FlashArray Asynchronous Replication Configuration and Best Practices Guide

FOR PURITY VERSIONS 5.X AND HIGHER
OCTOBER 2021

PURESTORAGE®

THIS GUIDE APPLIES TO PURITY ***VERSION 5.0.X AND HIGHER***

INTRODUCTION

Pure Storage enables customers to cost effectively deploy all-flash storage solutions in environments where traditional storage arrays can no longer meet the growing performance and efficiency requirements of today's workloads. These workloads are often the most important workloads in an enterprise environment, requiring data protection for business continuity and disaster recovery. Most all-flash storage systems require the use of host-based replication solutions that involve complex configuration, time-consuming recovery processes, and expensive licenses that increase the overall cost of the storage solution resulting in unexpected additional costs. The choice to implement an all-flash storage solution should not require compromises, should not add complexity to the environment, and should not increase the overall cost of the solution.

To meet these needs Pure Storage provides Asynchronous replication: a license-free, array-based, data reduction-aware asynchronous replication solution that's simple to configure and manage while providing robust capabilities for enterprise workloads.
Asynchronous replication is a snapshot-based asynchronous replication solution that leverages space efficient snapshots to replicate point-in-time consistent copies of multiple LUNs. Asynchronous combines snapshots and replication into a single solution allowing you to create multiple recovery points on the replication target.

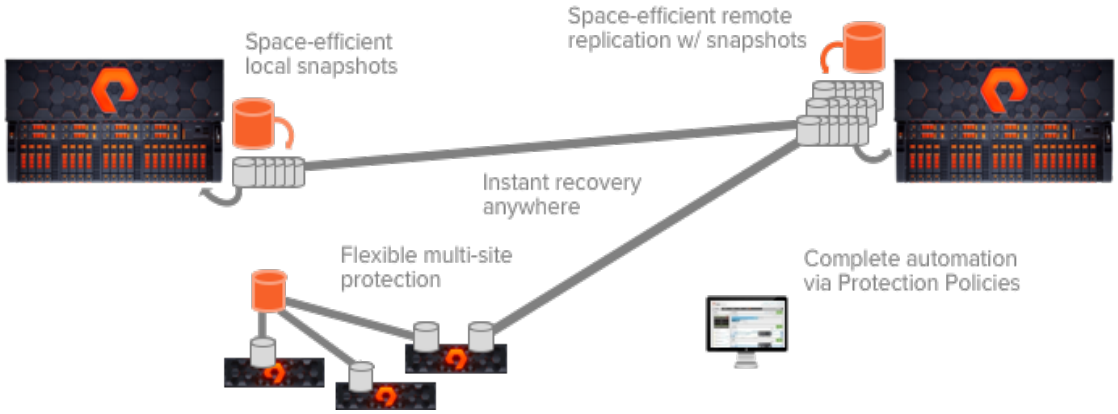


Figure 1 - Asynchronous Replication

EFFICIENCY

Purity snapshots do not require any space reservations and have no capacity or performance impact on creation. These snapshots can be replicated simultaneously to multiple remote sites over long distances. The replication engine compares the last replicated snapshot to the new one and sends only incremental data between the two snapshots.

Pure Storage's FlashArray is a data-reducing all-flash array. Asynchronous replication is designed to provide customers an array-based replication solution capable of preserving the data reductions they gain on the source array. Asynchronous replication translates those savings into decreased WAN and target array utilization as compared to other array-based replication solutions that do not replicate these efficiencies.

SIMPLICITY

Unlike typical array-based replication solutions that require the storage administrator to configure replication independently for each LUN or storage object on the system, Asynchronous replication allows configuring protection for all the LUNs provisioned to a host or group of hosts. This eliminates unnecessary configuration steps, reduces the chance of missing some of a host's LUNs when configuring protection, and removes the risk of forgetting to protect new LUNs added to a host after replication was already configured.

FlashArray replication can be set up within minutes in just a few simple steps – connect, configure and schedule. It requires no professional services or complex planning.

INSTANT RECOVERY

In addition to a low recovery point objective (RPO), a simple recovery procedure that enables a low recovery time objective (RTO) is of utmost importance. Achieving a low RPO while still managing a complex and time-consuming recovery procedure defeats the purpose of an efficient replication solution. Asynchronous Replication provides instantaneous recovery of multiple LUNs from any point-in-time replica preserved on the array. This enables a very low RTO, allowing customers to achieve faster recovery of multiple applications in case of disasters.

FLEXIBILITY

Asynchronous replication delivers flexibility to allow replication to be used for more than just data protection and disaster recovery. New LUNs can be created and provisioned for host IO independently of their parent replicas. Replicas can be used on other arrays to create multiple instances of the same environments without locking snapshots or being bound by dependencies between volumes. This enables the use of DR sites for application workloads or for Development and Test workloads without disrupting the protection policies or other workloads on the arrays. LUNs created from replicas preserve the data reduction of the replica they are recovered from, and they do not require any additional capacity at the time of creation. Replication can also be used to efficiently migrate workloads between arrays in the same data center or to another data center.

MULTI-ARRAY, MULTI-SITE RECOVERY

Purity Replication is a peer-to-peer multi-point disaster recovery solution that can be configured to replicate data between any arrays in the FA-400 family. Once connected, the arrays can participate in bi-directional replication.

Purity Replication can be configured in both fan-in and fan-out topologies. In such topologies, FlashArray's comprehensive and global deduplication can provide significant capacity savings for replicas on the DR FlashArray, making DR both more efficient and affordable.

DEDICATED CONNECTION & SECURITY

Asynchronous replication is IP based, enabling long-distance replication over standard existing IP infrastructures. Ensuring that replication does not interfere with the production front-end workload on the array is important, so FlashArrays are configured with dedicated Ethernet ports for replication.

Connecting arrays for replication requires administrator credentials and a secure connection from the peer array. Asynchronous replication does not encrypt data on the wire today; however, data is always encrypted at rest within the source and target FlashArray.

ASYNCHRONOUS REPLICATION

THE REPLICATION PROCESS

Asynchronous replication leverages Purity snapshots. A snapshot is created on the source, and that snapshot is replicated to the target array. The first transfer, known as the baseline, is a complete transfer of the entire contents of the volume. After the baseline copy is performed, the replication process performs incremental transfers by differencing the previously replicated snapshot and a newly created snapshot to determine which data to send to the target array.

The replication process consists of the following operations:

1. Determines the differences between a newly created snapshot on the source and the snapshot that was previously replicated from the source.
2. The differenced data is read from SSD at the source in compressed format.
3. Unique data is transmitted over the wire to the target array in compressed format.
4. The target array receives the compressed data and processes it to prepare for inline deduplication.
5. Inline deduplication reduces the incoming replication stream against other incoming replication streams replicating at the same time, and against data that already exists on the target array.
6. Deduplicated data is written to SSD in compressed format, in protected RAID-HA segments that are encrypted.
7. Background processes in Purity then work to ensure data is correctly managed on the flash media. This includes global wear leveling and refreshing, management of deletion of

data, data integrity checking, and continuous optimization in the form of background deep reduction to further reduce the footprint of all data on the array.

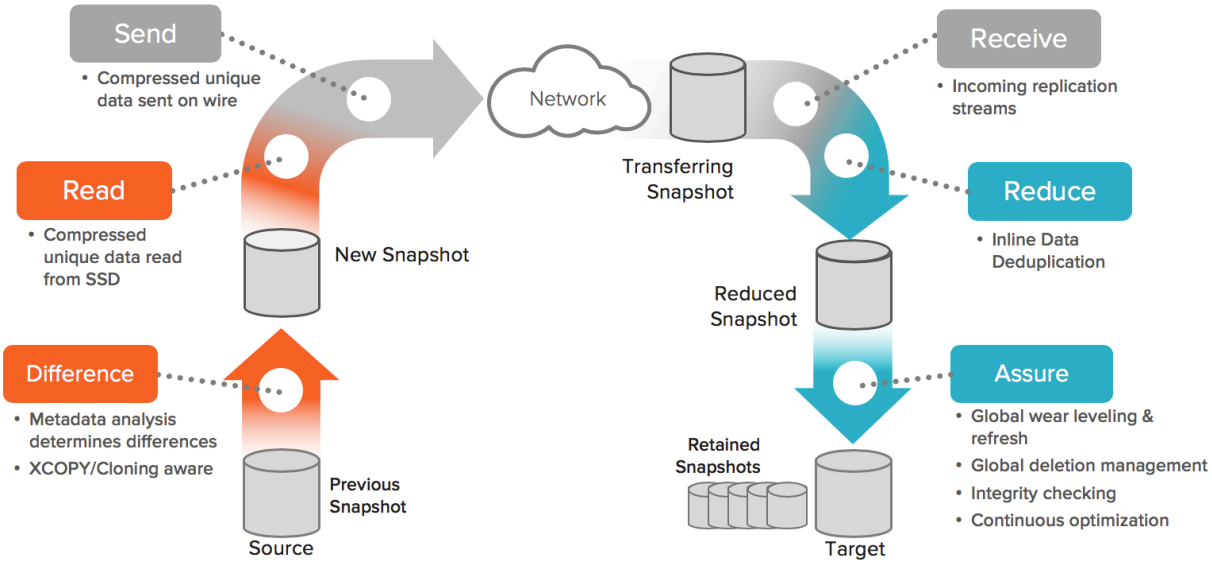


Figure 2 - Replication Process Flow

DATA REDUCTION SUPPORT

The Pure Storage FlashArray is a data reducing storage array. All volumes (or LUNs) on the array are completely virtualized devices that can share duplicate data from a common pool of variable sized blocks. Any blocks that are referenced by multiple volumes are said to be deduped, or shared. Variable sized blocks are also compressed before being written to SSD. Both dedupe and compression contribute to the overall data reduction in the array.

There are multiple forms of data reduction within the FlashArray. Asynchronous replication in Purity supports each type of data reduction as outlined below:

- **Data reduction from XCOPY & cloning** - Asynchronous replication is XCOPY & cloning aware and maintains XCOPY/cloning savings on the wire and target.
- **Compression** – Content transferred by Asynchronous replication is controlled by the target FlashArray based on whether or not the content (data) is already present on the target. Transferred data is always compressed on the wire and on the target array.
- **Data reduction from inline dedupe** – Starting in Purity 4.6 inline dedupe savings on the source is preserved on the wire for data replicated between a given source and target array.
- **Data reduction from background deep reduction** - After data is written on the target, Purity's deep reduction processes can further reduce the storage footprint of replicated data in the background.

Preserving XCOPY and Cloning with Replication

XCOPY and cloning are essentially the same form of data deduplication. They are both logical copies of data created by making copies of metadata that reference shared physical content. Asynchronous replication is designed to be XCOPY aware. However, replication cannot simply copy only metadata to the target array. The target array must have the required physical data for logical copies to reference. Replication tries to retain XCOPY savings when possible. There are some factors that determine how much savings can be retained:

- The amount of overwriting that has occurred in the XCOPY or cloned areas.
- Whether or not volumes containing the data referenced by XCOPY or cloned data are being replicated.

In the example shown in Figure 3 if you replicate only vol2, then replication must send at least one full copy of block B to the target or the target array will have no real data for logical blocks B^4, B^5, and B^6 to reference. Maintaining the data reduction within logical copies B^4, B^5, and B^6 can also depend on how much data within the XCOPY'ed ranges have been overwritten before replication occurs.

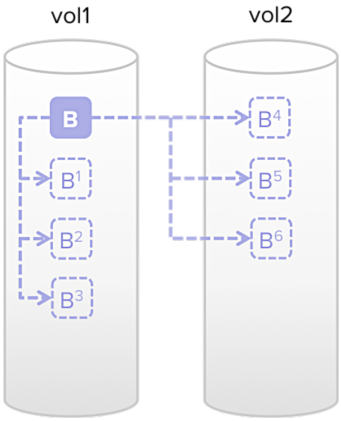


Figure 3 - XCOPY and Cloning

Preserving Compression with Replication

Data is always transferred in compressed format and written to SSD at the target array in compressed format. Purity always stores compressed data on SSD. Whenever data is read by a host, such as with the FC or iSCSI protocol, that data is always decompressed when sent in response to a host read request. In Asynchronous replication data always remains compressed. Compressed data is read from the source array SSDs and transferred over the wire in compressed format where it remains compressed and is written to the destination SSDs.

Preserving Deduplication with Replication

Starting in Purity 4.6 savings from inline dedupe on the source can be maintained on the wire for data that has already been transferred between a replication source and replication target array. This can reduce the network bandwidth required to perform replication transfers.

Asynchronous replication preserves dedupe on the wire by exchanging dedupe metadata between the source and target array. A replication target array sends remote content metadata to the source array as user data content is sent between arrays. The source array stores this metadata and can then reference it to determine whether or not a target array already has certain content before sending it on the wire. This exchange of metadata happens within large baseline transfers as well as in incremental transfers such that both types of transfers can get the benefit of dedupe preserving replication. As more data is sent to the target, more remote content metadata is generated that can be used to reduce the amount of data that must be sent during the in-progress transfer or during a later transfer.

Dedupe preserving replication provides the following benefits:

- Reduces the bandwidth required for large baseline transfers when replicating highly deduplicated data.
- Improves the performance of replication on low bandwidth WANs.
- Reduces the unnecessary buildup of duplicate content on a replication target.
- Reduced the utilization or resources to perform background deduplication on the replication target.

After data has been sent to the target array further data reduction might still be achieved by background deep reduction. Purity will continuously use background resources to scan for duplicate content to improve storage efficiency. Data sent to a target array might be further reduced in the background against other content that already existed on the target array.

Storage Efficient Replication

Let's look at a hypothetical example shown in Figure 4 of a volume on a replication source containing 100GB of logical data. There is some data reduction due to XCOPY/cloning operations. There is reduction from compression – let's assume the FlashArray compressed the data by 60%, so that gets us to 40GB. There is also some savings from inline dedupe of written data, as well as even more savings that was discovered later by the deep background reduction processing. So, in this example, after multiple forms of data reduction, the FlashArray has 25GB of unique data in this volume, which is being referenced by metadata representing 100GB of logical data to the host using this LUN.

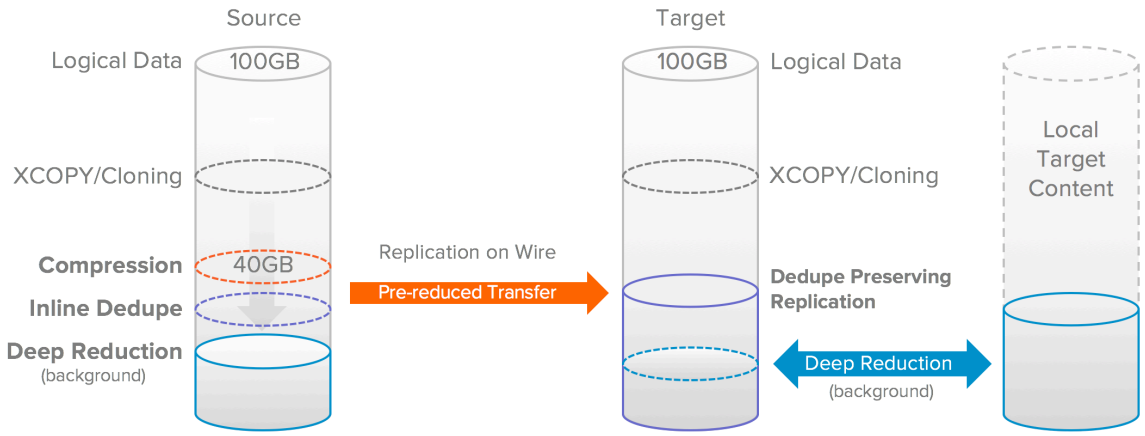


Figure 4 - Storage Efficient Replication

If you replicate this volume using Purity 4.6 or higher the FlashArray needs to send at least 25GB of unique data on the wire however it is unlikely that it will send only that much. If this is the first initial baseline transfer of data between these two arrays the arrays will need to build up some remote content metadata before any on-the-wire savings can be achieved depending on how much deduplication there is (the deduplication ratio) within the data being transferred. If the content of this volume is shared with other volumes that have already been replicated from this source to this target, then the bandwidth requirements could be significantly reduced.

The amount of data reduction that is maintained between a source and target depends on multiple factors:

- The versions of Purity running on the source and target arrays
- The amount of compression of the data on the source array
- How reducible the data is within the replication data stream, determined by the presence of duplicated blocks already on the target
- How reducible the data is against other replication streams running simultaneously to the same target
- The amount of data already existing on the target array that might contain blocks that are duplicates of those in the incoming replication streams

RPO MAINTENANCE AND FRONT-END WORKLOAD PRIORITY

A Recovery Point Objective (RPO) is the maximum amount of data loss deemed acceptable at the time of recovery after a disaster. For example, a RPO of one hour allows recovery on the target array with maximum possible loss of one hour of data. Asynchronous replication is an asynchronous replication technology. Changes that occur on the source system are replicated to the target system on some pre-defined interval (the replication frequency). As those changes are being replicated, other changes are being accumulated on the source to be replicated at the next interval. Any RPO requires a replication frequency of a maximum of half the RPO. For example, a RPO of one hour requires a replication frequency of 30 minutes. This is due to the need to fully

complete the current active transfer before it is useable. If a disaster occurs just before the current active transfer is complete, that snapshot will not be valid and recovery would have to be performed using the last snapshot that was completely replicated, which would have been created up to one hour ago.

Maintaining the configured replication frequency depends on much more than the minimum configurable replication frequency setting allowed. The primary factors affecting RPO maintenance are:

- Network bandwidth between the source and target
- Incoming write rate on the source volumes
- The front-end workload on the array

Purity 5.0.x and higher supports 1Gb/s, 10Gb/s, 25 Gb/s, and 40 Gb/s (passive DAC cable) options for the replication network. FlashArrays running Purity 5.0.x and higher are able to configure each of their replication ports with their own IP address (instead of Replbond). This allows load balancing across both replication ports on each controller and to facilitate faster failover between controllers and / or replication ports. This is referred to as a physical replication port configuration and is the recommended configuration methodology for FlashArrays running Purity version 5.0.x and higher. For compatibility with pre-5.0.x versions of Purity, replication ports continue to be configurable using a Replbond configuration.

Note: Replbond connections **cannot** be used by ActiveCluster. If replication will be changed from Asynchronous to Synchronous (ActiveCluster) then a physical replication port configuration is required.

More frequent replication frequencies will of course require more network bandwidth. In most cases the WAN connection between sites will be the slowest component of the replication network. The WAN bandwidth and latency should be considered when determining how much throughput can be achieved between sites for Asynchronous replication.

Each replication update deposits data onto the target array. After each transfer, processes such as deep reduction and capacity accounting will run in the background to account for this new data. Adequate time should be given in each replication interval to allow for the system to process these background tasks. This means that the incoming write rate of the front-end workload should be taken into account when determining how much time should be allowed between each transfer. It is recommended that no more than half the replication interval be used to send data to the target. This means that the replication network should support about 2x the bandwidth of incoming writes. When considering the amount of bandwidth available network compression should be considered. Asynchronous replication transfers compressed data on the wire. So, 200MB/s of incoming FC/iSCSI writes on the source would use about 100MB/s of replication bandwidth assuming a 2:1 compression of that data. The maximum replication throughput varies depending on the storage platform.

In Purity, priority is given to maintaining consistent performance for the front-end workload on the array over the performance of replication. In other words, replication is a lower priority process than the processes that serve iSCSI and FC front-end workloads. Background processes in Purity are used to optimize metadata to ensure front-end read latencies remain low. If these background processes fall behind, the creation of scheduled snapshots on the source may be delayed, which prevents replication from occurring. Once the background processes catch up, the system will allow scheduled snapshots to be created. In this way, Asynchronous replication maintains the best RPO it can, in an effort to replicate as often as configured, depending on the workload on the system. When snapshot creation is delayed, an alert is generated on the system.

ACTIVE-ACTIVE ASYNCHRONOUS REPLICATION

Active-Active Asynchronous for ActiveCluster
Leverages ActiveCluster FA as a source and provides asynchronous replication to a target FlashArray (at a 3rd site). A-A Asynchronous replication uses protection groups inside of ActiveCluster pods configured with an asynchronous replication target and schedule. When using Active-Active Async the target array transfers snapshot updates from both ActiveCluster source arrays at the same time. Asynchronous replication connections from both of the ActiveCluster source arrays to the async target array are required. In the event of an ActiveCluster failover, async replication continues automatically from the surviving ActiveCluster array, which allows async RPOs to be maintained regardless of which side of the ActiveCluster remains available. For more details about Active-Active Asynchronous replication click this link:

https://support.purestorage.com

and use key word ‘Active-Active Asynchronous Replication’ to locate related knowledge base articles.

CONFIGURING ASYNCHRONOUS REPLICATION

Asynchronous replication can be configured using the Purity web GUI, CLI, or by REST API. This guide demonstrates configuring replication using the GUI, explaining the configuration of various replication components and noting best practices along the way.

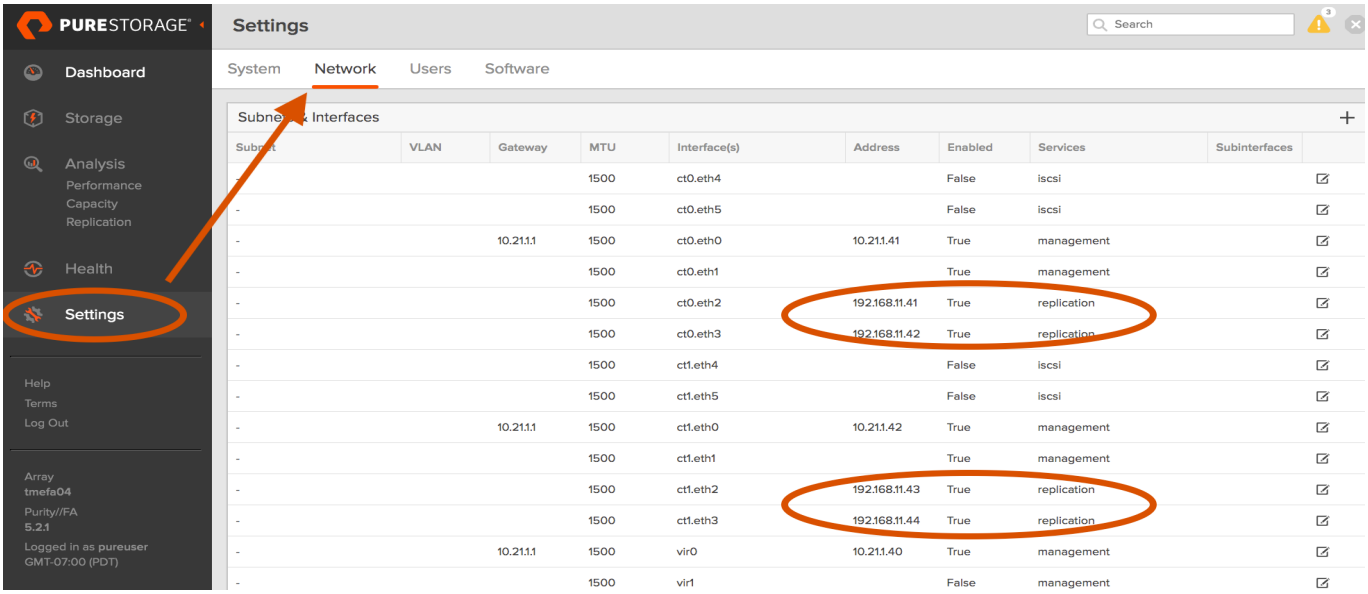
Configuring replication consists of the following high-level actions:

1. Review the requirements for replication
2. Connect the source and target array
3. Create a protection group
4. Configure a replication schedule and retention policy
5. Select the replication target arrays
6. Add volumes to the protection group
7. Allow the protection group to be replicated

REQUIREMENTS

Asynchronous replication requires the following configuration items before proceeding with replication.

1. The source and target arrays must be connected to Ethernet switches. **Direct connecting two storage arrays for replication is not supported.**
2. Each FlashArray must have 4 physical replication interfaces (2 per controller) or a replbond interface configured with the replication service assigned. The replication interfaces can be managed in the Settings > Network tab as shown below.



3. By default, the eth2 and eth3 interfaces are used for replication on each controller. They must be configured with an IP address, netmask, and gateway and be set to enabled on all replicating arrays. This is done in the Settings > Networking tab by clicking then Edit Interface button (the icon in the last column to the right) on each interface.
4. Pure recommends leaving MTU set at 1500 (default). Although jumbo frames (MTU size 9000) are supported for replication, they are of limited benefit compared to using 1500. If using jumbo frames, ensure the network devices between the arrays have been configured to support the larger MTU size.
5. Connecting arrays to each other requires access to port 443 and 8117. Ports 443 and 8117 are used for connect and disconnect operations. Replication data transfer uses only port 8117.

PURITY VERSION REPLICATION INTEROPERABILITY

Replication is supported in both directions between different minor versions of Purity with some limitations.

Starting with Purity 4.8, replicating to a 4.8 target array requires that the source array be running Purity 4.7.5 or newer. Replication is not possible between Purity 4.8 and versions of Purity older (lower) than 4.7.5. Purity version 5.0.x and higher are also backward compatible subject to the Purity Requirements and Interoperability Matrix. For the most up-to-date information about Purity replication interoperability see:

https://support.purestorage.com/FlashArray/PurityFA/Protect/Replication/Purity_Replication_Requirements_and_Interoperability_Matrix

REPLICATION NETWORKING

Asynchronous replication requires a redundant IP network for replication on the FlashArray. The ethernet ports used for replication must be dedicated and cannot be shared with other services on the array such as management or iSCSI. Two physical ethernet ports on each FA controller are assigned individual IP addresses or (for backward compatibility) bonded together using the Replbond interface. A physical replication port configuration is the recommended way to setup replication connectivity starting with Purity 5.0.x. This is because a physical replication port configuration has the advantage of load balancing across both ethernet interfaces on each controller (double the bandwidth of Replbond) and faster failover between controllers due to the elimination of the arp updating that Replbond requires.

Ports and Services

Purity assigns services to specific interfaces on the FlashArray. Firewall rules govern which protocols can use the ports on which the services are assigned. By default, only one service is assigned to each interface. The replication service may not share an interface with any other service.

Pure Storage Support handles configuration of network ports and services for replication on all FlashArrays. Customers may configure the settings on the ethernet interfaces such as the IP address, netmask, gateway and MTU size.

Depending on the FlashArray model and hardware configuration, replication ports may be redundant or non-redundant. The FlashArray supports options for 1GbE, 10GbE, and 25 GbE replication.

Note: If Replbond is used, 1GbE, 10GbE, and 25GbE ports are not supported in the same replication replbond interface.

Redundant configuration of the replication network requires two ports per FlashArray controller.

All FlashArray platforms support redundant replication network designs on each controller with the following exceptions:

- The FA-405 only supports redundant replication network configuration if using either the onboard 10GbE ports or an expansion 10GbE card for replication. If using 1GbE network for replication on the FA-405, only one port is available on each controller. Pure Storage Support must reconfigure the ports used for replication if necessary.
- The FA-450 only supports redundant replication if using the onboard 10GbE ports. A single onboard 1GbE port may be configured for replication, non-redundantly. Pure Storage Support must reconfigure the ports used for replication if necessary.
- The FlashArray//m10 iSCSI array supports replication only using a single 1GbE port (eth1) per controller.

Default replication ports are defined for systems running Purity 5.0.x or higher, or on systems upgraded to Purity 5.0.x from an earlier release.

The following table describes the default and optional replication ports configurations for each of the FlashArray 400 series arrays.

	FA-405	FA-420	FA-450
Default replication ports	1GbE (port eth1, non-redundant)	1GbE (ports eth2 and eth3, active/passive redundant)	10GbE (ports eth2 and eth3, active/passive redundant)
Optional replication ports	10GbE (dual port Slot 3, active/passive redundant) or 10GbE dual onboard ports	10GbE (dual port Slot 1, active/passive redundant)	1GbE (port eth1, non-redundant)

	(active/passive redundant)

The following table describes the default and optional replication ports configurations for each of the FlashArray//m arrays.

	m10 iSCSI	m10 FC, m20, m50, m70, //X10, //X20	//X50 //X70 //X90
Default replication ports	1GbE (port eth1, non-redundant)	10GbE (ports eth2 and eth3, active/active redundant)	10GbE, 25GbE (ports eth2 and eth3, active/active redundant)
Optional replication ports	None	1GbE (port eth1, non-redundant) or 1Gb RJ45 copper to SFP adapters in ports eth2 and eth3 as described below	1GbE (port eth1, non-redundant) or 1Gb RJ45 copper to SFP adapters in ports eth2 and eth3 as described below

1GbE RJ45 to SFP adapters may be used in the 10GbE replication ports ETH2 and ETH3 in order to configure redundant 1GbE replication while using the default replication ports. Pure Storage does not provide these adapters; they must be purchased by the customer. SFP support for the FlashArray is described here:

https://support.purestorage.com/FlashArray/FlashArray_Hardware/General_Hardware_Troubleshooting/FlashArray_SFP_Support

The following table describes the services and associated protocols and ports.

Service	Allowed Protocols
Management	SSH, HTTP, HTTPS, SNMP
iSCSI	TCP on port 3260
Replication	TCP on port 8117

Changing services and reconfiguring ports must be done by Pure Storage Support personnel.

Redundant Replication Network Design

Figure 5 illustrates the logical connectivity used to provide a redundant network connection for physical replication ports.

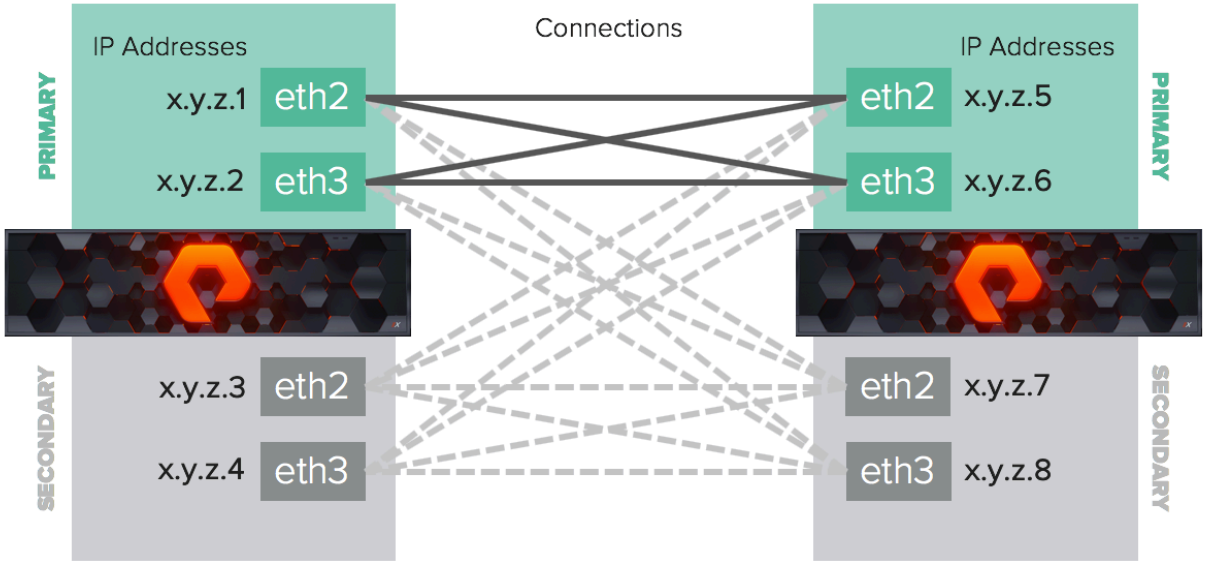


Figure 5 - Physical Replication Port Connections

The solid black connections represent the connections between the two active replication ports using a physical replication port configuration on each controller. The grey connections represent the logical replication port connections on the secondary controllers that are connected to the peer FlashArray, however these standby connections are only used if the secondary controller becomes the primary in the event of an HA failover (shown Figure 6).

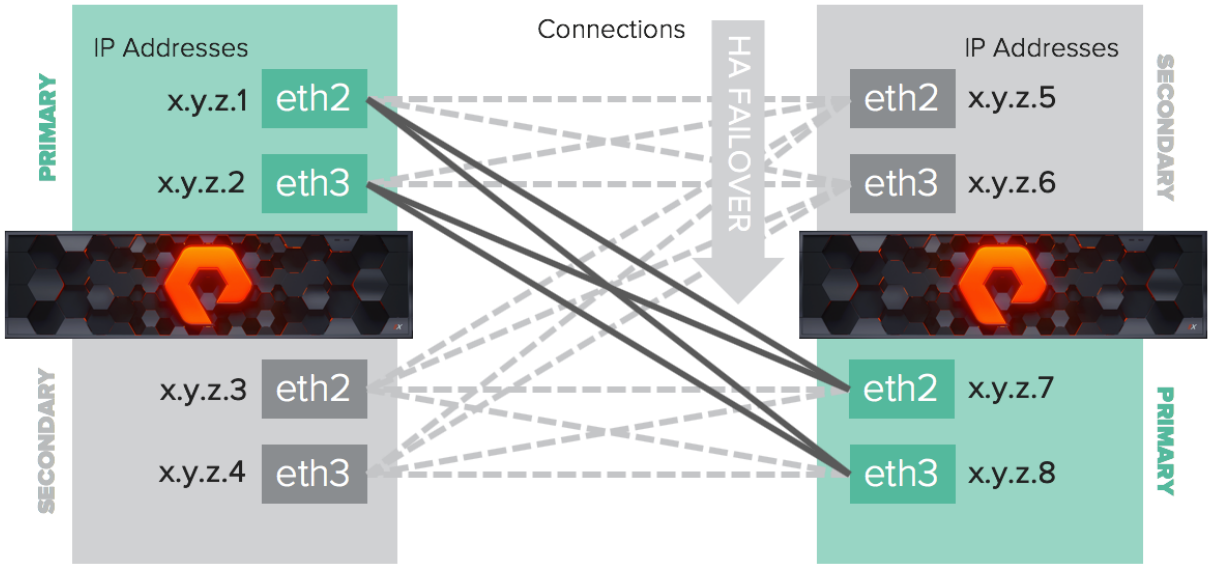


Figure 6 - HA Failover of physical replication ports

In order to achieve a fully redundant design each controller must have its replication ports attached to different redundant switches on the same network as shown in Figure 7.

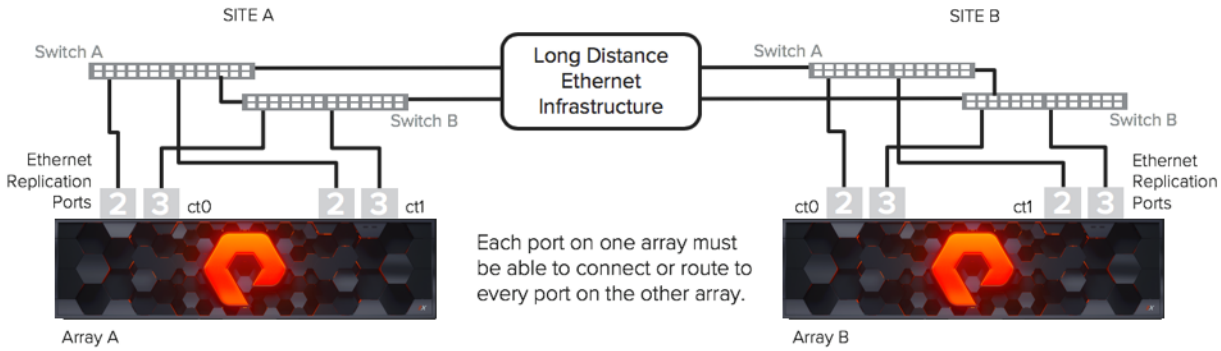


Figure 7 - Redundant Switched Network For Each Controller

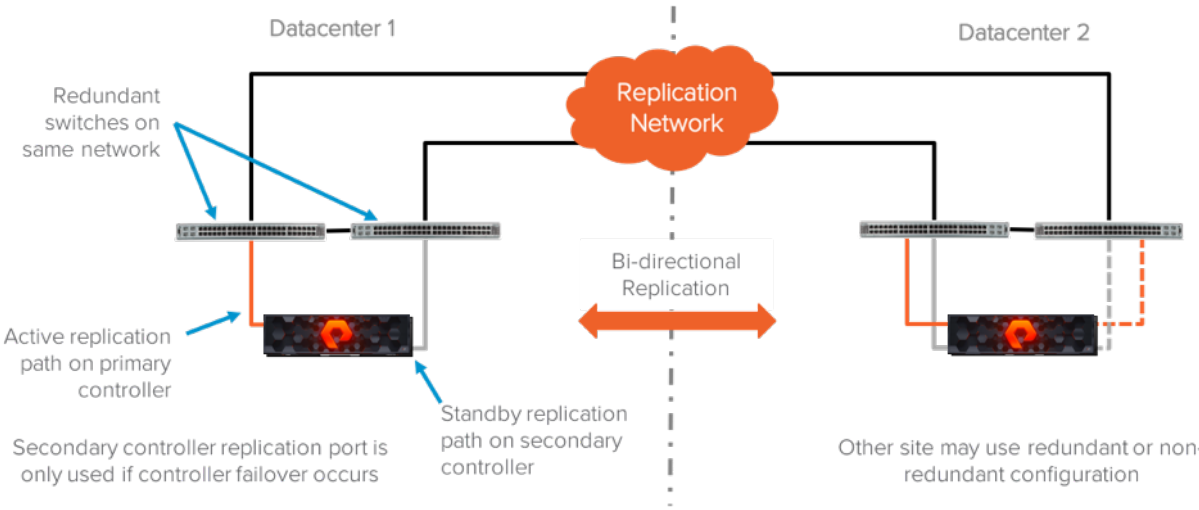
Note: Directly connecting the source and target storage arrays is not supported as this may prevent the primary controller in one storage array from being able to access the primary controller in the other storage array as the role of primary can dynamically move from one controller to the other.

Non-Redundant Replication Port Network Design

All available FlashArray platforms provide options for redundant replication ports, however in some situations it may not be possible to connect each controller redundantly. The non-redundant connection configuration should only be used temporarily and only if absolutely necessary to enable short-term connectivity between FA (such as during a PoC). The following image describes a non-redundant network configuration. In this configuration each controller has only one port connected for replication.

Note: Physical replication ports will alert if they are not both enabled on every controller and/or unable to connect to the peer FlashArray.

In the image below the orange connection represents the only replication port on the primary controller. The grey connection represents the standby connection on the secondary controller, which will only be used in the case of a controller failover when that controller becomes primary.



Note: The grey replication port only becomes active on HA failover.

Port Failover

Redundant configuration for replication ports requires use of two physical replication ports per controller (4 total per FlashArray). The logical network topology shown earlier in Figure 5, is a series of connections between all replication ports on each FlashArray. There are two active ports on the primary controller on each FA. If one of the active ports fail, or if a switch that the active ports is connected to fails in such a way to cause loss of link on the port, the other port remains active and carries on replication with the bandwidth of the remaining active port.

Controller Failover

Replication traffic only occurs on the primary controller, as the active replication interfaces exist only on the primary controller. When a failover occurs, the replication interfaces are automatically activated on the other controller when it becomes primary during a failover. Whatever ports were configured for replication on the primary controller, those same ports must exist on the secondary controller and be configured as replication interfaces to allow the secondary controller to become primary and to takeover replication. For example, if controller0 is using ports eth2 and eth3 as replication interfaces, then controller1 must have ports eth2 and eth3 configured for replication in order for replication updates to be able to continue after a failover.

CONFIGURING REPLICATION

Connect Arrays

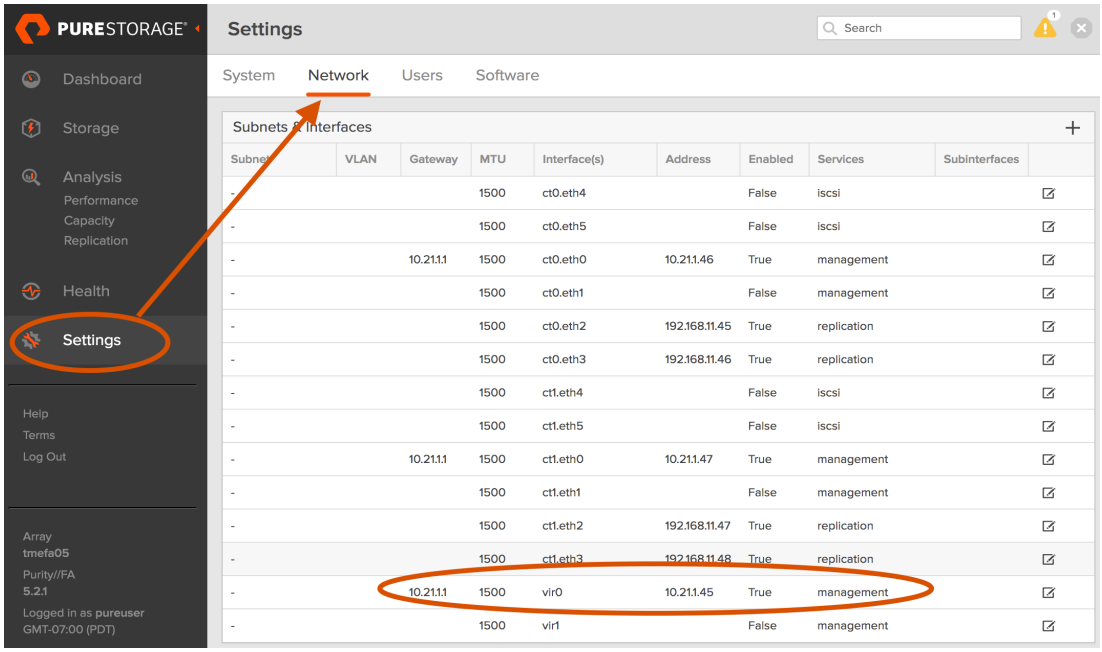
To configure Asynchronous replication, you will first need to connect the source and target arrays. Connecting arrays is a one-time process for each pair of peer arrays. For information about the maximum number of arrays that can be connected for replication see the Purity Replication Requirements and Interoperability Matrix KB article here:

https://support.purestorage.com/FlashArray/PurityOE/Asynchronous/Reference/Purity_Replication_Requirements_and_Interoperability_Matrix.

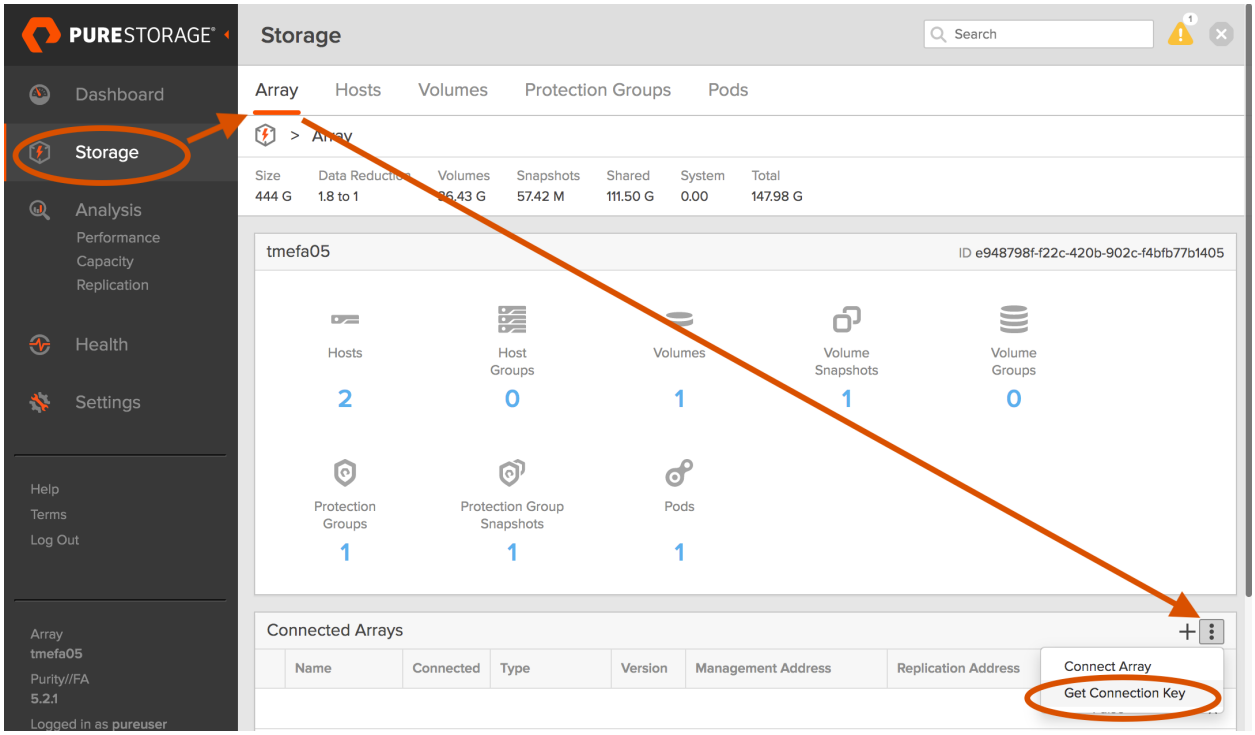
Arrays can be connected from either the source or target. Regardless of which array you use to create the connection you must make a note of the other array's management IP or fully qualified domain name (fqdn) and get a connection key from the array you are connecting to.

1. In this example, we will be performing the connection from a source FlashArray (**tmefa04**) to a target FlashArray (**tmefa05**). This means we will need to provide the management IP address (or fqdn) and obtain a connection key from the target FlashArray (**tmefa05)** array.

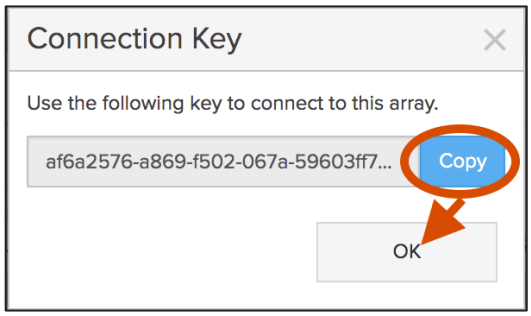
   On the target FlashArray (**tmefa05**) GUI click **Settings** and then select the **Network** tab. Make note of the IP address of the management port vir0.



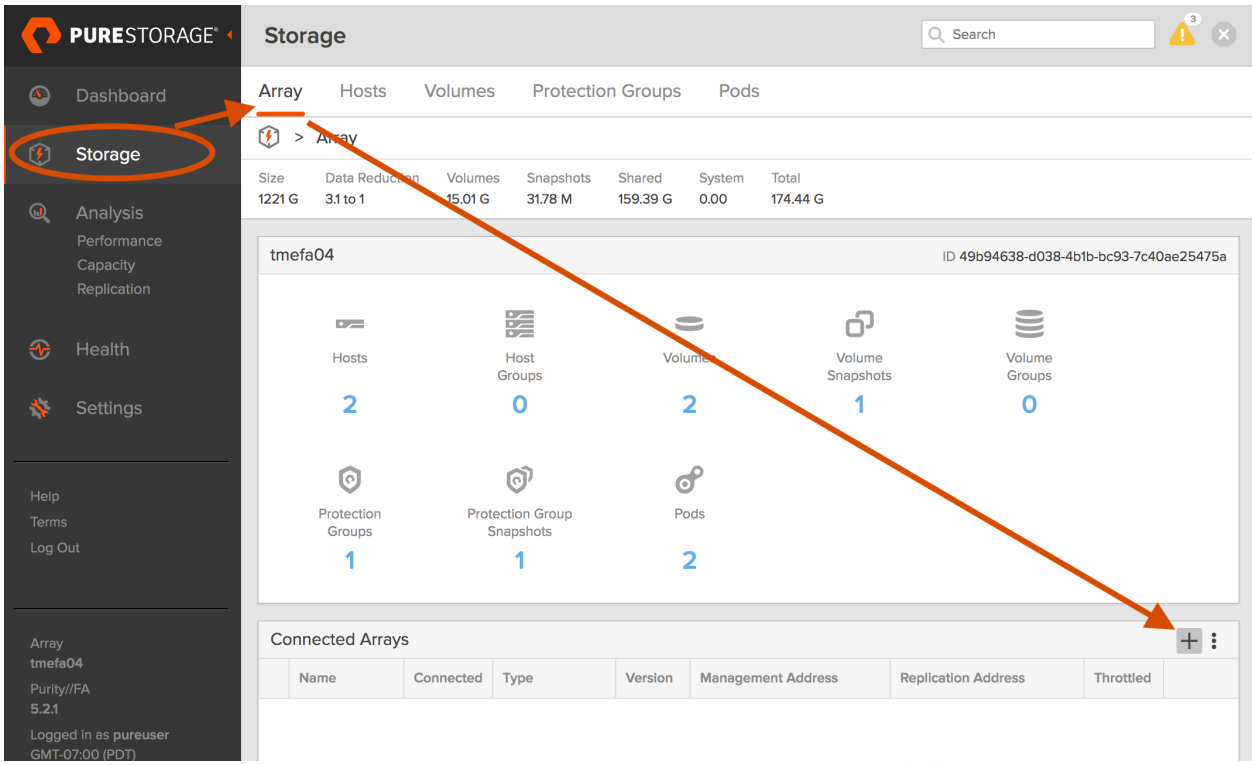
2. On the target FlashArray (**tmefa05**) click **Storage** then select the **Array** tab. Click the configuration button at the right corner of the **Connected Arrays** window and select **Get Connection Key**.



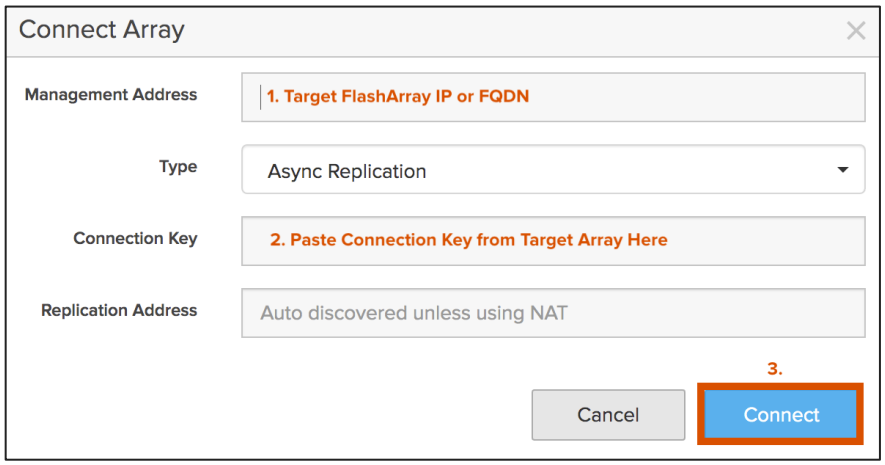
3. Select the displayed connection key and click **Copy**. Click the OK button to dismiss the Connection Key window.



4. On the source FlashArray (tmefa04) click **Storage** then select the **Array** tab. Click the **+** button in the right corner of the Connected Arrays panel. A **Connect Array** dialog window will open.

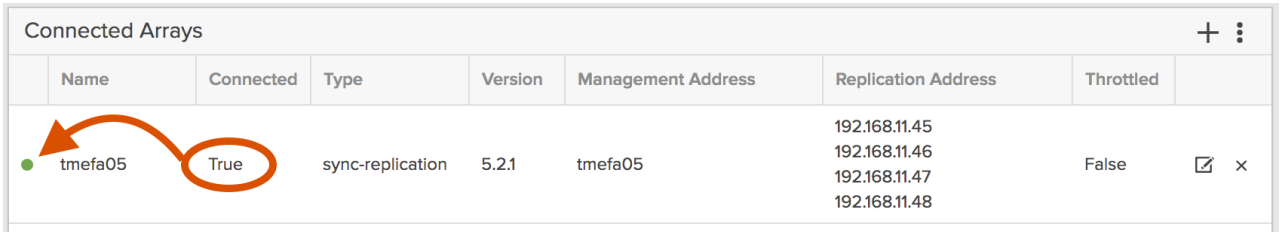


5. Enter the management IP address or fqdn of the target FlashArray (**tmefa05**) into the **Management Address** field. Leave the **Type** field set at Async Replication and paste the connection key from the target FlashArray into the **Connection Key** field. Finally, click **Connect**.



Important To Note	The management and/or replication IP addresses of the connected FlashArray may be changed. If the IP addresses are to be changed simply delete the array connection and re-create it with the new address. Existing replication relationships will not have to be recreated or re-replicated however replication will be interrupted while the old addresses are still configured.

After a few moments the target FlashArray will show up in the Connected Arrays panel. Ensure that the status under the **Connected** column is **True**. There will be a green dot indicating a healthy connection status as well.



Important To Note	If the connected status is not **True** then the arrays are not yet connected and replication will not occur. There may be a communication issue on the network preventing the arrays from connecting. You can verify connectivity of the replication network using the purenetwork ping command in the CLI. pureuser@array1> purenetwork ping --interface <source replication IP> <target replication IP> Using the --interface option on the ping command ensures the ping command uses the proper network interface to ping the other array. If there is no networking connectivity review the requirements section above and verify the replication network has been properly configured. The array monitors the replication network and will set the connected status green when there is connectivity. While at the CLI you can verify the arrays are connected with the purearray command. pureuser@array1> purearray list --connect Name ID Version Management Address Replication Address Connected Type array2 5b687... 99.9.9 10.21.16.63 10.21.16.65 True replication

6. Repeat the above steps to connect any additional arrays.

Once two arrays are connected, replication can be performed in either direction between those two arrays— there is no need to connect in the other direction.

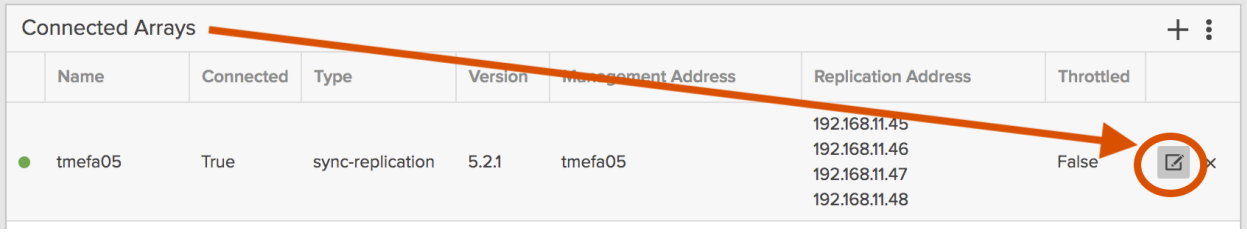
Important Note	Replication occurs through the replication ports on the primary controller only. The ports reserved for replication on the secondary controller will not be used for replication unless the secondary controller becomes primary in the event of a HA failover.

Configure a Replication Connection Throttle

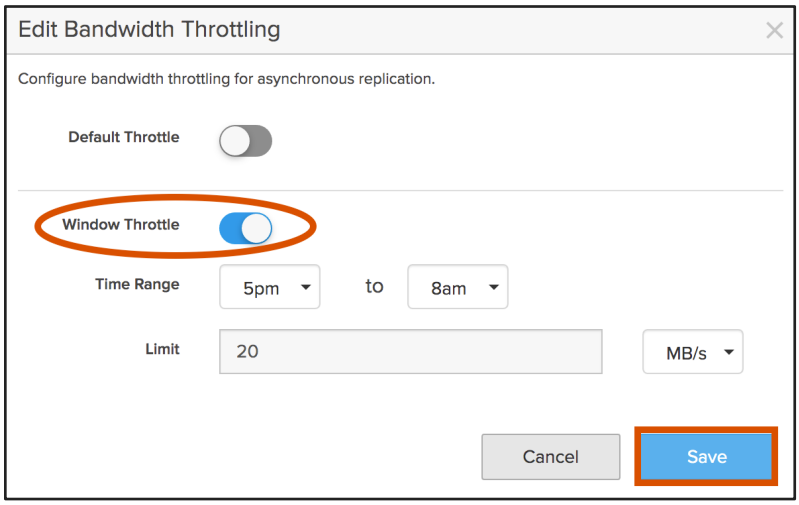
You may optionally configure an asynchronous replication throttle per connection and per direction between each pair of connected arrays. The throttle limits the amount of bandwidth that is used between two FlashArrays for asynchronous replication transfers. A throttle is configured on the source array of the direction in which you want to throttle the replication transfers. A different throttle may be configured in each direction.

Purity supports two replication throttles, a default throttle and a window throttle. The default throttle applies all the time. The window throttle applies only during the specified window. Both types of throttles may be applied to the same connection at the same time.

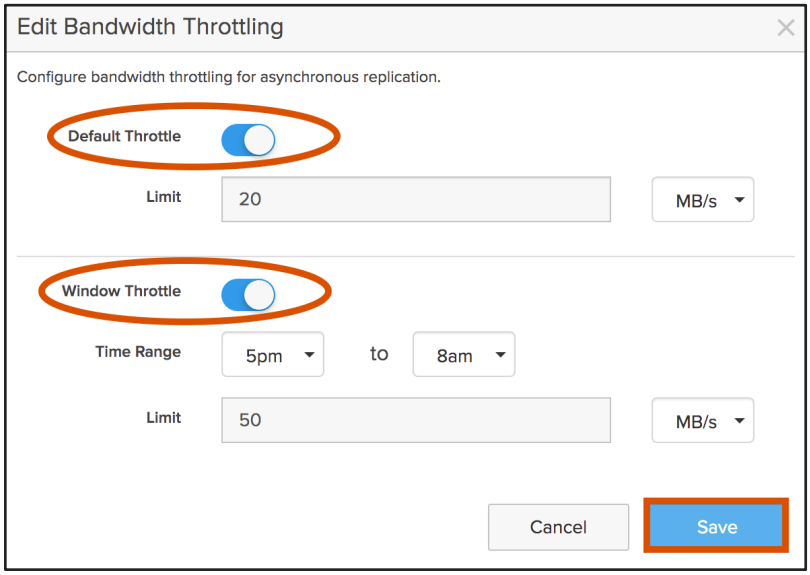
The throttle is configured in the Connected Arrays connections panel.



The following image is an example of applying a 20MB/s throttle that is in effect only between 5pm and 8am.



The following image is an example of using both throttles together. The default throttle limits bandwidth use to 20MB/s, except between the hours of 5pm and 8am, where replication is allowed to use 50MB/s of bandwidth.



Create a Protection Group

A protection group (a.k.a. **pgroup**) defines a set of volumes, hosts, or host groups, which will be replicated together with point-in-time consistency across the member volumes. Replication in Purity is snapshot based, so each time replication occurs a pgroup snapshot is created. Each pgroup snapshot consists of one snapshot per volume for each of the volumes contained within that pgroup. All the volumes in the pgroup will have a snapshot created simultaneously that is point-in-time consistent with all of the member volumes in that pgroup.

In Figure 8 below, Protection Group 1 contains three volumes, vol1, vol2, and vol3. Each time a replication update occurs a new pgroup snapshot is created containing a snapshot for each volume.

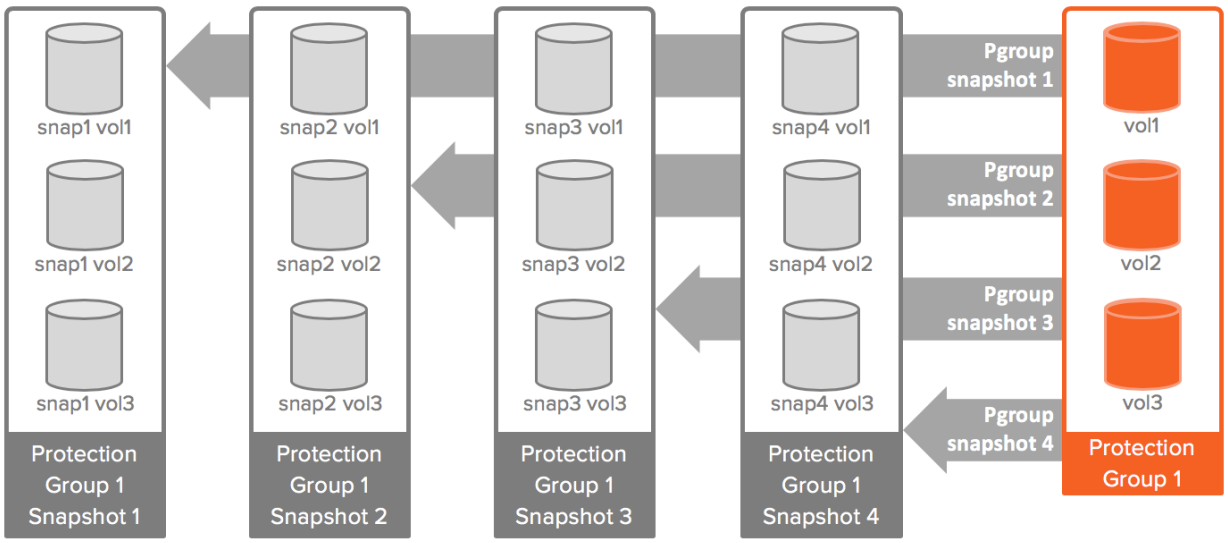
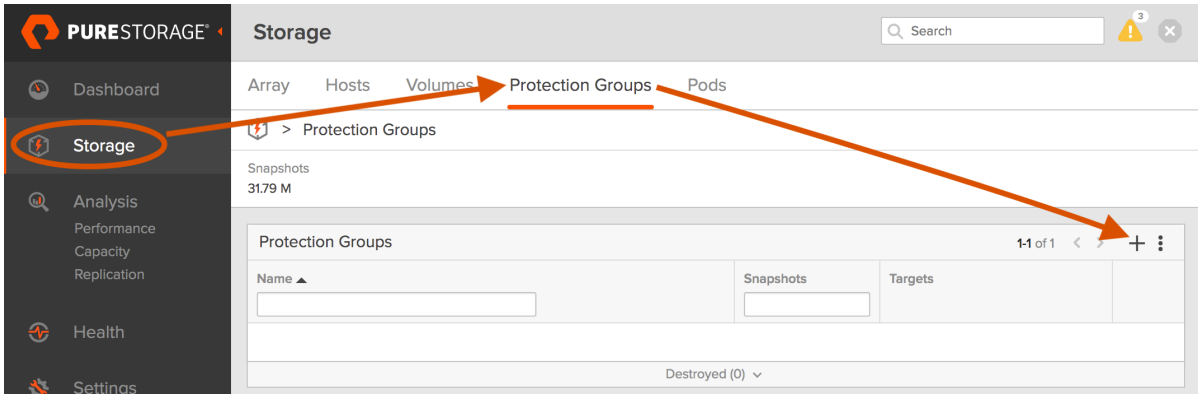


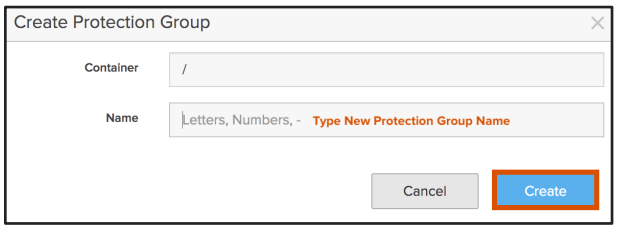
Figure 8 - Protection Group Snapshots

A protection group must be created on the source array.

7. On the source FlashArray (**tmefa04**) click **Storage** and then select the **Protection Groups** tab then click the **+** sign at the right corner of the **Protection Groups** panel.



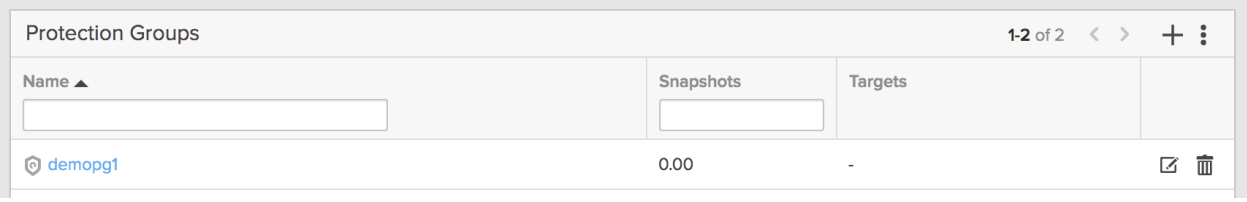
8. Enter a name for the Protection Group and click **Create**.



Best Practices – Protection Group Naming Conventions

- It's a good idea to use a naming convention for pgroups that helps identify them. For example, if the pgroup is being created for a given host or host group use a naming convention such as hostname_pg1 or clustername_pg1 to make it easily identifiable as a pgroup containing volumes attached to that host or cluster's host group.
- If you want to replicate the same volumes to different target FlashArrays but using a different schedule or retention policy per target array, then you must create a different protection group for each target FlashArray that requires a different schedule or retention. Using a pgroup naming convention that helps to identify the target array in the name of the pgroup can make it easier to differentiate the pgroups in the user interface.

The new Protection Group will be listed in the **Protection Groups** panel at the source FlashArray. For example, a newly created Protection Group named **demopg1** would be shown as:



Note: After a replication target is set (Step 10 below) for a Protection Group, the listing will show pgroups that replicate *from* this FlashArray *to* other FlashArray(s) as **'Allowed on 1 of # replication targets'** under the **Targets** column. Protection Groups listed with a **Targets** column showing '**Allowed on this array'** are pgroups that replicate *to* this FlashArray *from* the other FlashArray(s).

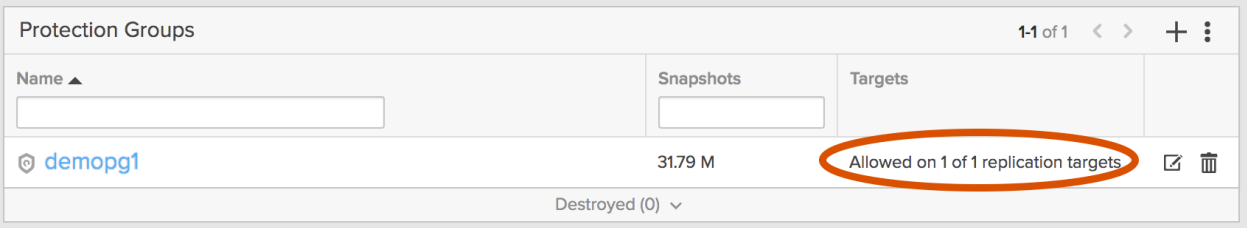


Figure 9 – Protection Group with a Target FlashArray defined.

Configure the Replication Schedule

The replication schedule determines the frequency for creating Protection Group snapshots on the source and then replicating them to the target array.

Purity allows two separate protection policies for a single protection group:

- **A snapshot schedule** - for snapshots to be created locally and retained only on the source array.
- **A replication schedule** - for a snapshot to be taken on the source, replicated to the target, and then retained on the target. Each schedule creates different snapshots.

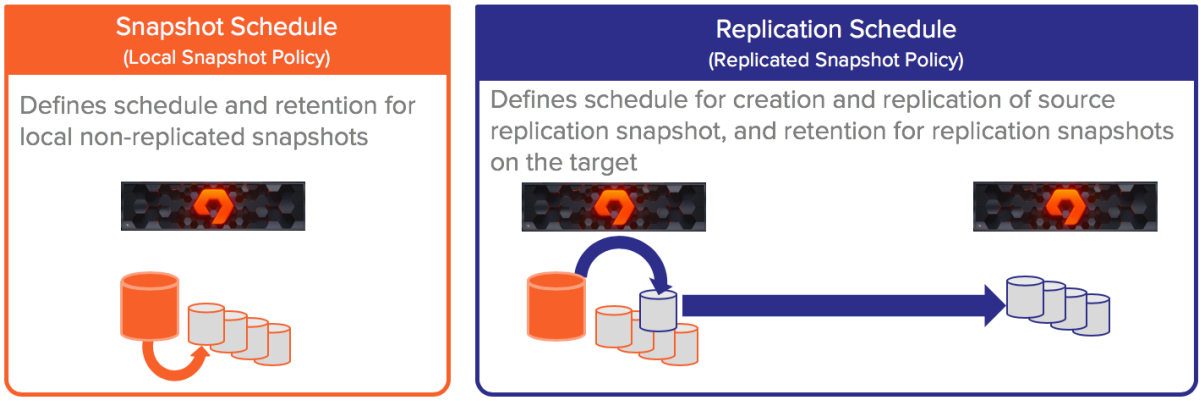


Figure 10 - Purity Protection Policies

Important Note	Local snapshots created by the snapshot schedule are intended for purposes such as creating local dev/test instances on the source, or for retaining local recovery points on the source. You **do not** need to configure the local snapshot schedule if you only wish to do remote replication, the replication schedule will create its own snapshots as needed.

Snapshot retention determines how many replication snapshots are maintained on the target. The retention policy is runs continuously and deletes any snapshots that exceed the configure retention policy. Scheduled replication will maintain at least one source and target snapshot after

the first replication cycle regardless of the retention policy to maintain efficient incremental transfers.

The example in Figure 11 (below) shows how many snapshots would exist on the replication target if replication were configured to occur every hour, keeping every hourly snapshot for 4 hours, and then keeping 2 snapshots per day for 5 more days.

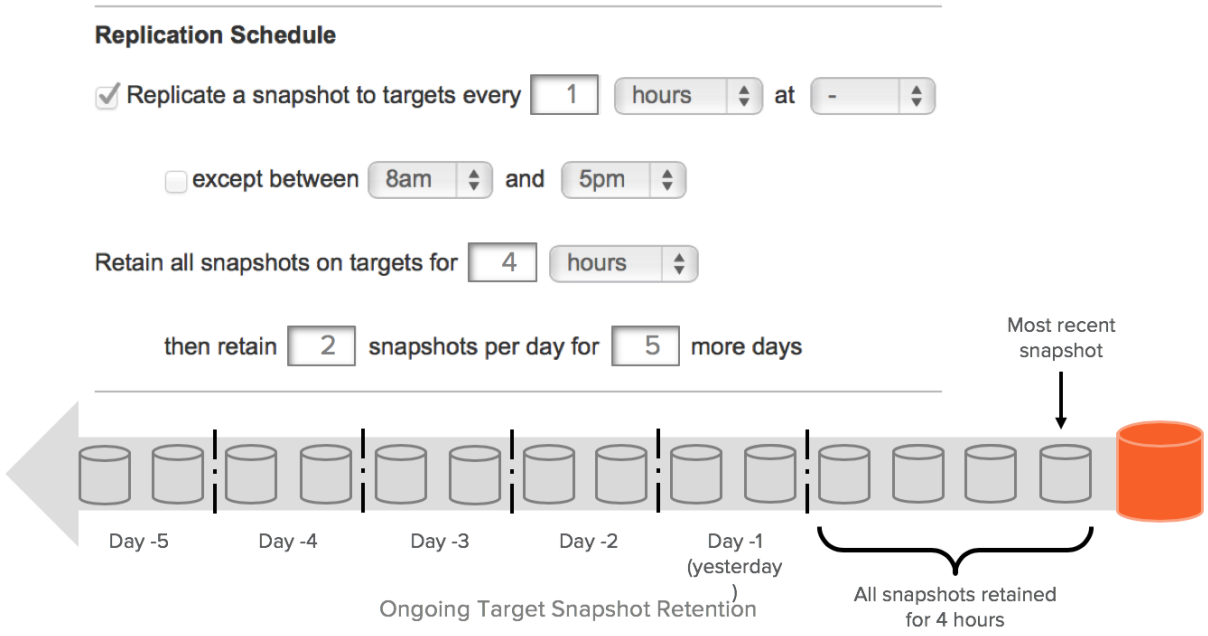
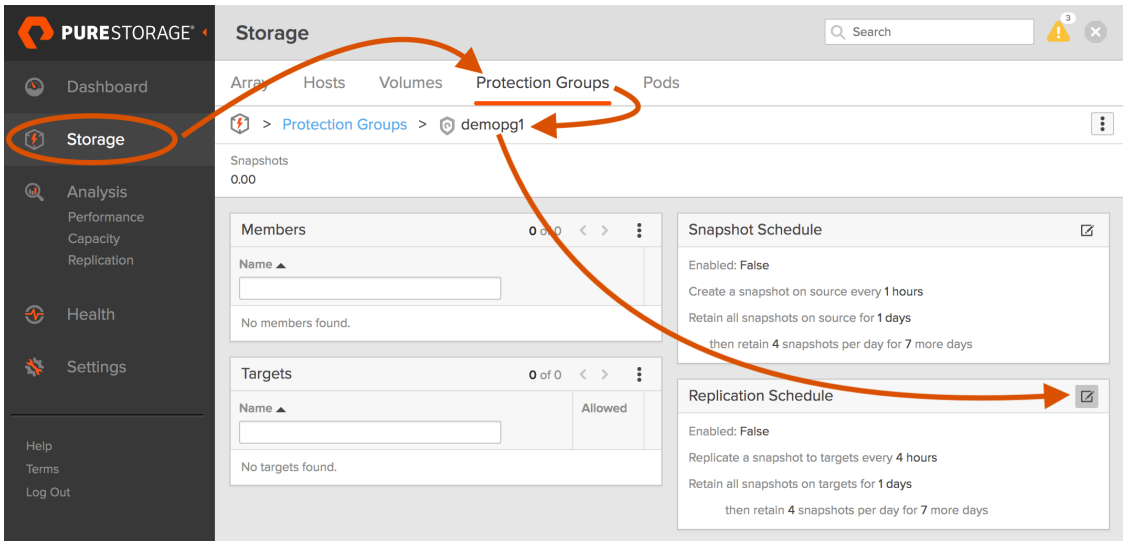


Figure 11 - Snapshot Retention

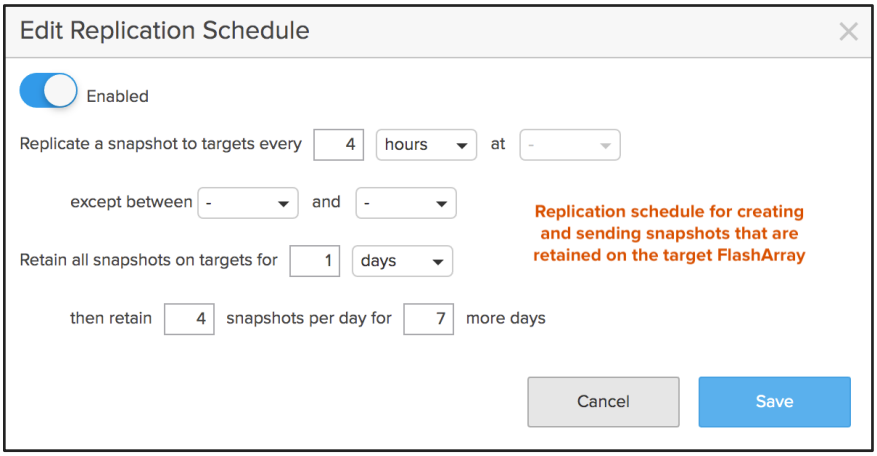
Initializing, or seeding (the first baseline copy) happens automatically as the first transfer. There is no special command required for replication seeding.

Important Note	Replication schedules and retention can only be configured on the source FlashArray.

9. Click **Storage** > **Protection Groups** and then click on the newly created protection group (i.e. **demopg1**). Next click the edit icon at the right corner of **Replication Schedule** panel within the Protection Group.



The Edit Replication Schedule dialog window will open.



Configure the Protection Group replication schedule and target snapshot retention policy.

The schedule must also be toggled from Disabled to **Enabled** in order for replication to take place. By enabling the schedule, the first transfer will begin as soon as you define a replication target (done in a later step). If you prefer to start replication at a later time, then leave the Replication Schedule **Disabled** until you wish to start replication.

Click **Save** when the schedule is set to the desired values.

Note: If you would like to schedule a blackout window to prevent replication transfers from starting during certain times, use the box next to the "except between" item and set the desired start and end times for the blackout window.

Important Note	Blackout windows only prevent new replication transfers from starting, in-progress transfers that started before the blackout window begins will continue into the blackout window until they finish.

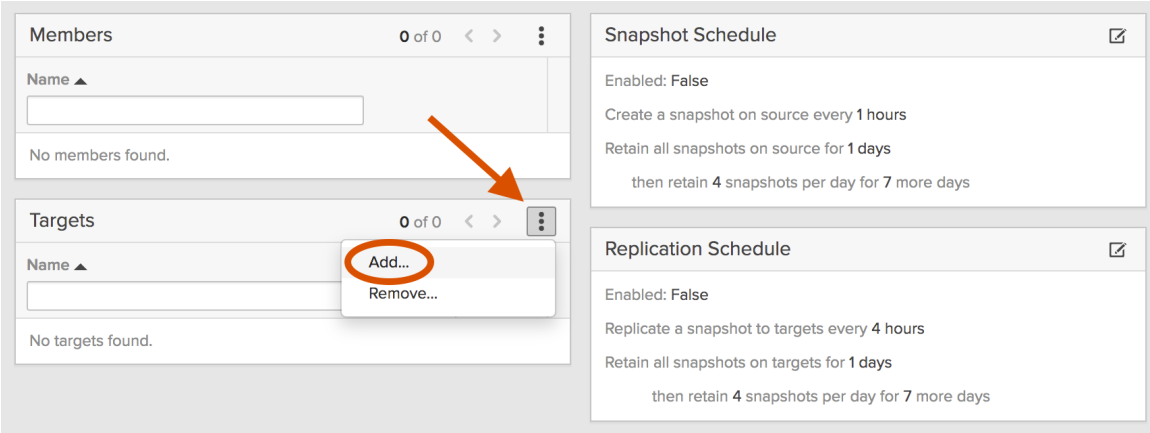
Best Practices – Replication & Snapshot Schedule Configuration

- The first transfer will be a baseline transfer of the entire volume. The replication schedule should initially be set to replicate once every 24 hours, then adjusted incrementally down to the desired frequency after smaller incremental transfers are occurring.
- In cases where the baseline transfer takes longer than the configured replication frequency, once the baseline is finished the replication schedule will continue creating snapshots and replicating them in an effort to create the number of snapshots it is configured to retain in the replication policy. To prevent this, when establishing replication configure the schedule to replicate infrequently until the baseline has been established, then increase the frequency to the desired interval after the baseline and any initial large increment transfers have been completed.
- The replication frequency should be configured such that no more than half of the replication interval is used to send data. This allows background processes on the target array to finish before the next transfer is started.

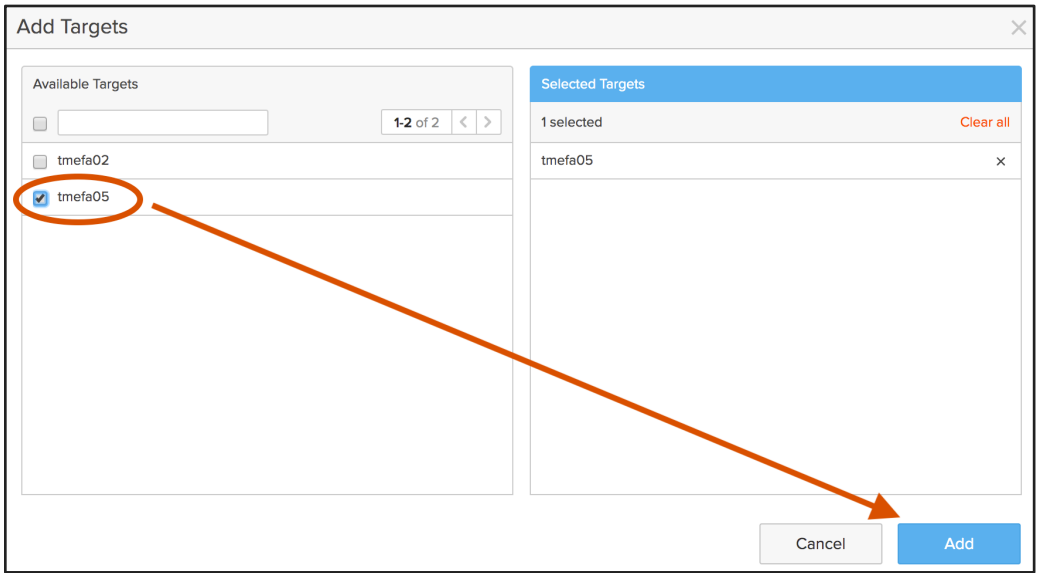
Select Replication Target Arrays

Target FlashArrays are the destination arrays that a Protection Group will be replicated to. A single FlashArray can have multiple target arrays, while simultaneously being the target for other arrays. When a Protection Group is replicated it will be replicated to all the targets configured in that Protection Group simultaneously.

10. On the source FlashArray within the individual Protection Group click on the setup button on the right of the **Targets** panel and select **Add**.

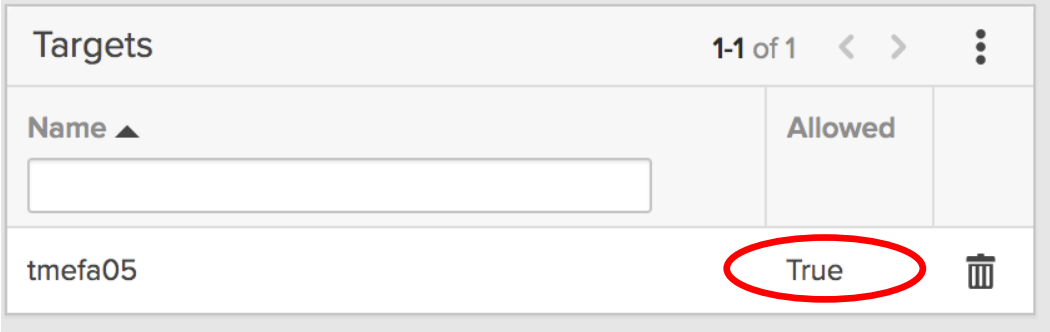


11. The **Add Targets** dialog box will open and list the available targets connected to this array. Put a check in the checkbox next to the desired targets for this Protection Group, then click **Add**.



Important Note	Each pgroup can have only one replication schedule. If you want to replicate the same volumes to different targets, but with different schedules or with a different retention per target, then you must make a different pgroup for each target. A single volume, host, or host group can be a member of multiple pgroups at the same time.

12. The target FlashArray (**tmefa05**) will be shown in the list of targets for this pgroup. You should see the Allowed column the status set to **True**.



Important Note	If Allowed is **False** or if there are no arrays listed in the Targets panel then **replication will not take place**. The allowed status determined automatically by FA Purity based on replication limits and connection health criteria.

Add Protection Group Members

Protection group members define which volumes get replicated. Asynchronous allows the storage administrator to configure protection of all the LUNs provisioned to a host or group of hosts. This eliminates unnecessary configuration steps, reduces the chance of missing some of a host's LUNs when configuring protection, and removes the risk of forgetting to protect new LUNs added to a host after replication was already configured.

Protection Group members can be individual volumes, hosts, or host groups. Only one type of member is allowed per pgroup. Whether or not individual volumes, hosts, or host groups should be used as members in pgroups depends on which features, or capabilities are important in your environment. Configuring replication on a per host or per host group basis reduces the number of replication relationships that need to be managed and ensures that new volumes connected to those hosts will be automatically replicated according to that pgroup's existing configuration. Managing replication of pgroups containing hosts or host groups provides simpler replication management, while replication of pgroups containing volumes provides more granularity of management as shown in

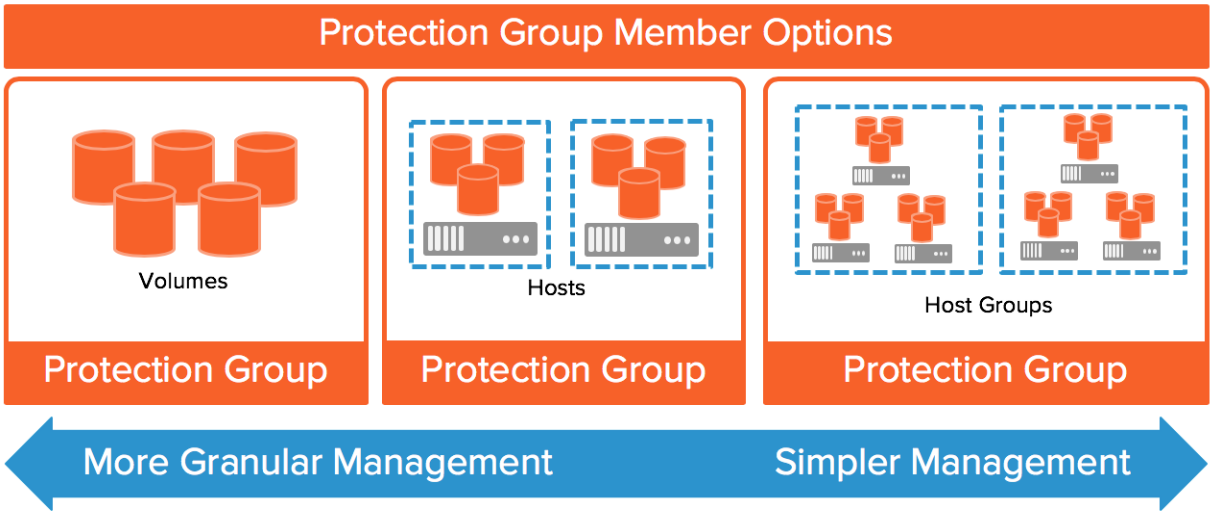


Figure 12.

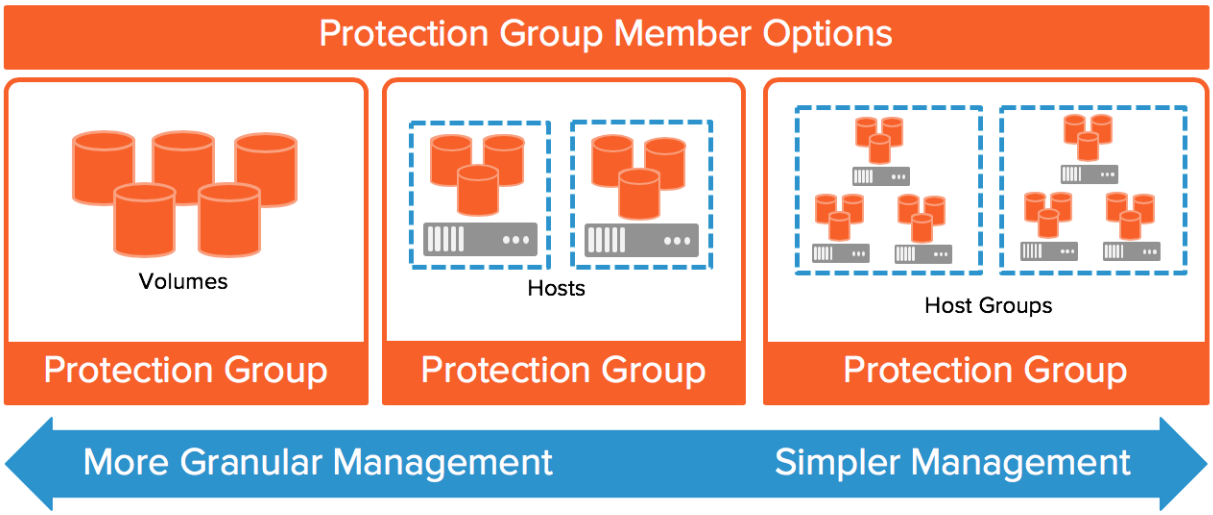


Figure 12 - Protection Group Member Options

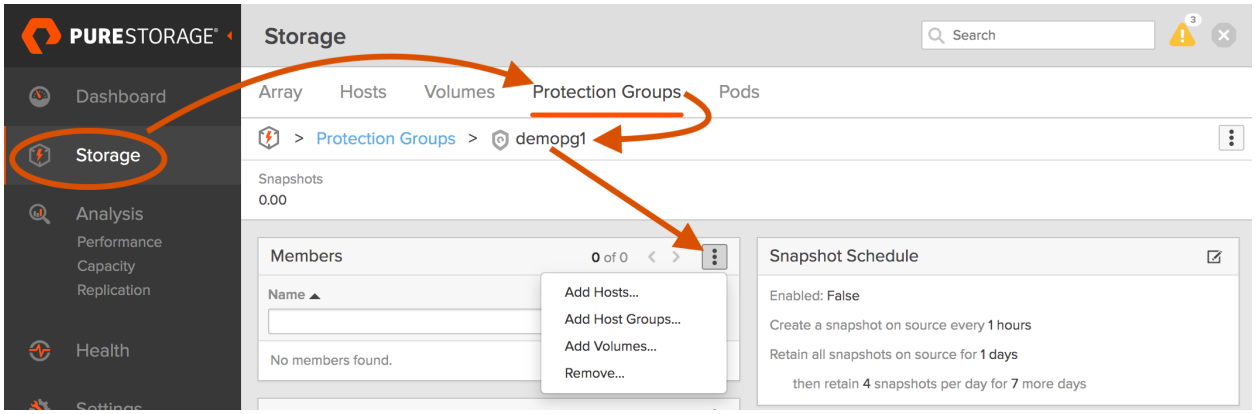
Adding a host or host group configures all volumes connected to those hosts to be replicated with the pgroup, this includes volumes connected directly to any hosts if host groups are added to a pgroup. In the future any new volumes connected to any of those hosts will automatically begin replicating on the next replication interval.

Best Practices – Protection Group Members

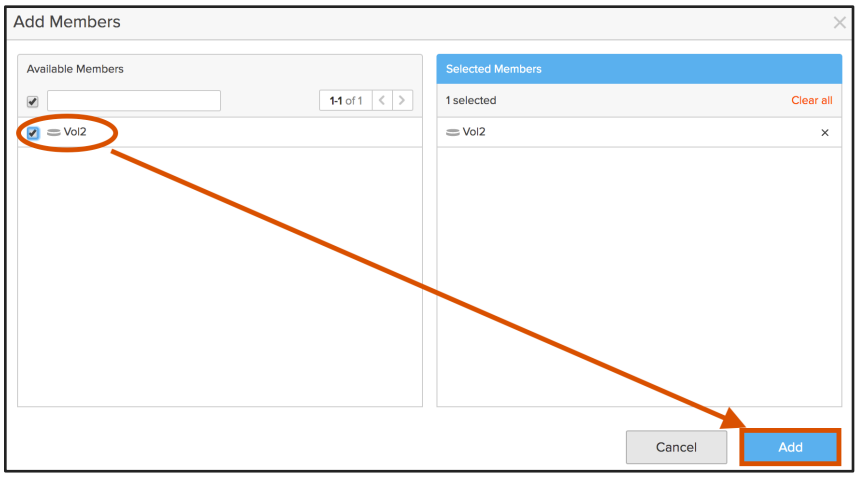
- Volumes that must be replicated with point-in-time consistency across the volumes, such as multiple volumes belonging to one database instance or applications that require consistent data is maintained between different LUNs, must be added to the same pgroup. This ensures that the base snapshot created for replication captures a snapshot that is consistent with respect to the order of writes across those member volumes.
- In environments where new volumes that must be replicated are created frequently, using hosts or host groups as pgroup members can simplify replication management. Any new volumes attached to hosts or host groups will automatically begin replicating at the next replication interval. Note however that if a large amount of data exists in the newly attached volume this could extend the replication time of the next transfer of that pgroup.
- For stand-alone application servers where all volumes need to be replicated together for consistency, add the host to the pgroup.
- For clustered servers where all volumes need to be replicated together for consistency, create a host group containing the nodes of the cluster. Add the host group to a pgroup.

In this configuration example we will simply add an individual volume to this pgroup.

13. On the source FlashArray within the Storage > Protection Group > [desired Protection Group] Members click the setup button on the right and select the type of member you want to add.

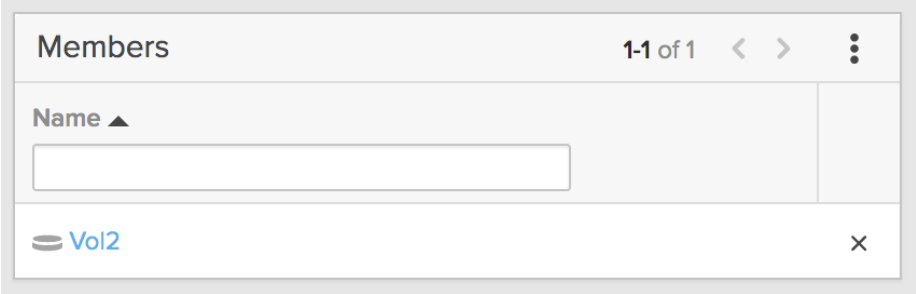


14. Click the checkbox next to each member to be added to this pgroup and click **Add.** You can add multiple members at the same time, however a pgroup can only contain one type of member at the same time.



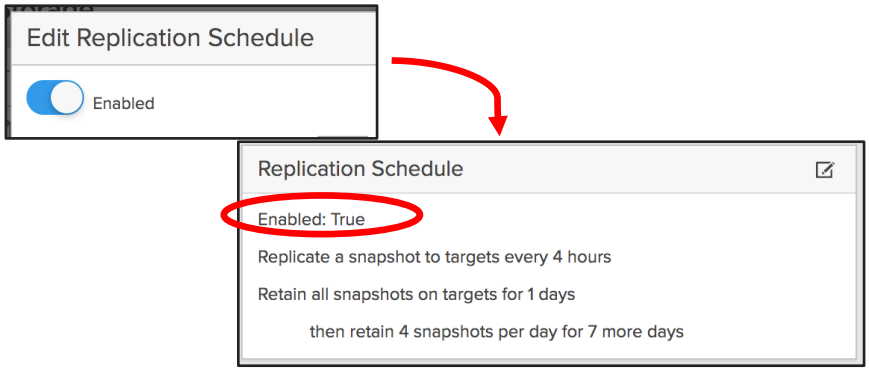
A Protection Group can contain multiple volumes, multiple hosts, or multiple host groups at the same time, but not volumes, hosts, and host groups together. Here is an example of adding volumes:

15. The **Members** panel of the Protection Group will show which members have been added.



Important Note	The replication process checks every 30 seconds to see if any pgroups need to be replicated according to their configured schedules. If you configured and enabled a replication schedule earlier, then the baseline transfer is performed as the first transfer. This will start within 30 seconds of the pgroup being allowed to replicate on the target, unless there is currently a blackout window in effect. If you do not want the transfer to start at this time, then wait to allow replication later or return to the Schedules tab of this pgroup on the source array and disable the replication schedule.

16. If you didn't Enable the Protection Group Replication Schedule initially, edit and toggle it to Enabled.



This will enable replication of the Protection Group to the target FlashArray.

MANAGING ASYNCHRONOUS REPLICATION

The following replication management topics are discussed in this section:

- Monitoring replication transfers
- Stopping replication
- Performing on-demand replication transfers
- Performing failover
- Reversing and resynchronizing replication

MONITORING REPLICATION TRANSFERS

Replication throughput for in-progress transfers can be reviewed on the Analysis tab in the UI. Replication status for in-progress transfers is tracked on the replication target. To review the replication on the target FlashArray click **Storage** and then select the **Protection Groups**. Next select **Transfers** in the **Protection Group Snapshots** panel to view the in-progress snapshot transfers as well as historical transfers.

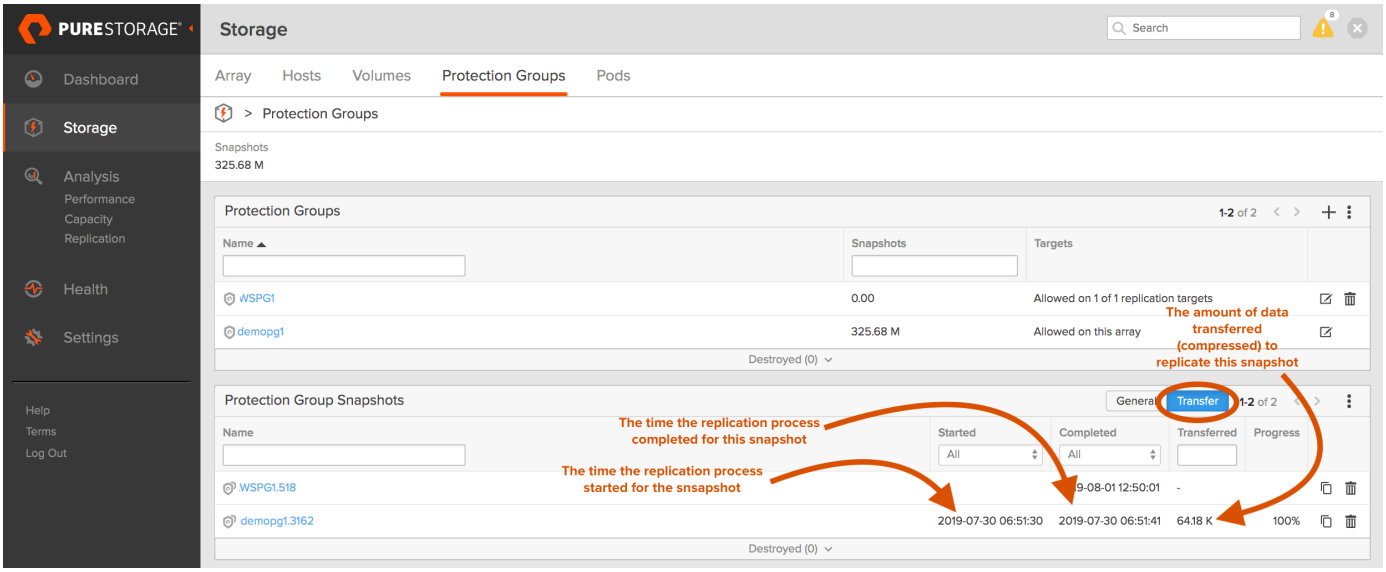


Figure 13 - Replication Start and Completion Times

As shown in Figure 13 the Protection Group Snapshots panel on the target array shows the time the replicated snapshot was created, the time the transfer started, and the time the transfer completed. If a transfer is currently in-progress or incomplete, the Completed column will not show a value.

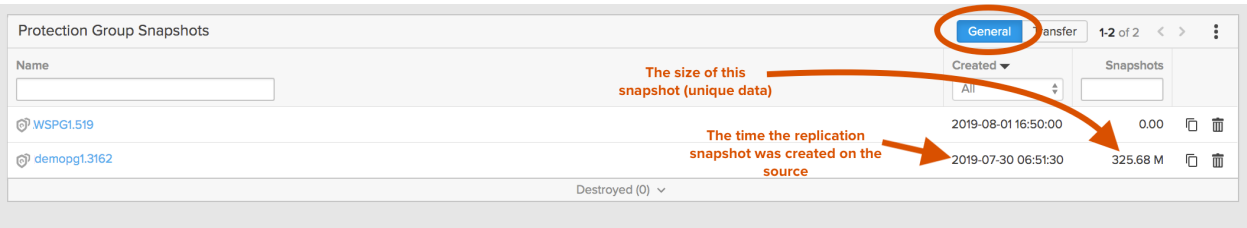


Figure 14 - Replicated Capacity Reporting

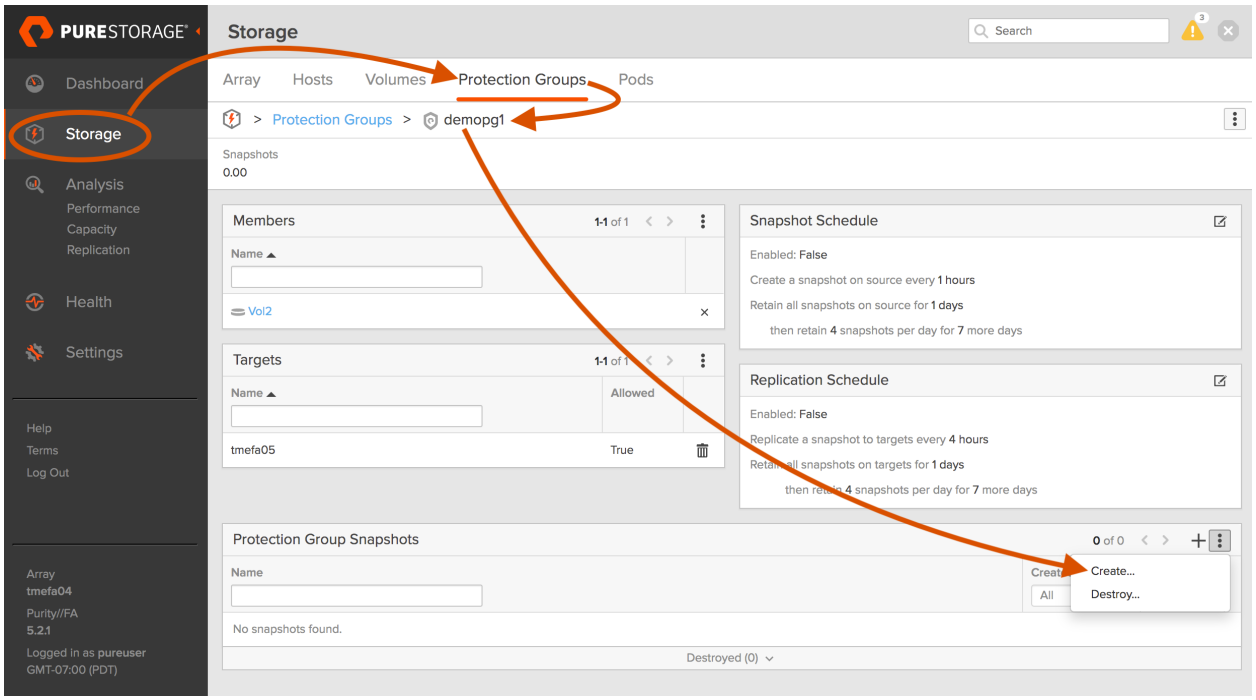
The Protection Group Snapshots panel also shows historical data for the previously replicated and retained pgroup snapshots. As shown in Figure 13, the Transferred column shows amount of physical compressed capacity that was sent over the network into the target array to complete that transfer. The Transferred value is after compression on the wire and target array.
Other snapshots listed show the capacity of each individual pgroup snapshot's unique data in the Snapshots column. This value does not represent the total amount of data replicated to build this snapshot as this does not report data that is shared with other snapshots or volumes on the array, nor does this value include new data added between snapshots. This is the amount of unique data owned only by this snapshot - the amount of capacity returned if the pgroup snapshot were to be deleted.

There is no destination volume object on the target array in which to report the unique capacity of the replicated volume. Instead, the capacity of the Protection Group's unique data is added to the amount shown in the Snapshots column of the latest completed snapshot (shown in Figure 14).

PERFORMING ON-DEMAND REPLICATION TRANSFERS
You may need to perform a replication transfer manually or on-demand for various reasons such as ensuring new changes are replicated to the other array before creating a clone there for a DR test.

Performing an on-demand transfer must be done from the source FlashArray.

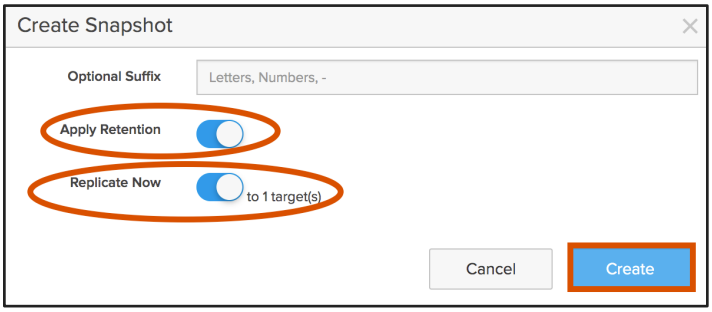
1. On the source FlashArray click **Storage** and then select the **Protection Group** tab then the specific Protection Group you want to create a snapshot of. Finally, click on the setup icon at the top right side of the **Protection Group Snapshots** panel and select **Create**.



2. A **Create Snapshot** dialog box will open. In order to replicate this snapshot to the target you must toggle the **Replicate Now** switch on. Toggle the **Apply Retention** switch to make sure this snapshot is removed from the **source** and **target** according to the retention settings configured for this pgroup.

Important Note	If you do not turn on the **Apply Retention** switch manually created snapshots will remain on the source and target arrays until they are manually deleted.

You may add a suffix that will replace the numerical suffix that the system would have otherwise added. This is useful for identifying this snapshot at the target array if you intend to use it for a DR test. Click the **Create** button to create the snapshot.



Using the Replicate Now switch will cause this transfer to begin now, even if another snapshot transfer for this pgroup is already in progress. The CLI offers an additional option for on-demand snapshot replication. Using the purepgroup snap command with the --replicate option tells the array to replicate this snapshot after any transfer currently in-progress is completed, rather than while any current transfer is in progress.

```
pureuser@array1> purepgroup snap --replicate <pgroup name>
```

STOPPING REPLICATION

You may stop replication by disabling the schedule on the source FlashArray, disallowing the pgroup on the target, or removing members from the Protection Group. Each of these methods will have a different effect on the system as explained below.

Stopping Replication By Disabling The Replication Schedule

This method will continue an in-progress transfer until it completes but stop upcoming transfers from being performed.

Important Note	The snapshot retention policy remains in effect while the replication schedule is disabled. Snapshots will continue to be deleted from the target according to the retention policy configured in the replication schedule.

To disable the replication schedule, select the specific Protection Group from the **Storage** > **Protection Group** tab. In the **Replication Schedules** panel click the **Edit** icon at the top right and toggle the **Disabled** switch within the Edit Replication Schedule dialog. Finally, click on the **Save** button to apply the change.

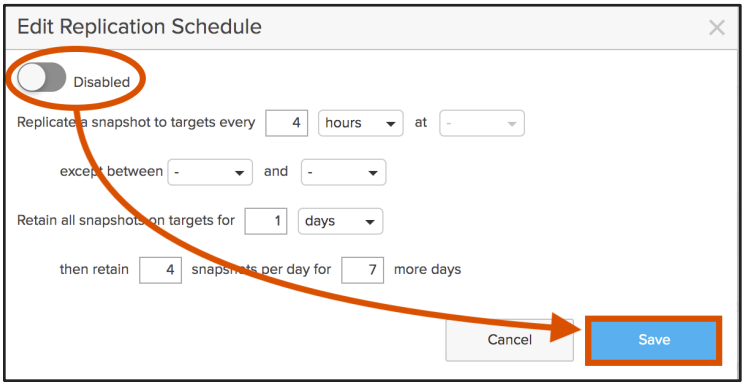


Figure 15 - Disabling the Replication Schedule

Stopping Replication By Disallowing The Protection Group At The Target

This method will stop an in-progress transfer. It may take a few minutes before data stops transferring on the network, as the system will attempt to finish any currently in progress replication segments. Snapshots are internally divided into segments to be transferred.

Important Note	The snapshot retention policy remains in effect while replication is disallowed for the pgroup. Snapshots will continue to be deleted from the target according to the retention policy configured in the replication schedule.

To disallow replication at target FlashArray, select the **Storage** > **Protection Group** tab, click the settings button on the right of the desired Protection Group within the **Protection Groups panel**.

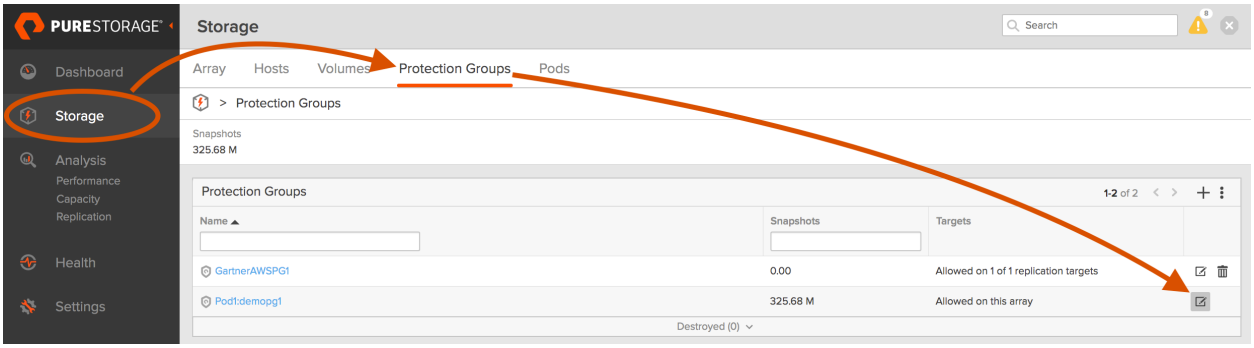
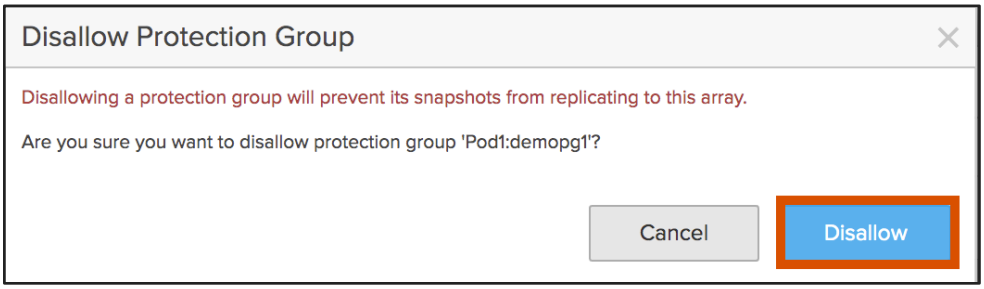


Figure 16 - Disallowing Protection Group Replication

Click **Disallow**:



Stopping Replication By Removing Members From A Protection Group
A method for stopping replication for a portion of a pgroup is to remove the respective member (volume, host, or host group) from the pgroup.

Important Note	Any pgroup snapshots replicated while the member is removed will not include a snapshot of those removed members. Existing pgroup snapshots on the target,

that were replicated prior to removing the member, will still contain snapshots of those members until they are removed according to the snapshot retention policy.

To remove a pgroup member select the pgroup from the Protection tab > Source Groups > pgroup name, in the Members panel check the box next to the member you want to remove, and then click on the settings button and choose Remove Selected Volumes.

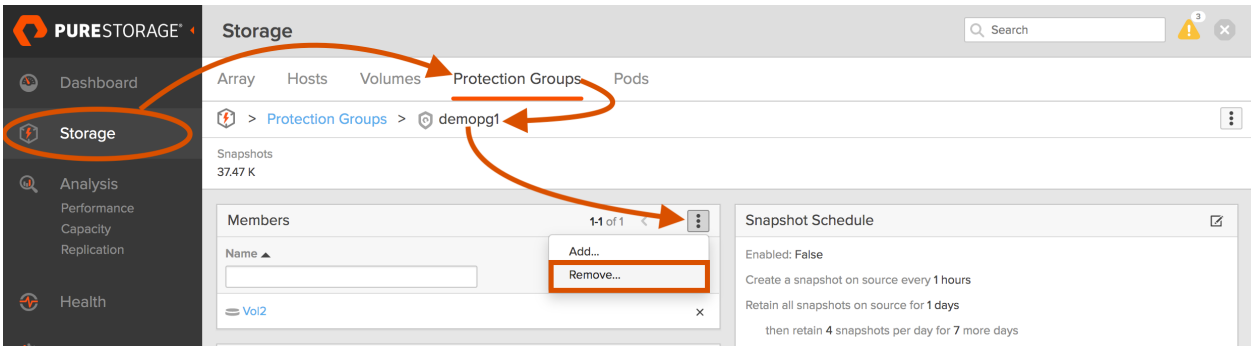


Figure 17 - Removing a member from a Protection Group

REPLICATION SCHEDULE BEHAVIOR

If an in-progress transfer is stopped or interrupted, for example if the FlashArrays lose connection with each other, the current in progress snapshot remains on both arrays. This snapshot will not expire or be automatically deleted until the arrays re-establish a connection and Purity is able to continue the transfer of that snapshot.

While replication is interrupted, no other replication snapshots will be created on the source array. That is, Purity does not continue to take replication snapshots in the background while an in-progress snapshot transfer is stalled due to the arrays being disconnected. Purity does not maintain a history or backlog of missed replication snapshots queued up to replicate when the connection is re-established.

Once the interrupted snapshot has finished replicating, if more than the replication interval time has passed, then Purity will immediately start another replication transfer. For example, if the configured replication frequency is configured for 30 minutes, and the last transfer took more than 30 minutes, another replication transfer will start immediately. If the last transfer completed in less than 30 minutes, then the next transfer will start 30 minutes after the start time of the last transfer.

PERFORMING FAILOVER

Typically, array-based replication solutions require that a replication relationship be "quiesced" or "broken" or in some other way stopped, before performing a failover, before connecting hosts to the target, or before reversing replication. This is not the case with Asynchronous replication. Because there are no dependencies between snapshots and volumes created from those snapshots, there is no requirement to stop replication from the source to the target when performing a failover.

However in most failover cases, for example a real DR failover, or permanently moving a workload to the target array, you would simply stop replication when performing a failover as there is likely no need to continue replicating to the target if there is no new data being generated at the source. If the workload has been moved to the target array, then replication updates would essentially be empty snapshot transfers. If this were a disaster recovery scenario source-to-target replication will not occur if the source is inaccessible.

Important Note	A failover might be performed for various reasons. There could be a large-scale outage that affects an entire site, causing all systems including application servers and storage to be inaccessible, or there could be a failure that affects the storage array only. Applications must not be started at a recovery site while older instances of those same applications might be servicing users at the original site. If uses are allowed to access two different instances of the same application, recovering from such a situation can be very difficult. Proper steps should be taken to ensure that older instances of applications that might be running are shut down, preventing access to them before a failover is performed.

Asynchronous replication does not create replica volumes at the target array; instead it replicates snapshots to the target array. Snapshots cannot be directly connected to hosts. A snapshot must be cloned to create a volume that the host can access. Volumes are cloned from snapshots using the volume copy command in the GUI, CLI or via REST API. As shown in Figure 18 each volume in the pgroup must be cloned individually. The volume copy command does not copy any data; only metadata is updated to create a new volume that simply shares all of its blocks with the snapshot. As data change occurs within this new volume, new blocks are allocated to store the new volume's unique data.

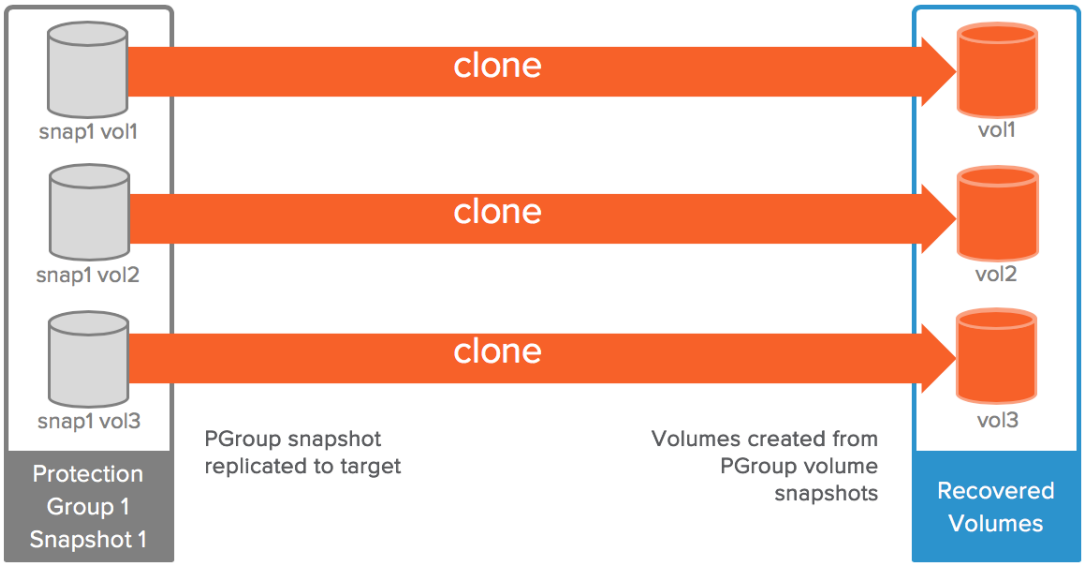
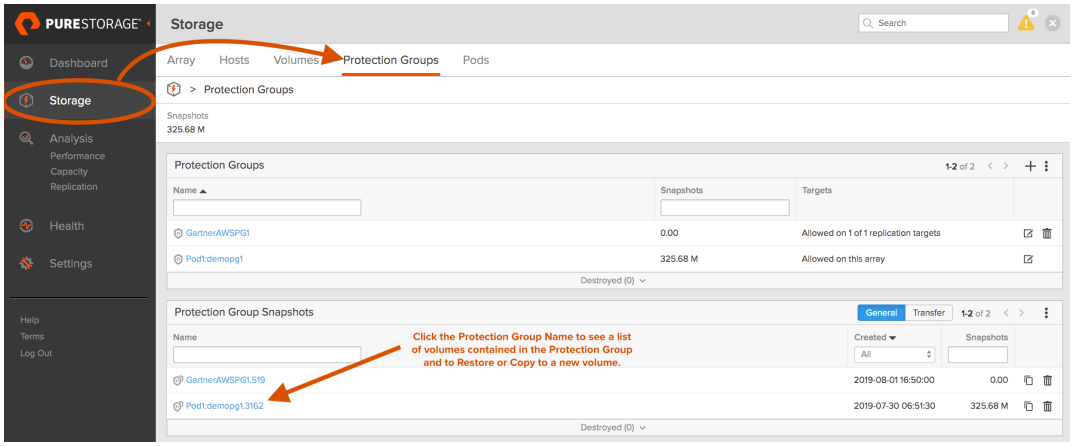


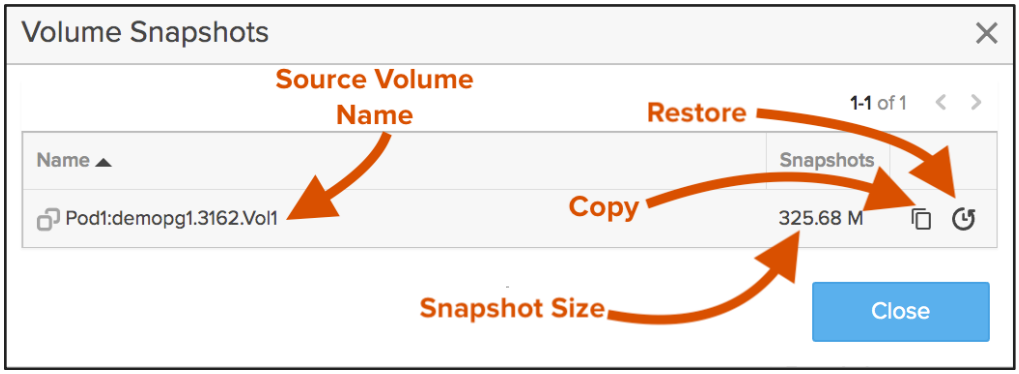
Figure 18 - Recovering Replicated Volumes

With Asynchronous replication performing a DR failover for a volume is a simple process. Essentially, you choose the desired recovery point (snapshot) for the volume you want to failover, clone a mountable volume from the desired recovery point, and then connect hosts to the new volume. Volumes cloned from snapshots are standard volumes in Purity; they do not have to be split from the snapshot they were created from, you can create snapshots of them, and you can replicate them just as any other volume on the array.

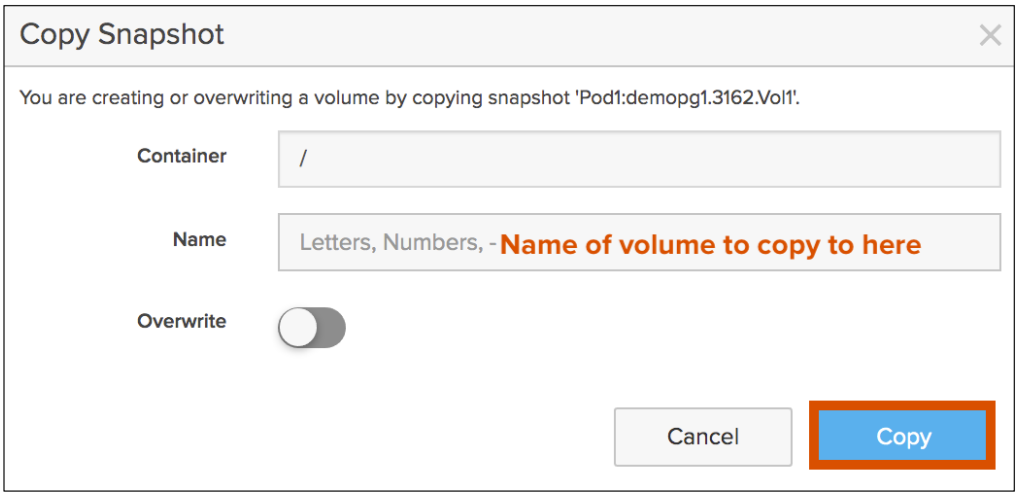
1. To failover or recover the volumes at the target FlashArray select the **Storage** > **Protection Groups** tab. **The Protection Group Snapshots** panel shows the historical list of snapshots for each Protection Group. Click the on the name of the snapshot button to view the volume snapshots.



2. The dialog that appears displays each of the volume snapshots contained within the select Protection Group snapshot. The icons to the right are to Copy to a new volume and to Restore to an existing volume, respectively. Click the Copy icon next to the volume snapshot you wish to copy out to a new volume and then click Copy.



3. Enter a name for the new volume created from this snapshot. In this example we use the same name as the original volume followed by _DR.



Steps 2 and 3 must be repeated for each volume in the pgroup if all volumes are required for failover.

Managing volume/LUN Serial Number Changes
By default, each time you create a volume from a snapshot a new serial number is assigned to the volume. Most host operating systems will detect this new serial number

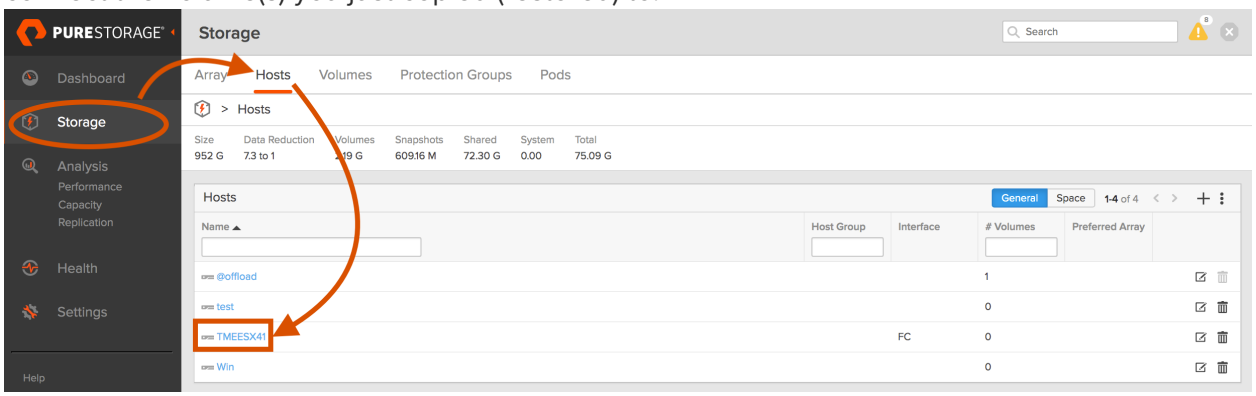
and identify the LUN as a different device each time a new volume is created and attached to the same host.
In some cases, you may want to create a volume from a snapshot replicated to a target array, connect that volume to a host, and then later refresh the data in that same volume using a more recently replicated snapshot. This is done using the overwrite option of purevol copy. In the CLI interface use the purevol copy command with the --overwrite option. The overwrite option will replace the volume with one having the same serial number as the volume being overwritten but based on the newer snapshot. Note that with most host operating systems you should unmount the LUN, or shut down hosts using the LUN, before performing the overwrite, and then remount the LUN. This is necessary to ensure that the host has released any data it had cached or buffered in memory before the LUN is changed on the array.

Example command to clone a snapshot to a volume, overwriting that volume:

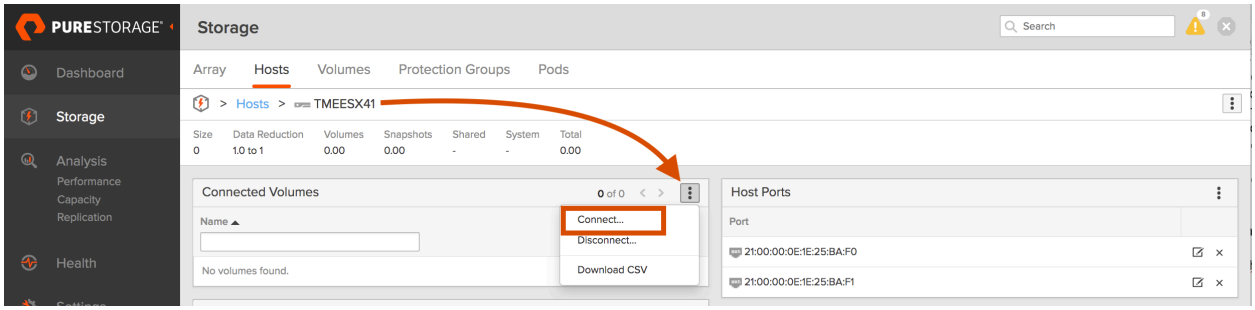
pureuser@array1>purevol copy --overwrite <snapshot_name> <overwritten_volume_name>

The --overwrite option is also useful when failing a workload back to its original size following a disaster recovery for example. If the original host at the primary site remains and you want to reconnect that same host to the volume replicated back to that array, use the purevol copy --overwrite command to ensure the serial number of the LUN is not changed. In some scenarios, such as failover and failback automated with VMware vCenter Site Recover Manager, maintaining the same serial number is not necessary as SRM manages reconnecting datastores to hosts regardless of changes in LUN serial numbers.

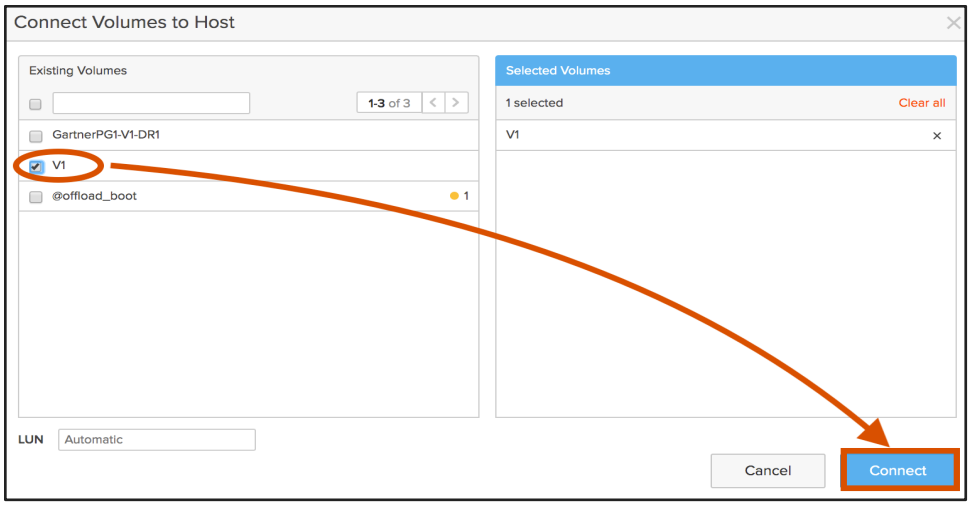
4. Connect the newly created volumes to the appropriate hosts. In this example we are connecting the volumes to a host called TMEESX41 that already exists on the target FlashArray. Select the **Storage** > **Hosts** tab, then click on the name of the host you wish to connect the volume(s) you just copied (restored) to.



5. Add each of the newly created volumes to the host by clicking the setup button at the top right corner of the **Connected Volumes** panel and clicking **Connect**.



6. A dialog box will open. At the left side of the dialog window select the volumes to add and then click **Connect** to add these volumes to this host.



7. The hosts may now rescan for storage to detect and access the recovered volumes.

REVERSING AND RESYNCHRONIZING REPLICATION

Asynchronous replication can be reversed without the need to resend all the data in the other direction, providing the original source volumes still remain at the source and have a common snapshot with the target volume.

Configuring Asynchronous replication to replicate volumes in the opposite direction simply involves creating a Protection Group at the original target array, adding the volumes you want to replicate back to the source, and then selecting the original source array as the target for this

pgroup. In other words, the same steps used to configuration replication in the original direction are simply performed again with the roles of the source and target reversed.

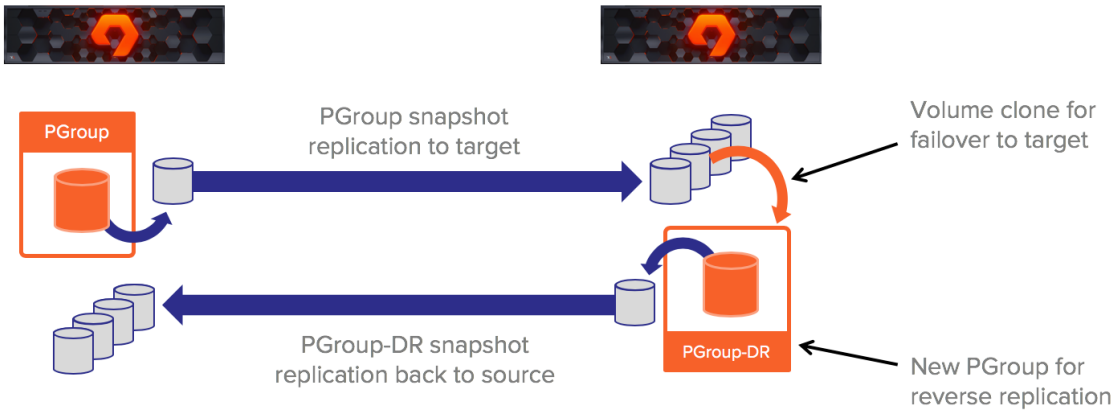
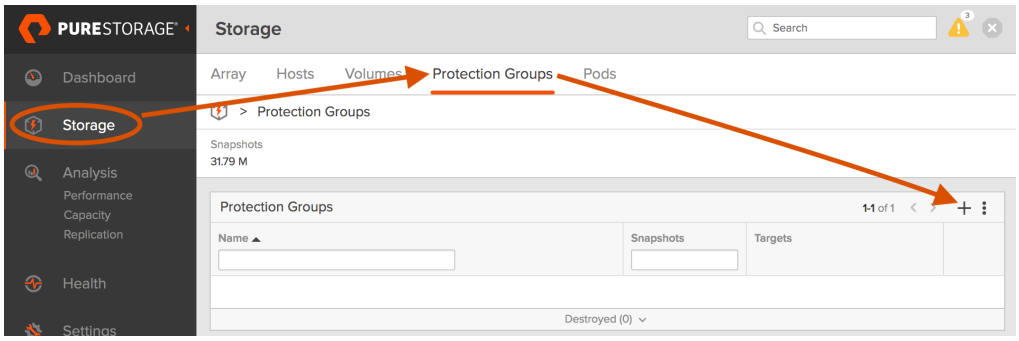


Figure 19 - Reversing and Resynchronizing Replication

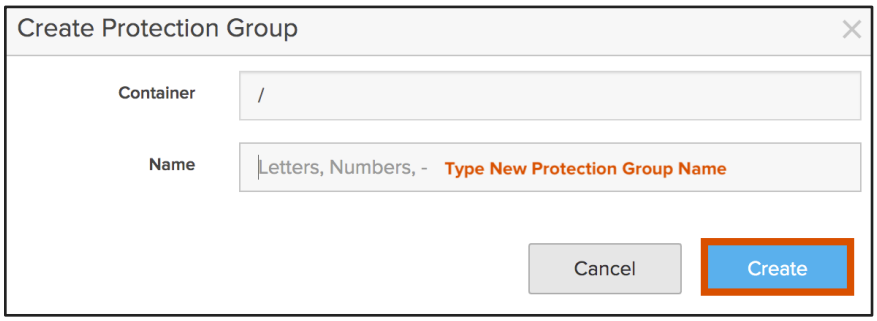
Asynchronous replication does not prevent the original source system from continuing to transfer snapshots to the target after replication is reversed. As mentioned earlier regarding failover of replication, because there are no dependencies between snapshots and volumes created from those snapshots, there is no requirement to stop replication from the source to the target when performing a failover. Likewise, there is no requirement to stop replication from the source to the target prior to configuring replication in the other direction. However, in most use cases where replication is to be reversed, replication in the other direction can be stopped.

Note, if you are recovering after a disaster in which the original source array was completely lost, the arrays must be reconnected following the procedure described in Connect Arrays section. Earlier in this guide we created a Protection Group from the **Storage** > **Protection Group** tab. In this example we will use that same methodology. This time, we are configuring replication in the reverse direction from the target FlashArray to the source FlashArray, so we need to create a Protection Group on the original target FlashArray—the array that will become the new source FlashArray for the new Protection Group.

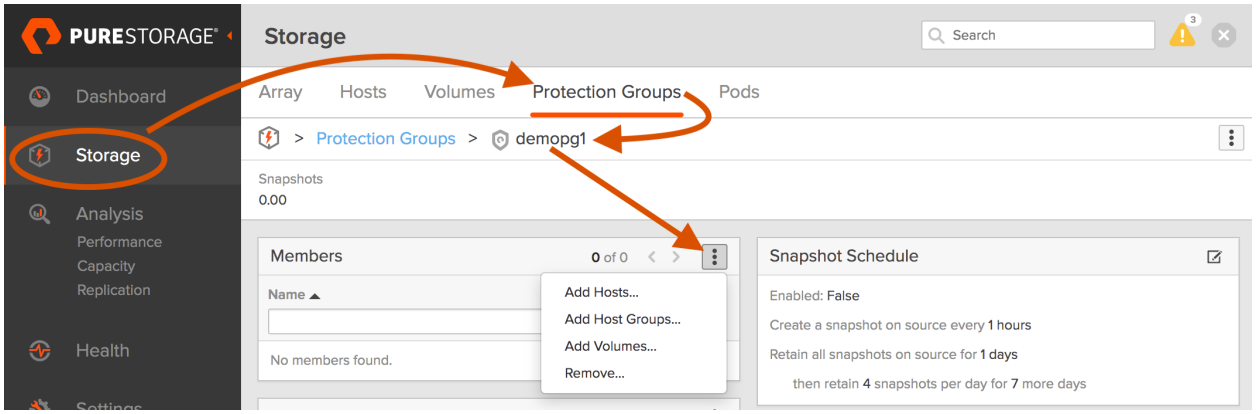
1. On the original target FlashArray click **Storage** and then select the **Protection Groups** tab then click the **+** sign at the right corner of the Protection Groups panel.



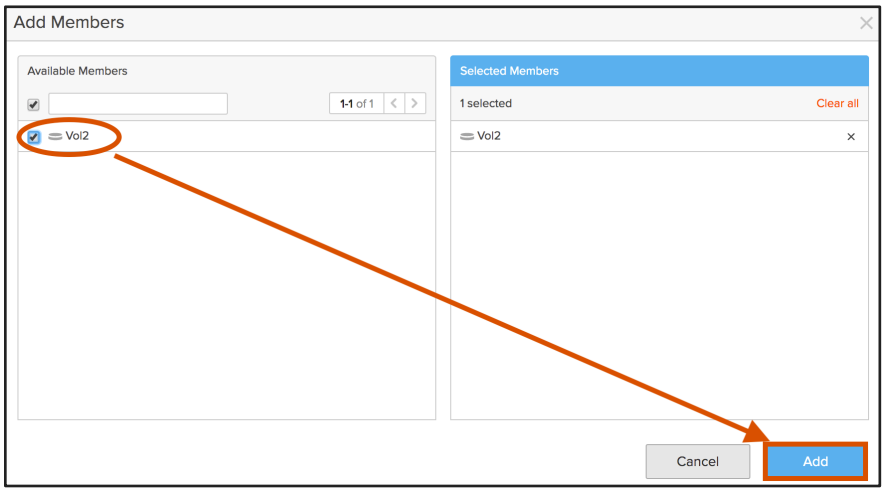
2. Enter a name for the new Protection Group and click **Create**.



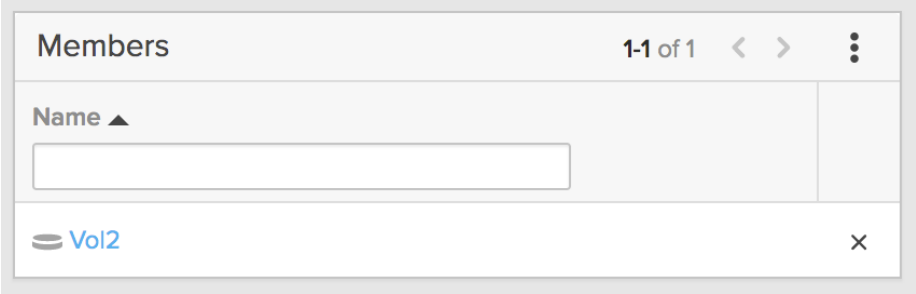
3. On the original target FlashArray, from the **Storage > Protection Group** > **[desired Protection Group] Members** panel click the setup icon in the top right right corner and select **Add Volumes**.



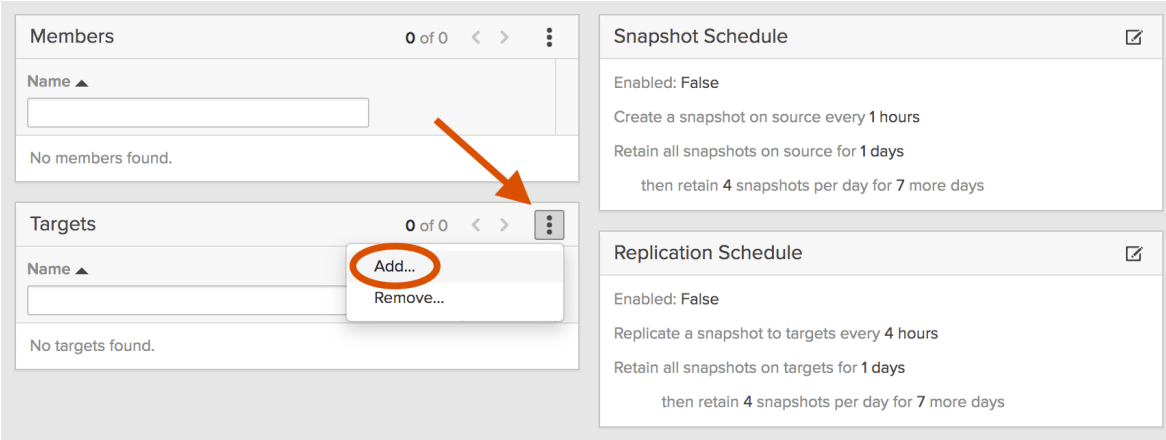
4. Click the checkbox next to each volume to be added to this protection group and click **Add.** You can add multiple members at the same time. The objective here is to select the volumes you wish to replicate back to the original source FlashArray.



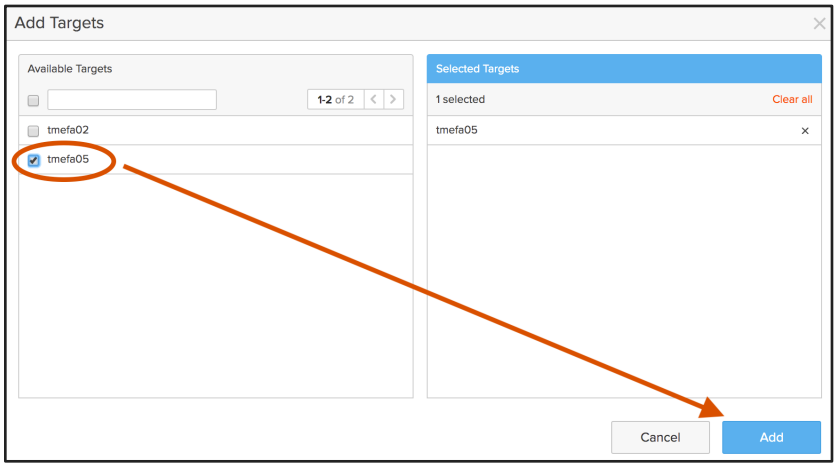
5. The **Members** panel of the Protection Group will show which members have been added.



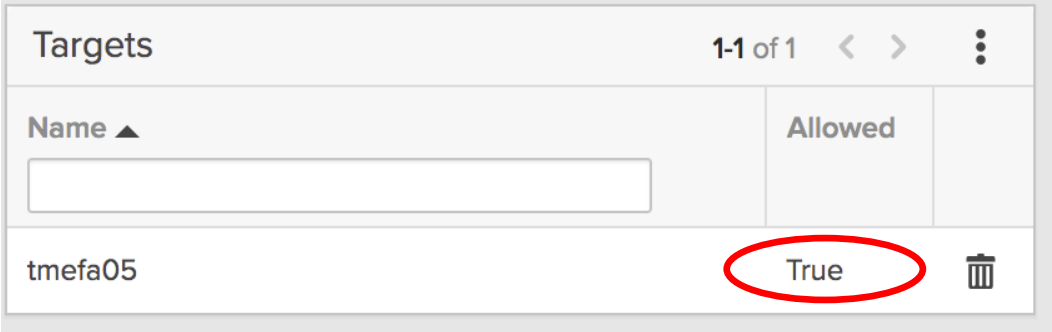
6. In the same FlashArray and Protection Group as step 5, click on the setup icon at the top right of the **Targets** panel and select **Add**.



7. The **Add Targets** dialog box will open and list the available targets connected to this array. Put a check in the checkbox next to the desired restore target FlashArray (originally the source FlashArray) for this Protection Group, then click **Add**.

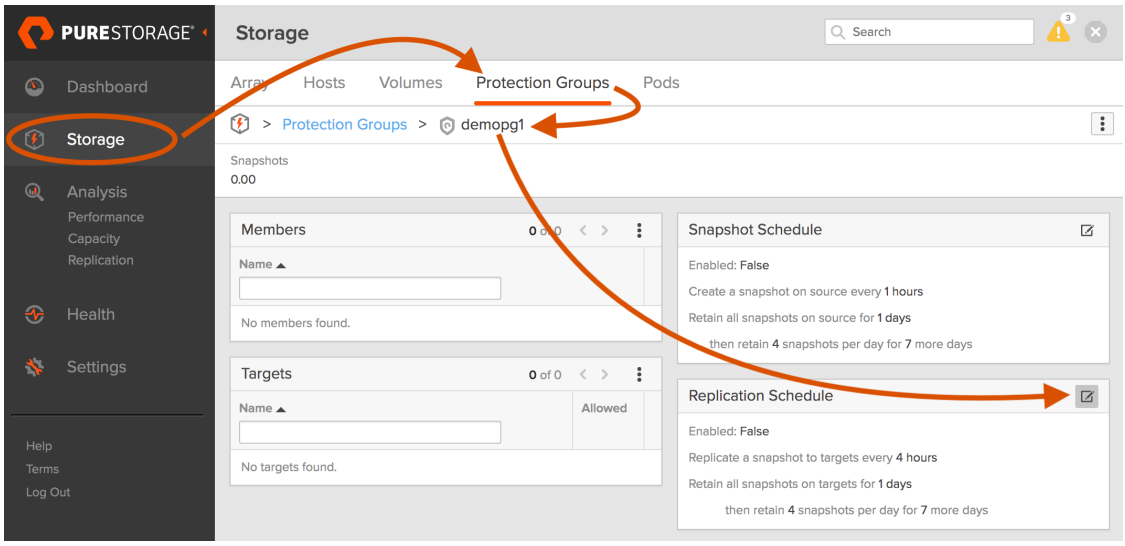


8. The restore target FlashArray (the original source FlashArray) will be shown in the list of targets for this Protection Group. You should see the **Allowed** column the status set to **True**.

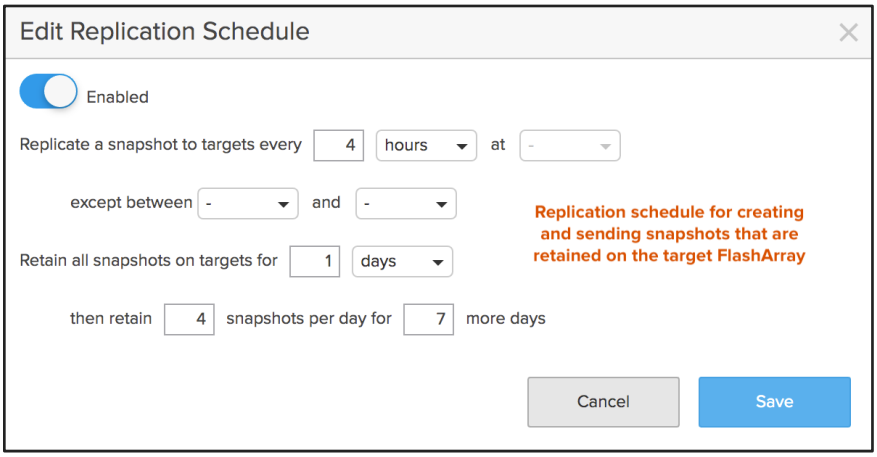


Note: If Allowed is **False** or if there are no arrays listed in the Targets panel then replication **will not take place**..

9. Click **Storage** > **Protection Groups** on the original target FlashArray (now the new source) and then click on the newly created protection group. Next click the edit icon at the right corner of **Replication Schedule** panel within the Protection Group.



The Edit Replication Schedule dialog window will open.



Configure the Protection Group replication schedule and target snapshot retention policy.

The schedule must also be toggled from Disabled to **Enabled** in order for replication to take place. By enabling the schedule, the first transfer will begin as soon as you define a replication target (done in a later step). If you prefer to start replication at a later time, then leave the Replication Schedule **Disabled** until you wish to start replication.

Click **Save** when the schedule is set to the desired values.

10. The new protection group will now replicate in the reverse direction (back to the original source FlashArray) allowing restore of volumes and data at the original source FlashArray.

Important Note	As discussed earlier, the purevol copy --overwrite command can be useful when failing a workload back to its original site following a disaster recovery. If the original host at the primary site remains, use the purevol copy --overwrite command when cloning the replicated snapshot to a volume to ensure the serial number of the LUN is not changed. Note that with most host operating systems you should unmount the LUN, or shut down hosts using the LUN, before performing the overwrite, and then remount the LUN. This is necessary to ensure that the host has released any data it had cached or buffered in memory before the LUN is changed on the array.

RESOURCES

FLASHARRAY USER GUIDE

- Pure Storage Community: https://support.purestorage.com/FlashArray/PurityOE/FlashArray_User_Guide
- Purity GUI: Select Help > User Guide
- Purity CLI: man <CLI command>. For example, man purearray displays the man pages for the purearray CLI command.

PURE STORAGE REST API GUIDE

- Pure Storage Community: https://support.purestorage.com/FlashArray/PurityOE/REST_API/REST_API_Reference_Guides
- Purity GUI: Select Help > REST API Guide

Pure Storage, Inc.

Sales | sales@purestorage.com
Media | pr@purestorage.com
General | info@purestorage.com

650 Castro St #400,
Mountain View, CA 94041
800-379-7873

PURESTORAGE®