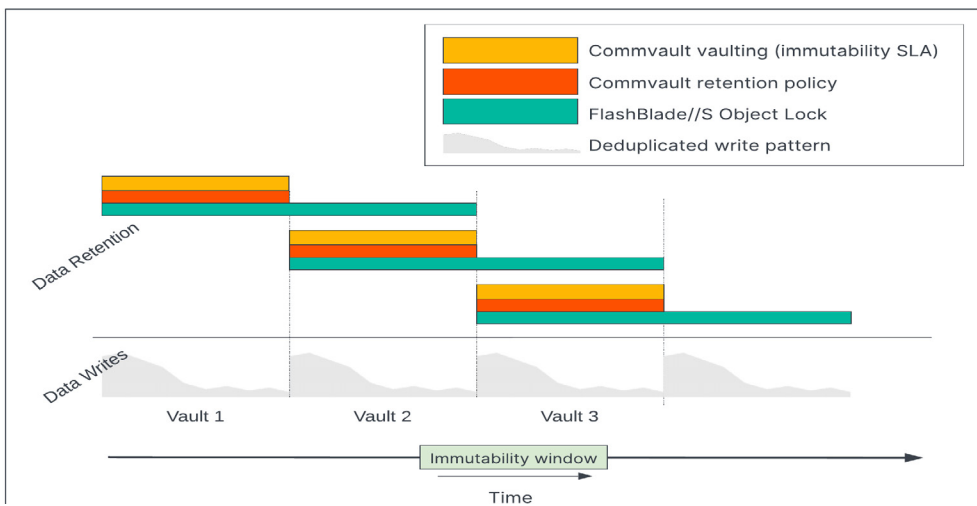


FIGURE 4 Layered data vaulting timeline

The layered vaulting system protects your data for the entire SLA period. Data is protected immediately as it is written and does not require any periodic point-in-time snapshots. You can get your production systems online sooner after an attack since your backup data is immediately available to start recovery without first having to roll anything back to an earlier recovery point. With hourly recovery costs ranging up to millions of dollars, this faster, simpler recovery can have a huge impact when it matters most.



Layered vaulting does require additional storage since each deduplication baseline will age after the next one is created, but it requires less storage and reduces downtime costs more than solutions based on snapshots and no deduplication.

The Freeze Objects feature of FlashBlade protects the vaulted backup data from direct attacks. FlashBlade SafeMode retention lock further prevents an attacker from deleting an object bucket, even if they were to gain administrative access.

Dynamic locking in Commvault aligns the object locks in the vault so they all expire at the same time. For example, to provide 14 days of protection, the first data into the vault must be locked for 28 days. Data written the second day is locked for 27 days, the third day is locked for 26 days, and so on until the 14th day, which is locked for 14 days. All object locks therefore align to day 28, and the vault can be deleted on day 29. This approach minimizes the extra storage consumption.

Estimating Capacity Requirements

In a Commvault environment, implementing object lock requires the capacity to store the data vaults until they expire. For most environments, you require between 2.2 and 2.5 times as much FlashBlade storage.

Be aware that this is a simplistic estimate. There are several factors that affect consumption, such as change rate, data reduction ratios, and whether you lock all or only part of your data, so the exact amount of storage you will need can vary widely. 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 two DDBs to ensure you don't run into issues. If you use partitioned DDBs, each partition will need enough storage available.

Detailed Best Practices

Consider the following key areas to ensure a comprehensive and effective implementation.

Use Commvault 11.32 or Later

Commvault 11.32 is the latest long-term support (LTS) release. It includes performance and management enhancements related to object storage and object lock, as well as improvements in integration with FlashBlade.

Use Purity//FB 4.3.4 or Later

While previous releases are compatible with Commvault, Purity//FB 4.3.4 includes a number of enhancements around security and performance.



Apply a Signed Array Certificate

To use Transport Layer Security (TLS) securely with FlashBlade object storage, you must apply a certificate to the FlashBlade that is signed by a certification authority (CA) that Commvault software trusts. The certificate must include the subject alternative name (SAN) field, and the service host name you configure in the Commvault storage pool must exactly match an entry in the SAN field and be resolvable by the MA. Each FlashBlade requires its own array certificate.

To add the SAN field, you must generate a new certificate signing request (CSR) and private key and have the CSR signed by the trusted CA. You must export the signed certificate and private key separately, in privacy enhanced mail (PEM) format, Base64 encoded. You may encrypt the private key with a passphrase. Refer to the documentation for your chosen tool for instructions on how to create the CSR, private key, and exported files.

To apply the certificate to the FlashBlade:

1. From the Settings page navigation sidebar, click Certificates under Security.
2. Click the More Options button in the Array Certificates panel. The Import Array Certificate pop-up window appears.
3. Complete or modify the following fields:
 - a. Certificate: Click Choose File and select the signed certificate.
 - b. Private Key: Click Choose File and select the private key.
 - c. Intermediate Certificate (Optional): Click Choose File and select the intermediate certificate.
 - d. Key Passphrase (Optional): If the private key is encrypted with a passphrase, enter the passphrase.
4. Click Import.

Once the certificate has been imported, Commvault's validation checks will succeed, allowing you to use TLS to encrypt communication with the FlashBlade.

NOTE: If your FlashBlade certificate is signed by a private CA that isn't tied to a trusted public root CA, Commvault certificate validation will fail. You must [add your CA to Commvault's trusted certificate list](#) on each MediaAgent or Storage Accelerator client for TLS to work. On any Linux systems and on Windows systems running agents older than release 2023E, you must check and update the list after each feature release update.

Use Commvault Workflow to Provision FlashBlade Object Storage

Pure Storage and Commvault developed an automated workflow to simplify provisioning object storage on FlashBlade. The workflow orchestrates the provisioning process from within Commvault's interfaces and applies the best practices from this guide. [Download and install](#) the [Provision Pure FlashBlade Object Storage](#) workflow from the Commvault store. You can install the workflow using either Command Center or the CommCell Console.

The workflow will create the following items on the FlashBlade and within Commvault:

- An object account on FlashBlade.
- Two object access policies on FlashBlade, with best practice permissions, one with and one without object delete access.
- Two object users attached to the object access policies.
- Object access keys for each object user.
- Saved credentials in Commvault to store the keys.
- One or more object buckets, with or without best practice object lock settings. Pure Storage recommends creating at least one bucket without object lock.



Before you execute the workflow, you must [create a saved credential](#) in Commvault to store an API token to access the FlashBlade [management REST API](#). Create the credential using the storage array account type.

To execute the workflow:

1. In the Commvault Command Center, navigate to the Workflows page. Locate the Provision Pure FlashBlade Object Storage workflow and click its title.
2. Complete the initial workflow form (Figure 5) as follows:
 - a. In the dropdown, select the saved credential containing the API token.
 - b. In the field, enter the management name or IP address of the FlashBlade. If you have applied a TLS certificate, use the name matching the certificate common name.

FIGURE 5 Initial workflow form

3. In the object account form (Figure 6), select the account to store the object users and buckets. If you want to create a new account, select New account and enter the name in the New account name field.

FIGURE 6 Object account form



4. On the IPs or subnets form (Figure 7), select the option you want to use to restrict bucket access in the FlashBlade privileged object access policy. You can choose to restrict MediaAgents, IP subnets, or no restrictions. Selecting MediaAgents will use specific IP addresses in the privileged access policy. Selecting an IP subnet will allow access for any IP address on the IP subnets you define. Selecting None will skip the access restrictions and allow any address to use the privileged keys—and is therefore not recommended.

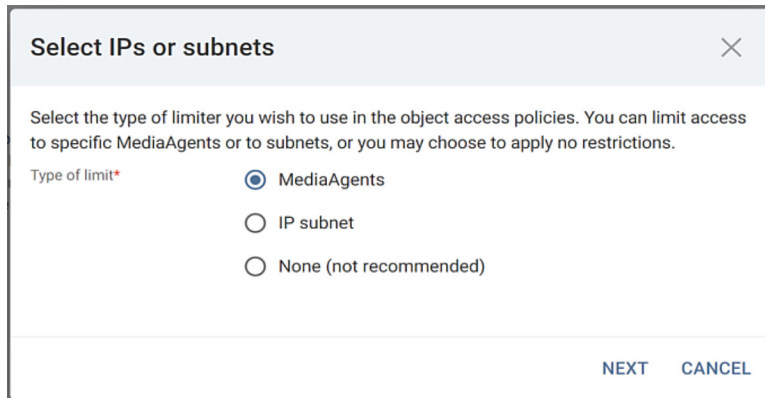


FIGURE 7 IPs or subnets form

- a. If you select MediaAgents, on the Select MediaAgents form (Figure 8), check the boxes for the MediaAgents that need access. Their IP addresses will be added to the policy.

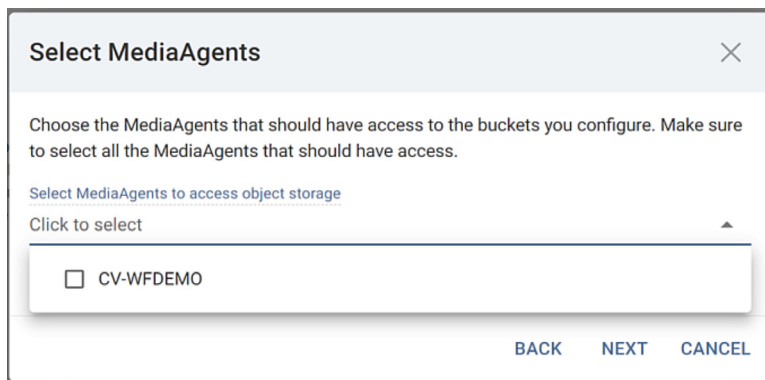
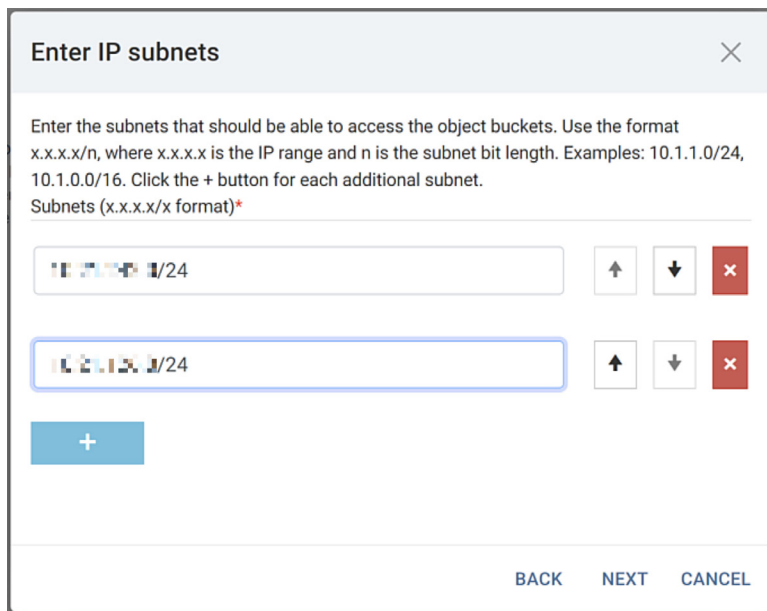


FIGURE 8 MediaAgent selection form

- b. If you select the IP subnet, on the IP subnets form (Figure 9), click the + button to add a field for each subnet. Enter the subnets in x.x.x.x/n format, where x.x.x.x is the IP range and n is the subnet mask bit length.



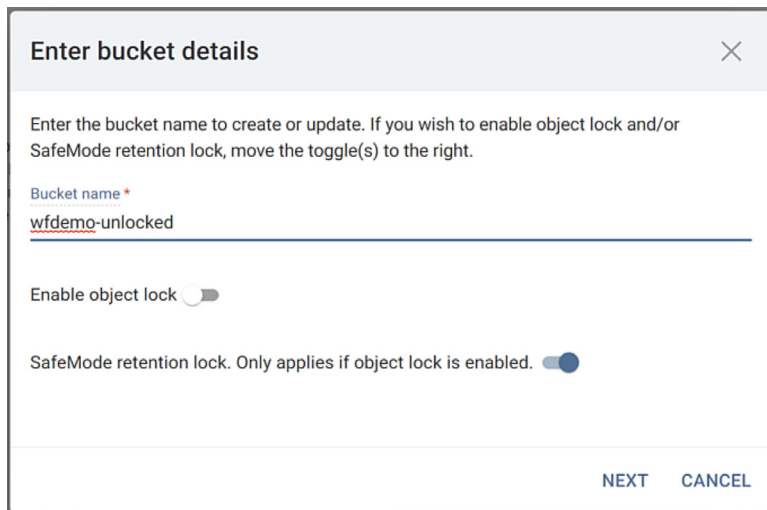


The form is titled "Enter IP subnets" and includes a close button (X) in the top right corner. It contains instructional text: "Enter the subnets that should be able to access the object buckets. Use the format x.x.x.x/n, where x.x.x.x is the IP range and n is the subnet bit length. Examples: 10.1.1.0/24, 10.1.0.0/16. Click the + button for each additional subnet." Below this is a label "Subnets (x.x.x.x/x format)*". There are two input fields, each containing a placeholder IP range "10.1.1.0/24". To the right of each input field are three buttons: an up arrow, a down arrow, and a red button with a white 'X'. A blue button with a white '+' is located below the first input field. At the bottom of the form are three buttons: "BACK", "NEXT", and "CANCEL".

FIGURE 9 IP subnets form

- On the bucket details form (Figure 10), enter a name for the bucket you want to create. To enable object lock on the bucket (optional), move the Enable object lock slider to the right. The form has SafeMode retention lock selected by default, but you can disable it by moving the SafeMode slider to the left; however, SafeMode retention lock will only be enabled if you also enable object lock.

If you choose to enable object lock, versioning will be disabled, and the freeze objects setting will be enabled.



The form is titled "Enter bucket details" and includes a close button (X) in the top right corner. It contains instructional text: "Enter the bucket name to create or update. If you wish to enable object lock and/or SafeMode retention lock, move the toggle(s) to the right." Below this is a label "Bucket name*". The input field contains the text "wfdemo-unlocked". Below the input field are two toggle switches. The first is labeled "Enable object lock" and is currently turned off. The second is labeled "SafeMode retention lock. Only applies if object lock is enabled." and is currently turned on. At the bottom of the form are two buttons: "NEXT" and "CANCEL".

FIGURE 10 Bucket details form

- The result form (Figure 11) will display success or failure. If you wish to create another bucket, move the slider to the right, click Next, then repeat step 5. To stop creating buckets, move the slider to the left and click Next.



Success

Do you wish to create or modify another bucket?

Create another bucket ☒

BACK

NEXT

CANCEL

FIGURE 11 Result form

7. A summary form (Figure 12) will show the objects that were created on the FlashBlade and in Commvault.

Summary

Provisioning succeeded. Objects created or configured:

Object account

wfdemo

Object access policies

wfdemo/mediaagent

wfdemo/storage-accelerator

Object users

wfdemo/mediaagent

wfdemo/storage-accelerator

Object buckets

wfdemo-unlocked

wfdemo-locked

Saved credentials

-wfdemo-Privileged

-wfdemo-Unprivileged

CANCEL

OK

FIGURE 12 Summary form

You are now ready to create storage pools and enable object lock in Commvault.

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. Following the principles of least required privilege, you should create a policy that is only applicable to the Commvault object account and grants only the access rights Commvault needs. If you use Storage Accelerator on Commvault release 11.32 or later, you should create a separate access policy and user to further restrict the bucket access rights for Commvault clients. The provisioning workflow will create policies as described in this section.

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

For more information, please refer to the [FlashBlade documentation](#). Then, create an object access policy associated with the Commvault object account on FlashBlade. Grant the access rights as shown in Later, when creating the object user, select this policy to restrict its access.

Without Object Lock	With Object Lock
s3:DeleteObject	s3:DeleteObject
s3:DeleteObjectVersion	s3:DeleteObjectVersion
s3:GetBucketVersioning	s3:GetBucketVersioning
s3:GetObject	s3:GetObject
s3:GetObjectAcl	s3:GetObjectAcl
s3:GetObjectTagging	s3:GetObjectRetention
s3:GetObjectVersion	s3:GetObjectTagging
s3:GetObjectVersionTagging	s3:GetObjectVersion
s3:ListBucket	s3:GetObjectVersionTagging
s3:PutObject	s3:ListBucket
	s3:PutObject
	s3:PutObjectRetention

TABLE 4 Required object access rights for Commvault

NOTE: These policies do not allow an object user to list buckets on the FlashBlade, which the CommCell Console interface uses for bucket detection. If you plan to use the CommCell Console interface, you can include the s3:ListAllMyBuckets access right in your policy to avoid setup issues.

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 5 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

*

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 13 Object access rule with IP address restriction

Store Object Access Keys in Commvault Saved Credentials

Commvault requires you to keep object storage access keys in its credentials vault. This allows you to easily reuse keys when you have storage pools that use multiple buckets in the same object account. It also simplifies updates when you rotate your object access keys. Create a saved credential in Commvault Cloud for each FlashBlade object user Commvault will use.

NOTE: The provisioning workflow will create saved credentials for both privileged and unprivileged accounts.



To create a saved credential for a FlashBlade object user:

1. In the FlashBlade UI, navigate to the Object Storage view. Click the user for which you want to create an access key. Click the context menu button for the **Access Keys** tile, then select **Create access key** from the menu. A dialog will appear with the details of the new key (Figure 7).

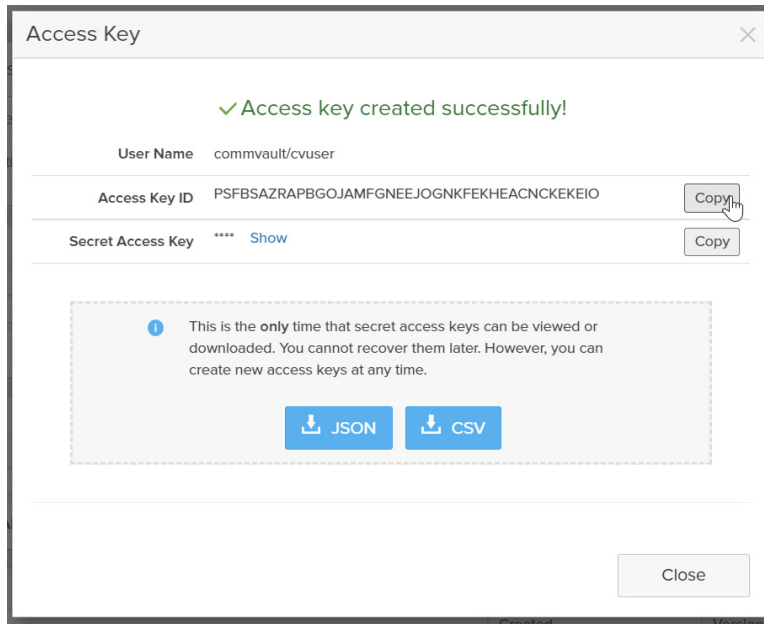


FIGURE 14 FlashBlade access key pair creation

2. Leave this page open and, in a separate browser window or tab, navigate to Manage > Security in Commvault Command Center. Click the Manage credentials tile (Credential vault in release 2024 and newer) to open the **Credential manager** view. Click the Add link to open the Add credential form.
3. In the **Add credential** form (Figure 9), enter a name for the credential. Switching between the FlashBlade and Commvault interfaces, click **Copy** in the FlashBlade interface (Figure 10) to copy the access key ID and secret key and paste them into the appropriate Command Center fields.
4. You may enter descriptive text in the **Description** field.
5. Click **Save** 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.



FIGURE 15 Add credential form

Create Only Required Object Buckets

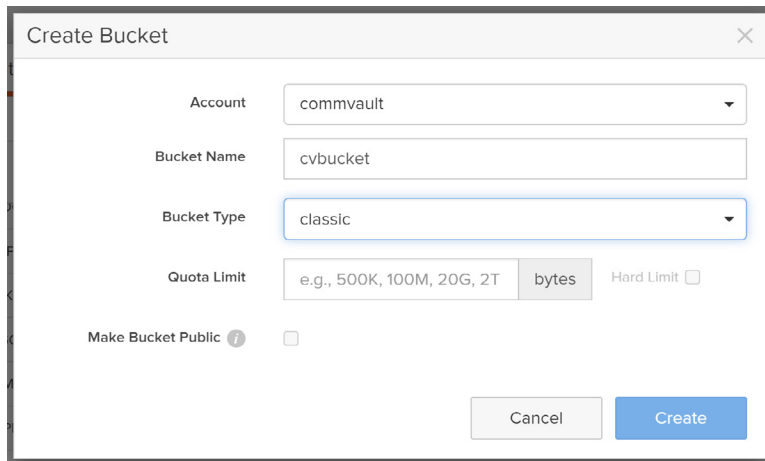
You must create at least one FlashBlade object bucket to hold Commvault data. When creating buckets, consider the following:

- You should create only as many storage pools as you need. Different retention policies do not necessarily need separate pools.
- Create only one bucket per pool. If object lock is not enabled, you can configure multiple pools to use the same bucket.
- Object lock is not enabled on new buckets. You must enable object lock after creating a bucket.
- When using object lock, you must create a separate storage pool for each retention period, and each pool should have its own bucket.

NOTE: The provisioning workflow will create buckets that match these best practices.

1. Access the Object Store view in the FlashBlade UI. In the Buckets tile, click the Add (+) button.
2. Complete the Create Bucket dialog box (Figure 16).
 - a. From the Account dropdown, select the object account you created.
 - b. In the Bucket Name field, enter a name for the bucket.
 - c. In the Bucket Type field, select classic.
 - d. Optionally, in the Quota Limit field, enter a maximum size for the bucket. With quotas enabled, the FlashBlade will generate alerts when consumption nears the quota size. Quotas may be useful if multiple applications share the FlashBlade and you want to limit how much data Commvault can store.
 - e. You should not enable the Hard Limit checkbox, as reaching the quota would cause Commvault operations to fail.
 - f. Do not enable the Make Bucket Public checkbox.
 - g. Click Create to finish creating the bucket.





The image shows a 'Create Bucket' dialog box with the following fields and options:

- Account:** A dropdown menu with 'commvault' selected.
- Bucket Name:** A text input field containing 'cvbucket'.
- Bucket Type:** A dropdown menu with 'classic' selected.
- Quota Limit:** A text input field with 'e.g., 500K, 100M, 20G, 2T' and a unit dropdown set to 'bytes'. There is also a 'Hard Limit' checkbox which is unchecked.
- Make Bucket Public:** A checkbox which is unchecked, with a help icon to its left.
- Buttons:** 'Cancel' and 'Create' buttons at the bottom right.

FIGURE 16 Create a bucket dialog box.

Enabling Object Lock on Buckets

If you plan to use the Commvault Hardware WORM feature to make backup data immutable, you must enable object lock on the bucket after you create it. You can only enable object lock on empty buckets, so you must enable it before writing any data to the bucket.

NOTE: The provisioning workflow will create buckets that match these best practices.

To enable object lock on an empty bucket:

1. Access the Object Store view in the FlashBlade UI. In the Buckets tile, locate the bucket where you want object lock enabled. Click the context menu button for the bucket, then click Edit from the popup menu.
2. In the Edit Bucket dialog box, click Retention Settings to expand the retention options.
3. Configure the following options in the dialog box (Figure 17):
 - Leave the Versioning dropdown set to none.
 - Enable the Object Lock checkbox.

IMPORTANT: Once enabled, object lock cannot be disabled for the bucket.

- Enable the Freeze Locked Objects checkbox.

IMPORTANT: Disabling Freeze Locked Objects requires assistance from Pure Storage Support.

- Leave the Default Retention field unset.
- Leave the Default Retention Mode dropdown set to none.
- Leave the Eradication Mode dropdown set to permission-based.

NOTE: Setting eradication mode to retention-based prevents both manual and automatic eradication of the bucket. Once configured to be retention-based, the bucket cannot be set back to permission-based, even by Pure Storage Support. While this conforms to SEC 17a-4 WORM compliance, it is outside the scope of this guide. Please consult with your Pure Storage SE and FSA resources if you require this capability.

- (Optional) From the Retention Lock dropdown, select ratcheted (locked) to enable SafeMode Retention Lock, preventing deletion of the bucket.

NOTE: Disabling retention lock requires assistance from Pure Storage Support.



- Click **Save** to commit the changes.

FIGURE 17 Bucket options for object lock with Commvault

Create Commvault Storage Pools

To enable Commvault to write backup data to FlashBlade, you need to create storage pools that connect Commvault deduplication stores with FlashBlade object buckets. Your server plans or storage policies will associate with the storage pools.

You should minimize the number of FlashBlade object buckets you create. You can use a single bucket to host multiple standard storage pools. If you choose to enable object lock, you will need to create additional buckets. For each retention period you define, Commvault will require a separate storage pool with Hardware WORM enabled. You should create a separate bucket for each of these pools.

IMPORTANT: Never add a bucket that uses object lock as the first storage pool in a new CommCell.

To create a cloud storage pool using FlashBlade in release 11.32 and later:

1. In Commvault Command Center, navigate to Storage>Cloud and click the **Add link** to open the Add cloud storage wizard.
2. Complete the first section of the form (Figure 13) as follows:



- a. From the **Type** dropdown, select **Pure Storage FlashBlade**.
(On 11.30 CommCells, select S3 Compatible Storage.)
- b. In the **Name** field, enter a display name for the storage pool.
- c. From the **MediaAgent** dropdown, select a media agent to be the owner of the storage pool.
- d. In the **Service host** field, enter the name of the FlashBlade data VIP.

NOTE: You may reuse the same data VIP and DNS name for any locked and unlocked FlashBlade storage pools.
You can disable TLS by including "http://" at the beginning of the service host. This is not recommended in a production environment without an isolated storage network.

- e. From the Credentials dropdown, select the privileged saved credential you created.
- f. In the **Bucket** field, enter the name of the bucket.

FIGURE 18 Add cloud storage form in Commvault Command Center

3. In the Deduplication DB location section (Figure 19), click the **Add** link to add a DDB partition.
4. In the Add Deduplication DB location popup window, select the MediaAgent that will host the DDB partition, then **enter or browse** to the path where the DDB should be stored. For optimal backup performance, this path should be located on high-speed, low-latency storage, such as a PCIe NVMe drive, an SSD, or a Pure Storage FlashArray//X.
5. Click **Add** to update the DDB location list.
6. Repeat Steps 3 to 5 for any additional partitions you want to add and expand, up to a maximum of four.
7. Click **Save** to create the storage pool.



Bucket *

wfdemo-unlocked

Use deduplication

☒

Deduplication DB location

Add

MediaAgent ↑

Deduplication DB location

CV-WFDEMO

E:\DDB-unlocked

EQUIVALENT API

CANCEL

SAVE

FIGURE 19 Deduplication DB location section

Repeat the above procedure for each FlashBlade bucket.

After the procedure is completed, you will have two or more storage pools, one for each bucket (Figure 20).

Cloud

Add

All

+ Add filter

Name ↑	Status	Capacity	Free space	Actions
FlashBlade object lock	Online	N/A	N/A	
FlashBlade unlocked	Online	N/A	N/A	

FIGURE 20 Commvault storage pools for locked and unlocked FlashBlade buckets

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 21).

Cloud / Cloud storage type

Add cloud storage

Configure cloud

Name *

FlashBlade/S

Storage

Type

S3 Compatible Storage

MediaAgent *

sn1-720-g08-07

Service host *

http://10.21.237.25

Credentials *

FlashBlade/S S3

Bucket *

cvbucket

FIGURE 21 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 22). Pure Storage, recommends creating partitions on at least two MAs to provide performance and resilience. You can create up to four partitions per storage pool.

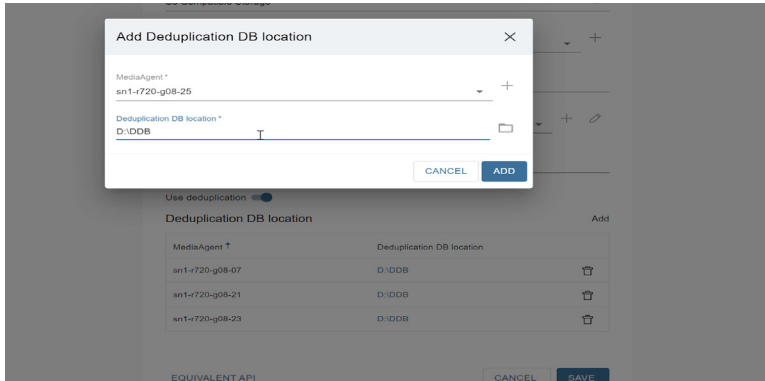


FIGURE 22 Adding a DDB

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

Share the Bucket across MediaAgents

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

1. In the Commvault Command Center, navigate to the cloud storage pool.
2. In the **Buckets** tile, select **Actions** for the bucket, and then click **Add MediaAgent**, as shown in Figure 23.

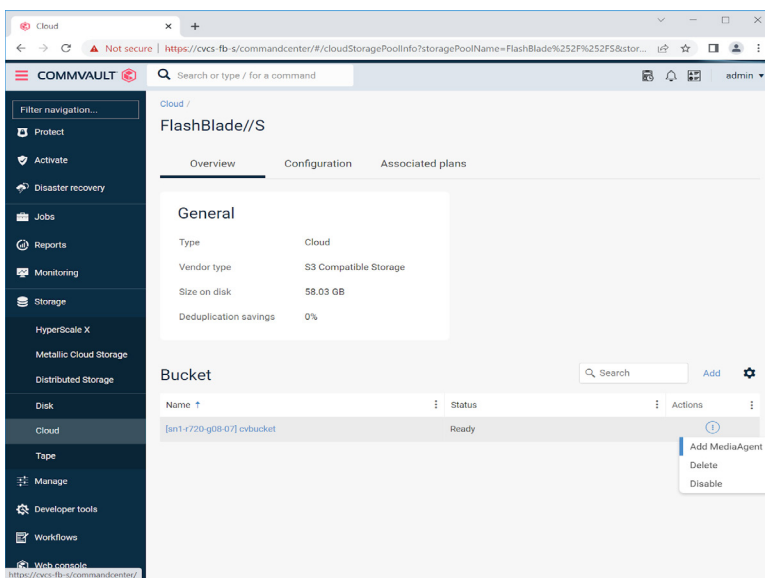


FIGURE 23 Cloud buckets



3. As shown in Figure 24, select the MediaAgents that should access the mount path, and then click **Save**. MediaAgents is added with read/write access.

FIGURE 24 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 25 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.

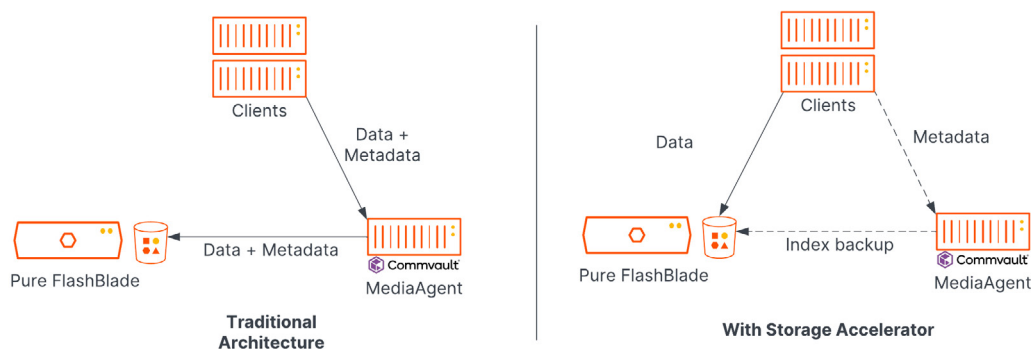


FIGURE 25 Traditional and Storage Accelerator architectures



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.

Beginning with Commvault release 11.26, the CommServe will automatically deploy the Storage Accelerator package. Clients backing up to FlashBlade will receive the package on their first backup job. You can disable automatic installation.

To disable the automatic installation of Storage Accelerator, you need to use the CommCell Console interface.

1. On the **Home** ribbon menu, select **Control Panel** and select 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.
3. Set a value of 1 enables automatic installation, while a value of 0 disables automatic installation.
4. Click **OK** to commit the change.

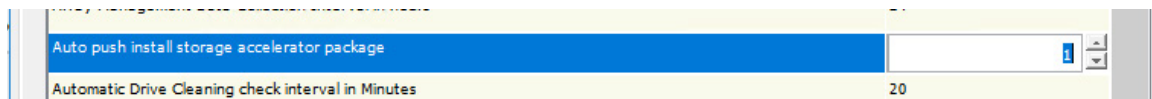


FIGURE 26 Enabling or disabling automatic push of Storage Accelerator

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.

To manually deploy Storage Accelerator, follow these steps:

1. In **Command Center**, navigate to **Manage > Servers** and select **All** from the **Type** dropdown to display all the clients.
2. Click **Actions(...)** next to the client where you wish to install Storage Accelerator, and then select **Add software** from the popup menu.
3. As shown in Figure 27, select the Storage Accelerator package from the dropdown menu. You can type in the search box to filter the list.
4. Click **Install** to start the installation.

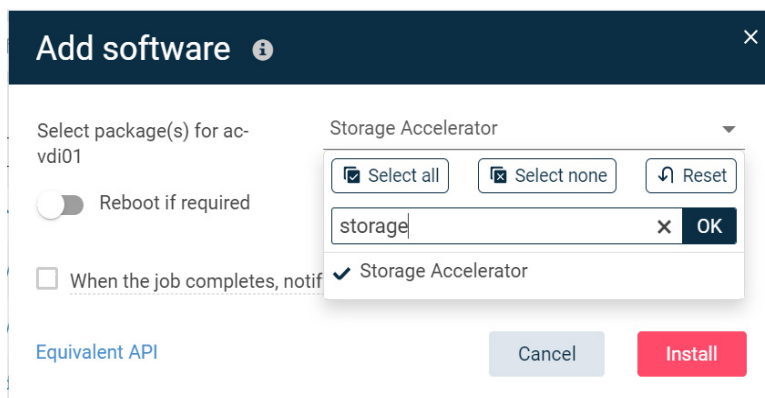


FIGURE 27 Installing the Storage Accelerator package

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



Use Separate Credentials for Storage Accelerator

You have the capability to assign a secondary credential to a bucket in a cloud storage pool, specifically for Storage Accelerator. This enables the use of a restricted user who can read and create objects but cannot delete them.

To assign a Storage Accelerator credential, follow these steps:

- 1. On the FlashBlade, 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 5. The policy should not restrict the source IP addresses. This would limit the effectiveness of Storage Accelerator and can cause backup and recovery failures.

Without Object Lock	With Object Lock
s3:GetBucketVersioning	s3:GetBucketVersioning
s3:GetObject	s3:GetObject
s3:GetObjectAcl	s3:GetObjectAcl
s3:GetObjectTagging	s3:GetObjectRetention
s3:GetObjectVersion	s3:GetObjectTagging
s3:GetObjectVersionTagging	s3:GetObjectVersion
s3:ListBucket	s3:GetObjectVersionTagging
s3:PutObject	s3:ListBucket
	s3:PutObject
	s3:PutObjectRetention

TABLE 5 Required object access rights for Commvault

- 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 cloud storage pool. In the Storage Accelerator credential field, click **Create (+)** to open the New credential form.
- 4. Give the credential a clear name. Copy the access key ID and secret key from the FlashBlade interface and paste them in the appropriate fields in Command Center (Figure 28).
- 5. Enter a description (optional) and click **Save** to finish creating the credential.

Add credential

Credential name

FlashBlade//S Storage Accelerator

Access key ID

PSFBSAZRKDAGMKCLPKMEGKMBOAJFKFHCFD

Secret access key

.....

Description

Restricted access for Storage Accelerator

Security

Owner

admin

User/Group

Enter users or user groups

Equivalent API

CancelSave

FIGURE 28 Creating a saved credential for Storage Accelerator

- The new credential appears in the Storage accelerator credentials field. Click the **check mark** button to apply the credential to the cloud storage pool.

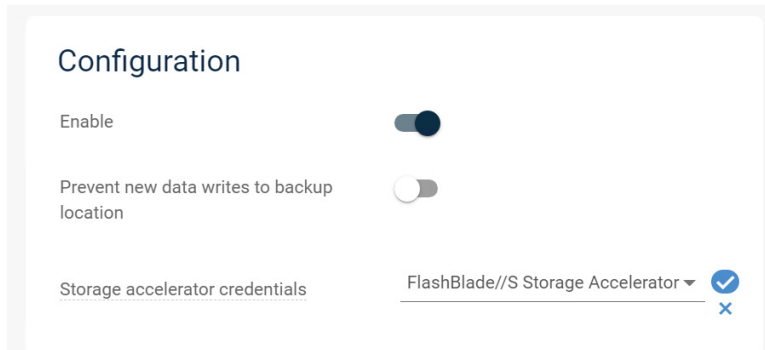


FIGURE 29 Applying Storage Accelerator credentials

Reduce Deduplication Block Size

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.

NOTE: Beginning with release 11.36, Commvault automatically sets the deduplication block size to 128KB on new storage pools with the Pure Storage FlashBlade storage type.

To set deduplication block size for a storage pool, follow these steps:

- In the **CommCell Browser** pane, navigate to **Storage Pools**, and then right-click the desired pool.
- Select **Properties**, then the **Storage Pool** to open the **Properties** dialog.
- Click the **Advanced** tab, as shown in Figure 30. Set the **Block level Deduplication** factor dropdown to 128 and click **OK** to commit the change.

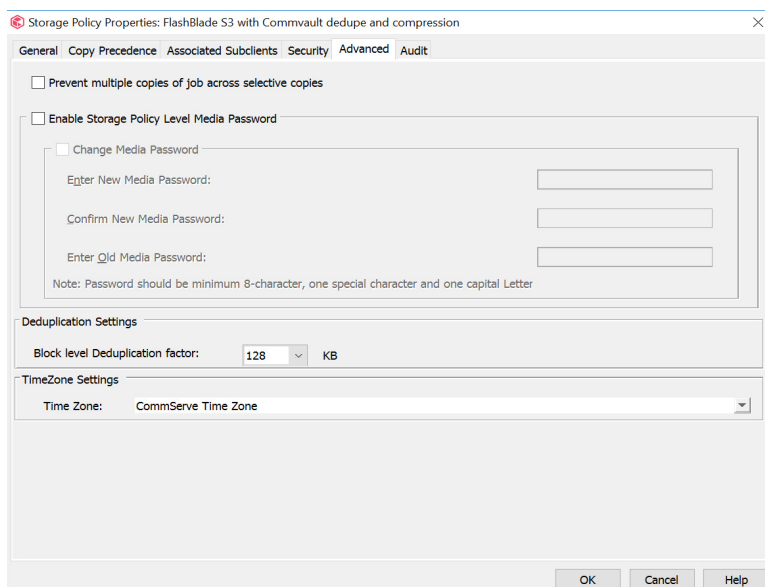


FIGURE 30 Storage policy deduplication factor



Rotate FlashBlade Access Keys Regularly

You should regularly rotate access keys for FlashBlade object storage—creating a new key and deleting the old one—to minimize the risk of key compromise. The credential manager in Commvault makes the rotation process simple. To rotate keys, first create a new key pair on the FlashBlade. Access the Object Store view in the FlashBlade UI, then navigate to the account that contains the Commvault bucket. Click the user that has the key you want to rotate. Click the context menu button for the **Access Keys** tile, then select **Create access key** from the menu. A dialog will appear with the details of the new key (Figure 31).

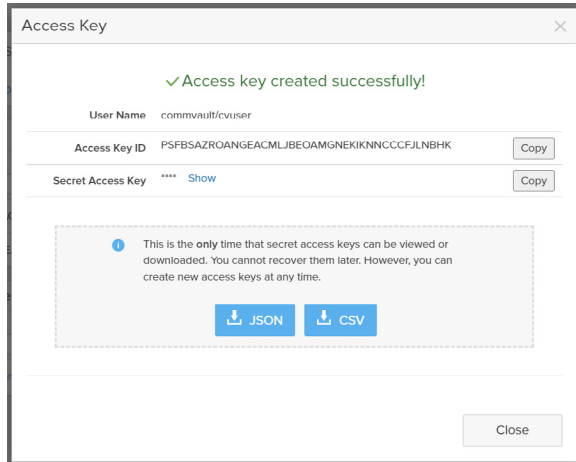


FIGURE 31 FlashBlade access key pair creation

Leave this page open and, in a separate browser window or tab, access the **Credential manager** view in Commvault Command Center. Click the stored credential for the FlashBlade object user to open the **Edit credential** form (Figure 32). Replace the **Access key ID** and **Secret access key** fields with the newly generated key values, then click the **Save** button. You may export the keys for future reference; however, copying the key values between consoles using the **Copy** buttons will prevent an attacker from compromising the stored file. Running the consoles side by side lets you copy the keys in seconds.

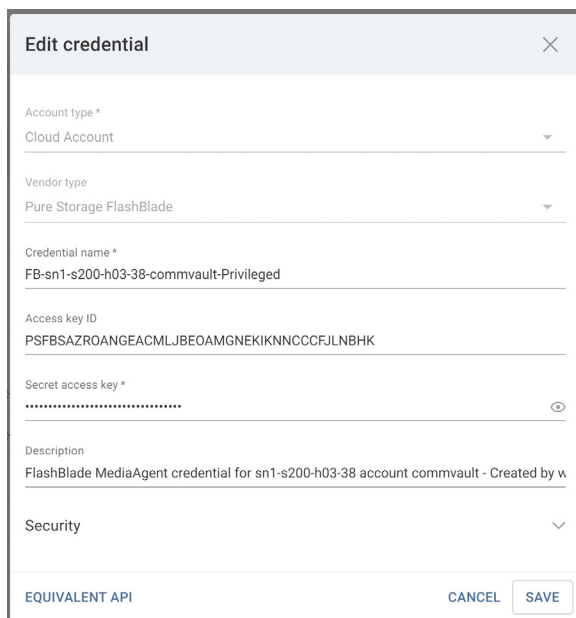


FIGURE 32 Updating access keys in Commvault Command Center



Once the key has been updated in Commvault, delete the old key from the FlashBlade to prevent anyone from using it. In the FlashBlade UI, Click the **Close** button to clear the dialog box. In the **Access Keys** tile, click the context menu for the old key, then select **Delete** (Figure 33). When prompted to confirm deletion, click the **Delete** button.

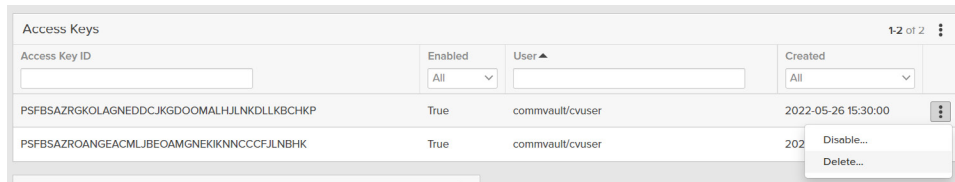


FIGURE 33 Deleting old access key

Optional: Configure Cloud Upload Thread Pool Size

By default, a MediaAgent or a client equipped with Storage Accelerator capability can establish up to 200 connections to object storage (50 connections if the agent was deployed prior to Commvault release 11.32). Backup jobs running on this MediaAgent or client at the same time will share these connections, and the number of active backup streams does not necessarily directly correspond to the number of threads in use.

You should set the maximum thread pool size based on the FlashBlade model.

- For first-generation FlashBlade, keep the default 200.
- For FlashBlade//S, set the limit to the number of chassis * 50. For example, a FlashBlade//S with a single chassis should have a limit of 50 threads, while a four-chassis array should have a limit of 200.
- For FlashBlade//E, set the limit to the number of EC chassis * 50, ignoring EX chassis.
- For MAs writing to multiple FlashBlade models, configure the limit for the model with the highest limit.

To change the thread pool size limit for a MediaAgent:

- In Command Center, navigate to the Manage>Infrastructure page. Click the MediaAgents tile.
- Click the name of the MediaAgent in the MediaAgents list to access its settings.
- In the **Settings** tile, locate the Configure number of concurrent upload threads for cloud storage setting and click its Edit (pencil) icon (Figure 34).
- Adjust the number, then click the Submit button to apply the change. You do not need to restart the MA, but the change will not affect any running jobs.

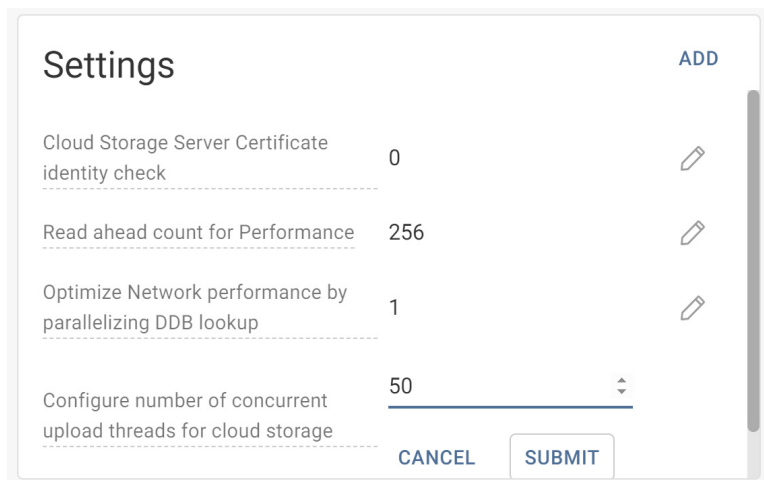
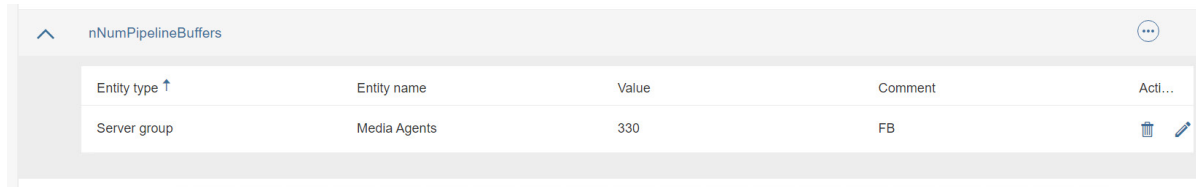


FIGURE 34 MediaAgent Settings tile



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.



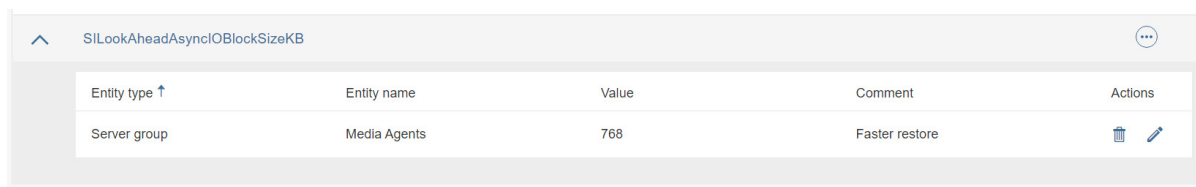
Entity type ↑	Entity name	Value	Comment	Acti...
Server group	Media Agents	330	FB	 

FIGURE 35 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. The impact may vary between deployments and data sets, so test the setting thoroughly in your own environment before deploying in production. Figure 36 shows the setting.



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

FIGURE 36 Increasing look-ahead block size

Optional: Limit Auxiliary Copy Streams with FlashBlade//E

When heavy read and write workloads run in parallel on FlashBlade//E, the read workload may experience a significant throughput decrease. This can slow down restore or auxiliary copy jobs that read from FlashBlade//E at critical times. To mitigate this, you can [limit the stream count](#) for the auxiliary copy jobs writing to FlashBlade//E. You must modify all automatic auxiliary copy schedules that write data to FlashBlade//E. If you have schedules that manage copies to multiple storage targets, create a separate schedule for FlashBlade//E targets and remove them from the existing schedules. Pure Storage recommends setting a limit of 20 streams per auxiliary copy job to FlashBlade//E.

Enabling Object Lock for Ransomware Mitigation

The following best practices apply only when using Object Lock for ransomware mitigation. Configuring object lock requires enabling the object lock feature on the FlashBlade bucket, then enabling Hardware WORM on the Commvault storage pool.

NOTE: Review the [Planning for Object Lock](#) section before you enable the feature to ensure your FlashBlade is sized correctly for immutability.

IMPORTANT: You can enable object lock on the FlashBlade bucket using the Commvault Workflow, or refer to the 'Enabling Object Lock on Buckets' section for instructions on manually configuring object lock.



Configure Hardware WORM on Storage Pool

You must [enable hardware WORM support](#) on the storage pool. This will perform several functions within Commvault. It will activate object lock commands, configure automatic sealing of the DDB for the storage pool, and apply an enforced retention policy. It will also enable and enforce compliance lock on all plan copies associated with the pool.

To enable hardware WORM on a storage pool:

1. In Commvault Command Center, navigate to the **Storage >Cloud** page and click the **Configuration** tab.
2. In the WORM tile (Figure 37), move the WORM storage lock toggle to the right.

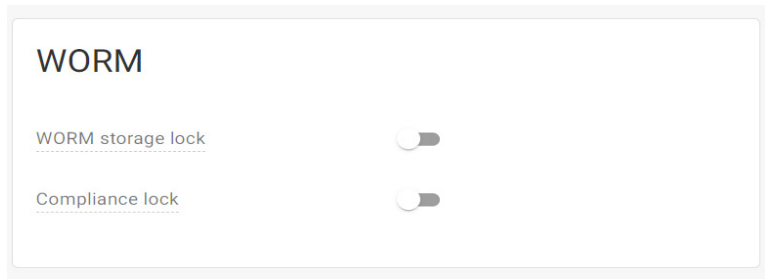


FIGURE 37 WORM tile

3. The Retention rules dialog box (Figure 38) appears. Set the Retention period to match the original retention that you set during plan creation, then click **OK**.

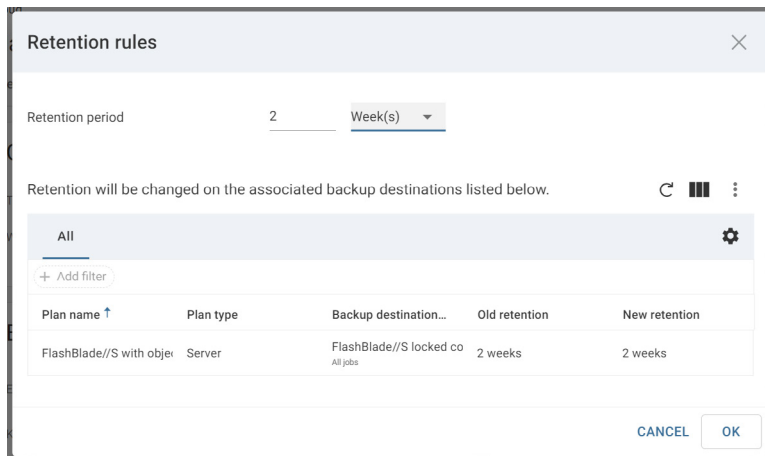


FIGURE 38 Retention rules dialog box

4. A confirmation dialog box (Figure 39) appears and do the following:
 - a. Enable both the checkboxes.
 - b. Type **Confirm** in the field.
 - c. Click **Confirm**.



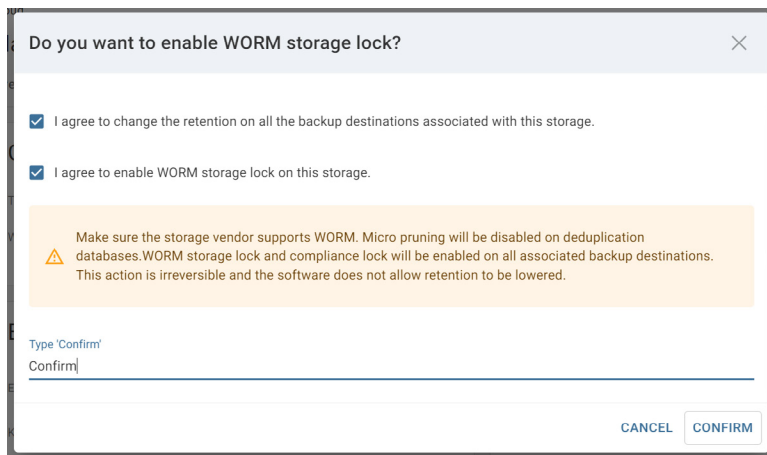


FIGURE 39 WORM confirmation dialog box

5. In the WORM tile (Figure 40), the WORM storage lock and Compliance lock toggles will move to the right, and both will be disabled for further changes. The retention period is also shown.

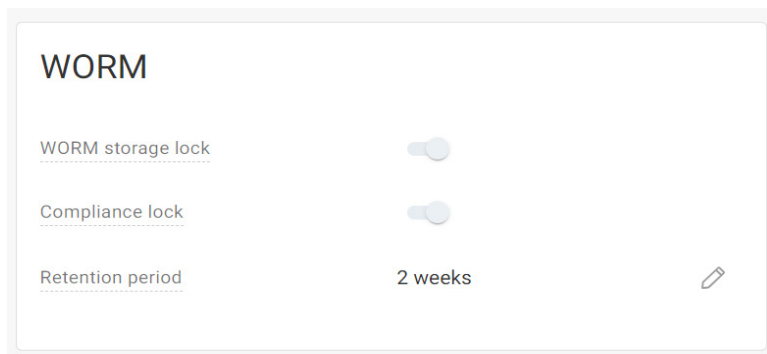


FIGURE 40 WORM tile reflecting changes

Object lock is now fully enabled, and all new backup data using the server backup plan will be protected with object lock on the FlashBlade, and the backup jobs will be protected within Commvault against deletion until they have reached the retention age. Any server backup plan copies you create later that use this storage pool will automatically inherit the hardware WORM settings.

Recommended: Modify DDB Backups

DDB backups follow a different retention scheme than normal backups, and you must avoid having these backups go to a locked storage pool to prevent pruning problems and excess capacity utilization. The backups should, however, still reside on FlashBlade for fast recovery and simplest protection. You can change the DDB backup settings using the CommCell Console.

NOTE: DDB backup settings for a MediaAgent are used by all the DDBs it hosts. If the MediaAgents hosting the FlashBlade DDBs already had other DDBs on them, the backups will already be configured to land on other storage.



To change the policy for DDB backups:

1. In the CommCell Browser pane, expand **Client Computers**, and then the **MediaAgent** you want to update.
2. Expand the **File System** node below the client, and then click the **defaultBackupSet** node.
3. In the right pane, right-click the **DDBBackup** subclient, and then click **Properties**.
4. In the Subclient Properties dialog box, click the **Storage Device** tab and select the unlocked FlashBlade server backup plan you created, and then click **OK**.

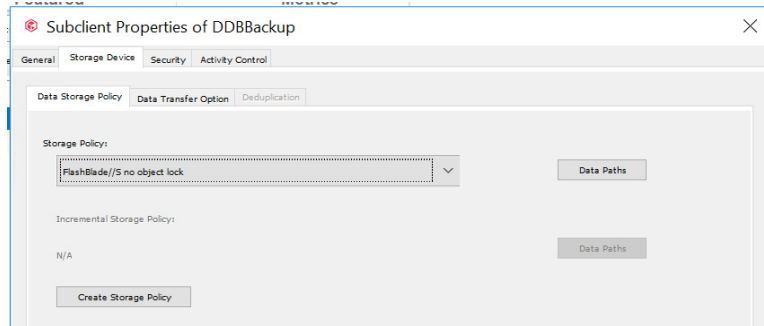


FIGURE 41 Setting DDBBackup storage policy

5. The Storage Policy column reflects the new setting as shown in Figure 42.

Subclient Name	Storage Policy
DDBBackup	FlashBlade//S no object lock
default	

FIGURE 42 Updated DDBBackup storage policy

Modify Index Backups

Similar to DDB backups, index backups follow their own schedule and retention settings, independent of any policies you create. You should associate these to an unlocked storage pool on the same FlashBlade system. You can change the index backup client settings using the CommCell Console.

NOTE: Commvault creates index backup clients automatically upon the first index backup for a storage policy. You may need to wait up to a day after your first backups in new server backup plans to be able to reconfigure them.

To change the policy for index backups:

1. In the CommCell Browser pane, right-click the **Client Computers** node, and then click **Customize View**.
2. In the User Preferences dialog box, enable the **Show Index Servers** checkbox, and then click **OK**.

NOTE: The checkboxes labeled Show Unlicensed with software and Show Unlicensed without software are selected by default.



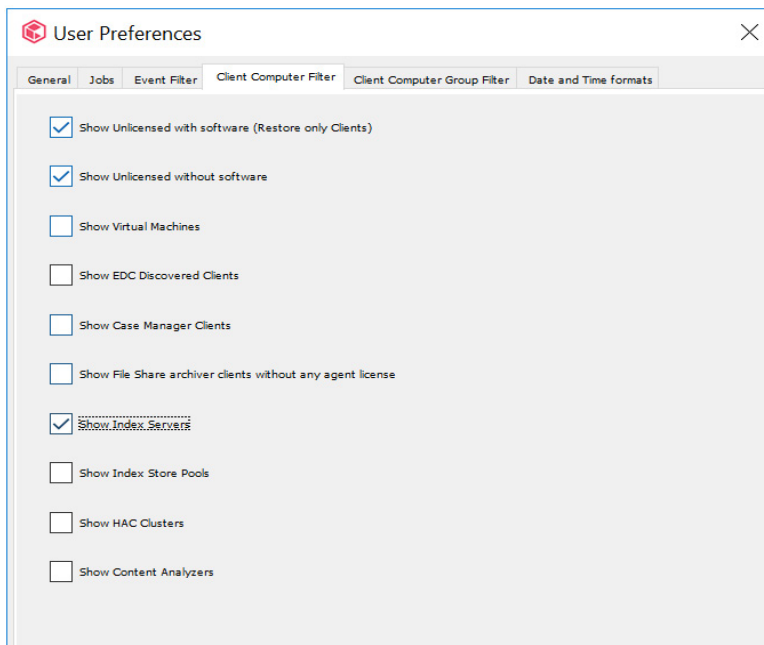


FIGURE 43 Showing index server clients

3. After the index server for the locked server backup plan is created, it appears in the Client Computers list.
 - a. It will follow the naming convention **<plan name>_IndexServer**, so the index server for a plan called "FlashBlade//S with object lock" would be named "FlashBlade//S with object lock_IndexServer."
 - b. You may need to refresh the client computers list before the index server is visible.
 - c. After it is visible, expand the **client** node, expand the **Big Data Apps** node, and then click the **classicIndexInstance** node.
 - d. In the right pane, right-click the default **subclient**, and then click **Properties** and click the **Storage Device** tab.
4. Click the **Storage Policy** dropdown (Figure 44).

A warning popup window appears.

5. Click **OK** to close the warning, then select the unlocked FlashBlade plan you created. A second warning popup window appears.
6. Click **OK** to close the warning and then Click **OK** on the Subclient Properties dialog box to commit the policy change.-

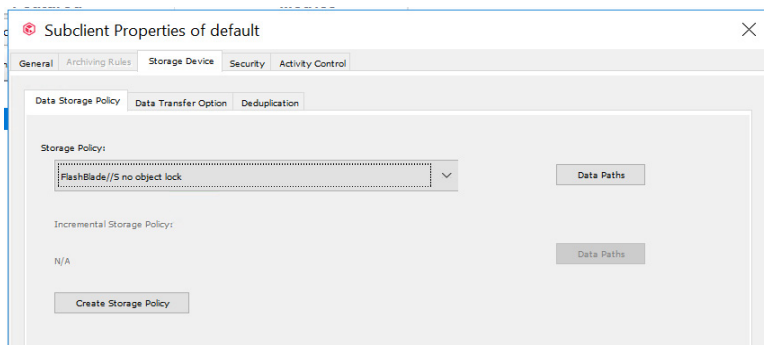


FIGURE 44 Setting index server storage policy



7. After the task is complete, the default subclient displays the unlocked FlashBlade plan as its storage policy as shown in Figure 45.

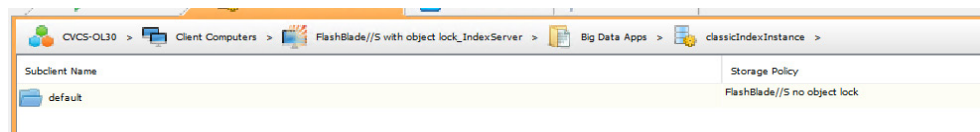


FIGURE 45 Updated index backup subclient

IMPORTANT: Commvault will create an index server for each server backup plan. If you create any new plans later that use the same object lock pool, you will need to reassociate their index servers the same way.

NFS

NFS is a proven protocol for FlashBlade 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.

The following points outline the best results achieved with NFS:

- 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

Use of NFS

You may opt to use NFS instead of object storage, as it is a more familiar protocol. This option is fully supported; however, it entails additional configuration work on your part.

Required MediaAgent Operating System

When opting to use NFS, Linux is the mandatory operating system for MediaAgents. Although Windows offers an NFS client compatible with FlashBlade, note that Commvault does not support new deployments utilizing 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.

Mount Path Configuration

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.

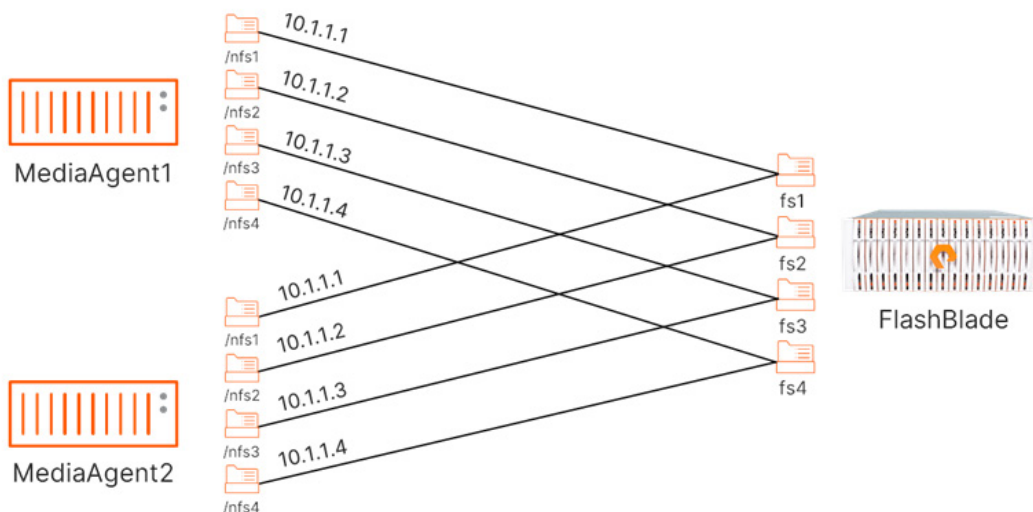


FIGURE 46 Consistent mounting of file systems across MediaAgents



Configure a restrictive export policy on the 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

As Figure 34 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).

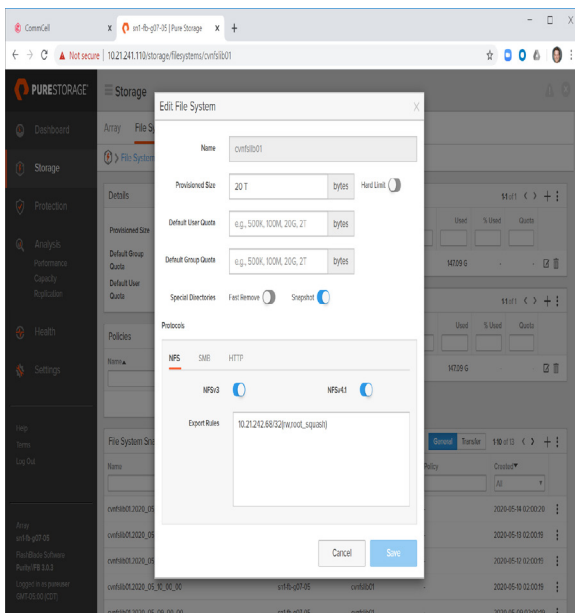


FIGURE 47 Mount path export policy

Mounting the Same File System

Do not 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 of NFS Buffers

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

On Linux or Unix, run the following command:

```
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>.
```



Mount Paths

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.

Creating 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 the Commvault Command Center, navigate to **Storage > Disk**. Click the **Add** link as shown in Figure 48.

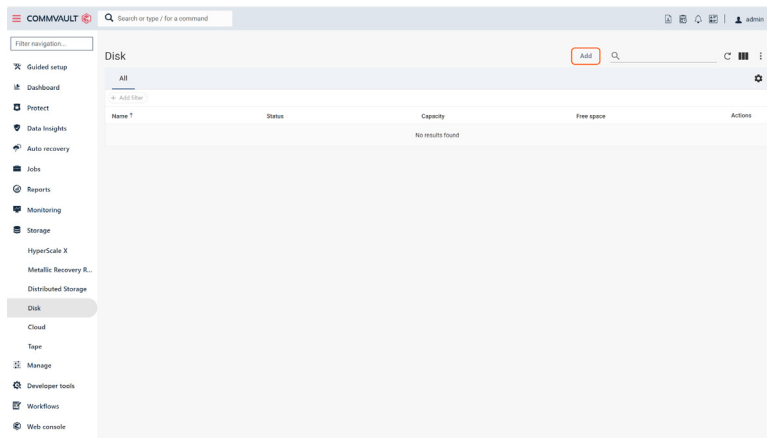


FIGURE 48 Adding a disk storage pool

2. In the **Name** field, enter a descriptive name for the storage pool.
3. In the **Backup location** section, click the **Add** link.
4. Perform the following steps as shown in Figure 49:
 - a. From the MediaAgent dropdown, select a Linux MediaAgent to act as the primary data path and perform pruning.
 - b. Enter or browse to the first NFS mount directory.
 - c. Click **Add** when complete.

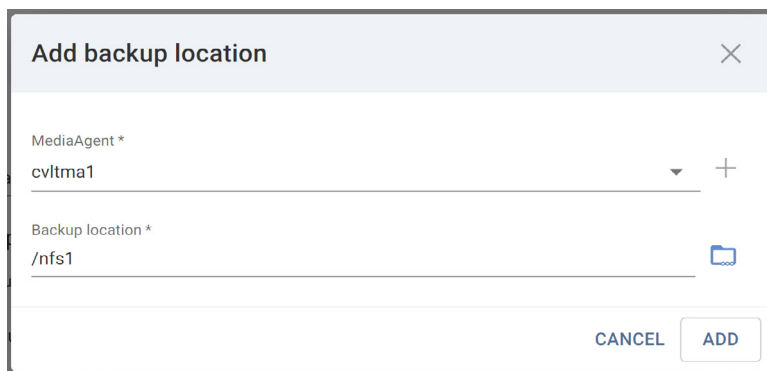


FIGURE 49 Add backup location form

5. Repeat **Steps 3 and 4** to add the remaining mounts.



For each DDB partition you want to create, perform steps 6 and 7.

- 6. In the Deduplication DDB section, click the Add link.
- 7. Perform the following steps as shown in Figure 50:
 - a. From the MediaAgent dropdown, select a **MediaAgent** to host the partition.
 - b. Enter or browse to the directory where the DDB partition should be stored.
 - c. Click **Add** when complete.

The screenshot shows a modal window titled "Add Deduplication DB location". It contains a dropdown menu for "MediaAgent *" with "cvltma1" selected. Below it is a text input field for "Deduplication DB location *" containing the path "/ddb". At the bottom right, there are two buttons: "CANCEL" and "ADD".

FIGURE 50 Add Deduplication DB location form

- 8. Figure 51 shows the expected result with four mount paths and two DDB partitions.

The screenshot displays the "Add disk storage" form. It includes a "Name *" field with "FlashBlade NFS". There are two main sections: "Backup location" and "Deduplication DB location". The "Backup location" section has a table with four rows, each containing a MediaAgent (cvltma1) and a path (/nfs1, /nfs2, /nfs3, /nfs4). The "Deduplication DB location" section has a table with two rows, each containing a MediaAgent (cvltma1) and a path (/ddb). A "Use deduplication" checkbox is checked. At the bottom, there are "EQUIVALENT API", "CANCEL", and "SAVE" buttons.

FIGURE 51 Add disk form with four mount paths

- 9. Click **Save** to create the disk storage pool. Figure 52 shows the created storage pool.

The screenshot shows the "Disk" management interface. It features a table with the following data:

Name ↑	Status	Capacity	Free space	Actions
FlashBlade NFS	Online	32 TB	31.8 TB	⋮

FIGURE 52 FlashBlade NFS storage pool

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 the **Storage Resources**, then **Libraries**.
2. Click the library matching the storage pool you created as shown in Figure 53). The mount paths will appear in the right pane.

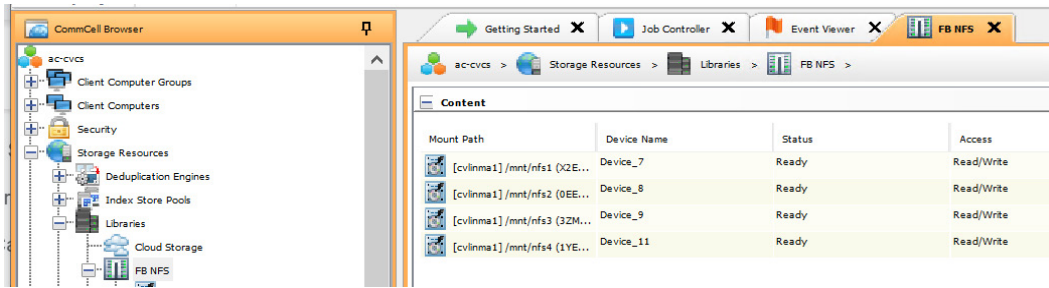


FIGURE 53 Library mount paths

3. Right-click the first **Mount Path**, then select the **Share Mount Path** from the context menu
4. Click the **Sharing** tab of the Mount Path Properties dialog (Figure 54) and then click **Share**.

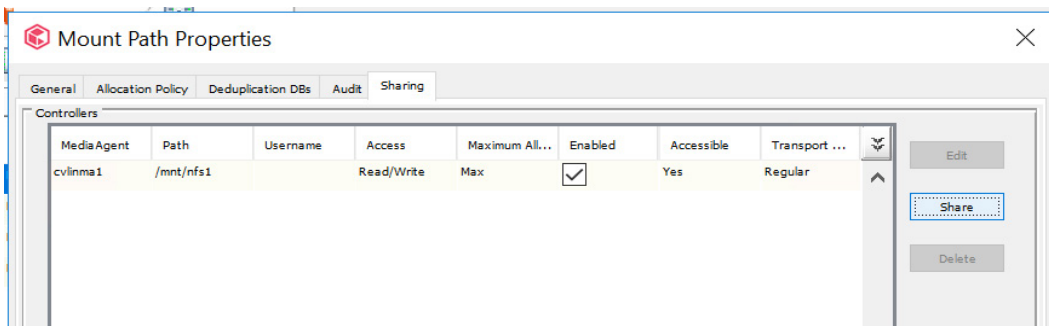


FIGURE 54 Mount path sharing

5. 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**.
6. Select the **Local Path** option (Figure 55), then enter or browse to the path where the file system is mounted and click **OK** to commit the change.



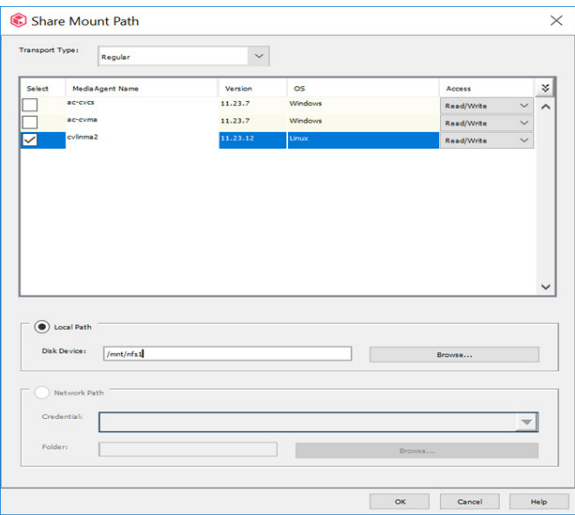


FIGURE 55 Share Mount Path dialog

7. The additional MediaAgents will appear in the list (Figure 56) and 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 update each path to use the Regular model.

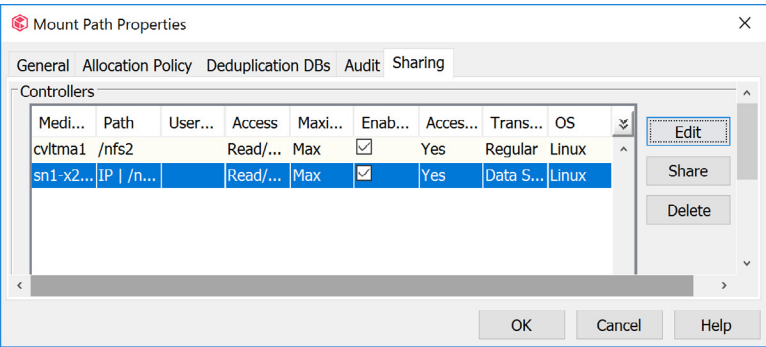
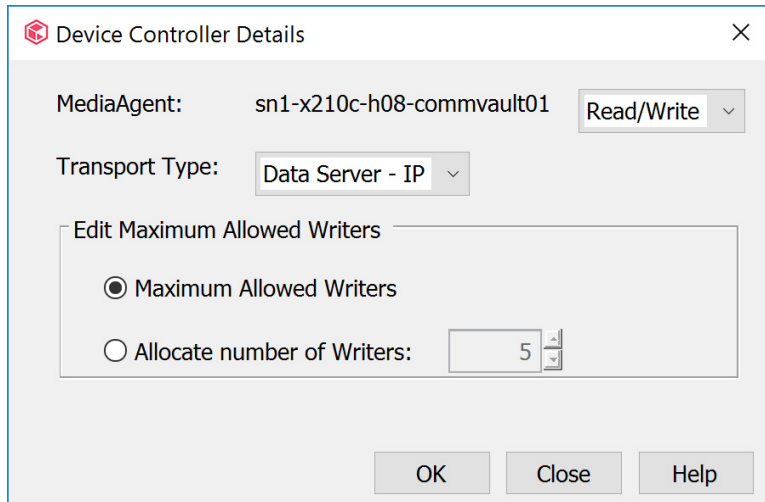


FIGURE 56 Updated mount path sharing list

Perform steps 8 to 13 for each MediaAgent using IP sharing.

- 8. Right-click the path, then select **Share Mount Path** to open the mount path sharing settings.
- 9. For each MediaAgent set with the IP sharing model, select the **MediaAgent** and click **Edit**.

10. In the dialog box that appears (Figure 57), change the Transport Type dropdown to Regular. The dialog box will expand to include additional options.

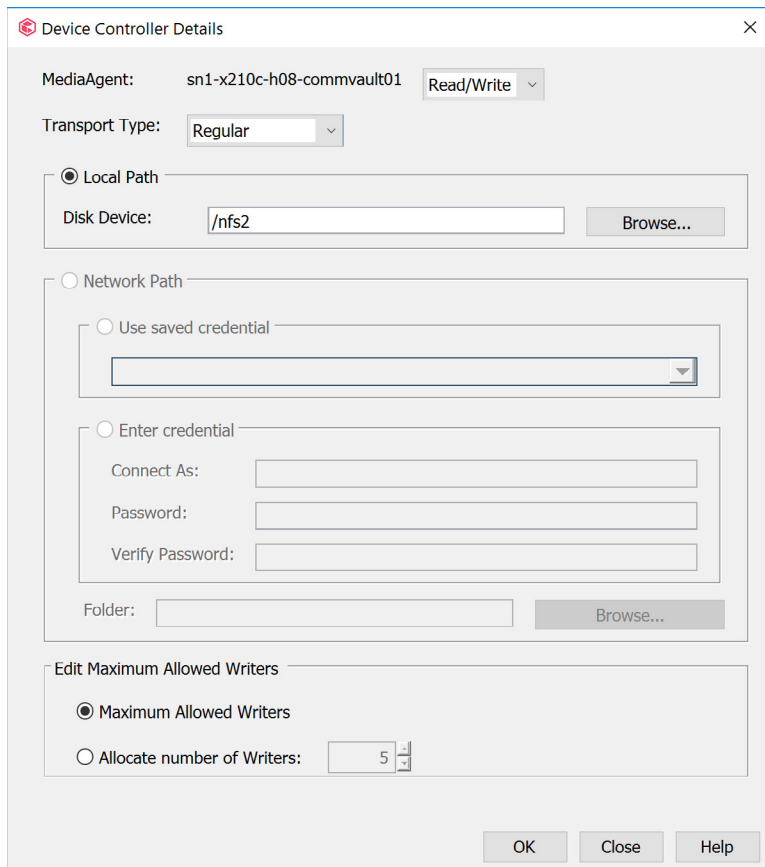


The dialog box titled "Device Controller Details" has a close button (X) in the top right corner. It contains the following fields and controls:

- MediaAgent:** A text field containing "sn1-x210c-h08-commvault01".
- Transport Type:** A dropdown menu currently showing "Data Server - IP".
- Read/Write:** A dropdown menu showing "Read/Write".
- Edit Maximum Allowed Writers:** A section with two radio buttons:
 - ☒ Maximum Allowed Writers
 - ☐ Allocate number of Writers: A numeric spinner box set to "5".
- Buttons:** "OK", "Close", and "Help" buttons at the bottom right.

FIGURE 57 Changing shared mount path transport type

11. In the **Disk Device** field (Figure 58), enter the local path where the FlashBlade file system is mounted, and click **OK**.



The dialog box titled "Device Controller Details" has a close button (X) in the top right corner. It contains the following fields and controls:

- MediaAgent:** A text field containing "sn1-x210c-h08-commvault01".
- Transport Type:** A dropdown menu currently showing "Regular".
- Read/Write:** A dropdown menu showing "Read/Write".
- Local Path:** A section with a radio button ☒ and a text field for "Disk Device" containing "/nfs2". A "Browse..." button is next to it.
- Network Path:** A section with a radio button ☐ and a sub-section with:
 - ☐ Use saved credential: A dropdown menu.
 - ☐ Enter credential: A sub-section with three text fields: "Connect As:", "Password:", and "Verify Password:".
 - Folder:** A text field with a "Browse..." button next to it.
- Edit Maximum Allowed Writers:** A section with two radio buttons:
 - ☒ Maximum Allowed Writers
 - ☐ Allocate number of Writers: A numeric spinner box set to "5".
- Buttons:** "OK", "Close", and "Help" buttons at the bottom right.

FIGURE 58 Updated Disk Device field



12. In the confirmation dialog box, click **Yes**.

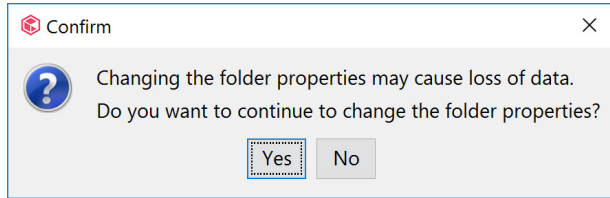


FIGURE 59 Confirm dialog box

13. Click **OK** to close the mount path properties.

14. When the mount paths have been shared correctly, the mount path properties in Command Center displays all the MediaAgents and their associated local mount.

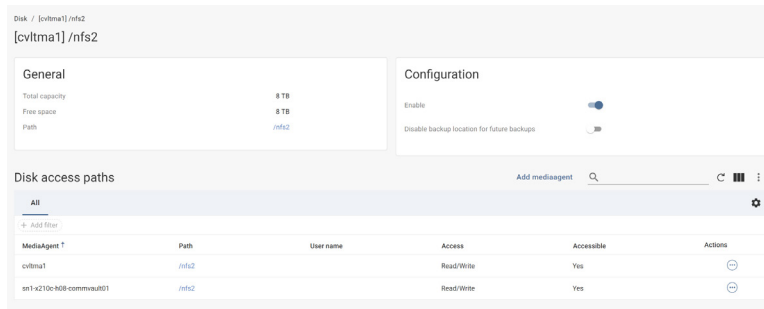


FIGURE 60 Mount Paths in Command Center

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 and 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**, and then click **Schedule Policies**.
2. Right-click on the **System Created DDB Space Reclamation schedule policy**, and then click **Edit** (Figure 48).
3. In the Tasks table, select the schedule, then click **Edit**.



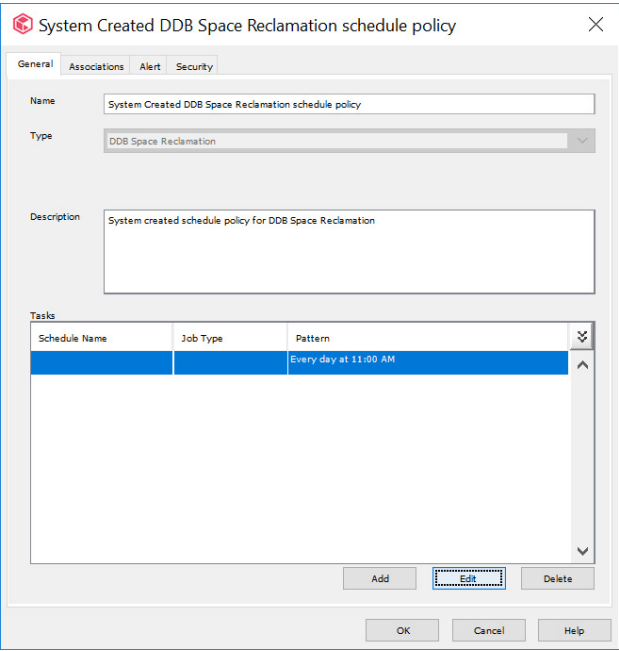


FIGURE 61 Editing the space reclamation schedule

4. You can change the frequency options on the **Schedule Pattern** tab. Pure Storage recommends a monthly schedule if you are using SafeMode snapshots, and at least weekly if you are not.

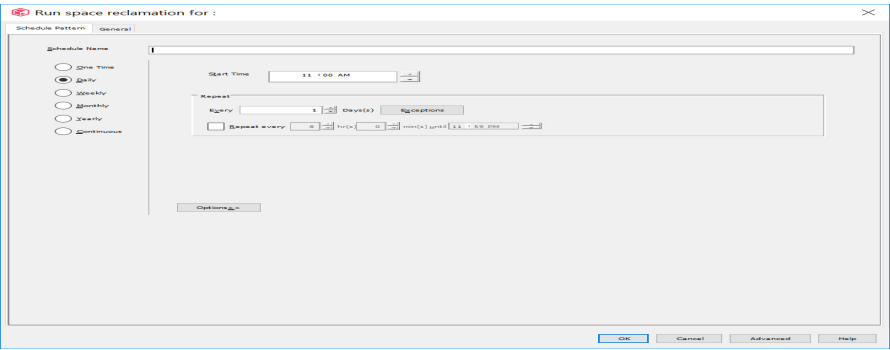


FIGURE 62 Reclamation schedule pattern

5. Click **OK** to set the new schedule, and then click **OK** 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**, and then select **New Schedule Policy**.
2. On the **General** tab, perform the following tasks:
 - d. enter a display name for the policy in the **Name** field.
 - a. Select DDB **Space Reclamation** from the **Type** dropdown.
 - b. You can also add a description in the **Description** field.
 - c. Click **Add** to create a schedule in the policy.

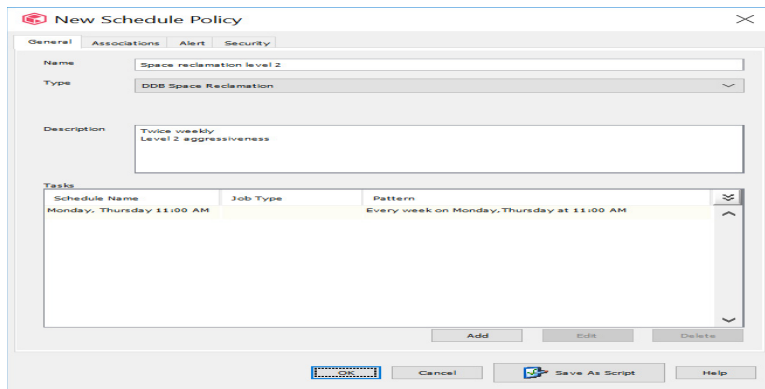


FIGURE 63 Schedule policy general settings

3. In the dialog that opens, on the **Schedule Pattern** tab, set the scheduling options you want. Pure Storage recommends a monthly schedule if you are using SafeMode snapshots, and at least weekly if you are not.

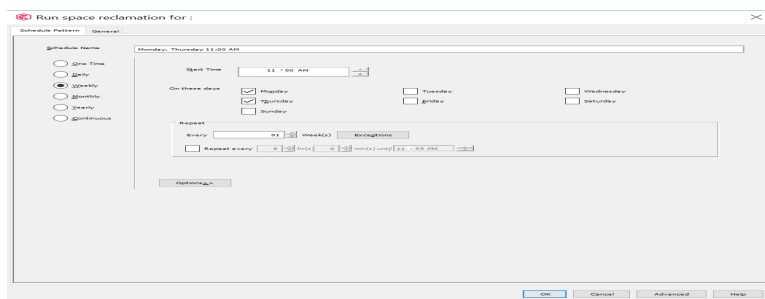


FIGURE 64 Schedule pattern



4. On the **General** tab, adjust the **Reclamation Level** slider to the desired aggressiveness.
 - a. 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.
 - b. Moving it to the right will cause files to be rewritten with fewer expired blocks, using less space but causing more I/O.
 - c. Click **OK** to add the schedule to the policy.

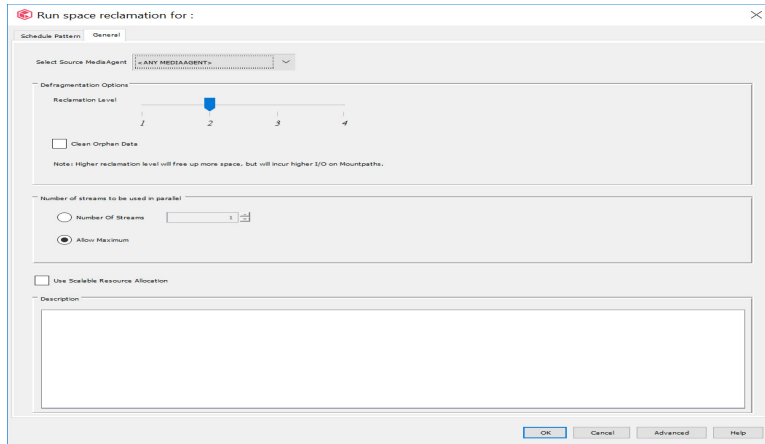


FIGURE 65 Adjusting reclamation level

5. On the **Associations** tab (Figure 65), enable the checkbox **Deduplication Engines** to apply the schedule to all DDBs.
 - a. You can also select specific DDBs to apply only to storage pools for FlashBlade.
 - b. Click **OK** to finish creating the schedule policy.

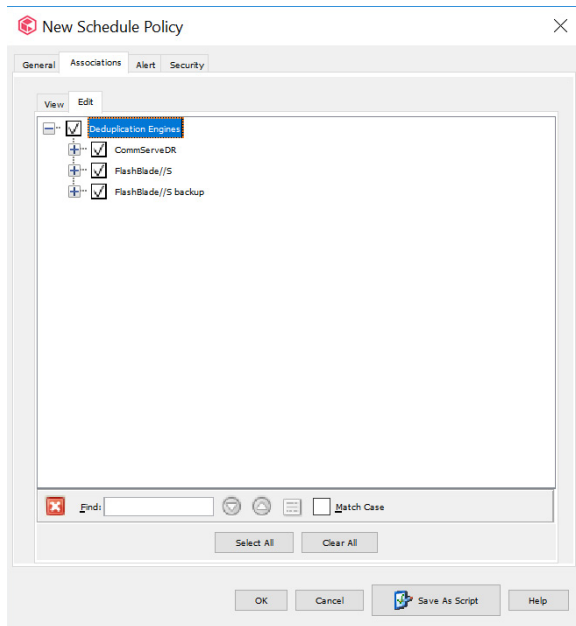


FIGURE 66 Setting schedule policy associations



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** and right-click the FlashBlade library and select **Properties**.
2. As shown in Figure 66, 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.

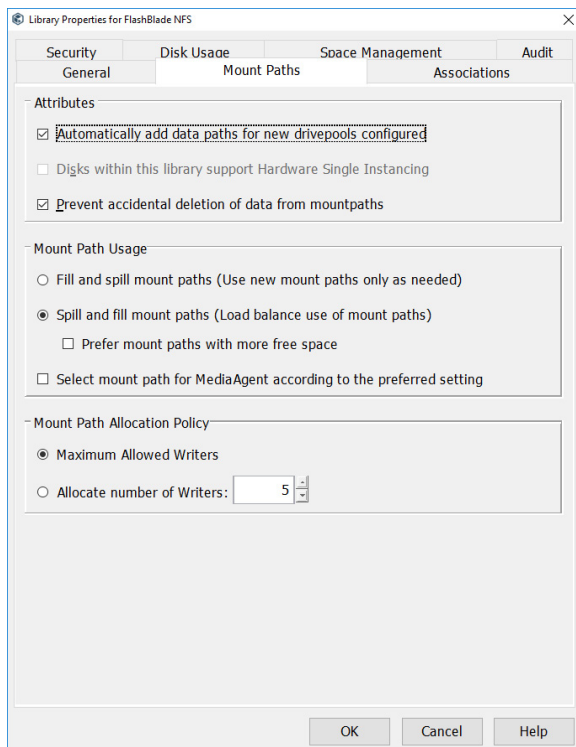


FIGURE 67 Mount path options



CommServe DR Backup

CommServe DR backups also benefit from SafeMode snapshot protection when using an SMB share on FlashBlade. Remember that SafeMode snapshots will apply to all file systems on the FlashBlade. For instructions on enabling SMB support on FlashBlade and connecting to Active Directory, refer to [FlashBlade 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 off site availability
- Upload backups to Metallic cloud
- Upload backups to the cloud library

Use a Dedicated Service Account

Using a dedicated service account ensures that the DR backups cannot 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 67, set up the DR file system with only SMB enabled.

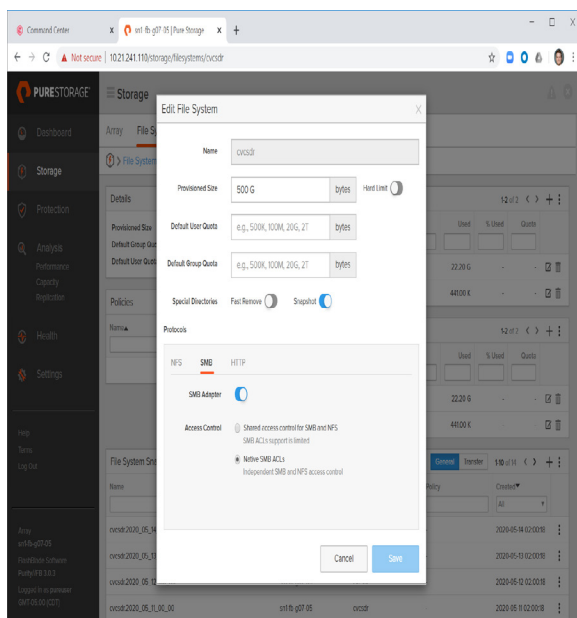


FIGURE 68 SMB export policy



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 68 shows the full ACL.

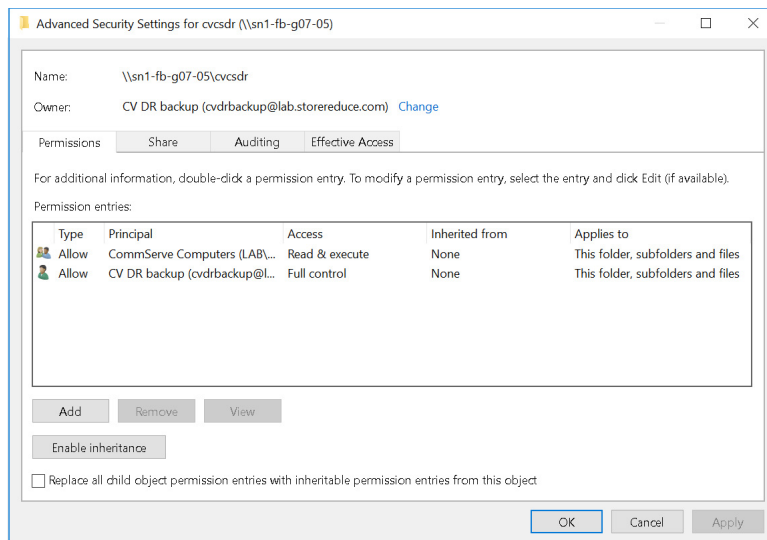


FIGURE 69 DR Backup SMB share ACL

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 Replication to Provide Off-site 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 the [FlashBlade documentation](#) for more detail on enabling replication.



Upload Backups to FlashBlade 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 Object storage.

To enable cloud library upload using Commvault Command Center,

1. In Commvault Command Center, navigate to the **Manage > System** view, then select **Maintenance**.
2. Click the **DR backup** (Daily) tile to fetch the settings, and then click **Edit** (gear icon) to open the properties. (Figure 69)
3. Enable the **Upload backup metadata to cloud library** option, then select the FlashBlade storage pool from the **Cloud library** dropdown.
4. Click **Save** to commit the changes.

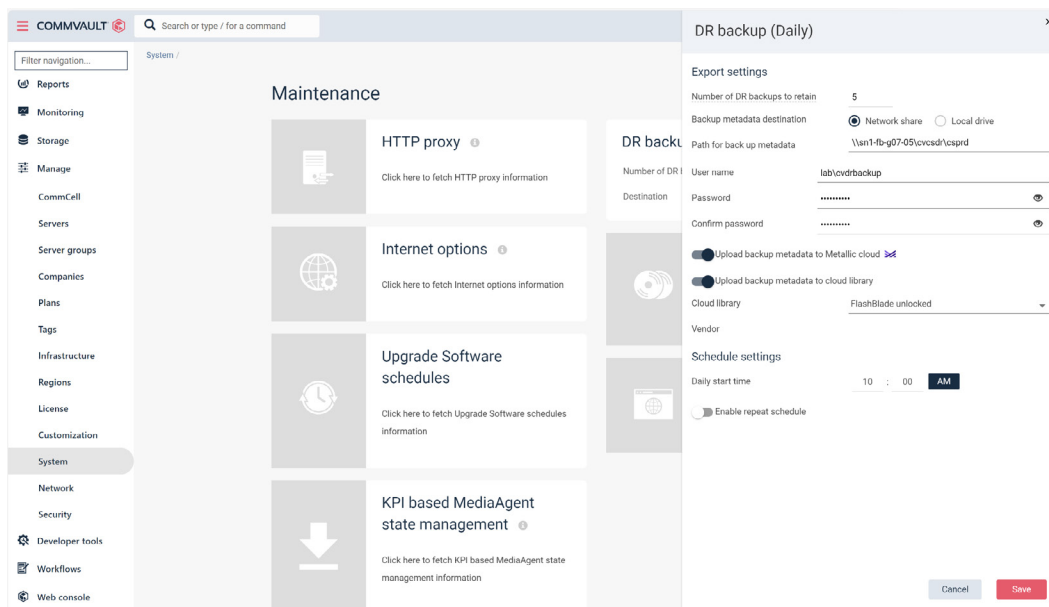


FIGURE 70 DR Backup configuration in Commvault Command Center

Upload Backups to Metallic 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 five 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 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 For more information, see Apply a Signed Array Certificate , and Create Commvault Storage Pools
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 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, 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 Create Commvault Storage Pools
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 6 Cloud target creation failure



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 Apply a Signed Array Certificate , and Create Commvault Storage Pools
Unsuccessful TLS certificate validation with CA-signed certificate	Could not access mount path	Name resolution failed on the MediaAgent for the FlashBlade 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 7 Backup and restore failures



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 or FlashBlade//E	Commvault thread pool is not large enough to maximize throughput.	Increase the thread pool maximum size. This also suggests a CPU and memory resource limitation on the MediaAgent. Consider upgrading the MediaAgent hardware. See (Optional) Configure Cloud Upload Thread Pool Size
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.	Separate front-end and backend network traffic on different physical adapters. See Segregate Client and Storage Networks
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.	Use Storage Accelerator to increase the overall available bandwidth. See Use Storage Accelerator for Additional Horizontal Scaling
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.	Increase CPU resources and/or reduce CPU load Enable source-side deduplication and compression or disable compression. Using Storage Accelerator can reduce the MediaAgent CPU load.

TABLE 8 Throughput failures.



Terms and Concepts

The following explains the terms and concepts that you will find useful.

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.
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.
Network File System (NFS)	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 9 Terms and concepts

purestorage.com

800.379.PURE

