FlashBlade User Guide

Version 3.1.1

PURESTORAGE®

Copyright Statement

Pure Storage, Inc. 650 Castro Street Mountain View, CA 94041 *http://www.purestorage.com*

Direct comments to *DocumentFeedback@purestorage.com*.

Table of Contents

Chapter 1
About this Guide

The Pure Storage FlashBlade® User Guide is written for array administrators who view and manage the Pure Storage FlashBlade storage system. FlashBlade arrays are administered through the Purity for FlashBlade (Purity//FB) graphical user interface (GUI) or command line interface (CLI). Users should be familiar with system, storage, and networking concepts, and have a working knowledge of Windows or UNIX.

Organization of this Guide

This guide is organized as follows:

Purity//FB Overview - Defines FlashBlade array naming and numbering conventions.

Purity//FB GUI Overview and (subsequent topics) - Describes the use of the browser-based Purity//FB graphical user interface (GUI) to administer a FlashBlade array.

Using the Purity//FB CLI to Administer a FlashBlade Array - Describes the Purity//FB command line interface (CLI) and its use in FlashBlade array administration.

Quick Reference Guide - Provides a quick reference for exporting file systems and creating alert watchers, as well as information about related FlashBlade documentation and how to contact Pure Storage.

SMB Port Assignments - Provides the minimum ports requiredo neach VLAN for SMB to function successfully.

About NFSv4.1 Support - Describes the subset of core NFSv4.1 protocol features supported by FlashBlade.

Replication and Recovery - Provides step-by-step instructions on how to set up file and object replication, as well as how to perform failover and failback, using the Purity//FB GUI and CLI.

A Note on Format and Content

FlashBlade technology is evolving rapidly. As with all advanced information technologies, once basic architecture is in place, implementation develops at different rates in its different facets, each preceded by feature-by-feature detailed design.

This edition of the guide describes the properties and behavior of arrays that run the latest Purity//FB release. It may include information about planned capabilities whose external form has been specified at the time of the release. Such material is included to provide users of this release with information for design planning purposes, and is subject to change as new functionality is implemented. Material relating to not-yet-implemented functionality is identified as such in the text.

Documentation and Support

All related guides are or will soon be available in the Pure Storage Knowledge Base, which is located at *http://support.purestorage.com*.

Contact Us

We welcome your feedback about Pure Storage documentation and encourage you to send your questions and comments to *documentfeedback@purestorage.com*.

Product Support

If you are a registered Pure Storage user, log in to *http://support.purestorage.com* to browse our knowledge base, view the status of your open support cases, and view the details of past support cases.

You can also contact Pure Technical Services at *support@purestorage.com*.

Chapter 2
Purity//FB Overview

Purity for FlashBlade (Purity//FB) is the operating environment that queries and manages the FlashBlade hardware, networking, and storage components. The Purity//FB software is distributed with the FlashBlade array.

Purity//FB provides two ways to administer the FlashBlade array: through the browser-based graphical user interface (Purity//FB GUI), and through the command line interface (Purity//FB CLI).

This chapter covers Purity//FB conventions and troubleshooting issues that may arise.

Purity//FB Conventions

Purity//FB follows certain conventions.

Object Names

Valid characters are letters (A-Z and a-z), digits (0-9), and the hyphen (-) character. File system names can also include the underscore (_) character. The first and last characters of a name must be alphanumeric, and a name must contain at least one letter.

Most objects in Purity//FB that can be named, including file systems, data vips, and subnets can be 1-63 characters in length.

Array names can be 1-56 characters in length. The array name length is limited to 56 characters so that the names of the individual controllers, which are assigned by Purity//FB based on the array name, do not exceed the maximum allowed by DNS.

Purity allows the use of lowercase and uppercase in names and preserves capitalization in command output. However, duplicate naming with different capitalization is not allowed. Additionally, search ignores capitalization differences.

Array and File System Sizes

Array and file system sizes are specified as integers followed by one of the suffix letters `K`, `M`, `G`, `T`, `P`, representing `KiB`, `MiB`, `GiB`, `TiB`, and `PiB`, respectively, where " `Ki` " denotes 2^10, " `Mi` " denotes 2^20, and so on.

Time Zones

Purity//FB GUI dates and times are displayed in the user's local time zone, which is determined by the browser's local time. The user's local time zone appears at the bottom of the navigation pane of the Purity//FB GUI.

Purity//FB CLI dates and times are displayed in the time zone of the array, which is set during FlashBlade installation. The array time zone appears and can be edited in the **Time** panel of the **System** > **Settings** page.

IP Addresses

FlashBlade supports two versions of the Internet Protocol: IP Version 4 (IPv4) and IP Version 6 (IPv6). IPv4 and IPv6 addresses follow the addressing architecture set by the Internet Engineering Task Force.

An IPv4 address consists of 32 bits and is entered in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from `0` to `255` representing a group of 8 bits. Here are some examples:

```
puredns setattr --domain mydomain.com --nameservers 192.0.2.10
purenetwork setattr --address 192.0.2.0 DataVip01
puresubnet create --prefix 192.0.2.0/24 --vlan 100 Sub01
```

An IPv6 address consists of 128 bits and is written in in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Colons separate each 16-bit field. Leading zeros can be omitted in each field. Furthermore, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`) . For example, IPv6 address `2001:0db8:85a3:0000:0000:8a2e:0370:7334` becomes `2001:db8:85a3::8a2e:370:7334`.

To use an IPv6 address in a URL or URI, enclose the entire address in square brackets (`[]`) . When specifying a URL or URI with a port number, append the port number after the end of the entire address. Here are some examples:

```
puredns setattr --domain mydomain.com --nameservers 2001:db8:85a3::8a2e:370:7334
purenetwork setattr --address 2001:db8:85a3::8a2e:370:7334 DataVip01
puresubnet create --prefix 2001:db8:85a3::/64 --vlan 100 Sub01
```

Pure File SafeMode

Pure File SafeMode is configured during installation and is designed to protect the system from accidental changes. File SafeMode prevents users from performing some operations such as file system

snapshot restore and file system snapshot eradication. For more information, contact Pure Technical Services.

Troubleshooting

Help! Who do I contact about problems with the FlashBlade Array?

Contact a member of your Pure Storage® account team, visit us at *http://support.purestorage.com*, or email Pure Storage Support at *support@purestorage.com*.

If you are contacting Pure Storage Support about a technical issue, they may ask you to open an RA session so that they can help you diagnose the issue.

Error Messages

The following error messages may appear when you perform operations through the Purity//FB GUI or CLI:

Service Temporarily Unavailable

> An internal service is currently unavailable. This is a temporary issue. Wait a few minutes and then try the Purity//FB GUI or CLI operation again. If you still get the error message, contact Pure Storage Support at *support@purestorage.com*.

Unexpected Error

> The Purity//FB GUI or CLI operation that you performed generated an unexpected error. Contact Pure Storage Support at *support@purestorage.com*.

Chapter 3
Purity//FB GUI Overview

The Purity//FB graphical user interface (GUI) is a browser-based system used to view and administer the FlashBlade array.

The screens of the Purity//FB GUI differ slightly depending upon if your FlashBlade array is a single-chassis or is a multi-chassis configuration. The following figure displays the GUI for a single-chassis configuration.

Figure 3.1 Purity//FB GUI

The Purity//FB GUI contains the following pages:

Dashboard

Represents a graphical overview of the array, including hardware status, recent alerts, storage capacity, replication bandwidth, and input/output (I/O) performance metrics.

Storage

Displays summary array details including connection and replication information. In addition, quick links are provided to access file systems, snapshots, object store user, account, bucket, and object information, and export rules. Additionally, a **File Systems** tab provides a list of file systems on the array and its attributes, including provisioned size, capacity usage details, and export rules. An **Object Store** tab is provided to quickly access object store accounts, buckets, and users.

Protection

Displays replication information. A list of file system snapshots, their source array, source file system, replication policy, and creation date is displayed. Tabs for **Policies**, **File Replica Links**, and **Object Replica** links are provided to view details about replication policies and file and object replication link information.

Analysis

Displays historical array information, including space and I/O performance metrics, from various viewpoints. Latency, IOPs, and bandwidth data for file systems and the object store can be viewed from the **File Systems** and **Object Store** tabs.

Health

Displays array health including alerts, hardware status, and parity.

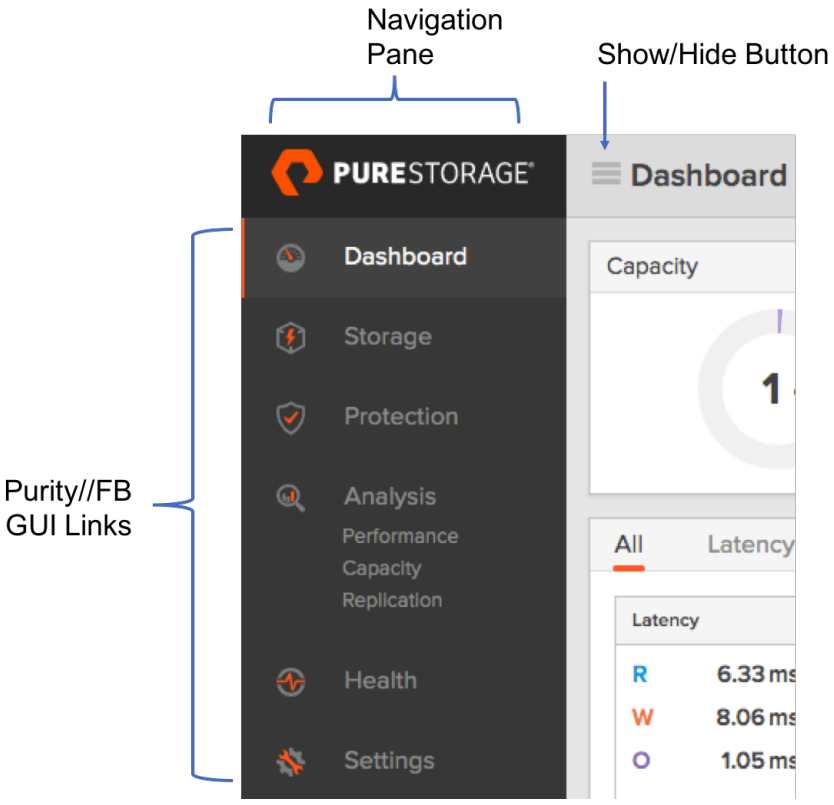
Settings

Displays array -wide system and network settings. Manage array -wide components, including network interfaces, system time, connectivity and connection configurations, directory services, support settings, SSL certificates, and alert settings.

Purity//FB GUI Navigation

The dark gray navigation pane that appears along the left side of the Purity//FB GUI contains links to the Purity//FB GUI pages.

Figure 3.2 Purity//FB GUI - Navigation Pane



Click a link in the navigation pane to analyze or configure the information that appears in the page to its right. For example, click the Storage link to view information about the FlashBlade file systems.

Click the **show/hide** button to toggle between the expanded and collapsed views of the navigation pane. The show/hide button appears to the right of the Pure Storage logo and resembles three horizontal lines.

The navigation pane also includes links to the following external sites:

Help

Launches the FlashBlade user guide, REST API guide, community and support websites.

Terms

Launches the Pure Storage End User Agreement page.

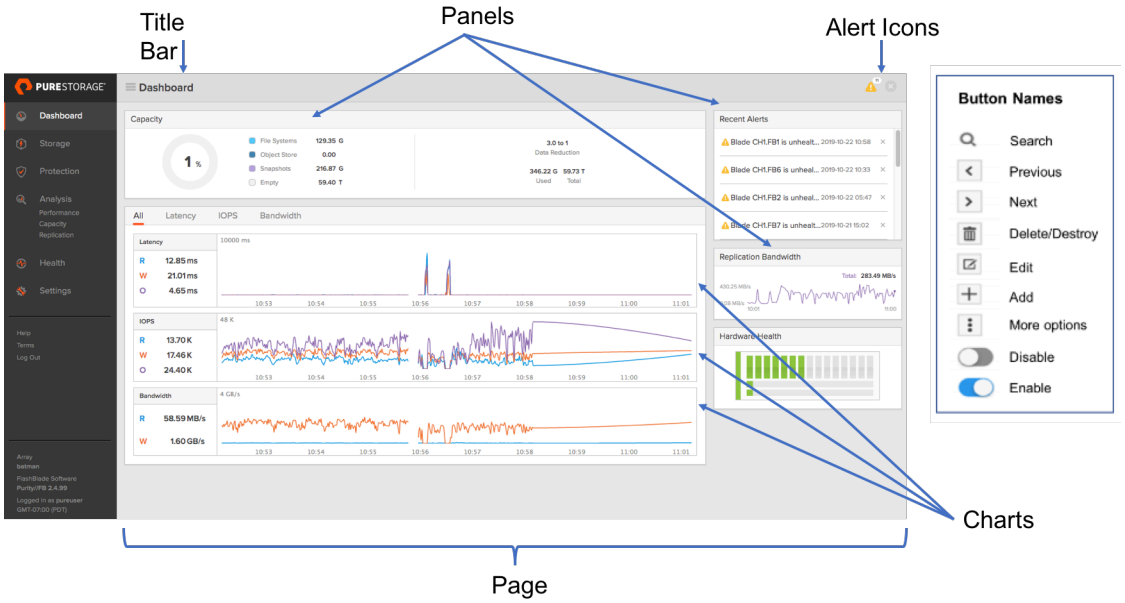
Log Out

Logs the current user out of the Purity//FB GUI.

The bottom of the navigation pane displays FlashBlade array name and Purity//FB version information, the name of the user who is currently logged into the Purity//FB GUI, and the user's local time zone. The dates and times that appear in the Purity//FB GUI are based on the user's local time zone, which is determined by the browser's local time.

The page to the right of the navigation pane displays the information and configuration options for the selected Purity//FB GUI link. The information on each page is organized into panels, charts, and lists.

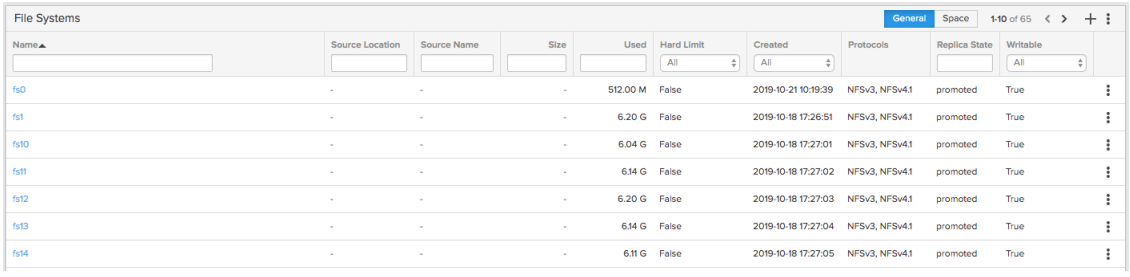
Figure 3.3 Purity//FB GUI - Page and Buttons



The alert icons that appear to the far right of the title bar indicate the number of open **Warning** and **Critical** alerts. Alerts in the `open` and `closing` states are included in the alert count displayed by the icons.

When an alert is in the `closing` state, it is still considered open. If after 24 hours there is no recurrence or increase in severity, the alert will close. However, if there is a recurrence or increase in severity, the alert will move from the `closing` state back to the `open` state. To analyze alerts in more detail, click the **Health** link.

Various panels, such as **Storage > File Systems** and **Health > Alerts**, contain lists of information. The total number of rows in a list output is displayed in the upper-right corner of the list. Some lists can be very large, extending beyond hundreds of rows.



Pagination divides a large list output into discrete pages. Pagination is in effect if the number of lines in the list output exceeds 25 rows for tabs displaying a single table, or 10 rows for tabs displaying multiple tables. To move through a paginated list, click **<** to go to the previous page, or click **>** to go to the next page.

Purity//FB GUI Login

Logging in to the Purity//FB GUI requires a virtual IP address or fully-qualified domain name (FQDN) and an Purity//FB login username and password; this information is provided during the FlashBlade installation.

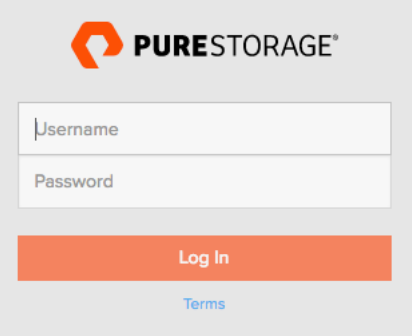
FlashBlade is installed with one administrative account with the username `pureuser`. The initial password for the account is `pureuser`. As a best practice, for security purposes Pure Storage recommends that the password for the account be changed immediately upon first log in using the `pureadmin` CLI command.

Pure Storage tests the Purity//FB GUI with the most recent version of the following web browsers:

- Apple Safari
- Google Chrome
- Mozilla Firefox

To launch the Purity//FB GUI login screen, open a browser and type the virtual IP address or FQDN into the address bar, and press **Enter**.

Figure 3.4 Purity//FB GUI - Login Screen



Logging in to the Purity//FB GUI

Log in to the Purity//FB GUI to view and administer the FlashBlade array.

To log in to the Purity//FB GUI:

1. Open a web browser.
2. Type the virtual IP address or fully-qualified domain name of the FlashBlade in the address bar and press **Enter**. The Purity//FB GUI login screen appears.
3. In the Username field, type the FlashBlade user name. For example, `pureuser`.
4. In the Password field, type the password for the FlashBlade user. For example, `pureuser`.

5 Click **Log In** to log in to the Purity//FB GUI.

Chapter 4
The Purity//FB GUI Dashboard Page

The **Dashboard** page displays a running graphical overview of the array 's storage capacity, performance, and hardware status.

Figure 4.1 Dashboard

The **Dashboard** page contains the following panels and charts:

- **Capacity**
- **Performance Charts**
- **Recent Alerts**
- **Replication Bandwidth** (only for replication setups)
- **Hardware Health**

Capacity

The **Capacity** panel displays array size and storage consumption details. All capacity values are rounded to two decimal places.

Figure 4.2 Dashboard - Capacity Panel

The capacity wheel displays the percentage of array space occupied by file system data. The percentage value in the center of the wheel is calculated as **Used** capacity/ **Total** capacity.

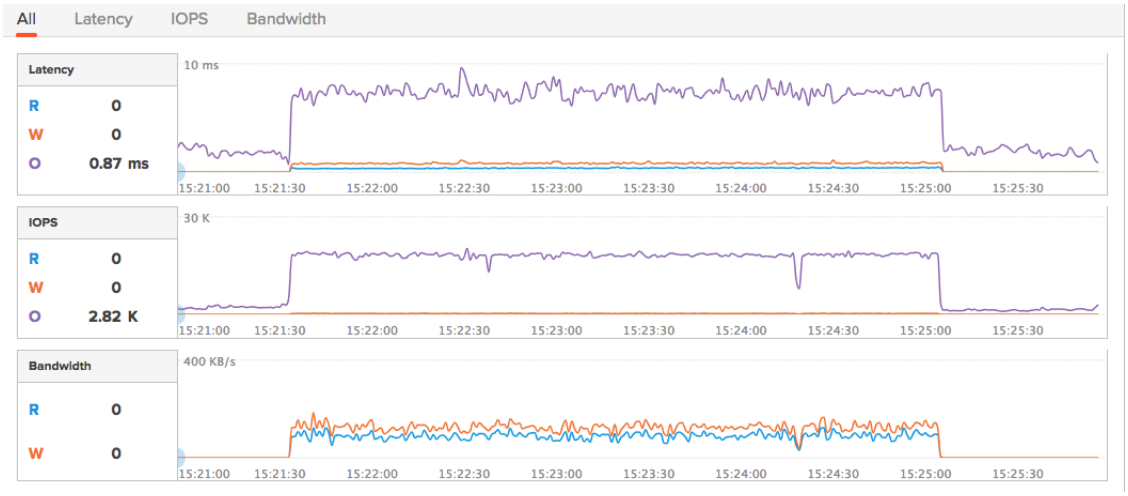
The capacity data is broken down into the following components:

- **File Systems**: Amount of space that the written data occupies on the array 's file systems after reduction via data compression.
- **Object Store**: Amount of space that the written data occupies on the array 's buckets after reduction via data compression.
- **Snapshots**: Amount of space that the array 's snapshots occupy after data compression.
- **Empty**: Unused space.
- **Used**: Amount of space that the written data occupies on the array after reduction via data compression.
- **Total**: Total usable capacity of the array.
- **Data Reduction**: Ratio of the size of the written data (i.e. Used) versus the amount of space the data occupies after data compression (i.e. Physical).

Performance

The performance panel displays latency, IOPS, and bandwidth values in real time.

Figure 4.3 Dashboard - Performance Charts



The performance metrics are displayed along a scrolling graph; incoming data appears along the right side of each graph every second as older data drops off the left side after 5 minutes. Each performance chart includes R, W, and O (if applicable) values, representing the most recent data samples.

Hover over any of the charts to display metrics for a specific point in time. The values that appear in the point-in-time tooltips are rounded to two decimal places.

The performance panel includes Latency, IOPS, and Bandwidth charts. The **All** chart displays all three performance charts in one view.

Latency

The **Latency** chart displays the average latency times for various operations.

- **Read Latency (R)** - Average arrival-to-completion time, measured in milliseconds, for a read operation.
- **Write Latency (W)** - Average arrival-to-completion time, measured in milliseconds, for a write operation.
- **Other Latency (O)** - Average arrival-to-completion time, measured in milliseconds, for all other metadata operations.

IOPS

The **IOPS** (Input/output Operations Per Second) chart displays I/O requests processed per second by the array. This metric counts requests per second, regardless of how much or how little data is transferred in each.

- **Read IOPS (R)** - Number of read requests processed per second.
- **Write IOPS (W)** - Number of write requests processed per second.
- **Other IOPS (O)** - Number of metadata operations processed per second.

- **Average IO Size** - Average I/O size per request processed. Requests include reads and writes.

Bandwidth

The **Bandwidth** chart displays the number of bytes transferred per second to and from the array (both file systems and buckets) . The data is counted in its expanded form rather than the reduced form stored in the array to truly reflect what is transferred over the storage network. Metadata bandwidth is not included in these numbers.

- **Read Bandwidth (R)** - Number of bytes read per second.
- **Write Bandwidth (W)** - Number of bytes written per second.

By default, the performance charts display performance metrics for the past 5 minutes. To display more than 5 minutes of historical data, select **Analysis** > **Performance**.

Note about the Performance Charts

The **Dashboard** and **Analysis** pages display the same latency, IOPS, and bandwidth performance charts, but the information is presented differently between the two pages.

In the **Dashboard** page:

- The performance charts are updated once every second.
- The performance charts display up to 5 minutes of historical data.

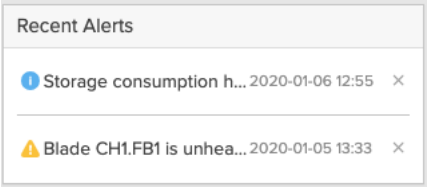
In the **Analysis** page:

- At its shortest range (5m), the performance charts are updated once every second. As the range increases, the update frequency (and resolution) decreases.
- The performance charts display up to 1 year of historical data.
- The performance charts can be filtered to display metrics by protocol.

Recent Alerts

The **Recent Alerts** panel displays a list of alerts that Purity//FB saw that is both flagged and not in the closed state. The list contains recent alerts of all severity levels. If no alerts are logged, the panel displays **No recent alerts**.

Figure 4.4 Dashboard - Recent Alerts Panel



To view the details of an alert, click the alert message.

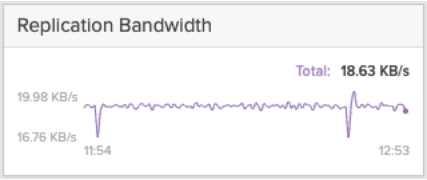
To remove an alert from the **Recent Alerts** panel, click the clear flag (**X**) button. The alert will no longer be displayed in the **Recent Alerts** panel, but will still appear on the **Health** page.

To view a list of all alerts, including ones that are in no longer open, go to the **Health** page.

Replication Bandwidth

The **Replication Bandwidth** panel provides a graph that displays the total replication bandwidth. Note that this panel appears only if replication has been set up.

Figure 4.5 Dashboard - Replication Bandwidth Panel



Clicking the graph opens the **Analysis > Replication** page where finer detail about bandwidth usage can be viewed (see *Replication*).

Hardware Health

The **Hardware Health** panel displays the operational state of the FlashBlade array chassis, blades, and fabric modules. Hover over the image to view the component details.

Depending on the configuration of your array, the **Hardware Health** panel displays either a single or multi-chassis FlashBlade graphic.

Figure 4.6 Hardware Health Panel - Single chassis

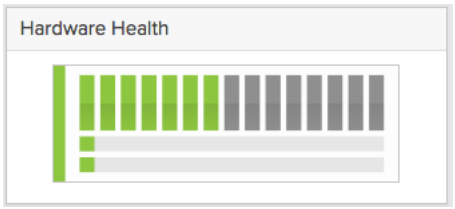
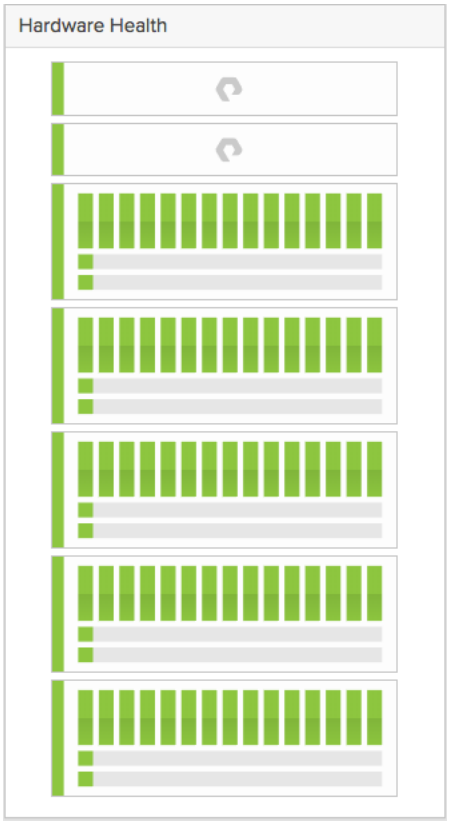


Figure 4.7 Hardware Health Panel - Multi-chassis



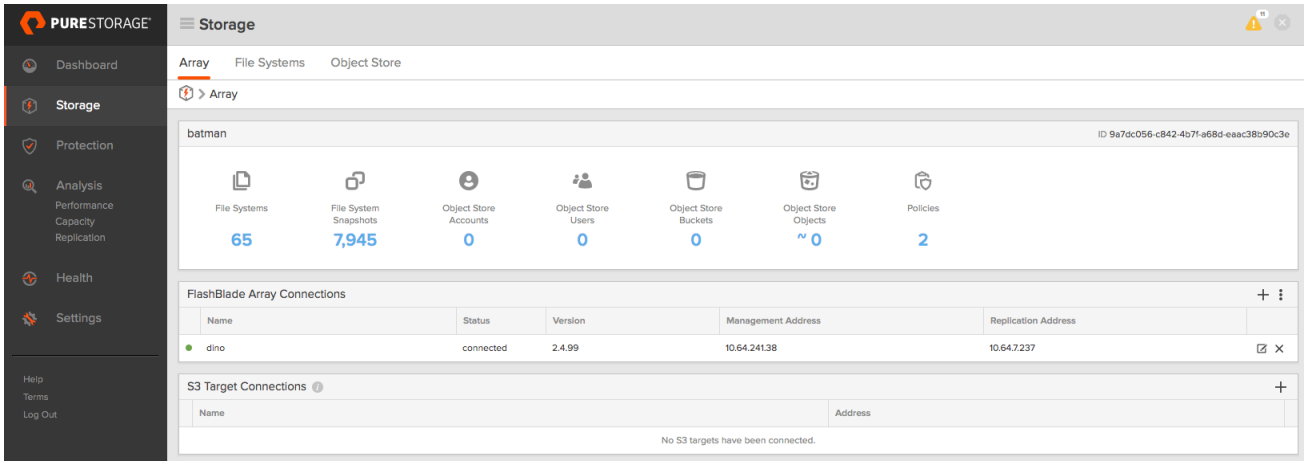
To analyze the hardware components in more detail, click the **Health** link.

Chapter 5
The Purity//FB GUI Storage Page

The **Storage** page displays and manages the array 's file systems and object store accounts, which contain users and buckets.

Accessing a file system or bucket requires at least one valid data virtual IP address (data vip). Data vips are configured through the **Settings** > **Network** page.

Figure 5.1 Purity//FB Storage Page

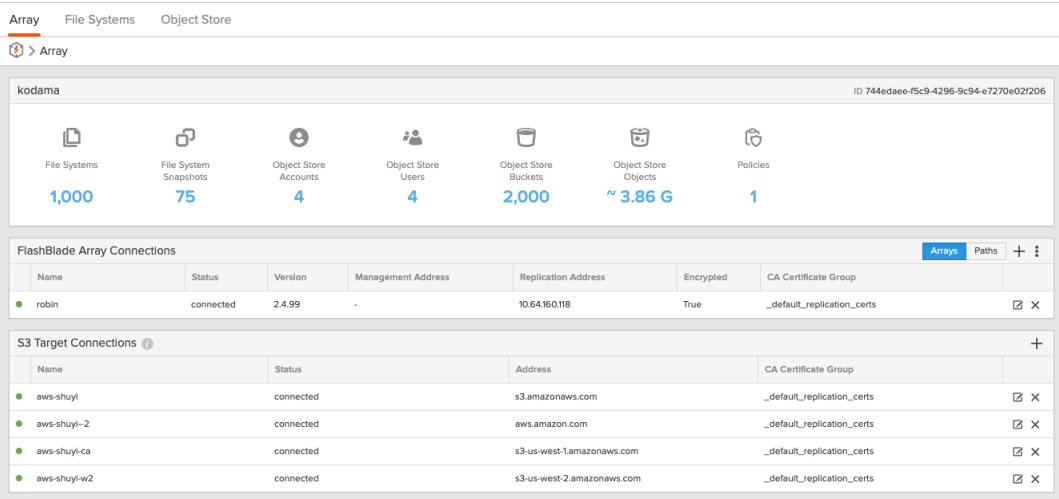


Array

By default the **Array** tab is displayed when navigating to the **Storage** page.

The **Array** tab displays an array summary pane, a **FlashBlade Array Connections** pane, and an **S3 Target Connections** pane.

Figure 5.2 Array Tab

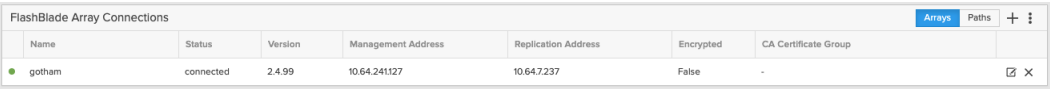


The array summary panel displays counts of various types of metadata objects. Click a count to jump to the page containing the details for associated objects. For example, clicking the **File System Snapshots** count opens the **Protection** > **Snapshots** tab in the GUI.

Array attributes such as array time and the array name are configured through the **Settings** > **System** page.

The **FlashBlade Array Connections** panel displays all arrays connected to the current FlashBlade array for replication.

Figure 5.3 FlashBlade Array Connections Panel



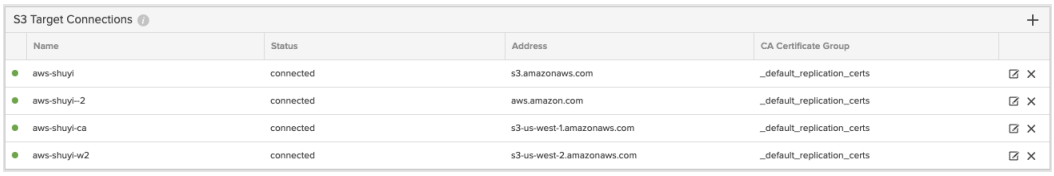
A connection must be established between two arrays in order for data replication to occur. FlashBlade supports asynchronous replication, which allows data to be replicated from one array to another. When two arrays are connected for asynchronous replication, the array where data is being transferred from is called the local (source) array, and the array where data is being transferred to is called the remote (target) array. For more information, see *The Purity//FB GUI Protection Page*.

Details about the array connection are provided including the current connection status, Purity software version, management and replication IP addresses, encryption status, and applied CA certificate group (if any). If you click **Paths**, the connected array and all connection paths (source and destination IP address) are displayed.

If the array connection is not healthy, the **Paths** view can be useful in determining which network paths are having problems.

The **S3 Target Connections** panel displays all S3 targets connected to the current FlashBlade array for object replication.

Figure 5.4 S3 Target Connections Panel



The names of connected S3 targets, connection status, connection address, and applied CA certificate groups are displayed. For more information, see *The Purity//FB GUI Protection Page*.

In addition to displaying details about connected S3 targets, this panel also allows you to rename existing S3 targets and connect new S3 targets.

Connecting Arrays

In order to connect two arrays to perform replication, you must have the following:

- Replication network interfaces on the source and target arrays.
- A network path from the source array to the target array's management vip.

Connecting two arrays is performed in the **FlashBlade Array Connections** panel of the **Array** page.

Note: Data can be optionally encrypted. However, file replication is not supported over an encrypted connection.

⚠ **Important note!** When connecting the source array to the target array, a secure connection is required even if encryption is not enabled. Certificates with small key sizes, weak signature algorithms, or that only support weak TLS algorithms are not supported.

To connect two arrays, perform the following:

1. From the **Storage** > **Array** page, click the add (+) button in the **FlashBlade Array Connections** panel. The **Connect FlashBlade Array** pop-up window appears.
2. Enter the target array's hostname or management address (unless using NAT) in the **Management Address** field. The address can be located from the **Subnets** table by navigating to the **Settings** > **Network** page.
3. Enter a connection key for the target array in the **Connection Key** field.
   - **a** On the target array, create the connection key (see *Creating a Connection Key*).
   - **b** Copy the new connection key.
   - **c** On the source array, paste the connection key in the **Connection Key** field in the **Connect FlashBlade** pop-up window.

4 The replication address is auto-discovered unless using NAT. The source and target arrays each need a replication network interface. If not already created, create the replication network interfaces on the source and target arrays (see *Creating a Network Interface*). Enter the target array's replication network address in the **Replication Address** field.

5 (Optional) Enable encryption by setting the **Encrypted** toggle to on. If set, the `_default_replication_certs` CA certificate group is applied. Before toggling, make sure to import the CA certificate to the source array if the target array's array-certificate is signed by a CA and add it to the certificate group. Note that encryption is set on the source array only. If enabled, the encryption setting displays as enabled on the target array.

Warning: Use of CA certificates to establish trust and secure communication for object replication is the recommended configuration. If the target array's array-certificate is self-signed, directly importing the target array certificate to the source and adding it to the certificate group will not provide as strict validation around aspects of the peer, such as certificate expiry.

6 Click **Connect**.

7 If using NAT, enter the source array's replication address on the target array.

a On the target array, in the **FlashBlade Array Conenctions** panel on the **Storage > Array** page, click the edit button at the end of the row displaying the connected source array. The **Edit Connected Array** pop-up window appears.

b Enter the replication address for the source array in the **Replication Address** field and click **Save**.

Creating a Connection Key

In order to connect two arrays for replication, a connection key must be created on the target array to be used on the source array during array connection.

Connection keys are created from the **FlashBlade Array Connections** panel of the **Array** page.

To create a connection key, perform the following:

1 On the target array, navigate to the **Storage** > **Array** page.

2 In the **FlashBlade Array Connections** pane, click **More Options** > **Create Connection Key**. The **Connection Key** pop-up window appears displaying the new connection key.

Once created, the key is active for two hours. If the source and target arrays have not been connected within that two hour period, a new connection key must be created.

Editing a Connected Array

A connected array must be edited if its management or replication address information has changed, or if you wish to enable or disable encryption (for object replication).

Editing a connected array is performed in the **FlashBlade Array Connections** panel of the **Array** page.

To edit a connected array, perform the following:

1. On the source array, navigate to the **Storage** > **Array** page.
2. In the **FlashBlade Array Connections** pane, click the **Edit** button located at the end of the row of the array you wish to edit. The **Edit Connected Array** pop-up window appears displaying the new connection key.
3. Enter a new management and/or replication address as needed.
4. Enable (blue) or disable (gray) encryption.
5. Click **Save**.

Disconnecting a Connected Array

Disconnecting an arry is performed in the **FlashBlade Array Connections** panel of the **Array** page.

To disconnect a connected target array, perform the following:

1. On the source array, navigate to the **Storage** > **Array** page.
2. In the **FlashBlade Array Connections** pane, click the Disconnect (**x**) button located at the end of the row of the array you wish to disconnect. The **Disconnect FlashBlade Array** pop-up window appears.
3. Click **Disconnect**.

Connecting an S3 Target

A FlashBlade array can be connected to an S3 target for object replication.

Connecting a FlashBlade to an S3 target is performed in the **S3 Target Connections** panel of the **Array** page.

To connect an S3 target, perform the following:

1. From the **Storage** > **Array** page, click the add (**+**) button in the **S3 Target Connections** pane. The **Connect S3 Target** pop-up window appears.
2. Enter the name of the S3 target and its address.
3. Click **Connect**.

Updating an S3 Target

The names and addresses S3 targets can be updated in the **S3 Target Connections** panel of the **Array** page.

To update an S3 target, perform the following:

1. On the source array, navigate to the **Storage** > **Array** page.
2. In the **S3 Target Connections** pane, click the **Edit** button located at the end of the row of the S3 target you wish to update. The **Edit S3 Target** pop-up window appears.
3. Enter a new name and/or new address.
4. Click **Save**.

Disconnecting an S3 Target

Disconnecting an S3 target from a FlashBlade array is performed in the **S3 Target Connections** panel of the **Array** page.

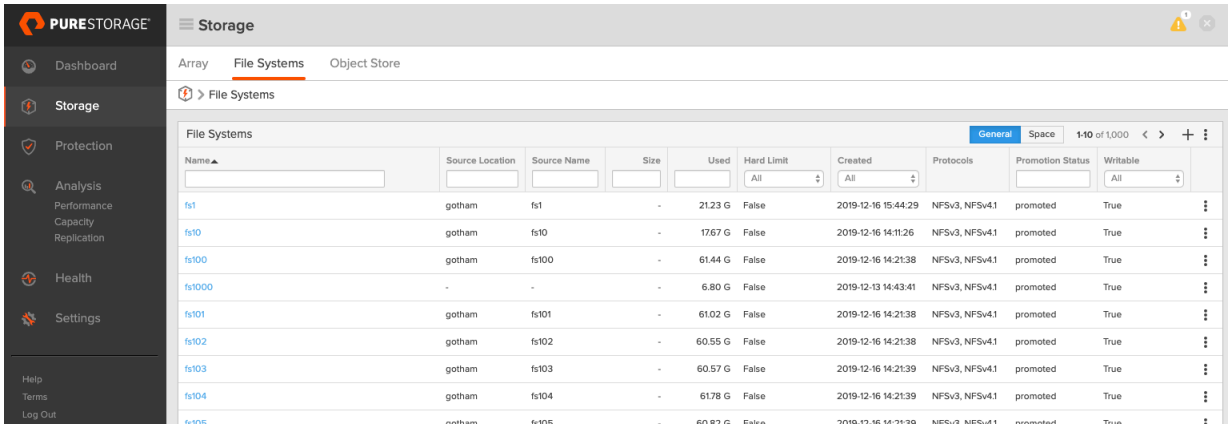
To disconnect an S3 target, perform the following:

1. On the source array, navigate to the **Storage** > **Array** page.
2. In the **S3 Target Connections** pane, click the Disconnect (**x**) button located at the end of the row of the S3 target you wish to disconnect. The **Disconnect S3 Target** pop-up window appears.
3. Click **Disconnect**.

File Systems

The **File Systems** tab displays information for all file systems on the array.

Figure 5.5 File Systems Tab



The **File Systems** panel displays configured file system information while the **Destroyed File Systems** panel displays destroyed file systems that are pending eradication.

File Systems Panel

By default general information about each file system is displayed, which includes the following information:

- **Name**: File system name. Clicking a file system name displays any snapshots belonging to that file system, and also provides the option of creating snapshots.
- **Source Location:** The location where the file system was created.
- **Source Name:** If created on the target array, the name of the source file system.
- **Size**: Provisioned size of the file system. If not set, the provisioned size is unlimited.
- **Used**: Total amount of pre-compressed data written to the file system.
- **Hard Limit**: Whether the file system's provisioned size has a hard limit (true) or not (false).
- **Created**: The file system creation date.
- **Protocols**: One or more protocols configured for the file system. A dash (-) means the file system has no protocols assigned.
- **Promotion Status**: Whether the source or replicated file system is promoted or demoted.
- **Writable**: Whether the file system is writeable (true) or not (false).

The size, rules, and protocols of each file system can be configured.

You can alternately click **Space** to view file system space information. The columns in the **File Systems** panel display the following information:

- **Name**: File system name. Clicking a file system name displays any snapshots belonging to that file system, and also provides the option of creating snapshots.
- **Size**: Provisioned size of the file system. If not set, the provisioned size is unlimited.
- **Used**: Total amount of pre-compressed data written to the file system.
- **% Used**: Percentage of provisioned size occupied by uncompressed system data. The percentage value is calculated as `Used/Size`.
- **Data Reduction**: Ratio of the size of the written data (i.e. Used) versus the amount of space the data occupies after data compression (i.e. Physical).
- **Physical**: Amount of space that the written data occupies on the array after reduction via data compression.
- **Snapshots**: Amount of space occupied by snapshots after reduction from data compression.
- **Total**: The total used space (physical space and snapshots).

You can search the file systems list by entering search criteria in the search box under each column.

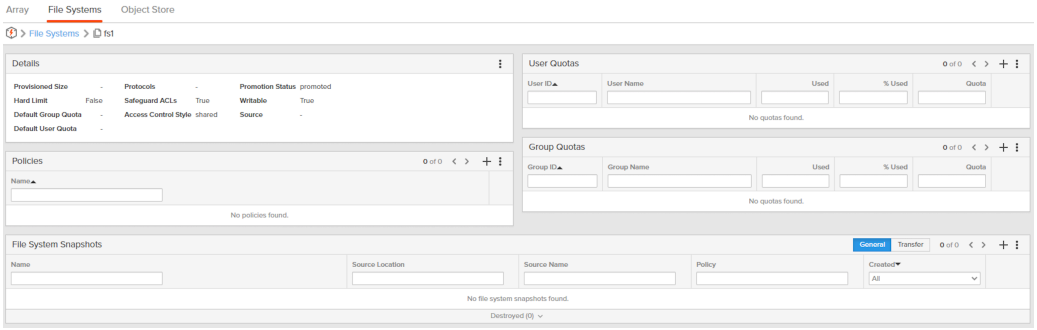
Clicking any file system directs you to that file system's details page.

In addition to displaying file systems, the **File Systems** panel allows you to create and maintain file systems.

File System Details

A file system's details page is accessed by clicking a file system from the **File Systems** panel.

Figure 5.6 File System Details Page



The file systems detail page provides the following panels:

- **Details**: Provides an overview of the file system's configured parameters, including: the file system's provisioned size, default group and quota sizes, if hard limits is enabled (True or False), export protocols, if ACLs safeguarding is enabled (True or False), the applied access control style, the file sytem's promotion status, if the file system is writable, and its source (for replication setups).

- **Policies**: Lists all policies applied to the file system and provides management of policies.
- **User Quotas**: Lists all configured user quotas and provides management of user quotas.
- **Group Quotas**: Lists all group quotas and provides management of group quotas.
- **File System Snapshots**: Lists all file system snapshots and provides management of snapshots.
- **Destroyed Snapshots**: Lists all destroyed snapshots and allows you to recover or eradicate destroyed snapshots.

Destroyed File Systems Panel

The **Destroyed File Systems** panel is located directly below the **File Systems** panel. Clicking the arrow beside the **Destroyed** heading expands the panel.

Once a file system has been destroyed, it is moved to the **Destroyed File System** panel making it inaccessible to clients. This is also known as an eradication pending period. During this period the file system and snapshot is left intact for 24 hours. If the destroyed file systems are not recovered within the 24-hour period, they will be eradicated and their space will be reclaimed by the array. If you need to reclaim space before the 24 hour expiration, file systems can be manually eradicated.

- **Name**: File system name.
- **Time Remaining**: Indicates the file systems expiration period in the eradication pending period. Time is represented in hours (**h**) and minutes (**m**) . When time expires the file systems will be eradicated and their space will be reclaimed by the array.

If a file system is destroyed, its snapshots are also destroyed and will appear in the **Destroyed Snapshots** panel. Those snapshots cannot be recovered unless the file system is recovered.

About Creating and Exporting File Systems

Creating and exporting a file system involves creating the file system and enabling protocol support for the file system. FlashBladefile systems support the Hypertext Transfer Protocol (HTTP), the Network File system (NFS), and Server Message Block (SMB) protocols.

Exporting a file system requires at least one data virtual IP address (data vip). The data vip can be created before or after creating the file system. Data vips are managed through the the **Settings > Network** page. For more information about data vips, refer to *Subnets*.

For more information about using directory services with file system protocols, refer to *Configuring Directory Services*.

When creating a file system, consider which protocol(s) you want to configure:

- **NFS**. File sharing over NFS requires that you configure the NFS export rules and then enable the protocol. Per NFS limitations, each user can belong to a maximum of 16 groups. If users belong to

more than 16 groups and a directory service contains the group information, configure file sharing over NFS with a directory service.

Follow these steps to set up file sharing over NFS without a directory service:

1. Create the data vip. Data vips are managed through the **Settings** > **Network** page.
2. Create the file system. Optionally set a provisioned size for the file system.
3. Define the NFS export rules.
4. Enable the NFS protocol.

- **NFS with a directory service**. For file sharing over NFS where a user might belong to more than 16 groups, configure the FlashBlade directory service to integrate with a directory server. This assumes that you already have a directory service, such as OpenLDAP or Active Directory set up.

  Follow these steps to set up file sharing over NFS with a directory service:

  1. Configure the directory service for protocol support. The directory service is managed through the **Settings** > **Users** page.
  2. Create the data vip. Data vips are managed through the **Settings** > **Network** page.
  3. Create the file system. Optionally set a provisioned size for the file system.
  4. Define the NFS export rules.
  5. Enable the NFS protocol.

- **SMB with Active Directory**. FlashBlade offers the following SMB authentication modes:

  - **AD RFC2307.** Recommended for customers where the same data needs to be accessed over SMB and NFS, and where access control is required. For more information, refer to the RFC 2307 specifications set by the Internet Engineering Task Force.
  - **AD Auto.** Recommended for customers with SMB-only clients or when SMB and NFS clients never access the same file systems.

  Before you enable the SMB protocol adapter, consider the following limitations:

  - Due to the nature of the SMB protocol, disruptions occur when a blade is added or removed, an upgrade is performed, or the system encounters network connectivity issues. Therefore, your applications must be able to tolerate I/O errors and disruptions.
  - File system permissions default to traditional UNIX-like permissions with an option to select the `native` ACL mode to include support for Windows ACLs. The SMB Access Control List (ACL) is

restricted to the following three entries: the file **owner**, the primary **group** to which the file owner belongs, and **others**.

- For file sharing over SMB, the FlashBlade array must be integrated with an Active Directory service. Before creating the file system, configure the directory service. For more information about the directory service, refer to *Configuring Directory Services*.

For more information about these and other limitations related to the SMB protocol, refer to the **FlashBlade SMB - Limitations in Purity//FB 2.x** KB article at *https://support.purestorage.com/FlashBlade/Purity////FB*.

The default SMB authentication mode is AD RFC2307. To configure file sharing over SMB in other authentication modes, such as AD Auto mode, contact Pure Storage Support.

Follow these steps to set up file sharing over SMB in AD RFC2307 mode:

1. Prepare Active Directory for multi-protocol support.

   Prepare Active Directory for multi-protocol support by setting the `gidNumber` and `uidNumber` attributes for each domain user and group that will be accessing the SMB shares. The `gidNumber` and `uidNumber` attributes can be set through the Microsoft Identity Management for UNIX service or PowerShell.

2. Configure the directory service for protocol support.

   The FlashBlade array must be integrated with an Active Directory service. The directory service is managed through the **Settings** > **System** page. For more information about the directory service, refer to *Directory Service*.

3. Create the data vip.

   Data vips are managed through the **Settings** > **Network** page.

4. Create the file system. Optionally set a provisioned size for the file system.

5. Enable the embedded SMB protocol adapter.

- **HTTP access**. Enabling the HTTP protocol turns on HTTP and HTTPS access to a file system. Before you enable the HTTP protocol, consider the following access control limitations:

  - All HTTP APIs by default are executed on behalf of a root user.
  - Certain HTTP APIs provide the ability to specify a user by specifying the UID as part of the request.
  - Access checking in HTTP is based solely on the client's crafted request. Any user can read, modify, and delete any files on a file system when HTTP is enabled.
  - The use of HTTP connections, as opposed to HTTPS connections, potentially gives unauthenticated users access to the file system. Anyone with network access to the data vip has access to the file system.

  Follow these steps to set up file system for HTTP access:

  1. Create the data vip. Data vips are managed through the **Settings** > **Network** page.
  2. Create the file system. Optionally set a provisioned size for the file system.
  3. Enable the HTTP protocol.

File System Size

File system size represents the provisioned size allocated to a file system. The size is a quota of space that is set to help gauge the fullness of a file system.

When the amount of data written to the file system reaches a certain percentage of its quota, alerts are generated notifying administrators of the threshold reached. For example, Purity//FB generates a `WARNING` alert when the amount of data written to the file system reaches 90% and then 100% of its provisioned size. This behavior differs slightly if a hard limit is set on the file system. When the file system usage reaches 90% of its provisioned size, the system generates a `WARNING` alert. When the file system usage reaches 100% of its provisioned size, the system generates a `CRITICAL` alert. See **File System Usage Limits** for more information. To view alert details, select **Health** > **Alerts**.

The file system size can be blank or set to any value between `0` and `8191` petabytes. If the field is left blank or set to `0`, the provisioned size will default to an unlimited size.

Purity//FB does not enforce any restrictions nor does it take any further action when space consumption reaches or exceeds the provisioned size. To address the issue, decrease the percentage of storage space used by either deleting files or increasing the size of the file system. In contrast, if a hard limit is set for the file system, all writes are halted once the provisioned size is reached. See **File System Usage Limits** for details.

File system sizes can be changed at any time.

File System Usage Limits

⚠ **Important note!** File system hard limits cannot be enabled for file systems with unlimited sizes.

To help manage resource usage, a hard limit can be enabled on the provisioned size of a file system. Enabling a hard limit effectively limits writes of the file system to its provisioned size. The hard limit can be enabled when creating or editing a file system.

When the system detects the file system has exceeded its provisioned size, all future writes are halted until usage decreases below the provisioned size. This occurs within minutes of the provisioned size being reached. Note that the provisioned size can be exceeded during this detection period when a hard limit is enabled. For example, a 5G write is in-process for a file system whose provisioned size is 10G and current usage is 9G. When the system halts the write, the file system ’s provisioned size will be slightly exceeded. Data writes only resume after data has been managed (deleted or truncated) to below the file system's provisioned size.

A hard limit cannot be enabled if the file system usage has already exceeded its provisioned size unless forced. You can force the hard limit by using the **Edit File System** dialog box from the **File Systems** screen

Additionally, if a hard limit is enabled, the file system ’s size cannot be reduced to a value less than its current space usage. You can force shrink the file system ’s size by using the **Edit File System** dialog box from the **File Systems** screen.

You can disable a hard limit from the **Edit File System** dialog box on the **File Systems** screen.

User and Group Usage Limits

Similar to file system hard limits, hard limits (quotas) can be set for users and groups on a per- file system basis to help manage file system resource usage.

When the system detects that a user or group has exceeded its quota, all future writes to the file system are halted until usage decreases below the set quota size via deletion or truncation of that user or group's data from the file system. Note that a quota can be exceeded during this detection period. For example, if a user is writing a large file that causes them to exceed the quota, their write will fail, but their total usage may have slightly exceed their set quota.

During file system creation, a default group quota and/or user quota can be set, which is applied to all users and/or groups in the file system. Additionally, once a file system has been created, quotas can be set per individual user ID or group ID (quotas can be set explicitly) for that file system. Note that explicit quotas take priority over default quotas. For example, if a file system was created with default user or group quota and then an explicit quota was set for a specific group or user of that file system, the explicit quota is used. If that explicit quota is later deleted, the default quota then takes effect.

User and group quotas are displayed on the **Details**, **User Quotas**, and **Group Quotas** panels from a file system 's detail screen. To access the detail screen, from the **File Systems** panel, click the file system whose details you wish to view.

NFS Export Rules

The NFS export rules define the access rights and privileges that an NFS client has to the file system. An NFS client's access rights and privileges include rules for users and groups. Any NFS export rule changes will be synchronously applied to all currently mounted clients. Note that NFS rules apply to both NFSv3 and NFSv4.1. The NFS export rules of a file system can be changed at any time.

Note that User and Group IDs that are zero (0) have root privilege and IDs that are non-zero do not have root privilege.

The default export rules will be applied to NFS clients if none are specified:

- `ro`
- `root_squash`
- `no_all_squash`
- `no_fileid_32bit`
- `anonuid=65534`
- `anongid=65534`
- `atime`
- `insecure`

Rules are applied in the form of `host(options)`, where `host` is an IP address or netmask and `options` are export rules enclosed in parenthesis.

The RULES argument represents the export rules that apply to the file system. Options can be applied to a single or multiple IP addresses, Netmasks, or LDAP group. See the syntax examples:

- Options applied to a single client: `'host(options)'`
- Options applied to multiple clients: `'-options host1 host2…'`
- Represent all clients with an asterisk: *

The `host` parameter must belong to one of the following categories:

- IPv4 IP address: `ddd.ddd.ddd.ddd`
  IPv4 Netmask: `ddd.ddd.ddd.ddd/dd`
- IPv6 IP address: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`
  IPv6 Netmask: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/ddd`

Valid export rules are listed below:

Export Rule	Rules Description
`ro`	Read-Only grants an NFS client and their users or groups the privilege to read the file system.
`rw`	Read-Write grants an NFS client and their users or groups the privilege to read and write to the file system.
`fileid_32bit`	Allows 32-bit inode support for clients.
`nofileid_32bit`	Disables 32-bit inode support for clients.
`root_squash`	Prevents client users and groups with root privilege from mapping their rootprivilege to a file system. All users with UID 0 will have their UID mapped to `anonuid`. All users with GID 0 will have their GID mapped to `anongid`.
`no_root_squash`	Allows root users and groups to access the file system with root privilege.
`all_squash`	Maps all UIDs (including root) to anonuid and all GIDs (including root) to anongid.
`no_all_squash`	Prevents the remapping of user and group IDs to `anonuid 65534` or `anongid 65534`. All users and groups will retain their IDs unless `root_squash` is also specified.
`anonuid`	Any user whose UID is affected by `root_squash` or `all_squash` will have their UID mapped to `anonuid`. The default `anonuid` is `65534`.
`anongid`	Any user whose GID is affected by `root_squash` or `all_squash` will have their GID mapped to `anongid`. The default `anongid` is `65534`.
`atime`	After a read operation has occurred, the inode access time is updated only if any of the following conditions is true: the previous access time is less than the inode modify time, the previous access time is less than the inode change time, or the previous access time is more than 24 hours ago.
`noatime`	Disables the update of inode access times after read operations.
`secure`	Prevents NFS access to client connections coming from non-reserved ports. Applies to NFSv3, NFSv4.1, and auxiliary protocols MOUNT and NLM.

Export Rule	Rules Description
`insecure`	Allows NFS access to client connections coming from non-reserved ports. Applies to NFSv3, NFSv4.1, and auxiliary protocols MOUNT and NLM.

Multiple rules may be specified by separating each rule with a space. For example,

```
10.60.50.83/32(rw,root_squash) 2620::/16(ro)
```

The following guidelines apply when creating NFS export rules:

- If rules are not defined before enabling the NFS protocol, then all clients will have `rw` access and `no_root_squash` privileges to the file system. This is equivalent to specifying export rule `*(rw,no_root_squash)`.
- Omitting the `(options)` portion of a rule is equivalent to specifying `(ro,root_squash)`. For example, an export rule that is defined as `*` will have `read-only, root_squash` access. This is equivalent to specifying export rule `*(ro,root_squash)`.
- If the rule is defined but one or more export options are not specified, then the undefined option(s) will default to `ro` access and `root_squash` privileges. For example, an export rule that is defined as `*(rw)` will have `root_squash` privileges. Likewise, an export rule that is defined as `*(no_root_squash)` will have `ro` access.
- A rule cannot include conflicting options. For example, `ro` and `rw` cannot be included in the same rule. Likewise, `root_squash` and `no_root_squash` cannot be included in the same rule.
- If a client IP address matches multiple rules, Purity//FB uses the first rule it encounters in priority based on rule category. IP address rules have the highest priority. Specifically, the priority of matching rules is as follows: IP address > Netmask > Wildcard.
- If an address matches multiple rules in the same category, then the first (leftmost) rule applies.
- By default, the FlashBlade file systems use 64-bit inode numbers. If your environment contains clients that can only support 32-bit inode numbers, add a third export option named `fileid_32bit` to the NFS export rule of the file system that requires 32-bit inode support. For example, `*(rw,no_root_squash,fileid_32bit)`

Here is an example of a file system list output taken from the Purity//FB CLI with various NFS export rules set for each file system:

```
purefs list --protocol-specific nfs
Name    Protocols Rules
File1   nfs       *
File2   nfs       *()
File3   nfs       -
File4   nfs       10.20.225.2(rw)
File5   nfs       2620::/16(no_root_squash)
File6   nfs       *(rw,no_root_squash)
```

```
File7   nfs       *(rw,root_squash,fileid_32bit)
File8   nfs       10.20.30.40(root_squash,anonuid=7777,anongid=8888)
```

In the list output above, the NFS export rules are described as follows:

Export Rules	Rules Description
`*`	All clients have default `ro` access and `root_squash` privileges to the file system.
`*()`	All clients have default `ro` access and `root_squash` privileges to the file system.
`-`	Export rules have been deleted, rendering the file system inaccessible.
`10.20.225.2(rw)`	Client `10.20.225.2` has `rw` access and `root_squash` (default) privileges.
`2620::/16(no_root_squash)`	Client `2620::/16` has `ro` (default) access and `no_root_squash` privileges.
`*(rw,no_root_squash)`	All clients have `rw` access and `no_root_squash` privileges. This is the default rule used when creating a file system.
`*(rw,root_squash,fileid_32bit)`	All clients have `rw` access and `root_squash` privileges. The file system also supports clients that require 32-bit inode support.
`10.20.30.40` `(root_squash,anonuid=7777,anongid=8888)`	Client 10.20.30.40 has `rw` access. Any access attempts with UID 0 will be mapped to UID 7777. Any access attempts with GID 0 will be mapped to GID 8888.

Netgroups

Purity//FB supports netgroups with LDAP and NIS.

Netgroups allows a set of hosts to be referenced as a single entity, or a *netgroup*.

Netgroups are used in export rules to grant NFS mount attributes to an entire set of hosts within a netgroup. This allows you to change the netgroup membership and affect every reference to that netgroup.

Netgroups are stored in LDAP as objects with the class nisNetgroup. Each netgroup has a name and may contain an unlimited number (or none) of member netgroups (sub-netgroups), as well as an unlimited number (or none) of tuples.

On a network using NIS, the ***`netgroup`*** input file on the master NIS server is used for generating the ***`netgroup`*** map. The ***`netgroup`*** map contains the basic information in the ***`netgroup`*** input file, which includes information in a format that speeds lookups of netgroup information among hosts.

FlashBlade queries directory service for a given netgroup, following any member netgroups and reading every tuple unitl the client attempting to mount the NFS file system is encountered within the netgroup, or until all referenced hosts are exhausted.

Referencing a netgroup with an export group must be expressed in the format `@<netgroup_name>(<export_rule>)`. The `@` indicates the specified `netgroup_name` is a netgroup. `(<export_rule>)` specifies the export rule to which the netgroup is referenced. Multiple export rules can be specified in a comma-separated list. Parentheses are required.

For example, `@netgroup1(rw,no_root_squash)` indicates that every host mentioned in the netgroup (`netgroup1`), may mount the given file system with read-write (`rw`) and root privilege (`no_root_squash`) attributes for that file system.

File System Snapshots

A file system snapshot is an immutable, point-in-time image of a file system. A snapshot can be created for any file system at any time. Snapshots are named after the file system it is copying with an assigned suffix to differentiate it from the source file system. A suffix is automatically assigned to a snapshot if one is not given at the time of creation.

Each file system has special directories named `.snapshot` that by default are enabled at file system creation to provide access to stored snapshots at the root and sub-directory level of that file system. The `.snapshot` directory provides a virtual repository for snapshots of a specified file system. By enabling a file system 's snapshot directory, the `.snapshot` directory will be visible in the root directory and accessible at the sub-directory level of that file system. The `.snapshot` directories allow access to their parent-level directories. All snapshots for a file system will be accessible in the `.snapshot` directory at the root level. The `.snapshot` directories at the sub-directory level allow access to snapshots taken after those sub-directories were created. For example, all snapshots are accessible through the root `/.snapshots` directory because it was created before any snapshots were taken (during the creation of the file system) . At the `/lolcats/.snapshot` sub-directory, snapshots taken before `/lolcats/` was created are not accessible.

The `.snapshot` directory can be optionally disabled. Once disabled, historical snapshots cannot be viewed and will not be accessible from the mount points until the directory feature is re-enabled. Disabling the the directory will not destroy any snapshots, they must be manually destroyed.

The `.snapshot` directories cannot be renamed, deleted, or moved into other directories. The directory name is reserved and you cannot create a regular directory with the `.snapshot` name, regardless of the enabled/disabled status.

ACLs

An Access Control List (ACL) is a list of security permissions associated with a directory or file. Specific rules can be added to the ACL to define who can access the file and what actions they can perform. When an ACL is applied to a directory or file, then that directory or file will reference its list of rules to determine whether to grant or deny access, including which actions are allowable, such as; read, write, and execute.

ACEs

An ACL consists of up to 1,820 Access Control Entries (ACEs), plus three OGE (owner, group, everyone) entries. Note that certain clients may have lower limits from what the server supports.

Each ACE contains detailed permissions information. ACEs consist of four fields separated by colons — type, flag, principal, and permissions.

- **type** - if the defined permissions are allowed (A) or denied (D).
- **flag** - defines both child object inheritance and if the ACE is defining group permissions.
- **principal** - defines to whom the permissions apply.
- **permissions** - defines the permissions that the specified principle will be allowed or denied.

For example, the ACE `A:d:user@purestorage.com:R` means that the user `user@purestorage.com` is allowed (A) read permissions (R) and that new sub-directories will inherit this ACE (d).

Access Control Styles

For implementations where more than one file system export protocol is enabled, Purity//FB 3.1.1 and later allows you to set the file system's access control style to dictate how the protocols will interact.

The following access control styles are available:

- **Shared** - NFS and SMB clients are both able to set permissions on, and access files and directories over any protocol.
- **NFS** - Only NFS clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- **SMB** - Only SMB clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- **Independent** - NFS and SMB clients are both able to set permissions on files and directories and can access files and directories created over any protocol. Permissions set by SMB clients will not affect NFS clients and vice versa. NFS clients will be restricted to only using mode bits to set permissions.

- **Mode Bits** - NFS and SMB clients are both able to set permissions on files and directories, but permissions are enforced in mode bits-only format. When SMB clients set an ACL it will be converted to mode bits.

Note the following behaviors and restrictions:

- If you set the access control style to **SMB** and then disable the SMB protocol, the access control style is automatically reset to **Shared**.
- If you set the access control style to **NFS** and then disable both NFSv3 and NFSv4.1, the access control style is automatically reset to **Shared**.
- If neither the NFSv3 or NFSv4.1 protocols are enabled, the access control style cannot be set to **NFS**.
- If the SMB protocol is not enabled, the access control style cannot be set to **SMB** or **Independent**.

File system attributes that were set prior to Purity//FB 3.1.1 will be transitioned to the following access control styles as described in the following table with Purity//FB 3.1.1 and later:

Table 5.3 Access Control Style Transitions

Pre Purity//FB 3.1.1 File System Attribute	Purity//FB 3.1.1 and Later Access Control Style
n/a	SMB
n/a	NFS
n/a	Shared
SMB native, NFSv3, NFSv4.1	Independent
SMB shared	Mode Bits

ACL Safeguarding

If using the **Shared** or **NFS** access control styles, consider enabling ACL safeguarding.

When using the SMB access control style, ACL safeguarding is automatically enabled.

ACL safeguarding controls how a mode bit change will impact the ACL on the file or directory. When disabled, modification of mode bits will result in the removal of any existing ACL. When enabled, modification of mode bits will result in the modification of the existing ACL to merge the permissions from the mode bits with the permissions in the ACL.

Note: ACL safeguarding is not available for **Independent** or **Mode Bits** access control styles.

Setting the access control style and enabling ACL safeguarding are performed when creating a file system and can be modified at a later date.

NFS and File Locking

The Network Lock Manager (NLM) protocol works with the NFS protocol to ensure file locks are visible across all NFS clients for I/O coordination and to help clients coordinate access to files. The NLM protocol allows all NFS clients mounting the same NFS shared file system to see file locks set by other clients.

NLM locks are considered advisory locks in that an NFS client application must check for the existence of a lock in order to coordinate access; the NFS service itself does not automatically prevent a client from accessing a file if it has the security permission to do so, and the service does not check for the existence of or try to obtain a lock on a file ahead of time.

The NLM service is enabled by default with the NFS protocol service and cannot be disabled.

Fast Remove

When a directory and its contents are no longer required, they can be removed to reclaim space.

Removing the contents of a directory recursively by running the `rm -r` command causes the client to delete files one at a time. This can be a time consuming and expensive operation.

The fast remove feature allows you to quickly remove large directories by offloading this work onto the server. When the fast remove feature is enabled, a special pseudo-directory named `.fast-remove` is created in the root directory of the NFS mount. To remove a directory and its contents, run the `mv` command to move the directory into the `.fast-remove` directory.

In the following example, from the root directory of the NFS mount, a directory called `big_dir` is being moved into the `.fast-remove` directory.

```
mv big_dir/.fast-remove/
```

Once a directory has been moved into the `.fast-remove` directory, it is immediately destroyed, and through background operations, the space it occupied will be freed.

The fast remove feature can be enabled or disabled at any time. Fast remove is disabled by default.

After you enable the fast remove feature, view the contents of the root directory of the NFS mount to verify that the `.fast-remove` directory has been created.

For example,

```
//Run on the NFS mount to view the content of the root directory.
ls -al
drwxrwxrwx   1  root  root      0 Oct 30 10:46 .
drwxrwxrwx  25  root  root  12288 Jun 30 16:44 ..
```

```
drwxr-xr-x   1  root  root      0 Oct 30 11:27 .fast-remove
…
```

The `.fast-remove` directory cannot be renamed, deleted, or moved into other directories. The `.fast-remove` directory name is a reserved name, so you cannot create a regular directory named `.fast-remove`, regardless of the fast remove enabled/disabled status.

You cannot move individual files to the `.fast-remove` directory. To remove a file, either use the `rm` command or move the directory enclosing the file to the `.fast-remove` directory.

The `.fast-remove` directory can only be removed by disabling the fast remove feature.

The fast remove feature is powerful and comes with the following cautionary notes:

- Once a directory has been moved into the `.fast-remove` directory, it is no longer recoverable.
- With fast remove, users can remove any files that are contained in a directory that they can rename, even if they do not have read, write, or execute permissions on the child directories. This breaks from the standard UNIX permissions model. To restrict access to the `.fast-remove` directory, set the directory permissions.

By default, the `.fast-remove` directory permissions are set to `drwxr-xr-x`. These permissions can be changed to grant or restrict access to the `.fast-remove` directory. Disabling and then re-enabling the fast remove feature will reset the settings to the default permissions.

File Replication Failover, Reprotect, and Failback

In file system replication setups you can stop writes to the local file system and redirect writes to the target file system (failover). Failover can be used if there is an issue with the local file system. During failover, replication stops until you reprotect.

Failover requires promotion of the target file system and redirection of all clients to the target array to return to operation.

Note that when demoting the local file system, any writes performed since the last replication will be discarded. If a local snapshot was taken since last replication, demoting the file system requires that snapshot be destroyed and eradicated. You must remove any data since the last snapshot of the file system was taken. The file system must either have no snapshots, or the most recent snapshot must be the last replicated snapshot. Additionally, it is suggested that any policies attached to the local file system and replica link be removed in order to expedite failover.

Once the local file system comes back up, re-enable the scheduled replication policy that was disabled before demoting the file system. Replication resumes as scheduled between the promoted remote file system and the demoted local file system. Data is written to the promoted remote file system, which is then replicated to the demoted local file system.

To resume the original replica link direction (failback), manually stop writes on the remote file system and send the remaining data to the local file system. Once data is sent to the local file system, promote the local file system and demote the remote file system.

Creating and Exporting a File System

Creating and exporting a file system involves creating the file system and configuring protocol support for the file system. A file system can support multiple protocols.

Before you create and export a file system, verify that a data vip has been created. Clients connect to the file system via the data vip. Data vips are managed through the **Settings** > **Network** page.

If you are integrating the FlashBlade array with a directory service, also verify that the directory service has been properly configured and enabled. The directory service is managed through the **Settings** > **Users** page.

If you are enabling the SMB protocol in AD RFC2307 (default) mode, prepare Active Directory for multi-protocol support by setting the `gidNumber` and `uidNumber` attributes for each domain user and group that will be accessing the SMB shares. The `gidNumber` and `uidNumber` attributes can be set through the Microsoft Identity Management for UNIX service or PowerShell.

To create and export a file system:

1. Create the file system.
   - **a** From the **Storage** > **File Systems** page, click the add (+) button in the heading of the **File Systems** list. The **Create File System** pop-up window appears.
   - **b** In the **Name** field, type the name of the directory to be exported.
   - **c** In the **Provisioned Size** field, specify the provisioned size allocated to the file system. The size is a quota of space that helps gauge the fullness of the file system. If left blank, the provisioned size will default to an unlimited size. To enable a hard limit for the file system to prevent exceeding the file system's provisioned space, set the **Hard Limit** toggle button to enable (blue).

     For more information about the provisioned size, refer to the **File System Size** section.

     ⚠ **Important note!** File system hard limits cannot be enabled for file systems with unlimited sizes.
   - **d** (Optional) In the **Default User Quota** field, specify the user quota size.
   - **e** (Optional) In the **Default Group Quota** field, specify the group quota size.
   - **f** To enable the ability to quickly remove large directories using the fast remove feature, set the **Fast Remove** toggle button to enable (blue). For more information about the fast remove feature, refer to the **Fast Remove** section.
   - **g** To enable the ability to access stored snapshots in a snapshot directory, set the **Snapshots** toggle button to enable (blue).

2 Perform the following protocol-specific configurations:

- **NFS**. Define the NFS export rules so the file system is accessible to the appropriate clients.

  a From the **Create File System** pop-up window, click **NFS** in the **Protocols** section.

  b Click the **NFSv3** or **NFSv4.1** toggle button to enable (blue).

  c In the **Export Rules** field, configure the NFS export rules to define the access rights and privileges a client has to the file system.

  The format is `host(options)`.

  For IPv4 addresses, `host` represents `ddd.ddd.ddd.ddd` for IP address, `ddd.ddd.ddd.ddd/dd` for netmask, or `*(options)` for all clients.

  For IPv6 addresses, `host` represents `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` for IP address, `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx` for netmask, or `*(options)` for all clients.

  The `options` within parenthesis represents a comma-separated list of NFS export options. Valid export options are `rw`, `ro`, `root_squash`, `no_root_squash`, and `fileid_32bit`. For example, `10.20.225.2(ro,root_squash)`.

  For more information about export rules, refer to the **NFS Export Rules** section.

- **SMB**. By default, file sharing over SMB is configured in AD RFC2307 mode for mixed Windows/ UNIX environments. To configure file sharing over SMB in other authentication modes, such as AD Auto mode, contact Pure Storage Support.

  a From the **Create File System** pop-up window, click **SMB** in the **Protocols** section.

  b Click the **SMB** toggle button to enable (blue) the SMB protocol adapter.

- **HTTP**. Enable HTTP and HTTPS access to a file system.

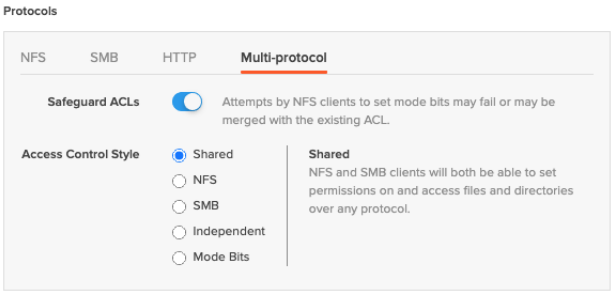
  a From the **Create File System** pop-up window, click **HTTP** in the **Protocols** section.

  b Click the **HTTP** toggle button to enable (blue) the HTTP protocol.

3 If you selected multiple export protocols, set the access control style. The access control style governs the interactions between protocols. From the **Create File System** pop-up window, click **Multi-protocol** in the **Protocols** section.

Note: With Purity//FB 3.1.1 and later, by default, all new file systems created are created with the **Shared** access control style with ACL safeguarding enabled.

Figure 5.7 Selecting an Access Control Style



Available access control styles are:

- **Shared** - NFS and SMB clients are both able to set permissions on, and access files and directories over any protocol.
- **NFS** - Only NFS clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- **SMB** - Only SMB clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- **Independent** - NFS and SMB clients are both able to set permissions on files and directories and can access files and directories created over any protocol. Permissions set by SMB clients will

not affect NFS clients and vice versa. NFS clients will be restricted to only using mode bits to set permissions.

- **Mode Bits** - NFS and SMB clients are both able to set permissions on files and directories, but permissions are enforced in mode bits-only format. When SMB clients set an ACL it will be converted to mode bits.

Note the following behaviors and restrictions:

- If you set the access control style to **SMB** and then disable the SMB protocol, the access control style is automatically reset to **Shared**.
- If you set the access control style to **NFS** and then disable both NFSv3 and NFSv4.1, the access control style is automatically reset to **Shared**.
- If neither the NFSv3 or NFSv4.1 protocols are enabled, the access control style cannot be set to **NFS**.
- If the SMB protocol is not enabled, the access control style cannot be set to **SMB** or **Independent**.

(Optional) If **Shared** or **NFS** are selected, you can click the Safeguard ACLs toggle to enable (blue) ACLs safeguarding.

ACL safeguarding controls how a mode bit change will impact the ACL on the file or directory. When disabled, modification of mode bits will result in the removal of any existing ACL. When enabled, modification of mode bits will result in the modfication of the existing ACL to merge the permissions from the mode bits with the permissions in the ACL. Note the following:

- If the access control style was set to **SMB**, ACL safeguarding is automatically enabled. Disabling ACL safeguarding is not allowed.
- ACL safeguarding is not available for **Independent** or **Mode Bits** access control styles.

4 Click **Create**.

Changing the NFS Export Rules of a File System

NFS export rule changes will be synchronously applied to all currently mounted clients.

To change the NFS export rules of a file system:

1 From the **Storage** > **File Systems** page, click the **Edit** button in the row of the file system you want to edit. The **Edit File System** pop-up window appears.

2 Click **NFS** in the **Protocols** section.

3 In the **Export Rules** field, configure the NFS export rules to define the access rights and privileges a client has to the file system.

   The format is `host(options)`.

   For IPv4 addresses, `host` represents `ddd.ddd.ddd.ddd` for IP address, `ddd.ddd.ddd.ddd/dd` for netmask, or `*(options)` for all clients.

   For IPv6 addresses, `host` represents `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` for IP address, `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx` for netmask, or `*(options)` for all clients.

   The `options` within parenthesis represents a comma-separated list of NFS export options. Valid export options are `rw`, `ro`, `root_squash`, `no_root_squash`, `all_squash`, `no_all_squash`, `anonuid`, `anongid`, `fileid_32bit`, and `no_fileid_32bit`. For example, `10.20.225.2(ro,root_squash)`.

4 Click **Save**.

Changing the Provisioned Size of a File System

To change the provisioned size of a file system:

⚠ **Important note!** If you are reducing a file system's provisioned size, but had previously enabled the hard limit for the file system and have already exceeded the new size, a warning will appear indicating subsequent data writes will fail. You must click **Update** to proceed with the operation.

1 From the **Storage** > **File Systems** page, click the **Edit** button in the row of the file system you want to edit. The **Edit File System** pop-up window appears.

2 In the **Provisioned Size** field, set the size to any value between `0` and `8191` petabytes. If the field is left blank or set to `0`, the provisioned size will default to an unlimited size and a hard limit cannot be set.

3 Click **Save**.

Modifying the Default User and Group Quotas

1 From the **Storage** > **File Systems** page, click the **Edit** button in the row of the file system you want to edit. The **Edit File System** pop-up window appears.

2 In the **Default User Quota** field, enter a new user quota size.

3 In the **Default Group Quota** field, enter a new group quota size.

4 Click **Save**.

Creating an Explicit User Quota

⚠ **Important note!** If a default user quota was created during file system creation, the explicit user quota will take priority over the default quota and will be used for that file system.

1. From the **Storage** > **File Systems** page, click the file system for which you wish to set an explicit user quota. The file system's details page appears.
2. In the **User Quotas** panel, click the add button (+) . The **Add User Quota** pop-up window appears.
3. Enter the user ID or user name, and quota size.
4. Click **Add**.

Modifying an Explicit User Quota

1. From the **Storage** > **File Systems** page, click the file system whose explicit user quota(s) you wish to modify. The file system's details page appears.
2. From the **User Quotas** panel, click the **Edit** button on the same row as the explicit user quota you wish to modify. The **Edit User Quota** pop-up window appears.

   If you wish to modify multiple explicit user quotas, click **More Options** > **Edit Quotas**. The **Edit User Quotas** pop-up window appears. From the **Existing Quotas** list, select the user quotas you wish to modify. You can optionally enter a partial name in the search box to filter the list. You can also optionally select all user quotas by clicking the checkbox next to the search box. All selected user quotas appear in the **Selected Quotas** list.
3. Enter a new quota size.
4. Click **Save**.

Deleting an Explicit User Quota

⚠ **Important note!** If a default user quota was created during file system creation, the default user quota will take effect once the explicit user quota is deleted.

1. From the **Storage** > **File Systems** page, click the file system whose explicit user quota(s) you wish to delete. The file system's details page appears.
2. From the **User Quotas** panel, click the trash can icon on the same row as the explicit user quota you wish to delete. The **Delete User Quota** pop-up window appears.

   If you wish to delete multiple explicit user quotas, click **More Options** > **Delete Quotas**. The **Delete User Quotas** pop-up window appears. From the **Existing Quotas** list, select the user quotas you wish to delete. You can optionally enter a partial name in the search box to filter the list. You can also

optionally select all user quotas by clicking the checkbox next to the search box. All selected user quotas appear in the **Selected Quotas** list.

3 Click **Delete**.

Creating an Explicit Group Quota

⚠ **Important note!** If a default group quota was created during file system creation, the explicit group quota will take priority over the default quota and will be used for that file system.

1 From the **Storage** > **File Systems** page, click the file system for which you wish to set an explicit group quota. The file system's details page appears.

2 In the **Group Quotas** panel, click the add button (+) . The **Add Group Quota** pop-up window appears.

3 Enter the group ID or group name, and quota size.

4 Click **Add**.

Modifying an Explicit Group Quota

1 From the **Storage** > **File Systems** page, click the file system whose explicit group quota(s) you wish to modify. The file system's details page appears.

2 From the **Group Quotas** panel, click the **Edit** button on the same row as the explicit group quota you wish to modify. The **Edit Group Quota** pop-up window appears.

If you wish to modify multiple explicit group quotas, click **More Options** > **Edit Quotas**. The **Edit Group Quotas** pop-up window appears. From the **Existing Quotas** list, select the group quotas you wish to modify. You can optionally enter a partial name in the search box to filter the list. You can also optionally select all group quotas by clicking the checkbox next to the search box. All selected group quotas appear in the **Selected Quotas** list.

3 Enter a new quota size.

4 Click **Save**.

Deleting an Explicit Group Quota

⚠ **Important note!** If a default group quota was created during file system creation, the default group quota will take effect once the explicit group quota is deleted.

1 From the **Storage** > **File Systems** page, click the file system whose explicit group quota(s) you wish to delete. The file system's details page appears.

2. From the **Group Quotas** panel, click the trash can icon on the same row as the explicit group quota you wish to delete. The **Delete Group Quota** pop-up window appears.

   If you wish to delete multiple explicit group quotas, click **More Options** > **Delete Quotas**. The **Delete Group Quotas** pop-up window appears. From the **Existing Quotas** list, select the group quotas you wish to delete. You can optionally enter a partial name in the search box to filter the list. You can also optionally select all group quotas by clicking the checkbox next to the search box. All selected group quotas appear in the **Selected Quotas** list.

3. Click **Delete**.

Enabling and Disabling the Hard Limit of a File System

⚠ **Important note!** File system hard limits cannot be enabled for file systems with unlimited sizes.

1. From the **Storage** > **File Systems** page, click the **Edit** button in the row of the file system you want to edit. The **Edit File System** dialog box appears.

2. Set the **Hard Limit** toggle button to enable (blue) or disable (gray) the feature.

Creating a File System Snapshot

Snapshots are immutable, point-in-time images of the contents of one or more file systems. Snapshot tasks include creating and deleting file system snapshots. Snapshots can be created or destroyed using the Purity//FB GUI (**Storage** > **File Systems**) or the CLI using the **`purefs`** command. They can also be created using policies.

To create a snapshot of a file system:

1. Select **Storage** > **File Systems**.

2. Click the file system. The snapshot list appears for that file system.

3. Click add (+) on the right. The **Create Snapshot** pop-up window appears.

4. Enter a suffix name for the snapshot. A suffix will be automatically assigned if this field is left empty.

5. Click **Create**.

Enabling and Disabling the Snapshot Directory

A snapshot directory is a special directory that can be enabled or disabled to store snapshots. By default, snapshot directories are enabled at file system creation.

To enable or disable the snapshot directory:

1. Select **Storage** > **File Systems**.

2. Click the **Edit** button in the row of the file system you want to edit. The **Edit File System** pop-up window appears.
3. Click the **Snapshot** toggle button to switch between enabled (blue) and disabled (gray) status.
4. Click **Save**.
5. If you are enabling the snapshot directory, in the root directory of the NFS mount, confirm that a pseudo-directory named `.snapshot` has been created.

   The `.snapshot` directory name is reserved. You cannot create a directory with the `.snapshot` name, regardless of the enabled/disabled status. The directory cannot be renamed, deleted, or moved into other directories. It can only be removed by disabling the directory.

Removing a Policy from a Snapshot

Policies can be removed from snapshots at any time.

Note: If a policy is removed from a snapshot, that snapshot is no longer managed by that policy and must be destroyed manually.

You can remove a policy from a snapshot from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

To remove a policy from a snapshot, perform the following:

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots.
2. If you are on the **Protection** page, select **Protection** > **Snapshots**.
3. From the table in the **File System Snapshots** panel, locate the file system snapshot from which you wish to remove a policy.
4. Click the **More Options** button located at the end of that snapshot's row and select **Remove Policy**. The **Remove Policy from Snapshot** pop-up window appears.
5. Click **Remove**.

Destroying File System Snapshots

You can destroy a snapshot from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

To destroy one or more file system snapshots, perform the following:

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.
2. From the table in the **File System Snapshots** panel, locate the file system snapshot you either wish to destroy, or is the point from which earlier snapshots are to be destroyed.
3. If you wish to destroy the snapshot, perform the following:
   a. Click the **More Options** button located at the end of that snapshot's row and select **Destroy**. The **Destroy Snapshot** pop-up window appears.
   b. Click **Destroy**
4. If you wish to destroy multiple snapshots, perform the following:
   a. Click **More Options** > **Destroy** in the **File System Snapshots** panel title bar. The **Destroy File System Snapshots** pop-up window appears.
   b. From the **File System Snapshots** list, select the snapshots you wish to destroy. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the check box next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
   c. Click **Destroy**.
5. If you wish to use the snapshot as the point from which earlier snapshots are destroyed, perform the following:
   a. Click the **More Options** button located at the end of that snapshot's row and select **Destroy older**. The **Destroy Snapshots older than...** pop-up window appears.
   b. Select the snapshots you wish to destroy from the **Older File System Snapshots** pane. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the check box next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
   c. Click **Destroy**.

The destroyed file system snapshot appears under the **Destroyed Snapshots** panel on the **Snapshots** tab. Destroyed snapshots can be recovered within a 24-hour period. Once the 24-hour period has passed, the snapshot is permanently destroyed (eradicated).

Restoring a File System to a Snapshot

Purity//FB allows you to restore a file system to a given point in time from a snapshot (via file system rollback). Restoring a file system to a snapshot overwrites the contents of that file system with data from the snapshot.

File system rollback is available for snapshots created on a FlashBlade array running Purity//FB 3.0.0 or later.

CAUTION: If Pure File SafeMode is enabled, contact Pure Technical Services to perform this action.

CAUTION: When restoring to a snapshot, the file system's configured quota, as well as any configured user/group quotas, do not change to reflect the limit when the snapshot was taken. This can cause the live space usage to increase, resulting in subsequent write failures due to exceeding the quota. You may need to re-adjust your quota settings accordingly.

Note: It is highly recommended that there be no I/O on the file system when performing a restore.

You can restore a file system from a snapshot from either the **Storage** or **Protection** page.

Note: If there are snapshots that are newer than the snapshot you wish to use as the restore point, the newer snapshots must be first destroyed and eradicated before you can proceed with the file system restoration. See *Destroying and Eradicating Newer Snapshots*.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

To restore a file system from a snapshot, perform the following:

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.
2. From the table in the **File System Snapshots** panel, locate the file system snapshot to be used to restore the file system.
3. Click the **More Options** button located at the end of row of the snapshot to be used for restore and select **Restore**. The **Restore File System from Snapshot** pop-up window appears.
4. Click **Restore**.

Destroying and Eradicating Newer Snapshots

Note: Destroyed snapshots can be recovered within a 24-hour period. Once the 24-hour period has passed, the snapshot is permanently destroyed (eradicated).

When restoring a file system from a snapshot, if there are snapshots that are newer than the snapshot you wish to use as the restore point, the newer snapshots must be first destroyed and eradicated before you can proceed with the file system restoration.

You can destroy and eradicate snapshots from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.
2. From the table in the **File System Snapshots** panel, locate the file system snapshot to be used to restore the file system.
3. Click the **More Options** button located at the end of that snapshot's row and select **Destroy newer**. The **Destroy Snapshots newer than...** pop-up window appears.
4. In the **Newer File System Snapshots** pane, select all snapshots by clicking the check box next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
5. Click **Destroy**. The snapshot(s) appear in the **Destroyed Snapshots** panel (directly below the **File System Snapshots** panel).
6. Click **More Options** > **Eradicate** in of the **Destroyed Snapshots** panel. The **Eradicate File System Snapshots** pop-up window appears.
7. From the **Destroyed File System Snapshots** list, select the snapshots you wish to eradicate. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the checkbox next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
8. Click **Eradicate**.

Recovering File System Snapshots

When a file system snapshot is destroyed, you have 24 hours to recover the data. This is known as the eradication pending state. When the eradication pending state expires, the snapshot will be eradicated and you will no longer be able to retrieve the snapshot data.

You can recover a snapshot from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

To recover a previously destroyed file system snapshot, perform the following:

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.

2 If you wish to recover a single snapshot:

   a From the **Destroyed Snapshots** panel, locate the file system snapshot you wish to recover.

   b Click **More Options** > **Recover** at the end of the row of the file system snapshot you wish to recover.

   c Click **Recover**.

3 If you wish to recover multiple file system snapshots:

   a Click **More Options** > **Recover** in the **Destroyed Snapshots** panel title bar. The **Recover File System Snapshots** pop-up window appears.

   b From the **Destroyed File System Snapshots** list, select the snapshots you wish to recover. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the check box next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.

   c Click **Recover**.

Eradicating File System Snapshots

During the eradication pending period, you can manually eradicate the destroyed file system to reclaim physical storage space occupied by the destroyed snapshot. Once reclamation starts, the destroyed snapshots can no longer be recovered.

You can eradicate a snapshot from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

CAUTION: If Pure File SafeMode is enabled, contact Pure Technical Services to perform this action.

To eradicate a destroyed snapshot:

1 If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's destroyed snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.

2 If you wish to eradicate a single snapshot:

   a In the **Destroyed Snapshots** panel under **File System Snapshots**, click **More Options** > **Recover** at the end of the row of the snapshot you want to eradicate. The **Eradicate Snapshot** pop-up window appears.

   b Click **Eradicate**. The array will begin reclaiming the eradicated space of the snapshot.

3 If you wish to eradicate mutliple snapshots:

   a Click **More Options > Eradicate** in of the **Destroyed Snapshots** panel. The **Eradicate File System Snapshots** pop-up window appears.

   b From the **Destroyed File System Snapshots** list, select the snapshots you wish to eradicate. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the checkbox next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.

   c Click **Eradicate**. The array will begin reclaiming the eradicated space of the snapshots.

Enabling and Disabling Fast Remove

⚠ **Important note!** The fast remove feature is available for removing directories only, not individual files.

Enable the fast remove feature to quickly remove large directories.

To enable or disable fast remove:

1 From the **Storage > File Systems** page, click the **Edit** button in the row of the file system you want to edit. The **Edit File System** pop-up window appears.

2 Click the **Fast Remove** toggle button to switch between enabled (blue) and disabled (gray) status.

3 Click **Save**.

4 In the root directory of the NFS mount, confirm that a pseudo-directory named `.fast-remove` has been created.

   By default, the `.fast-remove` directory permissions are set to `drwxr-xr-x`. Optionally use `chown` or `chmod` to set the desired access permissions.

   To remove a directory, run the `mv` command to move the directory into the `.fast-remove` directory.

   ⚠ **Important note!** Once a directory has been moved into the `.fast-remove` directory, it is immediately destroyed and unrecoverable.

Promoting and Demoting File Systems

Promoting and demoting file systems is required when performing file system failover and failback. In a file replication setup, data is written to the promoted local file system and then that data is replicated to the demoted remote file system on a remote array. During failover, the local file system is demoted and the remote file system is promoted. During failback, the original promoted/demoted filesystem designations are restored.

⚠ **CAUTION:** If Pure File SafeMode is enabled, contact Pure Technical Services to demote a file system.

For complete failover and failback instructions, see *Performing File Replication Failover and Reprotect* and *Performing File Replication Failback*.

To promote and demote file systems, perform the following:

1. Fom the **Storage** > **File Systems** page, locate the file system that you wish to promote or demote from the **File Systems** panel.
2. Click **More Options** > **Promote** or **More Options** >**Demote** at the end of the row in which the remote file system appears.
3. Click **Promote** or **Demote** when prompted for confirmation.

Destroying a File System

Destroying a file system will also destroy all related snapshots. All protocols of a file system must be disabled before it can be destroyed; disabling protocols terminates all client connections to the file system.

To destroy a file system:

1. Select **Storage** > **File Systems**.
2. Click the trash icon in the row of the file system you want to destroy. The **Destroy File System** pop-up window appears.

   If you wish to destroy mutliple file systems, click **More Options** > **Eradicate** in the **File Systems** panel. The **Destroy File Systems** pop-up window appears. From the **Existing File Systems** list, select the file systems you wish to destroy. You can optionally enter a partial name in the search box to filter the list. You can also optionally select all file systems by clicking the checkbox next to the search box. All selected file systems appear in the **Selected File Systems** list.
3. Click **Destroy**. The destroyed file system appears in the **Destroyed File Systems** panel and begins its 24-hour eradication pending period.

During the eradication pending period, you can recover the file system to bring it back from its previous state, or manually eradicate the destroyed file system to reclaim the physical storage space occupied by the file system.

When the 24-hour eradication pending period has lapsed, Purity//FB starts reclaiming the physical storage occupied by the file system. Once reclamation starts, either because you have manually eradicated the destroyed file system, or because the eradication pending period has lapsed, the destroyed file system can no longer be recovered.

Recovering a File System

Destroyed file systems have 24 hours to be recovered. When a file system is recovered, all related snapshots will be also be recovered unless the snapshot was explicitly destroyed. When the 24

hour eradication pending state expires, the file system and its implicitly destroyed snapshots will be eradicated and you will no longer be able to retrieve the data.

To recover a destroyed file system:

1. Select **Storage** > **File Systems**.
2. In the **Destroyed File System** panel, click the clock icon in the row of the file system you want to recover. The **Restore File System** pop-up window appears.

   If you wish to recover mutliple file systems, click **More Options** > **Recover** in the **Destroyed File Systems** panel under the **File Systems** panel. The **Recover File Systems** pop-up window appears. From the **Destroyed File Systems** list, select the file systems you wish to recover. You can optionally enter a partial name in the search box to filter the list. You can also optionally select all file systems by clicking the checkbox next to the search box. All selected file systems appear in the **Selected File Systems** list.
3. Click **Recover**. The recovered file system appears in the **File Systems** panel.

Eradicating a File System

During the eradication pending period, you can manually eradicate the destroyed file system to reclaim physical storage space occupied by the destroyed file system. Eradicating a file system will also eradicate all of its related snapshots. Once reclamation starts, the destroyed file system and snapshots can no longer be recovered.

CAUTION: If Pure File SafeMode is enabled, contact Pure Technical Services to perform this action.

To eradicate a destroyed file system:

1. Select **Storage** > **File Systems**.
2. In the **Destroyed File Systems** panel, click the trash icon at the end of the row of the file system you want to eradicate. The **Eradicate File System** pop-up window appears.

   If you wish to eradicate mutliple file systems, click **More Options** > **Eradicate** in the **Destroyed File Systems** panel under the **File Systems** panel. The **Eradicate File Systems** pop-up window appears. From the **Destroyed File Systems** list, select the file systems you wish to eradicate. You can optionally enter a partial name in the search box to filter the list. You can also optionally select all file systems by clicking the checkbox next to the search box. All selected file systems appear in the **Selected File Systems** list.
3. Click **Eradicate**. The array will begin reclaiming the eradicated space of the file system.

Performing File Replication Failover and Reprotect

To initiate failover in a file replication setup:

- The target file system must first be promoted.
- All clients must then be directed to the target array.
- To reprotect, the local file system is then demoted. Note that when a file system is demoted, any writes performed since the last replication will be discarded.

To initiate failover, perform the following:

1. On the target array, from the **Storage** > **File Systems** page, locate the remote file system that you wish to promote from the **File Systems** panel.
2. Click **More Options** > **Promote** at the end of the row in which the remote file system appears.
3. Click **Promote** when prompted for confirmation.
4. Manually redirect clients to the target array.

   To reprotect, you must remove snapshots since the last replicated snapshot. This can be difficult if the policy engine is creating new snapshots. You can stop the policy engine from creating new snapshots by removing the policies form the file system and replica link.
5. On the local array, remove policies attached to the to-be-demoted local file system and replica link.

   To remove the policy from the local file system:

   a. From the **Storage** > **File Systems** page, locate and click local file system that you wish to demote from the **File Systems** panel. The file system's details page appears.

   b. In the **Policies** panel, click the remove button (**X**) in the same row as the policy you wish to remove.

   c. Click **Remove** when prompted for confirmation.

   To remove the policy from the replica link:

   a. From the **Protection** > **File Replica Links** page, click **More Options** > **Remove Policies** at the end of the row of the replica link you wish to edit. The **Remove Policies** pop-up window appears.

   b. Select the policy or policies you wish to remove from the replica link from the **Available Policies** list.

   c. Click **Remove**.
6. Navigate back to the **Storage** > **File Systems** page and locate the local file system that you wish to demote from the **File Systems** panel.
7. Click **More Options** > **Demote** at the end of row in which the local file system appears.
8. Click **Demote** when prompted for confirmation.
   Writes to the local file system are stopped and are redirected to the promoted remote file system.

9 If you removed policies in step 5, add the removed policy or policies back to the file system and replica link.

To add the policy to the file system:

a From the **Storage > File Systems** page, locate and click the file system from the **File Systems** panel. The file system's details page appears.

b In the **Policies** panel, click the add button (+). The **Add Policies** pop-up window appears.

c Select the policy or policies you wish to add to the file system from the **Available Policies** list.

d Click **Add**.

To add the policy to the replica link:

a From the **Protection > File Replica Links** page, click **More Options > Add Policies** at the end of the row of the replica link you wish to edit. The **Add Policies** pop-up window appears.

b Select the policy or policies you wish to add back to the replica link from the **Available Policies** list.

c Click **Add**.

Once the local file system comes back up, replication resumes as scheduled between the newly promoted target file system and demoted local file system because the policy was re-added to the replica link. Because clients have been redirected to the target, data is written to the promoted remote file system, which is then replicated to the demoted local file system.

Performing File Replication Failback

Note: This procedure continues from the previous failover procedure. During the failover procedure, the remote file system was promoted and the local file system was demoted.

As noted in the failover procedure, if the replication schedule policy was re-added to the replica link in the replication setup, once the local file system comes back up, replication resumes as scheduled between the promoted target file system and demoted local file system. Performing failback restores the original local/target relationship between arrays and file systems in a replication setup. During failback, once the remote file system is demoted and the local file system is promoted and clients redirected to the local file system, the original replication setup resumes; data is written to the local file system and then replicated to the remote file system.

To initiate failback in a file replication setup:

- Stop writes on the remote file system and send the remaining data to the local file system.
- Promote the local file system.
- Add back any removed policy to the local file system.

- Demote the remote file system.

To initiate failback, perform the following:

1. On the target array, from the **Storage** > **File Systems** page, locate the remote file system that is to be demoted from the **File Systems** panel.
2. Click **More Options** > **Disable Writes** at the end of row in which the remote file system appears.
3. Click **Disable** when prompted for confirmation.
4. Ensure all writes are sent to the source file system by creating a snapshot on the target and send it to the source.
5. On the local array, from the **Storage** > **File Systems** page, locate the local file system that you wish to promote from the **File Systems** panel.
6. Click **More Options** > **Promote** at the end of row in which the local file system appears.
7. Manually redirect clients back to the to the local array.
8. Remove policies attached to the to-be-demoted remote file system and replica link.

   To remove the policy from the file system:

   a. From the **Storage** > **File Systems** page, locate and click local file system that you wish to demote from the **File Systems** panel. The file system's details page appears.

   b. In the **Policies** panel, click the remove button (**X**) in the same row as the policy you wish to remove.

   c. Click **Remove** when prompted for confirmation.

   To remove the policy from the replica link:

   a. From the **Protection** > **File Replica Links** page, click **More Options** > **Remove Policies** at the end of the row of the replica link you wish to edit. The **Remove Policies** pop-up window appears.

   b. Select the policy or policies you wish to remove from the replica link from the **Available Policies** list.

   c. Click **Remove**.
9. On the target array, from the **Storage** > **File Systems** page, locate the remote file system that you wish to demote from the **File Systems** panel.
10. Click **More Options** > **Demote** at the end of row in which the remote file system appears.
11. Click **Demote** when prompted for confirmation.
    Writes to the remote file system are stopped and are redirected to the promoted local file system.

12 If you removed policies in step 8, add the removed policy or policies back to the file system and replica link.

To add the policy to the file system:

a From the **Storage > File Systems** page, locate and click the file system from the **File Systems** panel. The file system's details page appears.

b In the **Policies** panel, click the add button (+). The **Add Policies** pop-up window appears.

c Select the policy or policies you wish to add to the file system from the **Available Policies** list.

d Click **Add**.

To add the policy to the replica link:

a From the **Protection > File Replica Links** page, click **More Options > Add Policies** at the end of the row of the replica link you wish to edit. The **Add Policies** pop-up window appears.

b Select the policy or policies you wish to add back to the replica link from the **Available Policies** list.

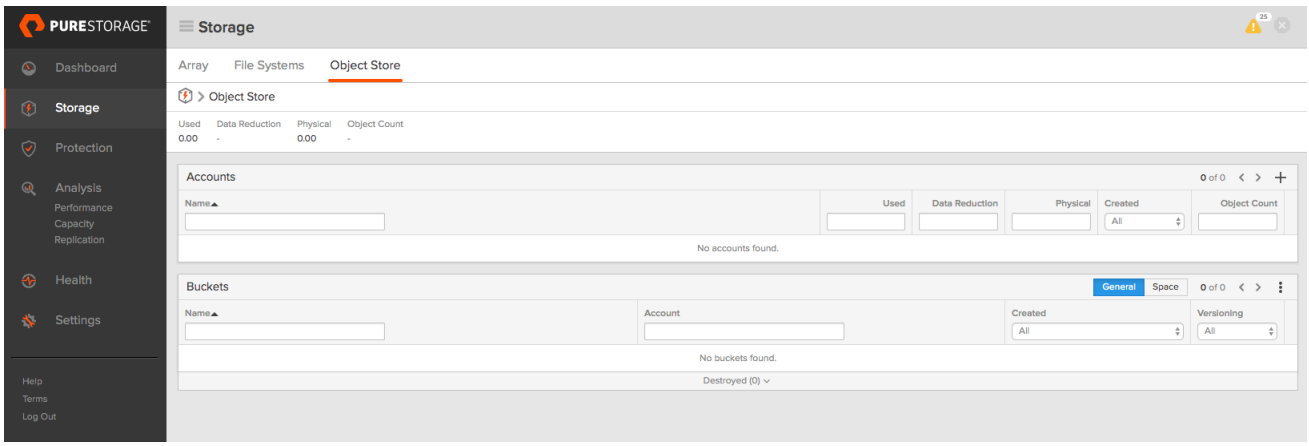
c Click **Add**.

The original replication setup has been restored.

Object Store

The **Object Store** tab is used to manage the array 's object store accounts, users, and buckets.

Figure 5.8 Object Store Tab



The **Object Store** tab is divided into an **Accounts** pane, which displays all object store accounts, and a **Buckets** pane, which displays all buckets.

In the **Accounts** pane, the following information is listed for each account:

- **Name**: The account name.
- **Used**: The amount of space used.
- **Data Reduction**: Ratio of mapped sectors within an account versus the amount of physical space the data occupies after data compression and deduplication.
- **Physical**: The amount of physical space.
- **Created**: The account creation date.
- **Object Count**: The number of objects in the bucket. Note that there can be a temporary inconsistency in the number of objects shown in the GUI versus the client if an HA event (for example an NDU upgrade, software restart, failover event, etc.) occurred within a 12-hour period.

You can search the accounts list by entering search criteria in the search box under the column names.

The **Buckets** pane displays the following information about buckets:

- **Name**: The bucket name.
- **Account**: The account to which the bucket belongs.
- **Created**: The bucket creation date.
- **Versioning**: Whether bucket versioning is enabled or disabled.

As with the **Accounts** pane, you can search the buckets list by entering search criteria in the search box under the column names.

Creating an Account

To create a new account:

1. In the **Accounts** section of the **Storage** > **Object Store** page, click the add (+) button. The **Create Account** pop-up window appears.
2. Enter the new account's name in the **Name** field.
3. Click **Create**.

Deleting an Account

To delete an account:

1. In the **Accounts** section of the **Storage** > **Object Store** page, click the **Delete** button on the same row as the account you wish to delete. The **Delete Account** pop-up window appears.

2 Click the **Delete** button to confirm the account deletion.

Users

To view information about an existing user, from the **Storage > Object Store** page, click the account to which the user you wish to view belongs. The user information page appears which provides a **Users** pane and **Buckets** pane.

The **Users** pane lists all account users. The following information is displayed about each account user:

- **Name**: The user name in the format <account name>/<user name>.
- **Access Key ID**: The user's access key ID.
- **Key Enabled**: Whether the access key is enabled (True) or disabled (False).

The **Buckets** pane lists all buckets associated with the selected account and displays the following information:

- **Name**: The bucket name.
- **Created**: The bucket creation date.
- **Versioning**: Whether bucket versioning is enabled or disabled.

Creating a User

To create a new user:

1 From the **Storage > Object Store** page, click the account to which you wish to add a new user.

2 In the **Users** section, click the add (+) button. The **Create User** pop-up window appears.

3 Enter the new user's name in the **User Name** field.

4 (Optional) Click the **Create Access Key** button to enable (blue) access key creation. Note that an access key can be created for the user at a later date (see *Creating an Access Key*).

5 Click **Create**.

Deleting a User

Deleting a user also deletes all access keys associated with that user. To delete a user:

1 From the **Storage > Object Store** page, click the account from which you wish to delete a user.

2 From the **Users** section, click the **More Options > Delete user** button on the same row as the user you wish to delete.

3 When prompted to confirm the deletion, click **Delete**.

Creating an Access Key

Creating an access key also creates the user's associated secret key.

Be sure to save off or download the access key and secret key information when you create a key.

To create an access key:

1 From the **Storage** > **Object Store** page, click the account to which you recently added a new user.

2 From the **Users** section, click the **More Options** > **Create access key** button on the same row as the user for which you wish to create an access key.

3 Make note of the access key and secret key information displayed in the confirmation pop-up. Optionally use the **CSV** or **JSON** button to download the key information.

⚠ **Important note!** You must save off or download your key information before closing the pop-up. The secret key can only be accessed during key creation confirmation and cannot be viewed or accessed after the confirmation pop-up is closed.

4 Use the access key and secret key in your `.s3cfg` or `.s3curl` files, for example.

Importing an Access Key

Note: FlashBlade does not validate if the access key being imported from another FlashBlade was a previously deleted access key during or after import.

Credentials generated on one FlashBlade can be imported on another FlashBlade for a single account, with data on both arrays being accessible using the same credentials. This can be useful in FlashBlade replication setups, or in environments where multiple FlashBlades are used in mutli-site deployments of Splunk SmartStore. When importing access keys, the following should be noted:

- Only new access keys created after Purity//FB 3.0.0 can be imported on another FlashBlade.
- The access key you are importing cannot already exist on the FlashBlade to which it is being imported.
- If there are two access keys present for a user, you cannot import a third access key.
- Credentials are not synchronized or validated against the FlashBlade from which they were imported; deleting or disabling a set of credentials on the source cluster will not delete or disable the credentials on any other clusters where they were imported.

As a best practice, Pure Storage recommends that customers assign a unique access key for each individual user and each application, and avoid sharing an access key across two FlashBlade arrays

except when specifically required by your application configuration (for example, Splunk SmartStore configurations on FlashBlade that use a second FlashBlade for disaster recovery).

To import an access key:

1. From the **Storage** > **Object Store** page of the FlashBlade to which you are importing the key, click the account containing the user to which you wish to import the access key.
2. From the **Users** section, click the **More Options** > **Import access key** button on the same row as the user to which you wish to import the access key. The **Import Access Key** pop-up window appears.
3. Enter the access key ID and secret access key.
4. Click **Import**.

Deleting an Access Key

Deleting an access key also deletes the user's associated secret key. Once an access key has been deleted, it cannot be recovered.

To delete an access key:

1. From the **Storage** > **Object Store** page, click the account containing the user who's access key you wish to delete.
2. From the **Users** section, click the **More Options** > **Delete access key** button on the same row as the user for which you wish to delete the access key.
3. When prompted to confirm the access key deletion, click the **Delete** button.

Disabling an Access Key

Access keys are enabled when they are created. To disable an access key:

1. From the **Storage** > **Object Store** page, click the account containing the user who's access key you wish to disable.
2. From the **Users** section, click the **More Options** > **Disable access key** button on the same row as the user for which you wish to disable the access key.
3. When prompted to confirm disabling the access key, click the **Disable** button.

You can optionally use the **`pureobj`** CLI command to disable access keys. See *pureobj*.

Enabling an Access Key

To enable a previously disabled an access key:

1. From the **Storage** > **Object Store** page, click the account containing the user who's access key you wish to enable.
2. From the **Users** section, click the **More Options** > **Enable access key** button on the same row as the user for which you wish to enable the access key.
3. When prompted to confirm enabling the access key, click the **Enable** button.

You can optionally use the **`pureobj`** CLI command to enable access keys. See *pureobj*.

Buckets

Buckets and their contents can be created and managed with the FlashBlade GUI, CLI, and management REST API. Buckets and their contents can also be managed with the object store REST API, with open source tools such as s3curl and s3cmd, and with Java, Python, or other languages.

To view buckets per account, from the **Storage** > **Object Store** page, click an account name. The buckets for that account are displayed in the **Buckets** panel. For each bucket, the amount of used space, data reduction, physical space, object count, and status of versioning (enabled or none) are displayed.

To view a bucket's replica link information, click the bucket name. For each replica link, the source bucket name, replication direction, remote bucket name, remote credentials, replication status, lags, and recovery point are displayed. See *Object Replica Links* for information.

To view a bucket's lifecycle rules, click the bucket name. For each lifecycle rule, the rule ID, rule status, object prefix filter, and number of days to keep non-current objects are displayed. See *Object Lifecycle Configuration Policy* for information.

Bucket Versioning

A bucket 's versioning status is displayed in the **Buckets** panel after clicking an account name from the **Storage** > **Object Store** page. Versioning status can also be displayed in the CLI using the **`purebucket list`** command.

Bucket versioning allows multiple variants of an object in the same bucket. Buckets can be in three different versioning states: None, Enabled, and Suspended. All buckets are initially created in a non-versioned (None) state. Once you enable versioning on a bucket, every object in the bucket automatically retains the entire history on every update, and that bucket can never return to the non-versioned state. However, you can suspend versioning on that bucket.

In a non-version bucket, each object name has only one version. This is always the current version. Deleting an existing object name permanently destroys the object.

In a versioning-enabled bucket, an object can have multiple versions for the same object name. Each object name has one current version and either zero or multiple non-current versions.

The versioning-suspended state can prevent the number of versions from growing.

Lifecycle rules can also prevent the number of versions from growing. Configuring a lifecycle rule, either for all objects or for objects matching a key name prefix, will cause noncurrent object versions older than the specified number of days to be permanently deleted from the bucket.

Creating a Bucket

To create a bucket:

1. From the **Storage > Object Store** page, click the account to which you wish to add a bucket.
2. From the **Buckets** pane, click the add (+) button. The **Create Bucket** pop-up window appears.
3. Enter a name for the bucket.
4. Click **Create**.

Enabling Bucket Versioning

To enable bucket versioning, perform the following:

1. From the **Storage > Object Store** page, click the account to which the bucket you wish to version belongs.
2. From the **Buckets** panel, click **More Options** > **Enable versioning** on the same row as the bucket you wish to version. The **Enable Versioning** pop-up window appears.
3. Click **Enable**.

Suspending Bucket Versioning

To suspend bucket versioning, perform the following:

1. From the **Storage > Object Store** page, click the account to which the bucket whose versioning you wish to suspend belongs.
2. From the **Buckets** panel, click **More Options** > **Suspend versioning** on the same row as the bucket whose versioning you wish to suspend.. The **Suspend Versioning** pop-up window appears.
3. Click **Suspend**.

Object Lifecycle Configuration Rules

When bucket versioning is enabled, multiple versions of an object in the same bucket are created that retain the object's entire history on every update. In order to help manage space usage, you can create object lifecycle configuration rules to automatically delete old, unneeded versions of the object. Without an object lifecycle rule, an object's versions are retained indefinitely until manually deleted.

Note: If bucket versioning was previously enabled, but currently suspended, the versions created while versioning was enabled are still subject to any lifecycle rules created after versioning was suspended.

When an object lifecycle rule is created, that rule manages the amount of time that the 'non-current' version of the object is retained. For example, you can specify that a non-current version be retained for 10 days. When the 10 day retention period is reached, that version will be deleted from the system within 24 hours.

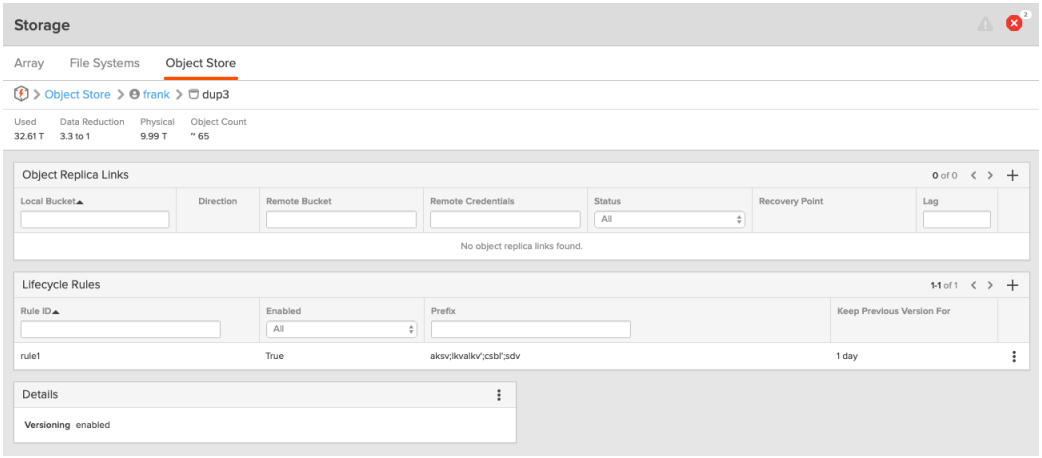
When a lifecycle policy is applied, then for all objects in the bucket or, if specified, for those objects matching the key prefix, all versions that have been noncurrent for more than the specified retention period are subject to immediate deletion.

Up to 1,000 object lifecycle rules can be created per bucket.

Object lifecycle rules can be viewed, created, edited, enabled, disabled, and deleted using the Purity//FB GUI, CLI, and Management REST API, as well as the Object Store REST API.

Object lifecycle rules are listed in the **Lifecycle Rules** list on a bucket's detail screen. To navigate to a bucket's detail screen, click **Storage** > **Object Store** > **Bucket** > **bucket name**.

Figure 5.9 Lifecycle Rules List



The **Lifecycle Rules** list displays the following information:

- **Rule ID** - The object lifecycle rule ID. This ID is either created by the system or specified by the user. If system-assigned, the ID is in the format `fbRuleIdX`, where X increases by one with each rule. The rule ID must be unique for the given bucket.
- **Enabled** - Indicates if the object lifecycle rule is enabled (**True**) or disabled (**False**).
- **Prefix** - An optional field that causes the rule to only be applied to objects beginning with this prefix.
- **Keep Previous Version For** - The duration that the non-current object version is retained expressed in a whole number day, for example: 10 days, 4 weeks, etc.

Creating an Object Lifecycle Rule

To create an object lifecycle rule, perform the following:

1. From the **Storage > Object Store** page, click the account that contains the bucket to which you wish to create a lifecycle configuration.
2. From the **Buckets** panel, select the bucket for which you wish to create an object lifecycle rule.
3. Click the **Add** (+) button in the heading of the **Lifecycle Rules** list. The **Create Lifecycle Rule** pop-up window appears.
4. (Optional) Enter a unique ID for the rule in the **Rule ID** field. The ID cannot exceed 255 characters. If no ID is entered, the system will assign a unique ID in the format `fbRuleIdX`, where X is the next available rule ID number in the `fbRuleIdX` format.
5. (Optional) Enter an object prefix in the **Prefix** field. The prefix cannot exceed 1,024 characters.

   An object prefix filter objects within the bucket to which the rule is applied. For example, if `/tmp/` is entered, the created rule will apply to only objects with the prefix `/tmp/`.
6. In the **Keep Previous Version For** field, enter the period for the non-current object version will be retained. The value must be entered in the format `N[m/h/d/w]`, where `N` is an integer followed by `m` (minutes), `h` (hours), `d` (days), or `w` (weeks). The minimum valid retention period is one day. Note that the value entered must be in a multiple of 1 day increments. For example: 24h, 1440m, etc.

   For example, entering 15d results in the retention of the non-current object version for 15 days; when the 15 day retention period is reached, that version will be deleted from the system within 24 hours.
7. Review the summary and click **Create**.

   The object lifecycle rule is automatically enabled upon creation.

Editing an Object Lifecycle Rule

To edit an object lifecycle rule, perform the following:

1. From the **Storage > Object Store** page, click the account that contains the bucket to which you wish to edit a lifecycle configuration.
2. From the **Buckets** panel, select the bucket with the object lifecycle rule you wish to edit.
3. From the **Lifecycle Rules** list, click the **More Options** button in the row of the object lifecycle rule you wish to edit and select **Edit**. The **Edit Lifecycle Rule** pop-up window appears.
4. (Optional) Enable or disable the rule using the **Enable** toggle.

5 (Optional) Enter or change the object prefix in the **Prefix** field. The prefix cannot exceed 1,024 characters.

   An object prefix filter objects within the bucket to which the rule is applied. For example, if `/tmp/` is entered, the created rule will apply to only objects with the prefix `/tmp/`.

6 (Optional) In the **Keep Previous Version For** field, enter the period for the non-current object version will be retained. The value must be entered in the format `N[m/h/d/w]`, where `N` is an integer followed by `m` (minutes), `h` (hours), `d` (days), or `w` (weeks). Note that the value entered must be in a multiple of 1 day increments. For example: 24h, 1440m, etc.

   For example, entering 15d results in the retention of the non-current object version for 15 days; when the 15 day retention period is reached, that version will be deleted from the system within 24 hours.

7 Review the summary and click **Save**.

   The object lifecycle rule is updated with your changes.

Deleting an Object Lifecycle Rule

To delete an object lifecycle rule, perform the following:

1 From the **Storage** > **Object Store** page, click the account to which the bucket whose object lifecycle rule you wish to delete belongs.

2 From the **Buckets** panel, select the bucket with the object lifecycle rule you wish to delete.

3 From the **Lifecycle Rules** list, click the **More Options** button in the row of the object lifecycle rule you wish to delete and select **Delete**. The **Delete Lifecycle Rule** pop-up window appears.

4 When prompted for confirmation, click **Delete**.

   The object lifecycle rule is deleted from the bucket.

Destroying a Bucket

When a bucket is destroyed, it enters a 24-hour eradication pending period. During this time, the bucket and its contents can be recovered. Note that also during this time, data in the bucket is inaccessible. When the 24-hour eradication pending period has lapsed, Purity//FB starts reclaiming storage space. During this time, the bucket can no longer be recovered.

To destroy a bucket:

1 From the **Storage** > **Object Store** page, click the account from which you wish to destroy a bucket.

2 From the **Buckets** panel, click the trash can icon on the same row as the bucket you wish to destroy. The **Destroy Bucket** pop-up window appears.

   If you wish to destroy multiple buckets, click **More Options > Destroy**. The **Destroy Buckets** pop-up window appears. From the **Existing Buckets** list, select the buckets you wish to destroy. You can optionally enter a partial name in the search box to filter the list. You can also optionally select all buckets by clicking the checkbox next to the search box. All selected file systems appear in the **Selected Buckets** list.

3 Click **Destroy**.

Recovering a Bucket

Destroyed buckets have 24 hours to be recovered. Once 24 hours has expired, the system begins eradicating the bucket, at which point the bucket is no longer recoverable.

To recover a destroyed bucket:

1 From the **Storage > Object Store** page, click the account from which you wish to recover a destroyed bucket.

2 From the **Destroyed Buckets** panel, click the clock icon on the same row as the bucket you wish to recover. The **Recover Bucket** pop-up window appears.

   If you wish to recover multiple buckets, click **More Options > Recover**. The **Recover Buckets** pop-up window appears. From the **Destroyed Buckets** list, select the buckets you wish to recover. You can optionally enter a partial name in the search box to filter the list. You can also optionally select all buckets by clicking the checkbox next to the search box. All selected file systems appear in the **Selected Buckets** list.

3 Click **Recover**.

Eradicating a Bucket

During the eradication pending period, you can manually eradicate the destroyed buckets to reclaim physical storage space. Once reclamation starts, the destroyed buckets can no longer be recovered.

To eradicate a destroyed bucket:

1 From the **Storage > Object Store** page, click the account from which you wish to eradicate a destroyed bucket.

2 From the **Destroyed Buckets** panel, click the clock icon on the same row as the bucket you wish to eradicate. The **Eradicate Bucket** pop-up window appears.

   If you wish to eradicate mutliple buckets, click **More Options > Eradicate**. The **Eradicate Buckets** pop-up window appears. From the **Destroyed Buckets** list, select the buckets you wish to eradicate. You can optionally enter a partial name in the search box to filter the list. You can also optionally

select all buckets by clicking the checkbox next to the search box. All selected file systems appear in the **Selected Buckets** list.

3 Click **Eradicate**.

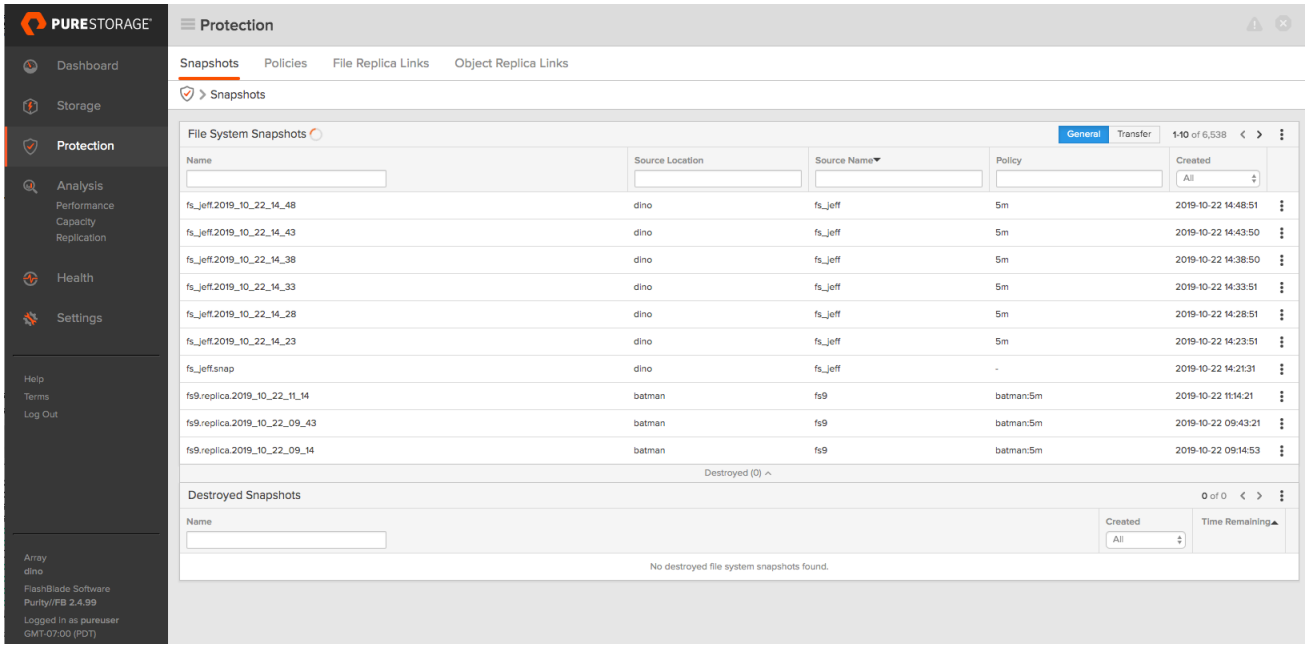
Chapter 6
The Purity//FB GUI Protection Page

The **Protection** page displays and manages the array's file system snapshots, replica links, and replication policies.

The **Protection** page provides the following four tabs:

- **Snapshots**: Displays and allows you to manage snapshots from all file systems.
- **Policies**: Displays and allows you to manage file system snapshot replication policies.
- **File Replica Links**: Displays and allows you to manage all file replica links used to schedule file system snapshot replication.
- **Object Replica Links**: Displays and allows you to manage all object replica links used to schedule object data replication.

Figure 6.1 Purity//FB Protection Page



Snapshots

By default, the **Protection** page displays the **Snapshots** tab. File system snapshots are created from the **Storage > File Systems** page (see *Creating a File System Snapshot*).

The top of the **Snapshots** tab contains the **File System Snapshots** panel, which displays general information about file system snapshots in a table with the following columns:

- **Name**: The name of the file system snapshot.
- **Source Location**: The name of the array from where the source file system was replicated.
- **Source Name**: The name of the source file system.
- **Policy**: The snapshot policy attached to the source file system.
- **Created**: The creation date and time of the snapshot.

Each column can be sorted by clicking the column title.

To view information about the replication transfer status of the file system snapshots, click **Transfer** from the title bar of the **File System Snapshots** panel. Note that replication transfer status information is shown only if replication has been configured.

Once selected, the table displayed in the **File System Snapshots** panel changes to display the following columns:

- **Name**: The name of the file system snapshot.
- **Started**: The date and time replication started.
- **Completed**: The date and time replication completed.
- **Transferred**: The amount of data replicated.
- **Progress**: The percentage of replication completed.

As with the **General** table, each column in the **Transfer** table can be sorted by clicking the column title.

The bottom of the **Snapshots** tab contains the **Destroyed Snapshots** panel, which displays all destroyed file system snapshots. Destroyed snapshots can be recovered within a 24-hour period. Once the 24-hour period has passed, the snapshot is permanently destroyed (eradicated). The **Time Remaining** column displays the time remaining for the snapshot to be restored.

Note that with scheduled snapshot replication, if there is a delay in snapshot completing the transfer of one snapshot, the next snapshot (per the policy schedule) is not created until the previous snapshot transfer is completed. For example, if snapshots are scheduled every 30 minutes and a snapshot completes transferring in 31 minutes, the next snapshot is created once that snapshot is completely transferred. The creation times as shown in the **File Systems Snapshots** panel will reflect this.

Destroying File System Snapshots

You can destroy a snapshot from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

To destroy one or more file system snapshots, perform the following:

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.
2. From the table in the **File System Snapshots** panel, locate the file system snapshot you either wish to destroy, or is the point from which earlier snapshots are to be destroyed.
3. If you wish to destroy the snapshot, perform the following:
   a. Click the **More Options** button located at the end of that snapshot's row and select **Destroy**. The **Destroy Snapshot** pop-up window appears.
   b. Click **Destroy**
4. If you wish to destroy multiple snapshots, perform the following:
   a. Click **More Options** > **Destroy** in the **File System Snapshots** panel title bar. The **Destroy File System Snapshots** pop-up window appears.
   b. From the **File System Snapshots** list, select the snapshots you wish to destroy. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the check box next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
   c. Click **Destroy**.
5. If you wish to use the snapshot as the point from which earlier snapshots are destroyed, perform the following:
   a. Click the **More Options** button located at the end of that snapshot's row and select **Destroy older**. The **Destroy Snapshots older than...** pop-up window appears.
   b. Select the snapshots you wish to destroy from the **Older File System Snapshots** pane. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the check box next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
   c. Click **Destroy**.

The destroyed file system snapshot appears under the **Destroyed Snapshots** panel on the **Snapshots** tab. Destroyed snapshots can be recovered within a 24-hour period. Once the 24-hour period has passed, the snapshot is permanently destroyed (eradicated).

Restoring a File System to a Snapshot

Purity//FB allows you to restore a file system to a given point in time from a snapshot (via file system rollback). Restoring a file system to a snapshot overwrites the contents of that file system with data from the snapshot.

File system rollback is available for snapshots created on a FlashBlade array running Purity//FB 3.0.0 or later.

CAUTION: If Pure File SafeMode is enabled, contact Pure Technical Services to perform this action.

CAUTION: When restoring to a snapshot, the file system's configured quota, as well as any configured user/group quotas, do not change to reflect the limit when the snapshot was taken. This can cause the live space usage to increase, resulting in subsequent write failures due to exceeding the quota. You may need to re-adjust your quota settings accordingly.

Note: It is highly recommended that there be no I/O on the file system when performing a restore.

You can restore a file system from a snapshot from either the **Storage** or **Protection** page.

Note: If there are snapshots that are newer than the snapshot you wish to use as the restore point, the newer snapshots must be first destroyed and eradicated before you can proceed with the file system restoration. See *Destroying and Eradicating Newer Snapshots*.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

To restore a file system from a snapshot, perform the following:

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.
2. From the table in the **File System Snapshots** panel, locate the file system snapshot to be used to restore the file system.
3. Click the **More Options** button located at the end of row of the snapshot to be used for restore and select **Restore**. The **Restore File System from Snapshot** pop-up window appears.
4. Click **Restore**.

Destroying and Eradicating Newer Snapshots

Note: Destroyed snapshots can be recovered within a 24-hour period. Once the 24-hour period has passed, the snapshot is permanently destroyed (eradicated).

When restoring a file system from a snapshot, if there are snapshots that are newer than the snapshot you wish to use as the restore point, the newer snapshots must be first destroyed and eradicated before you can proceed with the file system restoration.

You can destroy and eradicate snapshots from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.
2. From the table in the **File System Snapshots** panel, locate the file system snapshot to be used to restore the file system.
3. Click the **More Options** button located at the end of that snapshot's row and select **Destroy newer**. The **Destroy Snapshots newer than...** pop-up window appears.
4. In the **Newer File System Snapshots** pane, select all snapshots by clicking the check box next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
5. Click **Destroy**. The snapshot(s) appear in the **Destroyed Snapshots** panel (directly below the **File System Snapshots** panel).
6. Click **More Options** > **Eradicate** in of the **Destroyed Snapshots** panel. The **Eradicate File System Snapshots** pop-up window appears.
7. From the **Destroyed File System Snapshots** list, select the snapshots you wish to eradicate. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the checkbox next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
8. Click **Eradicate**.

Recovering File System Snapshots

When a file system snapshot is destroyed, you have 24 hours to recover the data. This is known as the eradication pending state. When the eradication pending state expires, the snapshot will be eradicated and you will no longer be able to retrieve the snapshot data.

You can recover a snapshot from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

To recover a previously destroyed file system snapshot, perform the following:

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.
2. If you wish to recover a single snapshot:
   a. From the **Destroyed Snapshots** panel, locate the file system snapshot you wish to recover.
   b. Click **More Options** > **Recover** at the end of the row of the file system snapshot you wish to recover.
   c. Click **Recover**.
3. If you wish to recover multiple file system snapshots:
   a. Click **More Options** > **Recover** in the **Destroyed Snapshots** panel title bar. The **Recover File System Snapshots** pop-up window appears.
   b. From the **Destroyed File System Snapshots** list, select the snapshots you wish to recover. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the check box next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
   c. Click **Recover**.

Eradicating File System Snapshots

During the eradication pending period, you can manually eradicate the destroyed file system to reclaim physical storage space occupied by the destroyed snapshot. Once reclamation starts, the destroyed snapshots can no longer be recovered.

You can eradicate a snapshot from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

CAUTION: If Pure File SafeMode is enabled, contact Pure Technical Services to perform this action.

To eradicate a destroyed snapshot:

1. If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's destroyed snapshots. If you are on the **Protection** page, select **Protection** > **Snapshots**.

2 If you wish to eradicate a single snapshot:

   a In the **Destroyed Snapshots** panel under **File System Snapshots**, click **More Options** > **Recover** at the end of the row of the snapshot you want to eradicate. The **Eradicate Snapshot** pop-up window appears.

   b Click **Eradicate**. The array will begin reclaiming the eradicated space of the snapshot.

3 If you wish to eradicate mutliple snapshots:

   a Click **More Options** > **Eradicate** in of the **Destroyed Snapshots** panel. The **Eradicate File System Snapshots** pop-up window appears.

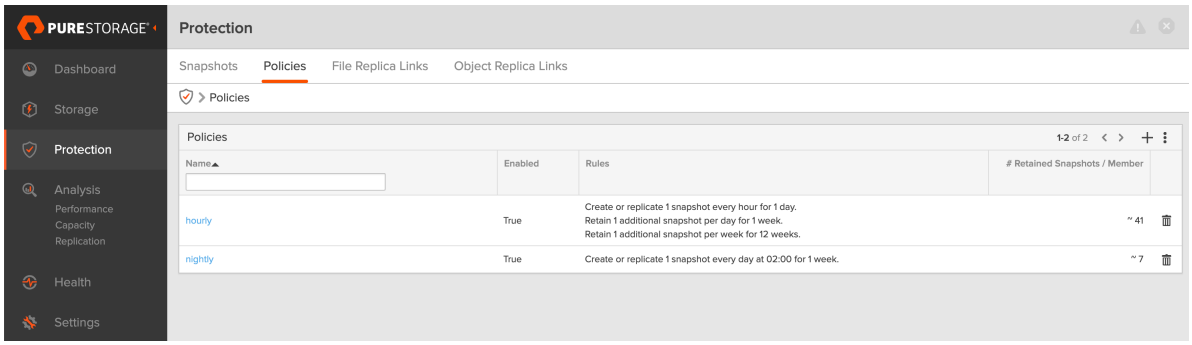
   b From the **Destroyed File System Snapshots** list, select the snapshots you wish to eradicate. You can optionally enter a partial name in the search box to filter the snapshot list. You can also optionally select all snapshots by clicking the checkbox next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.

   c Click **Eradicate**. The array will begin reclaiming the eradicated space of the snapshots.

Policies

A policy consists of one or more rules that govern when snapshots are created or replicated, and how long they are retained. When a policy is attached to a file system, it will create local file system snapshots. When a policy is attached to a file system replica link, it will replicate file system snapshots to a connected array. The **Policies** tab is used to manage the array's snapshot policies.

Figure 6.2 Policies



The **Policies** tab displays a list of all snapshot policies and the following rule information for each snapshot policy:

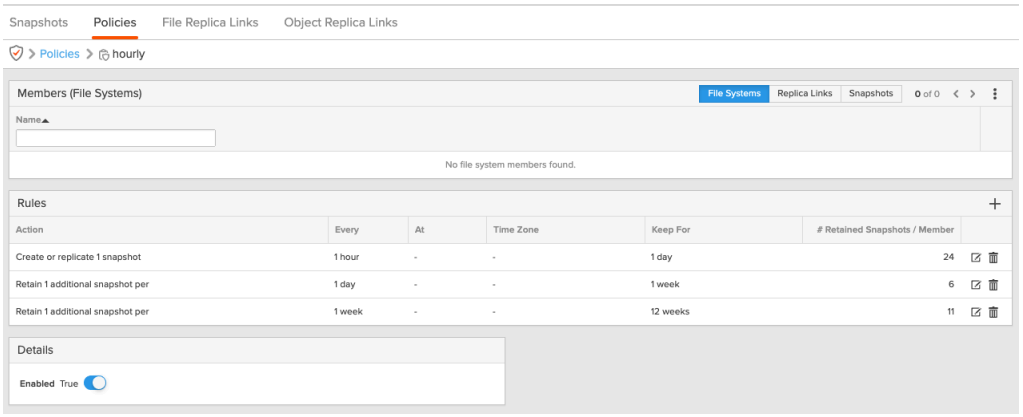
- **Name**: The policy name.
- **Enabled**: Whether the policy is enabled (**True**) or disabled (**False**).
- **Rules**: The snapshot creation and retention details of the policy rules.

- **# Retained Snapshots/Member**: The total number of snapshots retained per member per policy. If multiple rules exist for a policy, a summary is shown of the number of retained snapshots per member for each rule.

Viewing Policy Details

To view details about each policy in the array, from the **Protection** > **Policies** page, click a policy name to open a detail page about that policy.

Figure 6.3 Policy Details



The members, rules, and policy status are displayed in the **Members**, **Rules**, and **Details** panels, respectively.

Creating Policies

To create a policy, peform the following:

1. From the **Protection** > **Policies** page, click the **Add** (+) button in the heading of the **Policies** list. The **Create Policy** pop-up window appears.
2. In the **Name** field enter the name of the snapshot policy.
3. By default, the policy is set to enabled (blue). If you wish to disable the policy, set the **Enabled** toggle to disabled (gray).
4. Click the expand button next to **Create rule for policy** to add rules.
5. In the **Create or replicate 1 snapshot every** field, enter the frequency at which snapshots are created or replicated. The value must be entered in the format n{m|h|d|w}. For example, 15m, 3h, 2d, 1w, etc. The minimum value is 5m.
6. In the **At** field, enter the time of day the snapshot is created. The value must be entered in the format n:[m][am|pm], where n is a value of 0-23. If entering a minute (m) value, m must be a two digit number

between 00 and 59; for example 12:00am or 12:15pm. If entered without am or pm, n must be a value of 0-23. For example, 23, 5pm, 2:25am, 14:45, etc.

This field is only editable if the value specified in the **Create or replicate 1 snapshot every** field is a multiple of a day. For example, 24h, 48h, 72h, 1d, 2d, 3d, 1w, etc

7 In the **And keep for** field, enter the retention period for the snapshot. The value must be entered in the format n{m|h|d|w}. For example, 30m, 1h, 2d, 1w, etc. The retention period cannot be less than the snapshot interval.

8 Click **Create**.

Adding Policy Rules

After creating a snapshot policy, add snapshot creation and retention rules.

To add a pollicy rule, perform the following:

1 From the **Protection** > **Policies** page, click the policy to which you wish to add a rule. The policy's detail page appears.

2 In the **Rules** pane, click the **Add** (+) button. The **Add Rule to Policy <name>** pop-up window appears.

3 In the **Create or replicate 1 snapshot every** field, enter the frequency at which snapshots are created or replicated. The value must be entered in the format n{m|h|d|w}. For example, 15m, 3h, 2d, 1w, etc.

4 In the **At** field, enter the time of day the snapshot is created. The value must be entered in the format n:[m][am|pm], where n is a value of 0-23. If entering a minute (m) value, m must be a two digit number between 00 and 59; for example 12:00am or 12:15pm. If entered without am or pm, n must be a value of 0-23. For example, 23, 5pm, 2:25am, 14:45, etc.

This field is only editable if the value specified in the **Create or replicate 1 snapshot every** field is a multiple of a day. For example, 24h, 48h, 72h, 1d, 2d, 3d, 1w, etc

5 In the **And keep for** field, enter the retention period for the snapshot. The value must be entered in the format n{m|h|d|w}. For example, 30m, 1h, 2d, 1w, etc. The retention period cannot be less than the snapshot interval.

6 Click **Add**.

Adding Multiple Policy Rules

If you added a snapshot creation and retention rule when you created the policy, you can add up to four additional rules to the policy to save and retain snapshots from the original rule.

Consider the following example where a second rule is added to a policy with an existing rule:

- Rule 1 (existing) - One snapshot is created each hour and retained for a day.

- Rule 2 (new) - One snapshot per day will be retained for a week.

Rule 1 results in 24 snapshots being created in a 24-hour period. After 24 hours, the first snapshot that was created is destroyed and another snapshot is created; this cycle is repeated for each snapshot resulting in 24 snapshots each day. Per Rule 2, since one snapshot from Rule 1 is saved each day and retained for a week, six additional snapshots are retained for an additional six days. After one week, a total of 30 snapshots will be retained. This snapshot count is shown in the **#Retained Snapshots/ Member** column of the **Protection** > **Policies** page.

Note: If you are adding an additional retention rule to an existing rule, the values allowed for frequency of snapshot retention and the snapshot retention period are dependent on the snapshot creation rules that were initially created for the policy. For example, if a snapshot is created every 15 minutes and kept for one day, any additional retention frequency rule must be in multiples of 15 minutes and any additional retention periods must be greater than one day.

To add an additional rule, perform the following:

1. From the **Protection > Policies** page, click the policy to which you wish to add a rule. The policy's detail page appears.
2. In the **Rules** pane, click the **Add** (+) button. The **Add Rule to Policy <name>** pop-up window appears.
3. In the **Keep 1 snapshot for every** field, enter the frequency at which snapshots are retained. The value must be entered in the format n{m|h|d|w}, for example 15m.
4. In the **For** field, enter the retention period for the snapshot before it is eradicated. The value must be entered in the format n{m|h|d|w}, for example 30m.
5. Click **Add**.

Editing Policy Rules

Note: Existing snapshots managed by a modified policy will be destroyed if they do not comply with the policy's new snapshot schedule rules.

1. From the **Protection > Policies** page, click the policy whose rules you wish to edit. The policy's detail page appears.
2. In the **Rules** pane, click the **Edit** button on the row of the rule you wish to edit. The **Edit Rule for Policy <name>** pop-up window appears.
3. In the **Create or replicate 1 snapshot every** field, enter the frequency at which snapshots are created. The value must be entered in the format n{m|h|d|w}, for example 15m.
4. In the **At** field, enter the time of day the snapshot is created. The value must be entered in the format n:[m][am|pm], where n is a value of 0-23. If entering a minute (m) value, m must be a two digit number

between 0 and 59; for example 12:00am or 12:15pm. If entered without am or pm, n must be a value of 0-23. For example, 23, 5pm, 2:25am, 14:45, etc.
This field is only editable if the value specified in the **Create or replicate 1 snapshot every** field is a multiple of days. For example, 24h, 48h, 72h, 1d, 2d, 3d, 1w, etc.

5 In the **And keep for** field, enter the retention period for the snapshot. The value must be entered in the format n{m|h|d|w}, for example 30m.

6 Click **Save**.

Removing Policy Rules

Note: When a policy rule is removed, all snapshots managed by that rule are destroyed. Those destroyed snapshots will be eradicated after 24 hours.

1 From the **Protection** > **Policies** page, click the policy whose rules you wish to remove. The policy's detail page appears.

2 In the **Rules** pane, click the **Remove** button on the row of the rule you wish to remove. The **Remove Policy Rule** pop-up window appears.

3 Click **Remove**.

Adding File Systems to a Policy

1 From the **Protection** > **Policies** page, click the policy to which you wish to add file systems. The policy's detail page appears.

2 In the **Members** pane, click **More Options** > **Add File Systems**. The **Add File Systems** pop-up window appears.

3 From the **Available File Systems** pane, select one or more file systems from the list. The selected file system(s) appears in the **Selected File Systems** pane.

4 Click **Add**.

Removing File Systems from a Policy

⚠ **Important note!** After removing a file system from a policy, the policy will no longer create snapshots of that file system, but the policy will continue to destroy and eradicate the snapshots it created.

To remove a single file system:

1 From the **Protection** > **Policies** page, click the policy from which you wish to remove a file system. The policy's detail page appears.

2. In the **Members** pane, click the **Remove** (**X**) button in the row of the file system you wish to remove. The **Remove Member from Policy** dialog box appears.
3. Click **Remove**.

To remove multiple file systems:

4. From the **Protection** > **Policies** page, click the policy to which you wish to remove file systems. The policy's detail screen appears.
5. In the **Members** pane, click **More Options** > **Remove Members**. The **Remove Members** pop-up window appears.
6. From the **Current Members** pane, select one or more file systems from the list. The selected file system(s) appears in the **Selected File Systems** pane.
7. Click **Remove**.

Adding Replica Links to a Policy

1. From the **Protection** > **Policies** page, click the policy to which you wish to add file systems. The policy's detail page appears.
2. In the **Members** pane, click **More Options** > **Add Replica Links**. The **Add Replica Links** pop-up window appears.
3. From the **Available Replica Links** pane, select one or more replica links from the list. The selected replica link(s) appears in the **Selected Replica Links** pane.
4. Click **Add**.

Removing Replica Links from a Policy

To remove a single replica link from a policy:

1. From the **Protection** > **Policies** page, click the policy from which you wish to remove a replica link. The policy's detail page appears.
2. In the **Members** pane, click the **Remove** (**X**) button in the row of the replica link you wish to remove. The **Remove Member from Policy** dialog box appears.
3. Click **Remove**.

To remove multiple repolica links:

4. From the **Protection** > **Policies** page, click the policy from which you wish to remove freplica links. The policy's detail screen appears.

5 In the **Members** pane, click **More Options** > **Remove Members**. The **Remove Members** pop-up window appears.

6 From the **Current Members** pane, select one or more replica links from the list. The selected replica link(s) appears in the **Selected Members** pane.

7 Click **Remove**.

Enabling and Disabling Snapshot Policies

⚠ **Important note!** After a policy is disabled, that policy will no longer create snapshots or eradicate expired snapshots.

1 From the **Protection** > **Policies** page, click the policy you wish to enable or disable. The policy's detail screen appears.

2 In the **Details** pane, set **Enabled** to **True** (blue) to enable or **False** (gray) to disable the policy.

Removing a Policy from a Snapshot

Policies can be removed from snapshots at any time.

Note: If a policy is removed from a snapshot, that snapshot is no longer managed by that policy and must be destroyed manually.

You can remove a policy from a snapshot from either the **Storage** or **Protection** page.

The **Storage** page allows you to view snapshots per specific file system. The **Protection** page allows you to view all file system snapshots on the array.

To remove a policy from a snapshot, perform the following:

1 If you are on the **Storage** page, select **Storage** > **File Systems** and then click the file system from the table in the **File Systems** panel to access that file system's snapshots.

2 If you are on the **Protection** page, select **Protection** > **Snapshots**.

3 From the table in the **File System Snapshots** panel, locate the file system snapshot from which you wish to remove a policy.

4 Click the **More Options** button located at the end of that snapshot's row and select **Remove Policy**. The **Remove Policy from Snapshot** pop-up window appears.

5 Click **Remove**.

Deleting Policies

To delete a single policy:

1 From the **Protection** > **Policies** page, locate the policy you wish to delete from the policy list.

2 From the same row as the policy to be deleted, click the **Delete** button. The **Delete Policy** dialog box appears.

3 Click **Delete**.

To delete multiple policies:

4 From the **Protection** > **Policies** page, click **More Options** > **Delete**. The **Delete Policies** pop-up window appears.

5 From the **Existing Policies** pane, select the policies you wish to delete. Each selected policy appears in the **Selected Policies** pane.

For arrays with a large number of policies, you can search for a policy by entering a full or partial policy name in the search box to filter the policy list.

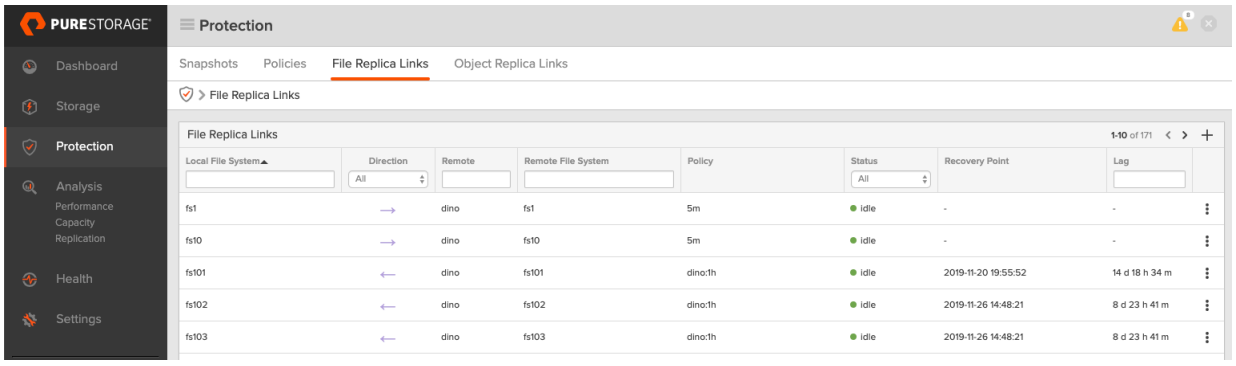
If you wish to delete all policies, from the **Existing Policies** pane, select the checkbox above the list of policies to select all policies.

6 Click **Delete**.

File Replica Links

File replica links are required for replicating file system snapshots between two connected FlashBlades. The **File Replica Links** tab is used to view and manage file replica links.

Figure 6.4 File Replica Links



The **File Replica Links** tab displays a list of all configured file replica links:

- **Local File System**: The name of the local (source) file system.
- **Direction**: The direction of replication, either inbound (←) or outbound (→).
- **Remote**: The name of the connected array.
- **Remote File System**: The name of the remote (target) file system.
- **Policy**: The name of the applied replication policy.
- **Status**: The current replication status.
  - **idle**: No errors, no current replication.
  - **replicating**: No errors, currently attempting replication.
  - **unhealthy**: No connectivity.
- **Recovery Point**: The creation time of the last snapshot that was completely replicated.
- **Lag**: The amount of time, if any, the replication target is behind the source, and is the time difference between the current time and the recovery point.

Clicking the file replica link on the local file system name directs you to the **Storage** > **File Systems** page, which provides detailed information about the file system to which the replica link is attached.

Creating a File Replica Link

File replica links are used to configure file system replication between two connected FlashBlade arrays.

To create a file replica link, perform the following:

1. From the **Protection** > **File Replica Links** page, click the add (+) button in the heading of the **File Replica Links** list. The **Create File Replica Link** pop-up window appears.
2. From the **Local File System** list, select the local (source) file system to be replicated.
3. From the **Remote Connection** list, select the remote (target) array.
4. (Optional) Enter a name for the remote file system to which data from the local file system will be replicated in the **Remote File System** field.
5. (Optional) From the **Policy** list, select a replication policy to attach to the replica link. If no policies are listed, click **Create Policy**. Follow the policy creation instructions as described in *Creating Policies*.
6. Click **Create**.

Adding a Policy to a File Replica Link

A snapshot policy can be added to any outbound (➔) replica link.

To add a snapshot policy to a replica link, perform the following:

1. From the **Protection** > **File Replica Links** page, locate the file replica link to which you wish to add a policy and click the **More Options** button located at the end of that replica link's row.
2. Select **Add Policies**. The **Add Policies** pop-up window appears.
3. Select the policy or policies you wish to add to the replica link from the **Available Policies** pane.
4. Click **Add**.

Removing a Policy from a File Replica Link

A snapshot policy can be removed from any outbound (➔) replica link.

To remove a snapshot policy from a replica link, perform the following:

1. From the **Protection** > **File Replica Links** page, locate the file replica link from which you wish to remove a policy and click the **More Options** button located at the end of that replica link's row.
2. Select **Remove Policies**. The **Remove Policies** pop-up window appears.
3. Select the policy or policies you wish to remove from the replica link from the **Available Policies** pane.
4. Click **Remove**.

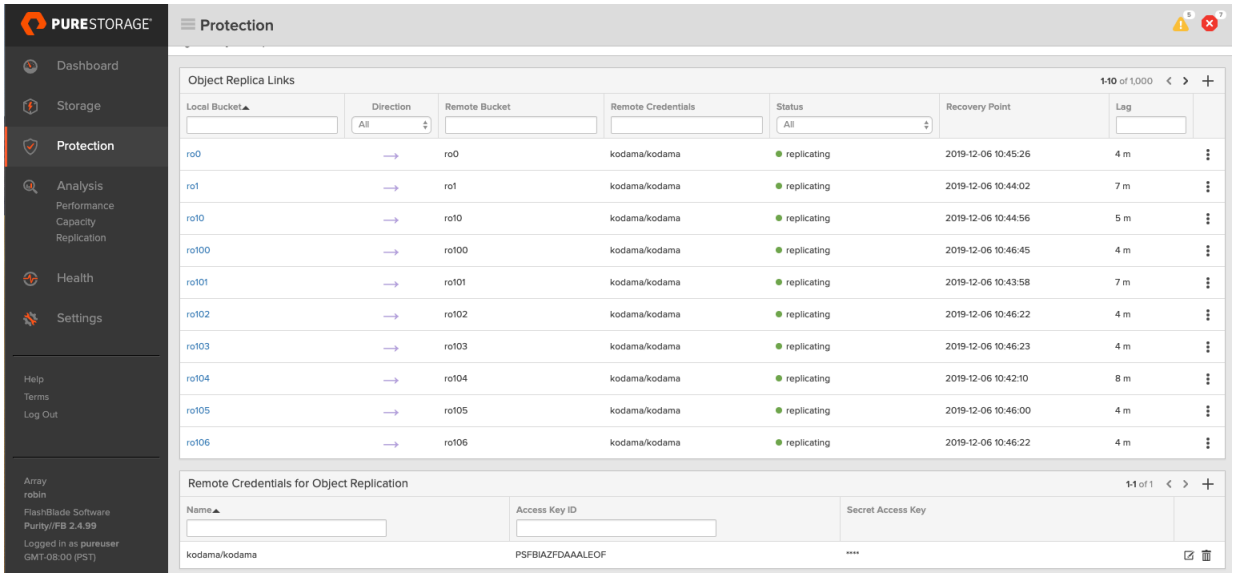
Object Replica Links

Object replica links and remote credentials are required for replicating object data between FlashBlade arrays and S3 targets.

Note: Objects that were created prior to the creation of the object replica link are not replicated. Objects created after the creation of the object replica link will be replicated from the source bucket to the target bucket.

The **Object Replica Links** tab is used to view and manage object replica links and remote credentials.

Figure 6.5 Object Replica Links



The **Object Replica Links** tab contains two panes, **Object Replica Links** and **Remote Credentials for Object Replication**.

The **Object Replica Links** pane displays a list of all configured object replica links:

- **Local Bucket**: The name of the local (source) bucket.
- **Direction**: The direction of replication, outbound (→).
- **Remote Bucket**: The name of the of the remote (target) bucket.
- **Remote Credentials**: The name of the remote credentials is the combined remote array name or S3 target with the remote credential name. For example `array2/cred1`.
- **Status**: The current replication status.
  - **replicating**: No errors, currently attempting replication.
  - **paused**: Data transfer is currently halted due to administrator action.
  - **unhealthy**: No connectivity.
- **Recovery Point**: A point in time where all object creations, updates, and deletions that occurred after the creation of the replica link and before this point in time are guaranteed to have been replicated. Object creations, updates, and deletions that occurred before the creation of a replica link are not replicated. Those that occurred after the recovery point may or may not have been replicated.
- **Lag**: The amount of time, if any, the replication target is behind the source, and is the time difference between the current time and the recovery point.

The **Remote Credentials for Object Replication** pane displays:

- **Name**: The name of the array and user is the combined remote array name or S3 target with the remote credential name. For example `array2/cred1`.
- **Access Key ID**: The access key ID created for the specific user.
- **Secret Access Key**: The secret key is encrypted and only revealed when first created.

Creating an Object Replica Link

Note: Objects created after the creation of the object replica link will be replicated from the source bucket to the target bucket. Any objects that were created prior to the creation of the object replica link are not replicated.

Object replica links allow object data to be replicated between FlashBlade arrays and S3 targets.

To create an object replica link, perform the following:

1. From the **Protection** > **Object Replica Links** page, click the **Add** (+) button in the heading of the **Object Replica Links** list. The **Create Object Replica Link** pop-up window appears.
2. From the **Local Bucket Name** list, select the local (source) bucket from which data will be replicated.
3. From the **Remote Array or Target** list, select the remote (target) array or S3 target.
4. From the **Remote Bucket** field, enter the name of the remote (target) bucket to which data will be replicated. Versioning must be enabled on the target bucket for replication to proceed.
5. From the **Remote Credential** list, select the proper remote credential to allow replication.
6. Click **Create**.

Editing an Object Replica Link

To edit (modify) an object replica link, perform the following:

1. From the **Protection** > **Object Replica Links** page, locate the object replica link you wish to modify and click the **More Options** button located at the end of that replica link's row.
2. Click **Edit**. The **Edit Object Replica Link** pop-up window appears.
3. From the **Remote Credential** list, select the proper remote credential to allow replication.

Pausing an Object Replica Link

Replication between a local bucket and target bucket can be paused at any time by pausing the replica link.

To pause an object replica link, perform the following:

1. From the **Protection** > **Object Replica Links** page, locate the object replica link you wish to pause and click the **More Options** button located at the end of that replica link's row.
2. Click **Pause**. The **Pause Replica Link** pop-up window appears.
3. Click **Pause** to confirm that you wish to pause the replica link.

Resuming an Object Replica Link

Paused replication between a local bucket and target bucket can be resumed at any time by resuming the replica link.

To resume a paused object replica link, perform the following:

1. From the **Protection** > **Object Replica Links** page, locate the paused object replica link you wish to resume and click the **More Options** button located at the end of that replica link's row.
2. Click **Resume**. The **Resume Replica Link** pop-up window appears.
3. Click **Resume** to confirm that you wish to resume the replica link.

Deleting an Object Replica Link

When an object replica link is deleted, replication between the local bucket and target bucket is suspended.

To delete an object replica link, perform the following:

1. From the **Protection** > **Object Replica Links** page, locate the object replica link you wish to delete and click the **More Options** button located at the end of that replica link's row.
2. Click **Delete**. The **Delete Replica Link** pop-up window appears.
3. Click **Delete** to confirm that you wish to delete the replica link.

Creating Remote Credentials

In addition to a replica link, a remote credential is required for replication between a local and target bucket. In order to create a remote credential, the following are required:

- The source and target array, or S3 target, must be connected.
- A previously created user access key and secret access key (see *Creating a User*).

To create remote credentials for a replica link, perform the following:

1. From the **Protection** > **Object Replica Links** page, click the **Add** (+) button in the heading of the **Remote Credentials for Object Replication** list. The **Create Remote Credentials** pop-up window appears.
2. From the **Remote Array or Target** list, select the connected remote (target) array or S3 target.
3. In the **Name** field, enter the credential name.
4. Enter the user access key from the target array or S3 target in the **Access Key ID** field.
5. Enter the user secret access key from the target array or S3 target in the **Secret Access Key** field.
6. Click **Create**.

Editing Remote Credentials

The access key ID and secret access key for remote credentials for object replication can be edited. In order to edit a remote credential, you must have created a new user access key and secret access key (see *Creating a User*).

To edit remote credentials, perform the following:

1. From the **Protection** > **Object Replica Links** page, click the edit button in the same row of the remote credential you wish to edit in the **Remote Credentials for Object Replication** list. The **Edit Remote Credentials** pop-up window appears.
2. Enter the user access key from the target array or S3 target in the **Access Key ID** field.
3. Enter the user secret access key from the target array or S3 target in the **Secret Access Key** field.
4. Click **Save**.

Deleting Remote Credentials

Deleting remote credentials in an object replication setup suspends replication between the array and its target.

To delete remote credentials, peform the following:

1. From the **Protection** > **Object Replica Links** page, click the delete button in the same row of the remote credential you wish to delete in the **Remote Credentials for Object Replication** list.
2. Click **Delete**.

Chapter 7
The Purity//FB GUI Analysis Page

The **Analysis** page displays historical array data, such as I/O performance trends and storage capacity and consumption data.

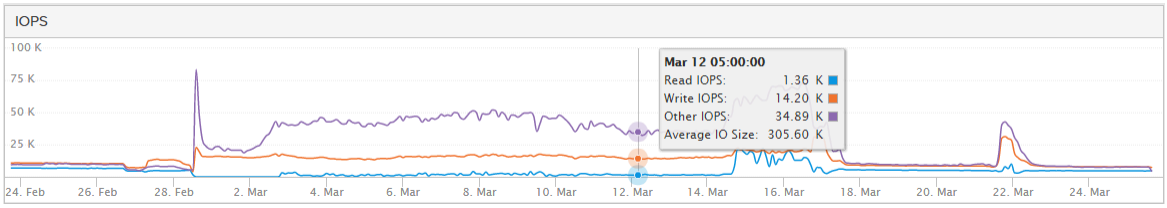
Performance

The **Analysis** > **Performance** panel displays a series of rolling charts consisting of real-time performance metrics for the array and its file systems and buckets. By default, performance metrics are displayed for the array.

The curves in each chart are comprised of a series of individual data points. Hover over any part of a chart to display values for a specific point in time. The values that appear in the point-in-time tooltips are rounded to two decimal places.

The Performance Chart

The following example displays the IOPS statistics on the array at precisely `5:00:00` on `March 12`.



The Performance panel includes the Latency, IOPS, and Bandwidth charts.

Latency

The Latency chart displays the average latency times for various operations.

- **Read Latency (R)** - Average arrival-to-completion time, measured in milliseconds, for a read operation.
- **Write Latency (W)** - Average arrival-to-completion time, measured in milliseconds, for a write operation.
- **Other Latency (O)** - Average arrival-to-completion time, measured in milliseconds, for all other metadata operations.

IOPS

The IOPS (Input/output Operations Per Second) chart displays I/O requests processed per second by the array. This metric counts requests per second, regardless of how much or how little data is transferred in each.

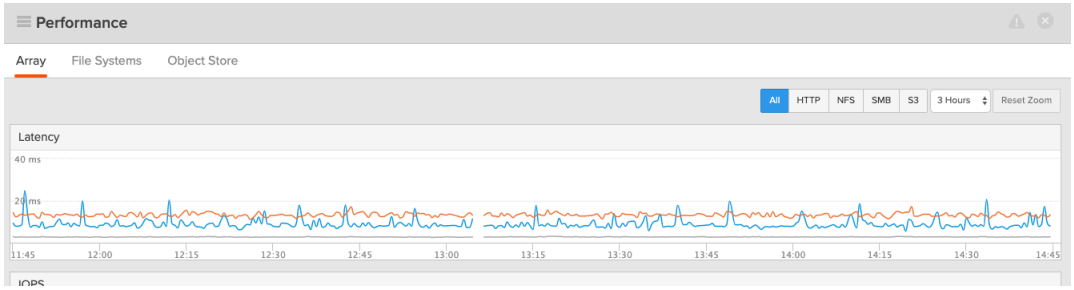
- **Read IOPS (R)** - Number of read requests processed per second.
- **Write IOPS (W)** - Number of write requests processed per second.
- **Other IOPS (O)** - Number of metadata operations processed per second.
- **Average IO Size** - Average I/O size per request processed. Requests include reads and writes.

Bandwidth

The Bandwidth chart displays the number of bytes transferred per second to and from all file systems. The data is counted in its expanded form rather than the reduced form stored in the array to truly reflect what is transferred over the storage network. Metadata bandwidth is not included in these numbers.

- **Read Bandwidth (R)** - Number of bytes read per second.
- **Write Bandwidth (W)** - Number of bytes written per second.

By default, the peformance charts, as described above, are displayed for the array. Click the **File Systems** or **Object Store** tabs to view file system or object store-specific latency, IOPs, and bandwidth performance charts.



The **File Systems** and **Object Store** tabs allow you to view the performance charts for one or more selected NFS file systems or buckets, respectively.

Displaying Protocol-Specific Metrics

By default, the performance charts display data for all supported protocols, as indicated by the **All** button in the upper-right corner of the **Array** tab. Click a protocol button to display only the performance information for that protocol.

The performance charts display data in a user-friendly view that is common for all protocols.

For HTTP and S3 protocol data, there are two display options:

- **General**: The Latency and IOPS graphs display operations that the array is doing in chunks. Default.
- **Protocol Specific**: The Latency and IOPS graphs display a protocol-specific number of user operations.

For example, for HTTP data, in the **General** view, operations such as listing a directory and deleting are both grouped in the Other line. In the **Protocol Specific** view, read and write operations on a directory are split out into their own line.

The file transfer can be performed in many chunks for one large file. The **General** view shows one operation per chunk transfer, while the **Protocol Specific** view shows only one operation per file. When transferring small files, the number of operations is the same in both the **General** and **Protocol Specific** views.

Displaying a Different Time Range

By default, the performance charts display data for the past 3 hours, as indicated by the time range drop-down button. Click the drop-down button to view performance data over a different time range.

Note about the Performance Charts

The **Dashboard** and **Analysis** pages display the same latency, IOPS, and bandwidth performance charts, but the information is presented differently between the two pages.

In the **Dashboard** page:

- The performance charts are displayed for the array.
- The performance charts are updated once every second.
- The performance charts display up to 5 minutes of historical data.

In the **Analysis** page:

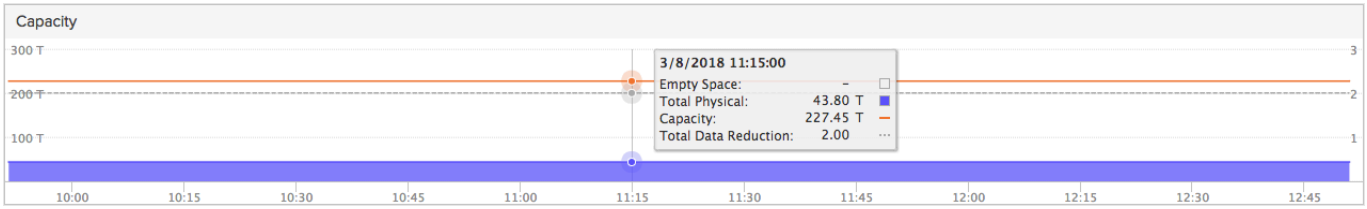
- The performance charts are provided for the array.
- At its shortest range (5m), the performance charts are updated once every second. As the range increases, the update frequency (and resolution) decreases.
- The performance charts display up to one year of historical data.
- The performance charts can be filtered to display metrics by protocol.

Capacity

By default, the **Capacity** graph displays information for all storage on the array. Click **File Systems** or **Object Store** to display information for only that storage type.

The **Capacity** graph displays the following array-wide capacity and consumption information:

- **Empty Space**: Total storage space available on the array.
- **Object Store**: Amount of space that the written data occupies on the array's buckets after reduction via data compression.
- **File Systems**: Amount of space that the written data occupies on the array's file systems after reduction via data compression.
- **Total Physical**: Total storage space used on the array.
- **Capacity**: Total storage capacity of the array.
- **Total Data Reduction**: Ratio of the size of the written data (i.e. Used) versus the amount of space the data occupies after data compression (i.e. Physical). Data reduction is given for all storage on the array, all file systems on the array, and all buckets on the array.
- **Unique**: Total space that the written data occupies on the array's file systems or buckets (Only shown if viewing the File systems or Object Store capacity graph).
- **Snapshots**: Amount of space that the written data occupies on the array's snapshots after reduction via data compression (Only shown if viewing the File systems capacity graph).



Hover over any part of a chart to display values for a specific point in time. The size values that appear in the point-in-time tooltips are rounded to two decimal places.

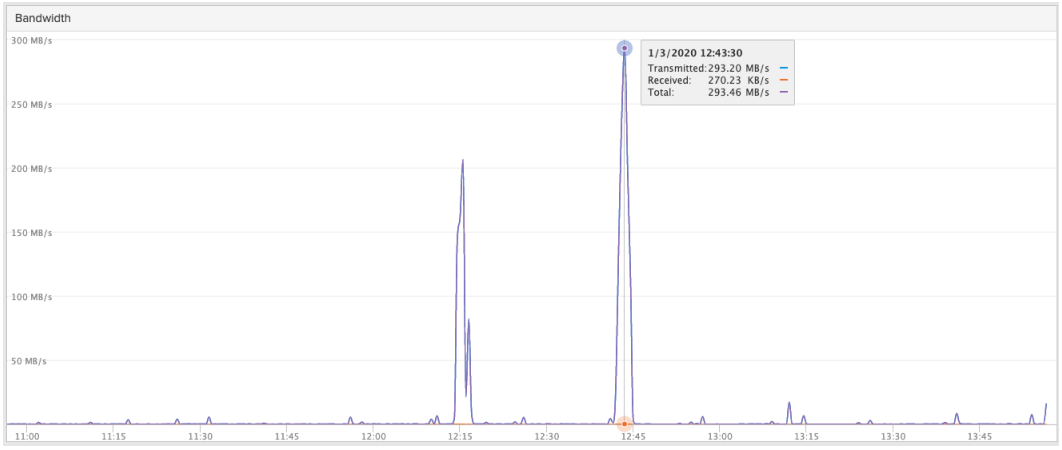
By default, the Capacity graph displays data for the past 3 hours, as indicated by the time range drop-down button. Click the drop-down button to view capacity data over a different time range.

Replication

The replication **Bandwidth** chart displays historical replication bandwidth on the array.

The replication **Bandwidth** chart displays the number of bytes of replication data transferred over the storage network per second between this array and its source arrays, target arrays, and external storage systems (such as S3 buckets), at certain points in time.

Hover over any part of a chart to display values for a specific point in time. The size values that appear in the point-in-time tooltips are rounded to two decimal places.



By default, the point-in-time pop-up in the replication **Bandwidth** chart displays the following metrics:

- **Transmitted**: The bandwidth of data transmitted to connected arrays and S3 targets.
- **Received**: The bandwidth of data received from connected arrays.
- **Total**: Total bandwidth of transmited and received data.

Click **File Systems** or **Object Store** to display information for only that storage type.

By default, the **Bandwidth** chart is displayed for the array. Click the **Connected Arrays** or **Connected Targets** tabs to view array or target-specific bandwidth charts.

The **Connected Arrays** and **Connected Targets** tabs allow you to view the performance charts for one or more selected arrays or targets, respectively.

Chapter 8
The Purity//FB GUI Health Page

The **Health** page displays information to help gauge the health of the array.

Hardware

The **Health > Hardware** page contains information about the array hardware components.

Note: The array displayed will vary depending upon if your FlashBlade array consists of a single or a multi-chassis configuration. For multi-chassis configurations, in addition to front and rear views of the chassis, the display also includes the array's external fabric modules (XFMs).

The hardware panel graphically displays the status of each hardware component, which is useful for diagnosing hardware-related problems.

Figure 8.1 Health - Hardware Page (single chassis)

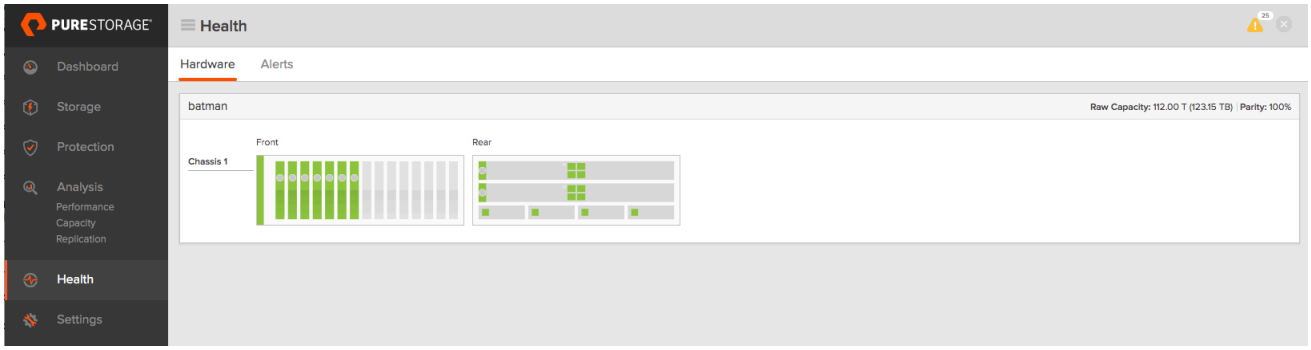
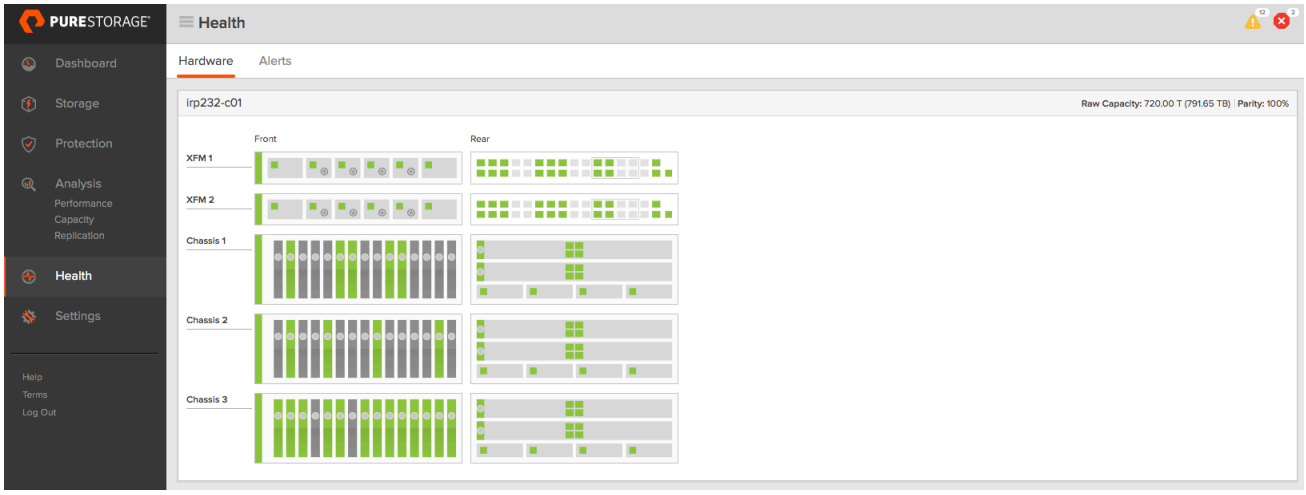


Figure 8.2 Health - Hardware Page (multi-chassis)



External Fabric Modules equipped with optional management ports will be displayed as follows:

Figure 8.3 Health - Management Ports

The view is a schematic representation of the array with colored indicators of each component's status.

The colors within each hardware component represent the component status:

- **Green**: Healthy and functioning properly.
- **Yellow**: Unhealthy, identifying, or unrecognized.
- **Red**: Critical or unknown.
- **Gray**: Disconnected, unused, or unclaimed.

Hover the mouse over a hardware component to display its status and details.

Figure 8.4 Health - Hardware Panel (Front and Rear Views)

The title bar of the hardware panel includes the array name, the raw capacity value, and parity information.

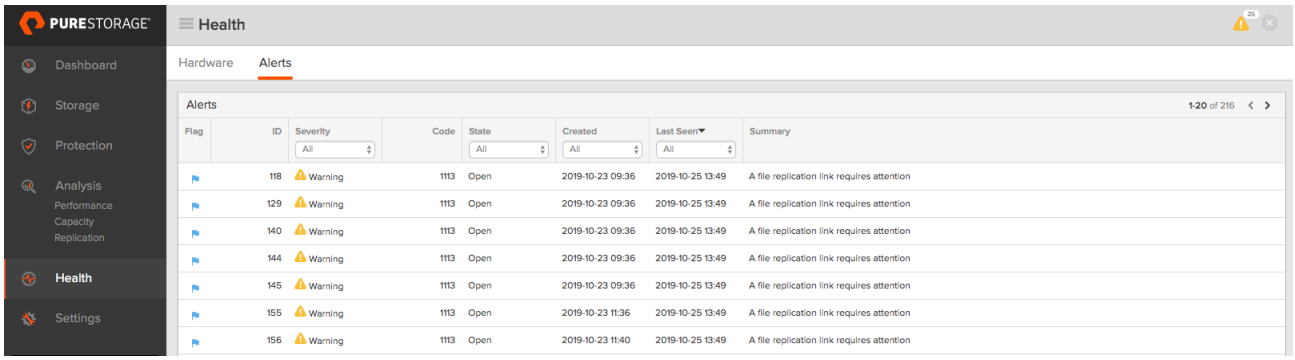
The raw capacity value represents the total usable capacity of the array, displayed in both tebibyte (T) and terabyte (TB) units.

The parity value represents the percentage of data that is fully protected. The value will drop below 100% if the data isn't fully protected, such as when a blade is pulled and the array is rebuilding the data to bring it back to full parity.

Alerts

Purity//FB generates an alert when there is a change to the array or to one of the Purity//FB hardware or software components. The **Health > Alerts** page displays a list of alerts that have been generated on the array.

Figure 8.5 Health - Alerts Page



To conserve space, Purity//FB stores a reasonable number of alert records on the array. Older entries are deleted from the log as new entries are added. To access the complete list of messages, contact Pure Storage Support.

Purity//FB assigns a unique numeric ID to each alert as it is created. By default, alerts are sorted in chronological descending order by "Last Seen" date.

The flag icon represents an alert that has been flagged by Purity or the user. Purity automatically flags all alerts. An alert remains flagged until you have manually cleared the flag to indicate that the alert has been addressed. If there are further changes to the condition that caused the alert (for example, the health of a blade has changed), Purity will set the flag again.

The icons that appear along the left side of each alert in the list output represent the alert severity level:

- **Blue (INFO)** icons represent informational messages generated due to a change in state. INFO messages can be used for reporting and analysis purposes. No action is required.
- **Yellow (WARNING)** icons represent important messages warning of an impending error if action is not taken.

- **Red (CRITICAL)** icons represent urgent messages that require immediate attention.

Click any of the column headings to change the sort order.

Each alert in the list output includes the following information:

- **ID**: Unique number assigned by the array to the alert. ID numbers are assigned to alerts in chronological ascending order.
- **Severity**: Alert severity, categorized as `critical`, `warning`, or `info`.

  Critical alerts are typically triggered by service interruptions, major performance issues, or risk of data loss, and require immediate attention. For example, Purity//FB triggers a critical alert if a blade has been removed from the chassis.

  Warning alerts are of low to medium severity and require attention, though not as urgently as critical alerts. For example, Purity//FB triggers a warning alert if it detects an unhealthy blade that is still processing data.

  Informational (Info) alerts inform users of a general behavior change and require no action. For example, Purity//FB triggers an informational alert if the NFS service is unhealthy.
- **Code**: Pure Storage alert code number that identifies the type of alert event.
- **State**: Current state of the alert. Possible states include: `open`, `closed`, `closing`, and `waiting to downgrade`.

  An alert goes from `open` state to `close` state when the issue is completely resolved, such as when a blade has been removed from the chassis. If the blade is then reinserted and pulled again, a new alert will be generated.

  Certain alerts are generated due to intermittent issues. Examples include temperature sensors reporting values outside of normal operating range and space usage exceeding certain capacity thresholds. These types of alerts can change from `open` state to `closing` or `waiting to downgrade` states.

  And alert changes from `open` state to `closing` state when it is no longer an issue. After the alert remains in `closing` state for 24 hours without change, it changes into `closed` state.

  An alert changes from `open` state to `waiting to downgrade` state when the issue becomes less severe. After the alert remains in `waiting to downgrade` state for 24 hours without change, the alert severity is downgraded. For example, when array capacity reaches 100%, a critical alert is issued with an `open` state. When array capacity drops below 100%, the alert changes to `waiting to downgrade` state. After the array has remained at less than 100%, but more than 90% capacity, for 24 hours, the alert severity is downgraded to `warning`.
- **Created**: Date and time the alert was first generated and initial alert email notifications were sent to alert watchers.
- **Last Seen**: Most recent date and time Purity//FB saw the issue that generated the alert.
- **Summary**: Alert summary.

Click anywhere in an alert row to display the alert details. Some alert detail descriptions include a **more info** link providing additional information about the alert through the Pure Storage Knowledge Base.

The identify light for blades and fabric modules can be illuminated from the GUI by clicking that individual component from the image of the array from the **Health** page. The identify light in the image will turn blue and the physical identify light will illuminate to assist with locating the physical component in the data center. Click the component again to turn off the indicator light.

Flagging Alert Messages

To flag alert messages:

1. Select **Health** > **Alerts**.
2. Locate the alert you want to flag.
3. Click the flag icon. When the icon becomes blue, the alert is flagged.

Unflagging Alert Messages

To unflag alert messages:

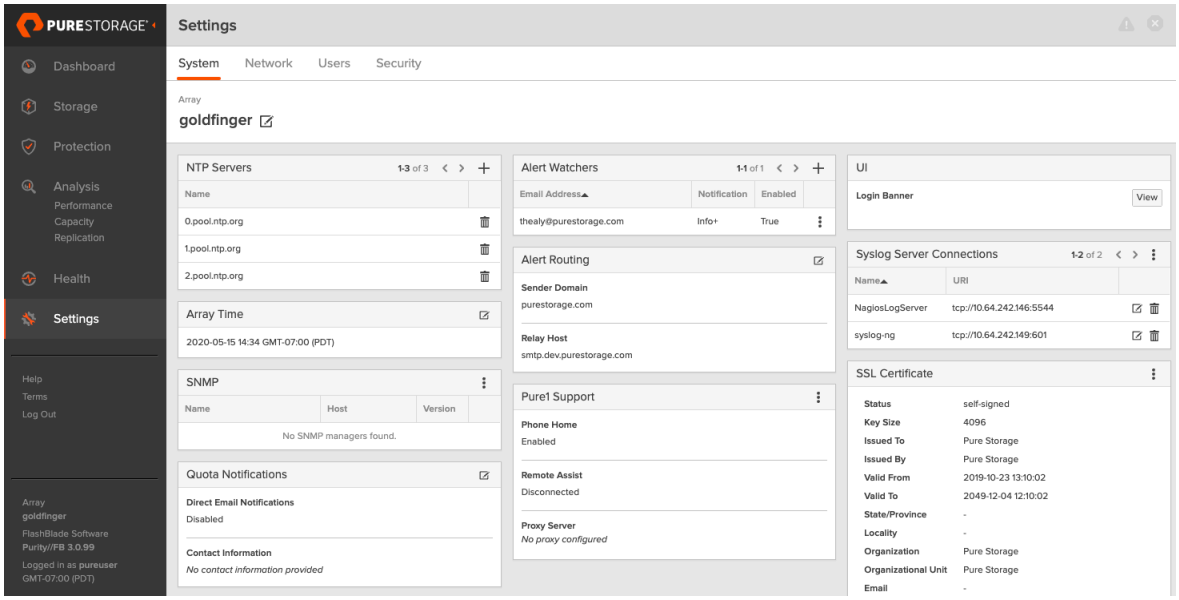
1. Select **Health** > **Alerts**.
2. Locate the alert you want to unflag.
3. Click the flag icon. When the icon becomes gray, the alert is unflagged.

Alerts can also be unflagged and dismissed from the **Recent Alerts** panel from the **Dashboard** page by clicking the X to the right of the alert.

Chapter 9
The Purity//FB GUI Settings Page

The **Settings** page displays and manages the general attributes, network settings, and directory service settings of an array.

Figure 9.1 Purity//FB Settings Page

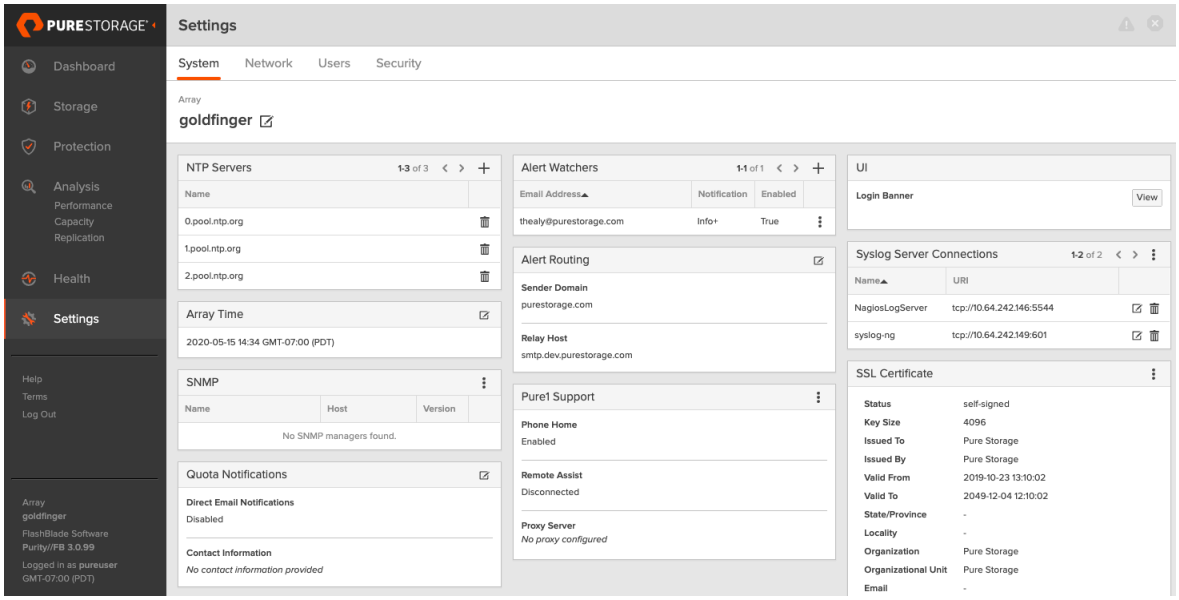


System

The **Settings > System** page displays and manages the general attributes of the FlashBlade array.

The page is made up of panels to view and configure NTP servers, SNMP settings, Pure Storage support settings, alert settings, login banner, and the SSL certificate.

Figure 9.2 Settings - System Page



Renaming the Array

1 Select **Settings** > **System**.

2 Click the edit icon next to the current array name. The array name becomes an editable box.

3 In the editable box, type the new array name.

4 Click the check mark icon to confirm the change.

Time

The **Time** panel displays and manages the time settings for the array.

Array Time

The array time is based on the time zone of the array, which is set during FlashBlade installation.

The array time zone is not related to the user's local time zone that appears in the bottom-left corner of the Navigation pane. The user's local time zone is determined by the browser's local time.

Purity//FB CLI dates and times are displayed in the time zone of the array, while Purity//FB GUI dates and times are displayed in the local user's time zone.

Changing the Time Zone

To change the time zone of the array:

1. Select **Settings** > **System**.
2. In the **Time** panel, click the edit button in the **Array Time** section. The **Edit Array Time** pop-up window appears.
3. Select a new time zone.
4. Click **Save**.

NTP Servers

The **NTP Servers** panel displays the hostnames or IP addresses of the Network Time Protocol (NTP) servers that are currently being used by the array to maintain reference time. The installation technician sets the proper time zone for an array when it is installed. During operation, arrays maintain time synchronization by interacting with the NTP server.

Configuring the NTP Server

The array maintains time synchronization by interacting with the NTP server.

1. Select **Settings** > **System**.
2. In the **NTP Servers** section of the **Time** panel, perform one of the following tasks:
   - To add an NTP server, in the New NTP Server(s) text box, type the hostname or IP address of the NTP server, and then click the Add (+) button. Enter multiple servers as comma-separated values.

     If specifying an IP address, for IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from `0` to `255` representing a group of 8 bits. For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`) .
   - To remove an NTP server, select the check box of the server you want to remove, and then click the **Delete** button.

Pure1 Support

The **Pure1 Support** panel displays and manages the features used to communicate with Pure Storage Support.

Phone Home

The phone home facility allows the array to transmit log and diagnostic information to Pure Storage Support via a secure network connection.

The phone home facility can be enabled and disabled at any time. The current phone home status appears just below the **Phone Home** label.

Enabling and Disabling Phone Home

Enabling phone home functionality allows the array to transmit log and diagnostic information to Pure Storage Support. It is highly recommended that you enable phone home.

To enable (or disable) phone home, perform the following:

1. Select **Settings** > **System**.
2. In the **Pure 1 Support** panel click the **Edit** button. The **Edit Pure1 Support Settings** pop-up window appears.
3. Click the **Phone Home** toggle button to switch between enabled (blue) and disabled (gray) status.
4. Click **Save**.

Remote Assist

In some cases, the most efficient way for Pure Storage Support to service a FlashBlade array or diagnose problems is through direct access to the array. A remote assistance (RA) session grants Pure Storage Support direct and secure access to the array through a reverse tunnel which you, the administrator, open. This is a two-way communication.

Opening an RA session gives Pure Storage Support the ability to log into the array, effectively establishing an administrative session. Once the RA session is successfully established, the array returns connection details, including the date and time when the session was opened, the date and time when the session expires, and the proxy status (true, if configured).

After the Pure Storage Support team has performed all of the necessary diagnostic or maintenance functions, close the RA session to terminate the connection.

RA sessions can be opened and closed at any time. The current status of the remote assistance session appears just below the Remote Assist label.

An open RA session automatically terminates (closes) after two days have lapsed.

Opening and Closing a Remote Assistance (RA) Session

A remote assistance (RA) session grants Pure Storage Support direct and secure access to the array through a reverse tunnel which you, the administrator, open. This is a two-way communication.

To open or close an RA session, perform the following:

1. Select **Settings** > **System**.
2. In the **Pure 1 Support** panel click the **Edit** button.The **Edit Pure1 Support Settings** pop-up window appears.
3. Click the **Remote Assist** toggle button to connect (blue) and disconnect (gray) an RA session. Opening an RA session gives Pure Storage Support direct and secure access to the array.
4. Click **Save**.

Support Logs

Purity//FB continuously logs a variety of array activity, including performance metrics, hardware and software operations, and administrative actions. The logs contain the activity of both fabric modules and all blades. The time stamp of each log is based on the array time.

If Phone Home is enabled, the logs are periodically transmitted to Pure Storage. The logs are also saved to the array with up to 30 days' worth of activity available for manual download.

When downloading support logs, specify an event date and time. The array generates a file containing a chronological list of array activity that occurred leading up to, during, and a little after the specified event time. Log files are password encrypted and can only be opened by Pure Storage Support.

Downloading Support Logs

Purity//FB continuously logs a variety of array activity. These logs contain the activity of both fabric modules and all blades. These logs are used by Pure Storage Support when assessing the array's performance and for diagnosting and troubleshooting any events or issues.

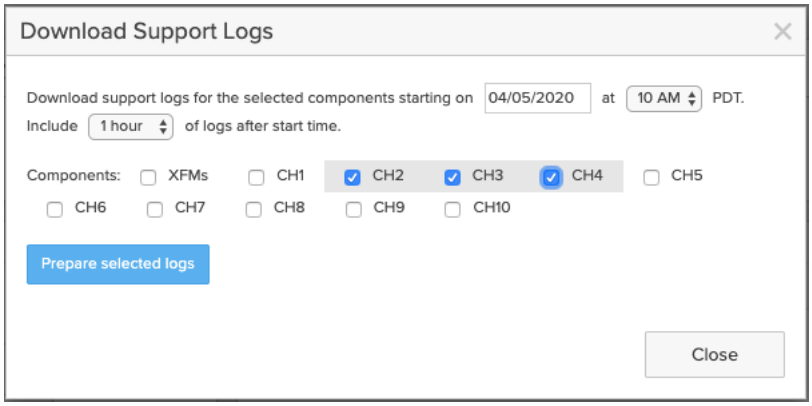
To download logs, perform the following:

1. Select **Settings** > **System**.
2. In the **Support Logs** section of the **Pure1 Support** panel, click **Download**. The **Download Support Logs** dialog box appears.
3. Specify the event date and time, representing the approximate array time the activity of interest occurred, and the duration of time for which you wish log information to be collected (1 to 6 hours). Note that the time is based on your local browser's time zone.

4 If you have a multi-chassis system, optionally select the component(s) for which you wish to prepare support logs.

Figure 9.3 Downloading Support Logs for Single-Chassis Systems

Figure 9.4 Downloading Support Logs for Multi-Chassis Systems



5 If you did not select a component, click **Prepare all logs**. If you selected one or more components, click **Prepare selected logs**.

6 Click **Download** to save the password encrypted file to your administrative workstation. The files can only be opened by Pure Storage Support.

Proxy Server

The proxy hostname, if set, represents the server to be used as the HTTP or HTTPS proxy. The format for the proxy host name is `http(s)://hostname:port`, where `hostname` is the name of the proxy host, and `port` is the TCP/IP port number used by the proxy host.

Configuring the Proxy Host Name

1 Select **Settings** > **System**.

2 In the **Pure 1 Support** panel click the **Edit** button. The **Edit Pure1 Support Settings** pop-up window appears.

3 In the **Proxy Server** field type the proxy host name.

The format for the host name is `http(s)://hostname:port`, where `hostname` is the name of the proxy host, and `port` is the TCP/IP port number used by the proxy host.

4 Click **Save**.

Deleting the Proxy Host Name

1 Select **Settings** > **System**.

2 In the **Pure 1 Support** panel click the **Edit** button. The **Edit Pure1 Support Settings** pop-up window appears.

3 In the **Proxy Server** field, delete the proxy host name.

4 Click **Save**.

Alert Watchers

Purity//FB generates an alert whenever the health of a component degrades or a capacity threshold is reached. Alerts can also be sent as email notifications to designated alert watchers.

The **Alert Watchers** panel displays the email addresses of designated alert watchers, the severity level of alert notifications that alert watchers will receive, and the status of each watcher. The sending account name for Purity//FB alert email notifications is the array name at the configured sender domain.

Once added, an alert watcher will start receiving alert email notifications.

By default, alert watchers receive all alert email notifications regardless of alert severity. You can optionally configure alert watchers to receive only critical and warning-level alert notifications, or only critical-level alert notifications.

Alert watchers can be in enabled (**True**) or disabled (**False**) status. Alert watchers who are in enabled status receive alert email notifications. When an alert watcher is created, its watcher status is automatically set to enabled status. Alert watchers who are in disabled status do not receive alert email notifications. Disabling an alert watcher does not delete the recipient's email address, it only stops the watcher from receiving alert notifications. Alert watchers can be enabled and disabled at any time.

Deleting an alert watcher completely removes the watcher from the list. Once an email address has been deleted, the corresponding alert watcher will no longer receive alert notifications.

Adding an Alert Watcher

Add an alert watcher to receive email alert notifications.

To add an alert watcher, perform the following:

1. Select **Settings** > **System**.
2. In the **Alert Watchers** panel, click the Add (+) button. The **Add Alert Watchers** pop-up window appears.
3. Enter the email address of the alert watcher in the **Email** field.
4. (Optional) Select a severity level (**Info+**, **Warning+**, or **Critical**) from the **Notification Severity** menu.
   - **Info+**: If selected, the alert watcher will receive all alert notifications for all severity levels.
   - **Warning+**: If selected, the alert watcher will receive any alert with a severity of Warning or more severe. Info-level alert notifications will not be received.
   - **Critical**: If selected, the alert watcher will only receive the most severe Critical-level alert notifications.
5. (Optional) Click the **Send Test Email** toggle to enable (blue) to send a test email to the configured email address when saving the alert watcher.
6. Click **Save**.

Enabling and Disabling an Alert Watcher

In order for an alert watcher to receive email alert notifications, the alert watcher must be enabled. By default, alert watchers are enabled at creation.

1. Select **Settings** > **System** and navigate to the **Alert Watchers** panel.
2. In the row of the alert watcher for whom you wish to enable or disable, click the **More Options** button and select **Edit**. The **Edit Alert Watchers** pop-up window appears.
3. Click the **Enabled** toggle switch to set the alert watcher to enabled (blue) or disabled (gray).
4. Click **Save**.

Setting the Severity Level for Alert Watcher Notifications

By default, an alert watcher receives all alert email notfications, regardless of severity. You can configure the severity level of alert that your alert watchers will receive.

To set the severity level for alert watcher notifications, perform the following:

1. Select **Settings** > **System** and navigate to the **Alert Watchers** panel.
2. In the row of the alert watcher for whom you wish to set a notification severity level, click the **More Options** button and select **Edit**. The **Edit Alert Watchers** pop-up window appears.

3 Select a severity level (**Info+**, **Warning+**, or **Critical**) from the **Notification Severity** menu.

- **Info+**: If selected, the alert watcher will receive all alert notifications for all severity levels.
- **Warning+**: If selected, the alert watcher will receive any alert with a severity of Warning or more severe. Info-level alert notifications will not be received.
- **Critical**: If selected, the alert watcher will only receive the most severe Critical-level alert notifications.

4 Click **Save**.

Sending a Test Email

You can test an array 's ability to send email notifications to alert watchers, designated or not. Two types of test messages can be issued:

- Before creating an alert watcher, send a test message to the alert watcher's email address to ensure alert notifications can reach the intended destination.
- At any time, send a test message to all of the enabled alert watchers to ensure alert notifications reach their intended destinations.

To send an test email, perform the following:

1 Select **Settings** > **System** and navigate to the **Alert Watchers** panel.

2 In the row of the alert watcher for whom you wish to send a test email, click the **More Options** button and select **Send Test Email**.
If the email is not valid, an error message appears.

Deleting an Alert Watcher

Once an email address has been deleted, the corresponding alert watcher will no longer receive alert notifications.

1 Select **Settings** > **System** and navigate to the **Alert Watchers** panel.

2 In the row of the alert watcher you wish to delete, click the **More Options** button and select **Delete**.

Alert Routing

The **Alert Routing** panel displays the ways in which alerts and logs are managed.

Relay Host

The relay host represents the hostname or IP address of the email relay server currently being used as a forwarding point for alert email notifications generated by the array. If a relay host is not configured, Purity//FB sends all alert email notifications directly to the recipient addresses rather than route them via the relay (mail forwarding) server.

Configuring the SMTP Relay Host

1. Select **Settings** > **System**.
2. In the **Relay Host** section of the **Alert Routing** panel, click the edit icon. The field becomes an editable box.
3. In the editable box, type the host name or IP address of the email relay server that is to be used as the forwarding point for alert email notifications generated by the array. If specifying an IP address, enter the IPv4 or IPv6 address.

   For IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from `0` to 255 representing a group of 8 bits. If a port number is also specified, append it to the end of the address in the form `ddd.ddd.ddd.ddd:PORT`, where `PORT` represents the port number.

   For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`) . If a port number is also specified, enclose the entire address in square brackets (`[]`) and append the port number to the end of the address. For example, `[xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]:PORT`, where `PORT` represents the port number.
4. Click the check mark icon to confirm the change.

Deleting the SMTP Relay Host

1. Select **Settings** > **System**.
2. In the **Alert Routing** panel, click the edit icon next to the relay host name or IP address. The host name or IP address becomes an editable box.
3. Delete the host name or IP address.
4. Click the check mark icon to confirm the deletion.

Sender Domain

The sender domain determines how logs are parsed and treated by Pure Storage Support and Escalations. The domain name is also used in the "from" address of outgoing alert email notifications. The domain name must be set to your company's domain name. For example, `mydomain.com`.

Configuring the Sender Domain

1. Select **Settings** > **System**.
2. In the **Sender Domain** section of the **Alert Routing** panel, click the edit icon. The field becomes an editable box.
3. In the editable box, type the sender domain name. The domain name must be set to your company's domain name. For example, `mydomain.com`.
4. Click the check mark icon to confirm the change.

SSL Certificate

Purity creates a self-signed certificate and private key when you start the system for the first time. The SSL Certificate panel allows you to view the certificate and import or export certificates and private keys. Intermediate certificates can also be imported to the system.

Imported SSL certificates must be PEM formatted (Base64 encoded), include the " `-----BEGIN CERTIFICATE-----` " and " `-----END CERTIFICATE-----` " lines.

Importing the SSL Certificate

1. Select **Settings** > **System**.
2. Click the **More Options** button from the **SSL Certificate** panel.
3. Click **Import Certificate**.
4. Complete or modify the following fields:
   - **Certificate** - Click **Choose File** and select the signed certificate. Verify the certificate is PEM formatted (Base64 encoded), include the " `-----BEGIN CERTIFICATE-----` " and " `-----END CERTIFICATE-----` " lines.
   - **Private Key** - Click **Choose File** and select the private key.
   - **Intermediate Certificate** - (Optional) Click **Choose File** and select the intermediate certificate.
   - **Key Passphrase** - (Optional) If the private key is encrypted with a passphrase, enter the passphrase.
5. Click **Import**. The page will refresh in several seconds.

Exporting the SSL Certificate

1. Select **Settings** > **System**.

2 Click the **More Options** button from the **SSL Certificate** panel.

3 Click **Export Certificate**.

4 Click **Download**.

Quota Notifications

Direct email notifications can optionally be enabled to send emails to users and groups when they approach or exceed their file system quota. An email with a warning message is sent when a quota reaches 80% capacity and again at 90% capacity. When a quota is at 100% capacity, a critical email is sent. Emails are sent every 24-hours until the quota falls below the 80% capacity threshold.

A direct email notification can optionally be configured to include the administrator's contact information which the user and group can use to inquire about their quota and manageability options, such as requesting an increase of the file system quota. The FlashBlade administrator can include a free-form string. The contact information can include a name, phone number, or email. For example **`John Doe, Storage Admin Team. jdoe@example.com (555-867-5309)`**.

Enabling Quota Notifications

If you wish for users and/or groups to receive notification when they approach or exceed their file system quota, configure (enable) quota notifications.

To enable quota notifications, perform the following:

1 Select **Settings** > **System**.

2 In the **Quota Notifications** panel, click the edit icon. The **Edit Quota Notifications** pop-up window appears.

3 Set the **Direct Email Notifications Enabled** toggle button to enable (blue) quota notifications.

4 In the **Contact Information** field, optionally enter the contact information of FlashBlade administrator that users and groups should reach out to if they have questions about their quota. This may be an email, a phone number, a name, or some combination thereof. The contact field cannot exceed 256 characters in length.

5 Click **Save**.

SNMP

The Simple Network Management Protocol (SNMP) is used by SNMP agents and SNMP managers to send and retrieve information. FlashBlade supports SNMP versions v2c and v3.

FlashBlade's built-in SNMP agent has local knowledge of the array. The agent collects and organizes this array information and translates it via SNMP to or from the SNMP managers. The agent cannot be

deleted. The managers are defined by creating SNMP manager objects on the array. The managers communicate with the agent via the standard TCP port 161, and they receive notifications on port 162.

The SNMP agent has two functions, namely, responding to GET-type SNMP requests and transmitting alert messages.

The agent responds to GET-type SNMP requests made by the SNMP managers, and return values for an information block, such as purePerformance, or individual variables within the block, depending on the type of request issued. The following variables are supported:

Table 9.1 Supported Variables

Variable	Description
pureArrayReadBandwidth	Array read bandwidth (bytes/s)
pureArayWriteBandwidth	Array write bandwidth (bytes/s)
pureArrayReadIOPS	Array read IOPS (op/s)
pureArrayWriteIOPS	Array write IOPS (op/s)
pureArrayReadLatency	Array read latency (us/op)
pureArrayWriteLatency	Array write latency (us/op)
pureArrayAverageOperationSize	Array average operation size (bytes)
pureArrayAverageReadSize	Array average read size per read operation (bytes)
pureArrayAverageWriteSize	Array average write size per write operation (bytes)
pureArrayOtherIOPS	Array other operations process per second (op/s)
pureArrayOtherLatency	Array other latency (us/op)

The FlashBlade Management Information Base (MIB) describes the purePerformance variables and can be downloaded from the array to your local machine using the GUI or displayed using the CLI or REST API.

Note: MIB-2 object identifiers (OIDs) are not supported.

SNMP managers are added to the array through the creation of SNMP manager objects. When creating an SNMP manager object, enter the Host, which represents the DNS hostname or IP address of the

computer that hosts the SNMP manager. Also specify the SNMP version from the Version drop-down list. Valid versions are v2c and v3.

The SNMP agent generates and transmits messages to the SNMP manager as traps or inform requests (informs), depending on the notification type that is configured on the manager. An SNMP trap is an unacknowledged SNMP message, meaning the SNMP manager does not acknowledge receipt of the message. An SNMP inform is an acknowledged trap.

If the SNMP manager notification type is set to `trap`, the agent sends the SNMP message (trap) without expecting a response. If the SNMP manager is set to `inform`, the agent sends the SNMP message (inform) and waits for a reply from the manager confirming message retrieval. If the agent does not receive a response within a certain timeframe, it will retry until the inform has passed through successfully. If the notification type is not set, the manager defaults to `trap`.

SNMPv2 uses a type of password called a community string to authenticate the messages that are passed between the agent and manager. The community string is sent in clear text, which is considered an unsecured form of communication. SNMPv3 supports secure communication between the agent and manager through the use of authentication and privacy encryption methods. As such, SNMPv2c and SNMPv3 have different security attributes.

To configure the SNMPv2c agent and managers, set the **Community** field to the community string under which the agent is to communicate with the managers. The agent and manager must belong to the same community; otherwise, the agent will not accept requests from the manager. When setting the community, Purity prompts twice for the community string. To remove the agent or manager from the community, leave the field blank.

To configure the SNMPv3 agent and managers, in the **User** field, specify the user ID that Purity uses to communicate with the SNMP manager. Also set the authentication and privacy encryption security levels for the agent and managers. SNMPv3 supports the following security levels:

- **noAuthNoPriv**: Authentication and privacy encryption is not set. Similar to SNMPv2c, communication between the SNMP agent and managers is not authenticated and not encrypted. noAuthNoPriv security requires no configuration.

- **authNoPriv**: Authentication is set, but privacy encryption is not set. Communication between the SNMP agent and managers is authenticated but not encrypted. Password authentication is based on MD5 or SHA hash authentication.

  To configure authNoPriv security, in the **Auth Protocol** field, set the authentication protocol to MD5 or SHA, and in the **Auth Passphrase** field, enter an authentication passphrase.

- **authPriv**. Communication between the SNMP agent and managers is authenticated and encrypted. Password authentication is based on MD5 or SHA hash authentication. Traffic between the FlashBlade and SNMP manager is encrypted using encryption protocol AES or DES.

  To configure authPriv security, in the **Auth Protocol** field, set the authentication protocol to MD5 or SHA, and in the **Auth Passphrase** field, enter an authentication passphrase. Also, in the **Privacy Protocol** field, set the privacy protocol to AES5 or DES, and in the **Privacy Passphrase** field, enter a privacy passphrase.

  Note that privacy cannot be configured without authentication

Once an SNMP manager object is created on the array, the FlashBlade immediately starts transmitting SNMP messages and alerts to the manager.

The **SNMP** panel displays configured SNMP managers. The **SNMP** panel also allows you to configure and manage SNMP managers and agents. From the panel you can also download the SNMP MIB.

The **SNMP** panel displays a list of configured SNMP managers with the following information:

- **Name**: The name of the SNMP manager.
- **Host**: The host address configured with the SNMP manager.
- **Version**: The SNMP version.

Downloading the Management Information Base (MIB) File

To download the MIB file, perform the following:

1. Select **Settings** > **System**.
2. From the header of the **SNMP** panel, click **More Options** > **Download MIB**.
   The `PURESTORAGE-MIB.txt` immediately downloads to the default download location on your local machine.

Creating an SNMP Manager Object

SNMP managers are added to the array through the creation of SNMP manager objects.

To add an SNMP manager, object perform the following:

1. Select **Settings** > **System**.
2. From the header of the **SNMP** panel, click **More Options** > **Add SNMP Manager**. The **Add SNMP Manager** pop-up window appears.
3. Complete the following fields:
   - **Name**: The SNMP manager's name.
   - **Host**: DNS hostname or IP address of a computer that hosts an SNMP manager to which Purity is to send messages when it generates alerts. If specifying an IP address, enter the IPv4 or IPv6 address.

     For IPv4, specify the IP address in the form ddd.ddd.ddd.ddd, where ddd is a number ranging from 0 to 255 representing a group of 8 bits. If a port number is also specified, append it to the end of the address in the format ddd.ddd.ddd.ddd:PORT, where PORT represents the port number.

     For IPv6, specify the IP address in the form xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx, where xxxx is a hexadecimal number representing a group of 16 bits. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (::). If a port number is also specified,

enclose the entire address in square brackets ([]) and append the port number to the end of the address. For example, [xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]:PORT, where PORT represents the port number.

- **SNMP Version**: Version of the SNMP protocol to be used by Purity in communications with the specified managers. Valid values are v2c and v3 (default).
- **Community**: SNMPv2c only. SNMP manager community ID under which Purity is to communicate with the specified managers.
- **User**: User ID recognized by the specified SNMP managers that Purity is to use in communications with them.
- **Auth Protocol**: SNMPv3 only. Hash algorithm used to validate the authentication passphrase. Valid values are **MD5**, **SHA**, or **None**.
- **Auth Passphrase**: SNMPv3 only. Passphrase used by Purity to authenticate the array with the specified managers. Required if the **Auth Protocol** option is set to **MD5** or **SHA**.
- **Privacy Protocol**: SNMPv3 only. Encryption protocol for SNMP messages. Valid values are **AES**, **DES**, or **None**.
- **Privacy Passphrase**: SNMPv3 only. Passphrase used to encrypt SNMP messages. The passphrase must be between 8 and 63 non-spaced ASCII characters.
- **Notification**: Notification type that determines whether the recipient (remote host) acknowledges receipt of SNMP messages. Valid options are **trap** and **inform**. An SNMP trap is an unacknowledged (asynchronous) SNMP message, meaning the recipient does not acknowledge receipt of the message. An SNMP inform request is an acknowledged trap. If not specified, the notification type defaults to trap.

4 Click **Save**.

Editing an SNMP Manager Object

To edit an SNMP manager object, perform the following:

1 Select **Settings** > **System**.

2 In **SNMP** panel, locate the SNMP manager you wish to edit and click **More Options** > **Edit** at the end of its row. The **Edit SNMP Manager** pop-up window appears.

3 Modify the following fields:

- **Host**: DNS hostname or IP address of a computer that hosts an SNMP manager to which Purity is to send messages when it generates alerts.
- **SNMP Version**: Version of the SNMP protocol to be used by Purity in communications with the specified managers. Valid values are v2c and v3 (default) .
- **Community**: SNMPv2c only. SNMP manager community ID under which Purity is to communicate with the specified managers.
- **User**: User ID recognized by the specified SNMP managers that Purity is to use in communications with them.
- **Auth Protocol**: SNMPv3 only. Hash algorithm used to validate the authentication passphrase. Valid values are **MD5**, **SHA**, or **None**.
- **Auth Passphrase**: SNMPv3 only. Passphrase used by Purity to authenticate the array with the specified managers. Required if the **Auth Protocol** option is set to **MD5** or **SHA**.
- **Privacy Protocol**: SNMPv3 only. Encryption protocol for SNMP messages. Valid values are **AES**, **DES**, or **None**.
- **Privacy Passphrase**: SNMPv3 only. Passphrase used to encrypt SNMP messages. The passphrase must be between 8 and 63 non-spaced ASCII characters.
- **Notification**: Notification type that determines whether the recipient (remote host) acknowledges receipt of SNMP messages. Valid options are **trap** and **inform**. An SNMP trap is an unacknowledged (asynchronous) SNMP message, meaning the recipient does not acknowledge receipt of the message. An SNMP inform request is an acknowledged trap. If not specified, the notification type defaults to trap.

4 Click Save.

Renaming an SNMP Manager Object

To rename an SNMP manager object, perform the following:

1 Select **Settings** > **System**.

2 In **SNMP** panel, locate the SNMP manager you wish to rename and click **More Options** > **Rename** at the end of its row. The **Rename SNMP Manager** pop-up window appears.

3 Enter a new name in the **Name** field.

4 Click **Rename**.

Deleting an SNMP Manager Object

To delete an SNMP manager object, perform the following:

1. Select **Settings** > **System**.
2. In **SNMP** panel, locate the SNMP manager you wish to delete and click **More Options** > **Delete** at the end of its row. The **Delete SNMP Manager** pop-up window appears.
3. Click **Delete**.

Sending a Test Trap

To send a test trap to an SNMP manager, perform the following:

1. Select **Settings** > **System**.
2. In **SNMP** panel, locate the SNMP manager to which you wish to send a test trap and click **More Options** > **Send Test Trap** at the end of its row. The **Send Test Trap** pop-up window appears notifying you that a test trap has been sent to the SNMP manager.
3. Click **Close**.

Editing an SNMP Agent

To edit an SNMP agent, perform the following:

1. Select **Settings** > **System**.
2. From the header of the **SNMP** panel, click **More Options** > **Edit SNMP Agent**. The **Edit SNMP Agent** pop-up window appears.

3 Modify the following fields:

- **SNMP Version**: Version of the SNMP protocol to be used by Purity in communications with the specified managers. Valid values are v2c (default) and v3.
- **Community**: SNMPv2c only. SNMP manager community ID under which Purity is to communicate with the specified managers.
- **User**: User ID recognized by the specified SNMP managers that Purity is to use in communications with them.
- **Auth Protocol**: SNMPv3 only. Hash algorithm used to validate the authentication passphrase. Valid values are **MD5**, **SHA**, or **None**.
- **Auth Passphrase**: SNMPv3 only. Passphrase used by Purity to authenticate the array with the specified managers. Required if the **Auth Protocol** option is set to **MD5** or **SHA**.
- **Privacy Protocol**: SNMPv3 only. Encryption protocol for SNMP messages. Valid values are **AES**, **DES**, or **None**.
- **Privacy Passphrase**: SNMPv3 only. Passphrase used to encrypt SNMP messages. The passphrase must be between 8 and 63 non-spaced ASCII characters.

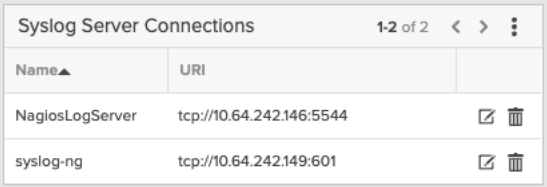
4 Click **Save**.

Syslog Server

The Syslog Server feature enables you to forward syslog messages to remote syslog servers.

Up to three remote syslog servers can be connected. All connected syslog servers can be viewed from the **Syslog Server Connections** pane on the **Settings** > **System** page of the GUI.

Figure 9.5 Syslog Server Connections



The Purity//FB syslog logging facility generates messages of major events within the FlashBlade and forwards the messages to remote servers. Purity//FB generates syslog messages for three types of events:

- Alerts (`purity.alert`)
- Audit (`purity.audit`)
- Tests (`purity.test`)

Alerts

Purity//FB generates alerts when there is a change to the array or to one of the Purity//FB hardware or software components. There are three alert severity levels:

- **INFO**: Informational messages that are generated due to a change in state. INFO messages can be used for reporting and analysis purposes. No action is required.
- **WARNING**: Important messages that warn of an impending error if action is not taken.
- **CRITICAL**: Urgent messages that require immediate attention.

Syslog alerts are broken down into the following format:

```
<Event Timestamp> <FM Host Name> purity.alert: <Alert Severity> <Alert State>
 <Alert Details>
```

In the following example, Purity//FB generated a Warning alert due to an unhealthy blade:

```
May 15 13:38:08 irp125a-v08g08-sim-ch1-fm1 purity.alert: WARNING Open - irp125a-
v08g08-sim: Blade CH1.FB1 is unhealthy [1000] #012A blade is unhealthy. No
 data has been lost, but performance may be impacted and additional wear may
 be occurring. There will be daily notifications for 7 days.#012#012Severity:
 Warning#012State: Open#012First Seen UTC Time: 2020 May 15 13:29:34 UTC#012First
 Seen Array Time: 2020 May 15 06:29:34 PDT#012Array Name: irp125a-v08g08-
sim#012Array ID: 00000000-0000-4000-8000-000000000000#012Purity//FB Version:
 3.1.99#012#012Suggested Action: This alert should automatically generate a
 support case. Please review your support case status by logging into Pure1
 at https://pure1.purestorage.com/. #012Knowledge Base Article: https://
support.purestorage.com/?cid=Alert_1000#012
```

Alerts are also sent via the phone home facility to the Pure Storage Support team. If configured, alerts can also be sent to designated email recipients and SNMP trap managers. Alerts can also be viewed from the GUI **Health** > **Alerts** page and from the **`purealert list`** CLI command.

Audit

An audit trail represents a chronological history of the GUI, CLI, or REST API operations that a user has performed to modify the configuration of the array. Each message within an audit trail includes the name of the Purity//FB user who performed the operation and the Purity//FB operation that was performed.

Syslog audit trail messages are broken down into the following format:

```
<Event Timestamp> <FM Host Name> purity.audit: <GUID> <ID> <Timestamp> <User>
 <Location> <User Interface> <Command Details>
```

In the following example, **pureuser** performed the **`purelog`** CLI command:

```
May 15 13:43:24 irp125a-v08g08-sim-ch1-fm1 purity.audit: - Adding audit entry.
 GUID=b1936eff-c4da-4582-89ad-9f5ab2d57777, ID=2, Time=2020-05-15 13:43:24.584225,
```

```
User=pureuser, Location=10.255.8.1, UI=CLI, Command=purelog, Subcommand=delete,
Arguments=test_syslog_alerts_remote
```

The audit trail can also be viewed from the GUI **Settings** > **User** page and from the the **`pureaudit list`** CLI command.

Tests

Test messages are generated by users to verify that the array can send messages to remote syslog servers. The message does not indicate whether or not the test message successfully reached the recipients; you must manually check the remote syslog server to confirm that the message arrived at the destination.

Syslog test messages are broken down into the following format:

```
<Event Timestamp> <FM Host Name> purity.test: <Test Message Details>
```

In the following example, a test was perfomed to determine if the array could send messages to a remote syslog server.

```
May 15 13:39:03 irp552-c01-ch1-fm2 purity.test: - This is a test message generated
 by Pure Storage FlashBlade. UTC Time: 2020 May 15 13:39:03 from FM Name: fm2
```

Creating a Syslog Server Connection

To create a remote syslog server connection, perform the following:

1. Select **Settings** > **System**.
2. In the **Syslog Server Connections** pane click the **More Options** button and select **Create Connection**. The **Create Syslog Server Connection** pop-up window appears.
3. In the **Name** field, enter the name for the remote syslog server connection.
4. In the **URI** field, enter the URI of the remote syslog server.

   Specify the URI in the format **PROTOCOL://HOSTNAME:PORT**, where

   - **PROTOCOL** is TCP, TLS, or UDP.
     Note that when establishing TLS communication, the syslog server must be configured with either a CA certificate or CA certificate group (for multiple remote syslog servers). See *Selecting*

CA Certificates for Syslog Server Connections. Additionally, your firewall must be set to allow in-bound and out-bound traffic.

- **HOSTNAME** is the syslog server's hostname or IP address. If specifying an IP address, for IPv4, specify the IP address in the form ddd.ddd.ddd.ddd, where ddd is a number ranging from 0 to 255 representing a group of 8 bits.

  For IPv6, specify the IP address in the form [xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx], where xxxx is a hexadecimal number representing a group of 16 bits. Enclose the entire address in square brackets ([]). Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (::).

- **PORT** is the port at which the server is listening. Append the port number after the end of the entire address. This is optional if the standard port number 514 for UDP or TCP, or the standard port number 6514 for TLS, is used.

5 Click **Create**.

The **Syslog Server Connections** pane is updated with the new syslog server

When establishing TLS communication, each remote syslog servers' certificate must have been signed by the configured CA certifcate, or by one of the CA certificates in the configured CA certificate group. See *Selecting CA Certificates for Syslog Server Connections*. Additionally, your firewall must be set to allow in-bound and out-bound traffic.

Selecting CA Certificates for Syslog Server Connections

For TLS connections, you must select a CA certificate for the remote syslog server.

If you are setting up TLS connections for mutliple remote syslog servers, select a CA certificate group.

1 Select **Settings** > **System**.

2 In the **Syslog Server Connections** pane click the **More Options** button and select **Select Certificate**. The **Select CA Certificate** pop-up window appears.

3 Select either a CA certificate or CA certifcate group.

As a best practice, it is recommended that you create a separate CA certificate group for syslog certificates.

4 Click **Save**.

Testing Syslog Server Connections

To test connected remote syslog servers, perform the following:

1 Select **Settings** > **System**.

2. In the **Syslog Server Connections** pane click the **More Options** button and select **Test**. The **Test Syslog Server Connection** pop-up window appears displaying the results of the syslog server connection test.

   A test message is sent to the remote syslog server. You must manually check the remote syslog server to confirm that the message at the destination.

   If the test message is not received, possible issues can include:

   - URI typos
   - Issues with the remote syslog server's configuration
   - Issues with firewall settings
   - Issues IPv6 connnectivity (if the FlashBlade is conrigured with IPv6 support)
   - Invalid certificate configurations

3. Click **Close** after you have reviewed the test results.

Editing a Syslog Server Connection

To edit a connected remote syslog server, perform the following:

1. Select **Settings** > **System**.
2. In the **Syslog Server Connections** pane click the **Edit** button to the right of the remote syslog server you wish to edit. The **Edit Syslog Server Connection** pop-up window appears.
3. In the **URI** field, enter the new URI of the remote syslog server.

   Specify the URI in the format **PROTOCOL://HOSTNAME:PORT**, where

   - **PROTOCOL** is TCP, TLS, or UDP.
     Note that when establishing TLS communication, the syslog server must be configured with either a CA certificate or CA certificate group (for multiple remote syslog servers). See *Selecting CA Certificates for Syslog Server Connections*. Additionally, your firewall must be set to allow in-bound and out-bound traffic.
   - **HOSTNAME** is the syslog server's hostname or IP address. If specifying an IP address, for IPv4, specify the IP address in the form ddd.ddd.ddd.ddd, where ddd is a number ranging from 0 to 255 representing a group of 8 bits.
     For IPv6, specify the IP address in the form [xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx], where xxxx is a hexadecimal number representing a group of 16 bits. Enclose the entire address in square

brackets ([]). Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (::).

- **PORT** is the port at which the server is listening. Append the port number after the end of the entire address. This is optional if the standard port number 514 for UDP or TCP, or the standard port number 6514 for TLS, is used.

4 Click **Save**.

The **Syslog Server Connections** pane is updated with your changes.

Deleting a Syslog Server Connection

To delete a remote syslog server connection, perform the following:

1 Select **Settings** > **System**.

2 In the **Syslog Server Connections** pane click the **Delete** button to the right of the remote syslog server whose connection you wish to delete. The **Delete Syslog Server Connection** pop-up window appears.

3 Click **Delete**.

The **Syslog Server Connections** pane is updated with your changes.

Login Banner

To help identify the FlashBlade system being logged into, you can set a login banner. The login banner is managed in the **UI** panel. From here, you can create and edit the array's login banner.

Creating a Login Banner

Create a login banner to identify your FlashBlade array.

To create a login banner, perform the following:

1 Select **Settings** > **System**.

2 In the **UI** panel, click **Create** next to **Login Banner**. The **Edit Login Banner** pop-up window appears.

3 Enter the login banner text. The limit is 8,000 characters.

4 Click **Save**.

Editing a Login Banner

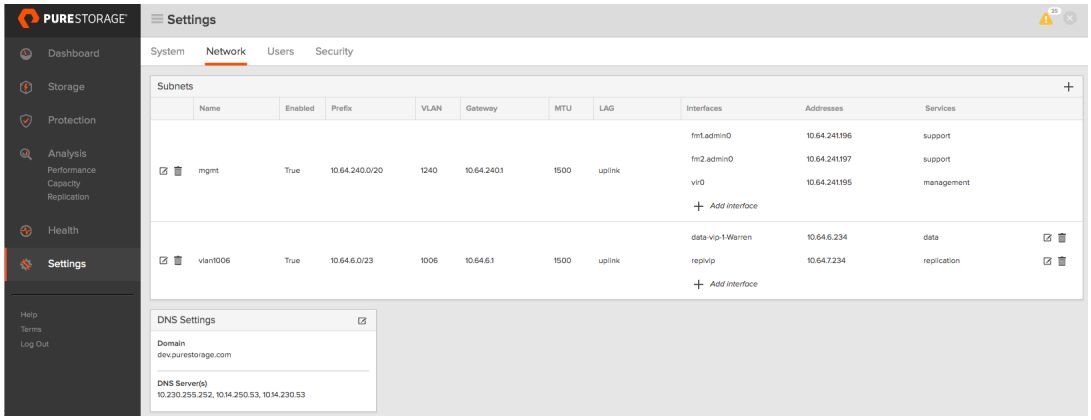
To edit an existing login banner, perform the following:

1. Select **Settings** > **System**.
2. In the **UI** panel, click **View** next to **Login Banner**. The **View Login Banner** pop-up window appears.
3. Click **Edit**.
4. Enter the new login banner text. The limit is 8,000 characters.
5. Click **Save**.

Network

The **Settings** > **Network** page displays and manages the subnets and data vips required for file system export and replication.

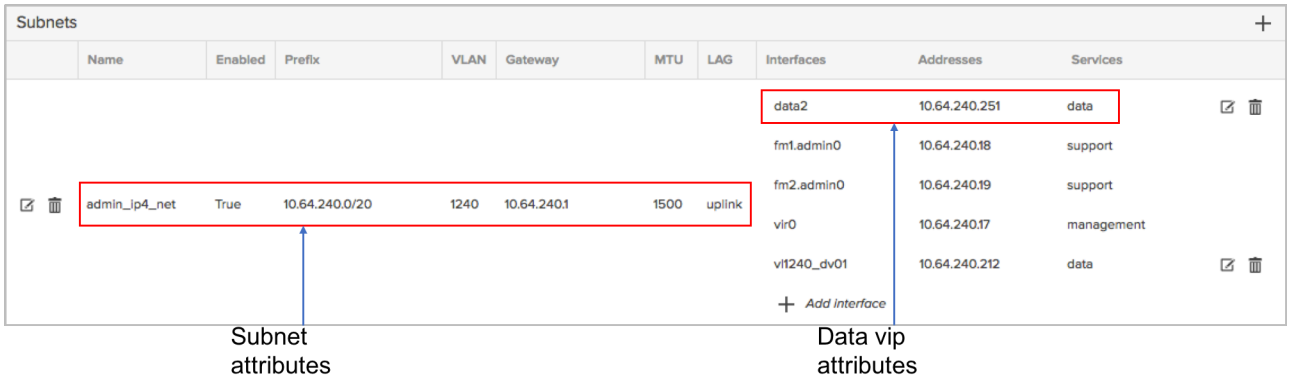
Figure 9.6 Settings - Network Page



Subnets

In the FlashBlade array, data virtual IP addresses (data vips) are organized into subnetworks (subnets). Exporting a file system requires a data vip which, in turn, must be attached to a subnet. Data vips that share the same network prefix and gateway are grouped into the same subnet.

In the following example, data vip **10.64.240.251** is attached to subnet **admin_ip4_net**.



Each subnet in the list includes the following information:

- **Name**: Subnet name.
- **Enabled**: Subnet status. When a subnet is enabled, the data vips within the subnet that are enabled are automatically available and ready to connect. Newly created subnets are automatically enabled.

  The **Subnets** panel displays the subnets on the array. Attached to each subnet are the data virtual IP addresses (data vips) that are used to export file systems.
- **Prefix**: IP address of the network prefix and prefix length.
- **VLAN**: VLAN ID number.
- **Gateway**: IP address of the gateway through which the data vip communicates with the network.
- **MTU**: Maximum transmission unit (MTU) for the data vips, in bytes.
- **LAG**: Link aggregation group (LAG) for the data vips.
- **Interfaces**: Data virtual IP address (data vip).
- **Addresses**: IP address associated with the data vip.
- **Services**: Service types for which the data vip is configured. The supported service for data vips is `data`.

Subnets can be created, modified, and deleted at any time. Create a new subnet if the desired one does not appear in the **Subnets** panel.

Creating a subnet requires a prefix and VLAN interface. Specify the IP address of the network prefix and prefix length in the form `ddd.ddd.ddd.ddd/dd` for IPv4, or `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx` for IPv6. Specify the VLAN ID to which the subnet is configured. Valid VLAN ID numbers are between 1 and 4094.

Optionally specify the gateway address in the form `ddd.ddd.ddd.ddd` for IPv4, or `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` for IPv6.

Optionally specify the maximum transmission unit (MTU), in bytes, of the data vip. If the MTU is not specified during subnet creation, the value defaults to `1500`.

When a data vip is attached to a subnet, it inherits the gateway, MTU and VLAN ID of the subnet. Likewise, the subnet inherits the `data` service type from the data vip.

Delete a subnet if it is no longer needed. Note that exporting a file system and file system replication require at least one valid data vip, which must be attached to a subnet. Therefore, there must always be at least one subnet (for service type `data` or `replication`) on the array.

Replication requires creating a replication vip. Replication vips can be in the same or different VLANs/subnets as data vips, and hence using different physical ports if required. The FlashBlade MTU setting applies to a subnet (VLAN). The MTU configured for replication VLAN must be the minimum MTU along the link between the source array and the target array.

Creating a Subnet

Data vips that share the same network prefix and gateway are grouped into the same subnet. Exporting a file system requires a data vip, which in turn, must be attached to a subnet. Create a subnet if none of the existing ones meet your requirements.

Replication vips can be in the same or different VLANs/subnets as data vips, and hence using different physical ports if required. The FlashBlade MTU setting applies to a subnet (VLAN). The MTU configured for the replication VLAN must be the minimum MTU along the link between the source array and the target array. If you notice the arrays are connected, but replication shows little throughput, ensure that the MTU configured for the replication VLAN is the correct size.

It is recommended to create separate VLANs for data and replication so that data and replication can use different MTU settings.

1. Select **Settings** > **Network**.
2. In the **Subnets** list, click the Add (+) button in the Subnets title bar. The **Create Subnet** pop-up window appears.
3. In the **Name** field, type the name of the subnet.
4. In the **Prefix** field, type the IP address of the network prefix and prefix length in the form `ddd.ddd.ddd.ddd/dd` for IPv4, or `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx` for IPv6.
5. In the **VLAN** field, specify the VLAN ID to which the subnet is configured. Valid VLAN ID numbers are between 1 and 4094.
6. In the **Gateway** field, type the IP address of the gateway through which the data vip communicates with the network in the form `ddd.ddd.ddd.ddd` for IPv4, or `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` for IPv6
7. In the **MTU** field, specify the maximum transmission unit (MTU) of the data vip. If the MTU is not specified during subnet creation, the value defaults to `1500`.

8 Click **Create**.

Creating a Network Interface

Once you have created a subnet you can create and add network interfaces.

To creat a network interface, perform the following:

1 Select **Settings** > **Network**.

2 In the **Subnets** list, locate the subnet to which you wish to add a network interface and click the **Add interface** (+) button. The **Create Network Interface** pop-up window appears.

3 In the **Name** field, enter the interface name.

4 In the address field, enter the network address.

5 From the **Services** list, select if the interface will be used for `data` or `replication`.

6 The subnet defaults to the subnet in which you are creating the interface. Click **Create**.

The **Interfaces** column of the **Subnets** list is updated with the new interface.

Deleting a Network Interface

1 Select **Settings** > **Network**.

2 In the Subnets list, click the **Delete** icon next to the network interface you want to delete.

Deleting a Subnet

A subnet can only be deleted if it is empty.

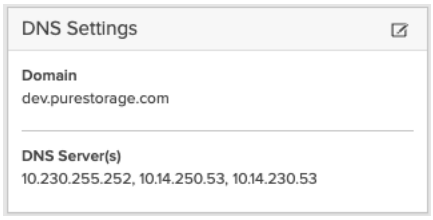
1 Select **Settings** > **Network**.

2 In the Subnets list, click the **Delete** icon next to the subnet you want to delete.

DNS Settings

The **DNS Settings** panel displays and manages the DNS attributes for an array 's administrative network.

Figure 9.7 DNS Settings Panel



Domain

The DNS domain represents the domain suffix to be appended by the array when performing DNS lookups. For example, `mydomain.com`.

DNS Server(s)

The DNS servers manages the DNS domains that are configured for the array. Each DNS domain can include up to three static DNS server IP addresses. Purity//FB queries the DNS servers in the order in which the IP addresses are listed in this option.

Editing DNS Settings

The DNS servers manage the DNS domains that are configured for the array.

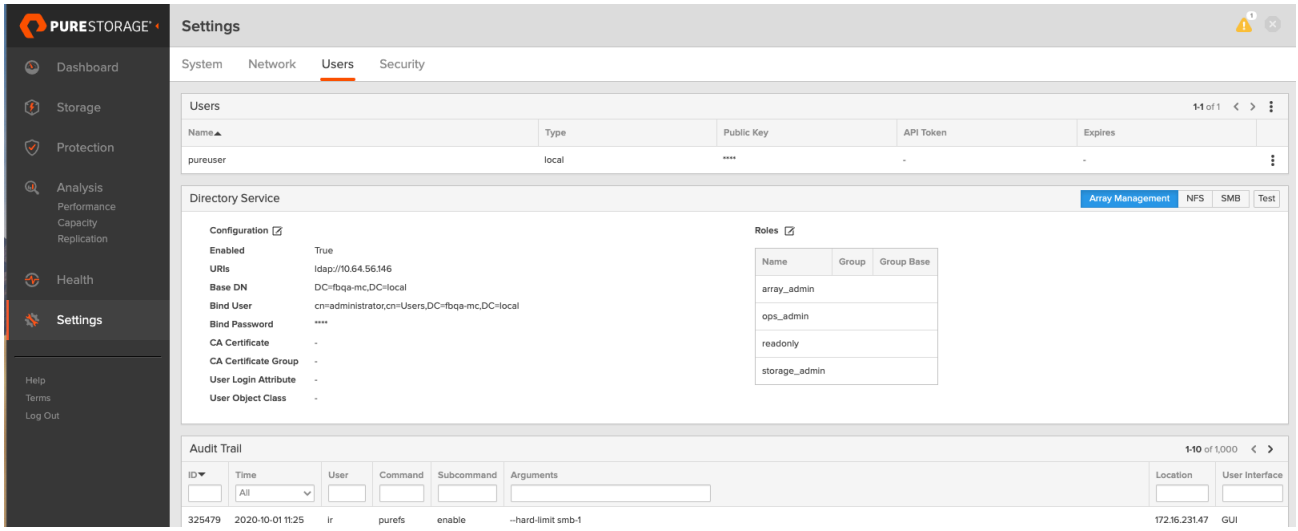
To edit DNS settings, perform the following:

1. Select **Settings** > **Network**.
2. In the **DNS Settings** panel, click the **Edit** icon. The **Edit DNS** pop-up window appears.
3. (Optional) In the **Domain** field, type the domain suffix to be appended by the array when doing DNS lookups. For example, `mydomain.com`.
4. (Optional) In the **DNS #** fields, specify up to three DNS server IP addresses for Purity//FB to use to resolve hostnames to IP addresses. Enter one IP address in each DNS # field. Purity//FB queries the DNS servers in the order that the IP addresses are listed.
5. Click **Save**.

Users

The **Settings** > **Users** page provides configuration and management of users and directory services. Additionally, an **Audit Trail** panel is provided, which lists the create, update, and delete commands issued on the array.

Figure 9.8 Settings-Users Page



Users

The **Users** panel manages local and remote FlashBlade array users.

Figure 9.9 Users - Users Panel

The panel displays the following:

- **Name**: The user name.
- **Type**: Displays `local` for users local to the array (pureuser), or `remote` for any other LDAP users.
- **Public Key**: The user's public key displayed as *****.
- **API Token**: Active tokens for API access.
- **Expires**: Expiration time of the API token.

In addition to displaying user attribute details, you can also create and manage user API tokens and user array authentication in the **Users** panel.

API Tokens

The REST API requires permission to run commands. To access the server to run REST API commands, an authentication API token must be generated for the user. Each API token is unique and has an optional expiration setting. When a token expires, it can no longer be used and a new token must be

generated to continue running REST API commands. It is a best practice to set an expiration on a token for the duration of time it will be used.

Array Authentication

Note: Setting other users' public keys is available to administrators and users with admin permissions only. The public key should be in the OpenSSH format, which begins with `ssh-rsa` or `ssh-ed25519`. A maximum of 100 users may have a public key set.

Administrators and users with administrator privileges can set passwords for array users. Additionally, public key authentication can be configured allowing users to SSH to the array.

Creating an API Token

API tokens allow users to run REST API commands. To access the server to run REST API commands, you must generate an authentication API token for the user. As a best practice set an expiration on a token for the duration of time it will be used.

To create an API token for a user, perform the following:

1. Select **Settings** > **Users**.
2. In the **Users** panel, click the **More Options** button and select **Create API Token**. The **Create API Token** pop-up window appears.
3. Enter user for whom the token will be created.
4. Enter the token's expiration duration in the format N[m/h/d/w}. For example, **15m**, **1h**, **2d**, **3w**. If not set, the API token will not expire.
5. Click **Create**.

The API Token pop-up window appears displaying the token and its creation and expiration dates. Click **Copy** to copy the token to your clipboard for later use.

Displaying an API Token

If you previously created an API token you can later display and copy the token for use with the REST API.

To display a user's API token for a user, perform the following:

1. Select **Settings** > **Users**.
2. In the **Users** panel, locate the user whose API token you wish to view. Click the **More Options** button and select **Show API Token**. The **API Token** pop-up window appears.
3. Click **Copy** to copy the token if needed.

4 Click **Close.**.

Removing an API Token

To remove an API token from a user, perform the following:

1 Select **Settings** > **Users**.

2 In the **Users** panel, locate the user whose API token you wish to remove. Click the **More Options** button and select **Remove API Token**. The **Remove API Token** pop-up window appears.

3 Click **Remove**.

Updating a User's Public Key

A public key can be added or updated for user authentication in order to allow users to SSH to the FlashBlade.

Note: Setting other users' public keys is available to administrators and users with admin permissions only. The public key should be in the OpenSSH format, which begins with `ssh-rsa` or `ssh-ed25519`. A maximum of 100 users may have a public key set.

To add or update a user's public key, perform the following:

1 Select **Settings** > **Users**.

2 In the **Users** panel, click the **More Options** button and select **Update Public Key**. The **Update Public Key** pop-up window appears.

3 Enter the user for whom you are adding a public key.

4 Enter the public key into the **Public Key** field.

5 Click **Update**.

Editing Users

When editing a user, you can set and change the user's password and manage the user's public key configuration.

Note: Setting other users' public keys is available to administrators and users with admin permissions only. The public key should be in the OpenSSH format, which begins with `ssh-rsa` or `ssh-ed25519`. A maximum of 100 users may have a public key set.

To edit a user, perform the following:

1 Select **Settings** > **Users**.

2 In the **Users** panel, locate the user you wish edit and click the **More Options** button and select **Edit User**. The **Edit User** pop-up window appears.

3 (Optional) Enter and confirm a new password.

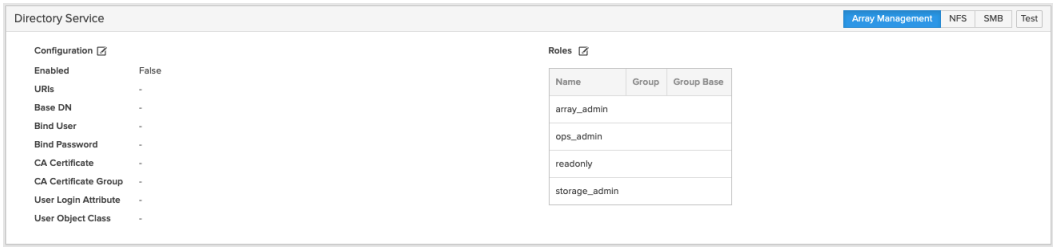
4 If a user's public key is set up you can optionally **Retain**, **Overwrite**, or **Remove** the public key. If you select **Overwrite**, enter the new public key in the displayed field.

5 Click **Save**.

Directory Service

The **Directory Service** panel manages the integration of FlashBlades with an existing directory service.

Figure 9.10 Users - Directory Service Panel



When the **Directory Service** panel's user roles and directory service protocols are configured and enabled, the FlashBlade leverages a directory service to perform user account and permission level searches.

The FlashBlade is delivered with a single local user, named `pureuser`, with array -wide (Array Admin) permissions. Users can be added to the array through or Lightweight Directory Access Protocol (LDAP) by integrating the array with a directory service such as Active Directory or Open LDAP.

To integrate the FlashBlade array with a directory service:

1 Configure the FlashBlade directory service.

2 Test the directory service configuration.

3 Enable the directory service.

When configuring the directory service for array management, SMB, or NFS protocol integration with LDAP, specify the URIs (optional for SMB), base DN, and bind username and password of the directory service. Note that for file sharing over SMB, the base DN of the directory service is used to represent the domain that is joined, and the URIs can be configured for SMB to control what servers are communicated with when joining the domain. If selecting the NIS directory service for NFS, clear any existing LDAP configurations before specifying the NIS servers and NIS domain.

⚠ **Important note!** Anonymous binding is supported for NFS and Management. If configuring anonymous binding for NFS or Management, neither the bind user nor the bind password should be specified.

A "bind" account must be configured to allow the array to read the directory. The bind account should be tied to a service account (rather than an individual), and the account password should never expire or be required to change periodically. The bind account must have read privileges for users and groups and the privileges should be configured at the Organizational Unit.

For Active Directory, users with disabled accounts are not able to access the file systems.

After the directory service has been configured, test the configuration to verify that the URIs can be resolved and the FlashBlade array can bind and query the tree using the bind user credentials.

Enable the directory service after it has passed the directory service test. The directory service can be disabled at any time. Disabling the directory service stops any users in the directory from accessing the file system. In the case of management, disabling directory service prevents users from logging into the array.

Roles

Directory services for array management can configure role-based access control (RBAC) for LDAP users by configuring groups in the directory that corresponds to the following permission groups (roles) on the array.

Role	Description
readonly	Read Only users have read-only privileges to run commands that convey the state of the array. Read Only users cannot alter the state of the array
storage_admin	Storage Admin users have all the privileges of Read Only users, plus the ability to run commands related to storage operations, such as administering file systems and snapshots, as well as object store accounts, users, and access keys. Storage Admin users cannot perform operations that deal with global and system configurations.
array_admin	Array Admin users can perform all FlashBlade operations.
ops_admin	Ops Admin users have all the privileges of Read Only users, plus the ability to run commands related to support operations, such as managing remote assistance sessions, phonehome, and proxy settings. Ops Admin users cannot perform operations that deal with storage or non-support global and system configurations.

When an array management user connects to the FlashBlade, the array confirms the user's identity from the directory service. The response from the directory service includes the user's group, which Purity//FB maps to a role on the array, granting access accordingly.

If users are assigned to more than one role, an alert is issued. For security purposes, a user belonging to more than one FlashBlade role will inherit the permissions of the most restrictive role. For example, a user who has been assigned both the Storage Admin and Read Only roles will inherit the privileges of the more restrictive Read Only role. Note that the user will continue to have privileges from the original role until the directory service role cache is cleared (cache is automatically cleared every 8 hours). To make new permissions take immediate effect, run the **`pureadmin refresh`** command.

One exception to this rule is users assigned the Ops Admin role—users assigned the Ops Admin role and any other role are denied the ability to log into the FlashBlade. If there is an error with the directory service configuration on the FlashBlade array, contact a user with Array Admin privileges to resolve the issue. If the LDAP server is configured incorrectly, contact your LDAP server administrator. The alert issued in the case where a user is denied access due to being assigned the Ops Admin and another role, provides additional information to help resolve the issue.

As a best practice, after changing a user's role or management directory service configuration settings, use the **`pureadmin refresh`** command to refresh the user's role or clear directory service role cache to ensure all user privileges are up to date.

For directory service-enabled accounts, user passwords to the FlashBlade are managed through the directory service while API tokens are configured through Purity//FB. The password can be a maximum of 100 characters in length and include any character that is entered from a US keyboard.

In the FlashBlade GUI, buttons and menu items for unauthorized actions are disabled per the user's configured role.

Protocols

To export a file system over NFS where a user might belong to more than 16 groups, the FlashBlade array must be integrated with a directory service such as Active Directory, OpenLDAP, or Network Information Service (NIS). If selecting the NIS directory service for NFS, clear any existing LDAP configurations before specifying the NIS servers and NIS domain. Note that NIS is only supported for NFS. To export a file system over SMB, the FlashBlade array must be integrated with an Active Directory service. The SMB account must have read privileges for users and groups, including write permissions to add computer objects in Active Directory.

Configuring the directory service for SMB or NFS is a one-time process and is only required if you are integrating the FlashBlade array with a directory service for protocol support.

Before you configure the directory service, verify that the DNS settings can be used to resolve the directory server domain and server. DNS settings are managed through the **Settings** > **Network** page.

For file sharing over SMB, also verify that the FlashBlade array can access the directory service through a management interface. Interfaces are managed through the **Settings** > **Network** page.

Configuring Directory Services

Configuring the directory service for protocol support is a one-time process and is only required if you are integrating the FlashBlade array with a directory service.

To configure the directory service for protocol support:

1. Verify that the directory server is accessible through a management interface. For file sharing over SMB, verify that the Active Directory server is accessible through a management interface.
2. Verify that the DNS settings can be used to resolve the directory server domain and server.
   a. Select **Settings** > **Users**.
   b. In the **Directory Service** panel, verify that the DNS settings can be used to resolve the directory server domain and server.
3. Configure the directory service.
   a. Select **Settings** > **Users**.
   b. In the **Directory Service** panel, select **Array Management**, **NFS**, or **SMB**.
   c. Click the **Edit** icon to the right of **Configuration**. The **Edit Directory Service Configuration** pop-up window appears.
   d. In the **URIs** field, type the comma-separated list of up to 30 URIs of the directory servers.

      For file sharing over SMB, this field is optional. If not specified, the base DN of the SMB directory service is used to try to discover domain controllers within an Active Directory domain. If URIs are configured for SMB, they will be treated as the preferences for the servers with which FlashBlade will communicate when joining the domain.

      Each URI must include the scheme `ldap://` or `ldaps://` (for LDAP over SSL), a hostname, and a domain name or IP address. For example, `ldap://ad.company.com` configures the directory service with the hostname "ad" in the domain "company.com" while specifying the unencrypted LDAP protocol.

      We highly recommend either configuring StartTLS by enabling a certificate or certificate group, or configuring URIs by using the `ldaps://` scheme to use LDAP over SSL, unless there is no need for secure communication in your environment.

      If specifying a domain name, it should be resolvable by the configured DNS servers.

      If specifying an IP address, for IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from `0` to `255` representing a group of 8 bits.

      For IPv6, specify the IP address in the form `[xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]`, where `xxxx` is a hexadecimal number

representing a group of 16 bits. Enclose the entire address in square brackets (`[]`) . Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`).

If the base DN is not configured and a URI is provided, the base DN will automatically default to the domain components of the URIs.

Optionally specify a port. Append the port number after the end of the entire address. Default ports are 389 for ldap, and 636 for ldaps. Non-standard ports can be specified in the URI if they are in use.

e In the **Base DN** field, type the base distinguished name (DN) of the directory service. The Base DN is built from the domain and should consist of only domain components (DCs). For example, for `ldap://ad.storage.company.com`, the Base DN would be: `DC=storage,DC=company,DC=com`.

f In the **Bind User** field, type the username used to bind to and query the directory. For OpenLDAP and Active Directory servers, you can use the full DN of the user account that is used to perform lookups. For example, `CN=John,OU=Users,DC=example,DC=com`. For Active Directory servers, you may instead choose to enter the username - often referred to as the sAMAccountName or user login name - of the account that is used to perform directory lookups. The username cannot contain the characters `" [] : ; | = + * ? < > / \`, and cannot exceed 104 characters in length.

g In the **Bind Password** field, type the password for the bind user account.

h If configuring SMB, in the **Join OU (SMB)** field, enter the relative DN of the organizational unit (OU) within your domain where the system machine account should be created when joining the domain for SMB. For example, `OU=Arrays,OU=Storage,OU=ServiceMachines`.

i Optionally, select a CA certificate or certificate group to verify the authenticity of configured LDAP servers to establish a TLS (Transport Layer Security) communication for directory services. CA certificates are supported for **Array Management** or **NFS** only.

j If configuring Array Management, the **User Login Attribute** defaults to the sAMAccountName for Active Directory, or uid for other server types.

k If configuring Array Management, the **User Object Class** defaults to User for AD, posixAccount or shadowAccount for OpenLDAP depending on the server's schema, posixAccount for posix-compliant servers, or person for all other server types.

⚠ **Important note!** Anonymous binding is supported for NFS and Management. If configuring anonymous binding for NFS or Management, neither the bind user nor the bind password should be specified.

4 Click **Save**.

5 If you selected and configured the **Array Management** service, click the **Edit** icon next to **Roles** to configure the role(s) you wish to assign the service. The **Edit Directory Service Roles** pop-up window appears. Enter the following for each role you wish to assign:

   a In the **Group** field, enter the common name (CN) of the directory group. The name should be the Common Name of the group without the "CN=" specifier. If the configured groups are not in the same OU, also specify the OU. For example, "pureusers,OU=PureStorage", where pureusers is the common name of the directory service group.

   b In the **Group Base** field, enter the organizational unit (OU) to the configured groups in the directory tree. The Group Base consists of OUs that, when combined with the base DN attribute and the configured group CNs, complete the full distinguished name of each group. The group base should specify "OU=" for each OU and multiple OUs should be separated by commas. The order of OUs should get larger in scope from left to right. In the following example, SANManagers contains the sub-organizational unit PureGroups: "OU=PureGroups,OU=SANManagers"

   c Click **Test** to verify the configuration information. If the test fails, please update the configuration information and retest.

   d If the test passes, you may change the **Enabled** toggle to enable (blue) the directory service. Alternatively, you may save the configuration and enable the directory service at a later time.

   e Click **Save**.

Note: Directory service testing is applicable to the selected directory service (**Array Management**, **NFS**, or **SMB**) .

After you configure the directory service settings, test and enable the directory service.

Configuring Directory Services for NFS with NIS

Selecting NIS (network information system) as the directory service protocol for NFS.

1 Verify that the directory server is accessible through a management interface.

2 Verify that the DNS settings can be used to resolve the directory server domain and server.

   a Select **Settings** > **Users**.

   b In the **Directory Service** panel, verify that the DNS settings can be used to resolve the directory server domain and server.

3 Select **NFS** on the right of the Directory Service panel.

4 Click the **Edit** icon next to **Configuration**. The **Edit Directory Service Configuration** pop-up window appears.

5 Select the **NIS** service.

6 In the **NIS Servers** field, type the comma-separated list of server host names or IP addresses. For example, `myserver.example.com,188.121.4.56`.

7 In the **NIS Domain** field, type the domain name. For example, `yp-example-domain`.

8 Click **Test** to verify the configuration information. If the test fails, please update the configuration information and retest.

9 If the test passes, you may change the **Enabled** toggle to enable (blue) the directory service. Alternatively, you may save the configuration and enable the directory service at a later time.

10 Click **Save**.

Testing the Directory Service Settings

After you configure the directory service settings, test the directory service configuration to verify that the URI can be resolved and that the directory service can successfully bind and query the tree using the bind user credentials.

To test the directory service configuration:

1 Select **Settings** > **Users**.

2 In the **Directory Service** panel, select the directory service you wish to test.

3 Click **Test**. The **Test <Directory Service> Configuration** pop-up window appears, displaying the output of the test. During the directory service test, Purity//FB tests the directory service configuration to verify that the URI can be resolved and that the directory service can successfully bind and query the tree using the bind user credentials.

   Once the test passes, enable the directory service.

Enabling the Directory Service

Before enabling the directory service, test the configuration to make sure the URI can be resolved and that the directory service can successfully bind and query the tree using the bind user credentials.

To enable the directory service:

1 Select **Settings** > **Users**.

2 In the **Directory Service** panel, select the directory service you wish to enable and click the **Edit** icon.

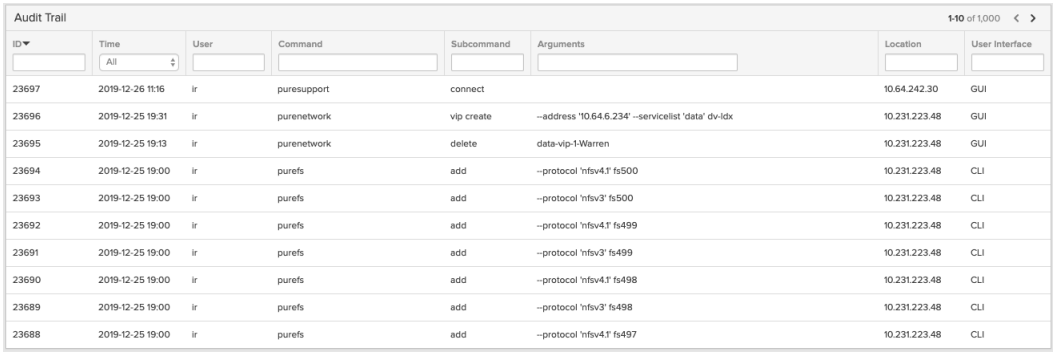
3 Set the **Enabled** toggle button to enable (blue) the directory service.

4 Click **Save**.

Viewing the Audit Trail

The **Audit Trail** panel provides a chronological history of modification operations (for example, commands that create, update, and delete) that were run on the array.

Figure 9.11 Users - Audit Trail Panel



To view the history of modification operations issued on the array, select **Settings** > **Users**.

The **Audit Trail** panel is located directly below the **Directory Service** panel and provides the following information:

- **ID**: The chronological number the command was used.
- **Time**: The date and time the command was run.
- **User**: The user who ran the command.
- **Command**: The name of the command.
- **Subcommand**: The subcommand used with the command.
- **Arguments**: Any arguments used with the command and their associated user-supplied input.
- **Location**: The IP address where the command was issued.
- **User Interface**: The interface from which the operation was run. For example, CLI, GUI, REST.

Security

LDAP over TLS (Transport Layer Security) can be optionally configured to encrypt communication between the FlashBlade array, acting as an LDAP client, and LDAP servers. In order to establish LDAP over TLS, the LDAP server and FlashBlade must have mutually supported versions of TLS. FlashBlade supports TLS versions 1.0., 1.1, and 1.2. The LDAP server must support at least one of these versions. The highest matching TLS version on the LDAP server and FlashBlade is used.

⚠ **Important note!** The LDAP over TLS configuration does not support NFS with NIS (Network Information Service).

LDAP over TLS requires that each LDAP server's CA certificate be imported to the FlashBlade where they can be used individually by the FlashBlade or grouped into CA certificate groups to verify the LDAP server's identity and establish communication over TLS. CA certificate groups are used to manage multiple trusted certificates issued to different servers as a directory of certificates. A CA certificate group can contain one or more CA certificates. A CA certificate can be used in multiple CA certificate groups. CA certificates can be added, removed, or deleted without disruption to the services using them as long as the certificates for any configured servers remain in the group. While a CA certificate group is in use, it cannot be deleted, nor can you remove or delete all CA certificates in that group.

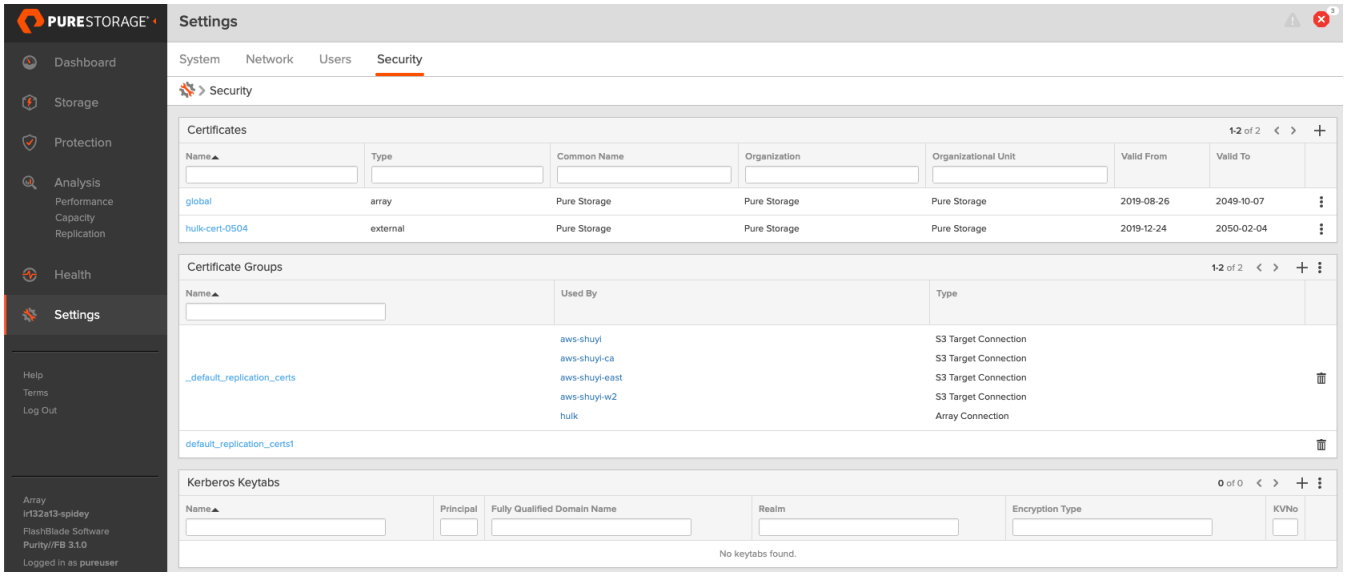
⚠ **Important note!** Adding multiple CA certificates in a certificate group which secure the same hostname can produce undefined behavior --there is no guarantee which certificate will be used.

Purity//FB supports the Kerberos network authentication protocol (krb5) for NFSv4.1. clients, in addition to auth_sys. With Kerberos, NFSv4.1 clients have to be authenticated by a Key Distribution Center (KDC) to get a session ticket. They have to present this session ticket to FlashBlade array in order to get access to the NFSv4.1 file system. Keytab files can be generated for a FlashBlade array by a KDC, and can then be imported and managed by the FlashBlade. Keytab files contain list of Kerberos principal names and encryption keys to automatically authenticate clients without requiring human interaction or access to password stored in a plain-text file.

The **Security** page allows you to manage certficates, certificate groups, and Kerberos keytabs via three panels:

- **Certifcates**
- **Certificate Groups**
- **Kerberos Keytabs**

Figure 9.12 Settings - Security Page



Importing a CA Certificate

CA certificates provide the security authentication between two identities. This allows the FlashBlade array to communicate securely with servers and clients.

1. Select **Settings** > **Security**.
2. In the **Certificates** panel, click the Add interface button (+) . The **Import CA Certificate** pop-up window appears for entering the certificate details.
3. In the **Name** field, type the name you want to assign to the certificate.
4. In the **Type** field, enter **array** or **external**. The type indicates the type of certificate to create. An **array** certificate type is used by clients to validate the array and requires a private key. For example, the FlashBlade array shows this type of certificate to an S3 client. An **external** certificate type is used when the FlashBlade array acts as a client to validate a server.
5. In the **CA Certificate** field, click **Choose File** and select the certificate file.
6. Click **Import**.

Viewing Certificate Details

To view a certificate's details, perform the following:

1. Select **Settings** > **Security**.

2 From the **Certifcates** panel, click the certificate for which you wish to view more details. **More Options** button in the same row as the certificate for which you wish to view more details.

The **Certficates Details** panel appears displaying the certficate's information. Additionally, details about who is using the certficate is displayed in the **Used By** panel.

Exporting a Certificate

To export a certificate, perform the following:

1 Select **Settings** > **Security**.

2 From the **Certifcates** panel, click the **More Options** button in the same row as the certificate you wish to export.

3 Click **Export Certificate**. The **Export Certificate** pop-up window appears.

4 Click **Download**.

Deleting a Certificate

To delete a certificate, perform the following:

1 Select **Settings** > **Security**.

2 From the **Certifcates** panel, click the **More Options** button in the same row as the certificate you wish to delete.

3 Click **Delete Certificate**. The **Delete Certificate** pop-up window appears.

4 Click **Delete**.

Creating a CA Certificate Group

A group can be created to house a collection of CA certificates. After you create a group, certificates can be added to that group.

1 Select **Settings > Security**.

2 In the **Certificate Groups** panel, click the Add interface button (**+**) . The **Create Certificate Group** pop-up window appears for entering the group name.

3 In the **Name** field, type the name you want to assign to the certificate group.

4 Click **Create**.

Deleting a CA Certificate Group

To delete a CA certificate group, perform the following:

1. Select **Settings** > **Security**.
2. From the **Certifcate Groups** panel, click the **Delete** button in the same row as the certificate group you wish to delete. The **Delete Certificate Group** pop-up window appears.
3. Click **Delete**.

Adding a CA Certificate to a Group

Once a certificate group is created, you may add certificates to that group.

1. Select **Settings > Security**.
2. In the **Certificate Groups** panel, click the group name you wish to add certificates. The selected group certificates page opens.
3. In the **Certificates** field of the certificates group, click the Add interface button (**+**) . The **Add Certificates** pop-up window appears.
4. Select the certificates you wish to add to the group.
5. Click **Add**.

Removing a CA Certificate from a Group

Certificates can be removed from a certificate group if the certificate has expired or is no longer used.

1. Select **Settings > Security**.
2. In the **Certificate Groups** panel, click the group name you wish to remove certificates. The selected group page opens.
3. In the **Certificates** field of the certificates group, click the **More Options** button from the **Certificates** panel.
4. Click **Remove**. The **Remove Certificates** pop-up window appears.
5. Select the certificates you wish to remove from the group.
6. Click **Remove**.

Importing a Keytab File

To import a keytab file, perform the following:

1. Select **Settings** > **Security**.
2. From the **Kerberos Keytabs** panel, click the Add (+) button. The **Import Keytab File** pop-up window appears.
3. Enter a prefix in the **Name Prefix** field. Specifying a file entry prefix is required because a single keytab file can contain multiple keytab entries in multiple slots.
4. Click **Choose File** in the **Keytab File** field. Navigate to, and select the keytab file you wish to import.
5. Click **Import**.

Exporting a Keytab File

To export a keytab file, perform the following:

1. Select **Settings** > **Security**.
2. From the **Kerberos Keytabs** panel, click the **More Options** button in the same row as the certificate you wish to export.
3. Click **Export**. The **Export Keytabs** pop-up window appears.
4. Select the keytab entries you wish to export to a file from the **Existing Keytabs** pane.
5. Click **Export**.

Deleting a Keytab Entry

To delete a keytab entrh, perform the following:

1. Select **Settings** > **Security**.
2. From the **Kerberos Keytabs** panel, click the **Delete** button in the same row as the keytab you wish to delete. The **Delete Keytabs** pop-up window appears.
3. Select the keytab entries you wish to delete from the **Existing Keytabs** pane.
4. Click **Delete**.

Chapter 10
Using the Purity//FB CLI to Administer a FlashBlade Array

Purity//FB is the operating environment that queries and manages the FlashBlade hardware, networking, and storage components. The Purity//FB software is distributed with the FlashBlade array.

Purity//FB provides two ways to administer the FlashBlade: through the browser-based graphical user interface (Purity//FB GUI), and through the command-driven interface (Purity//FB CLI).

The Purity//FB command line interface (CLI) is a non-graphical, command-driven interface used to query and administer the FlashBlade.

Visibilty of FB CLI commands is dependent on user role.

This chapter covers general Purity//FB CLI concepts and conventions.

CLI Command Syntax

The Purity//FB CLI is comprised of built-in commands specific to the Purity//FB operating environment.

Purity//FB CLI commands have the general form:

```
command subcommand --options OBJECT-LIST
```

The parts of a command are:

COMMAND

> Type of FlashBlade object to be acted upon, prefixed by "pure". For example, the **`purefs`** command acts on Purity//FB file systems. Run the **`purehelp`** command to see a list of Purity//FB CLI commands.

SUBCOMMAND

> Action to be performed on the specified object. Most CLI subcommands are common to some or all object types. For example, **`puresubnet list`** lists all subnets on the array.

OPTIONS

> Options that specify attribute values or modify the action performed by the subcommand.

For example, in the following command, the **`--size`** option sets the provisioned size of file system `MyFiles` to 100 gigabytes.

```
purefs setattr --size 100G MyFiles
```

OBJECT-LIST

Object or list of objects upon which the command is to be operated. If a subcommand changes the object state, then at least one object must be specified. Examples of subcommands that change the object state include **`create`**, **`delete`**, and **`setattr`**. For example, **`purefs create MyFiles`** creates file system `MyFiles`. In the command synopsis, OBJECT specifications that are not enclosed in square brackets (for example, ***NAME*** in `purenetwork` commands) represent ones that are required.

Passive subcommands, such as **`list`**, which do not change object state, do not require object specification. Leaving out the object is equivalent to specifying all objects of the type. For example, **`purefs list`** with no file systems specified displays information about all file systems in an array. In the command synopsis, OBJECT specifications enclosed in square brackets (for example, ***[NAME]***) represent ones that are optional.

Most subcommands act on a single object. For example, in the following command, the **`create`** subcommand can only be run on a single file system to change the attributes of that file system.

```
purefs setattr --size 10T MyFiles
```

Certain subcommands can operate on multiple objects. For example, the following command creates two alert watchers.

```
purealert create watcher wendywatcher@example.com,walterwatcher@example.com
```

In the command synopsis, OBJECT specifications that take in a list of objects are appended with ellipses (for example, ***ADDRESS...***) .

The following list describes the conventions used in CLI documentation:

- Text in fixed-width (Courier) font must be entered exactly as shown. For example, **`puresubnet list`**.
- Text not enclosed in brackets represents mandatory text.
- Text inside square brackets (“[]”) represents optional items. Do not type the brackets.
- Text inside curly braces (“{ }”) represents text, where one (and only one) item must be specified. Do not type the braces.
- The vertical bar (|) separates mutually exclusive items.
- Uppercase italic text represents entered text whose value is based on the nature of the subcommand or option. For example, **`--size SIZE`** , where SIZE represents the value to be entered, such as `10T`.

CLI Output

Various Purity//FB CLI subcommands, such as `list`, generate list outputs. With the ability to create and manage hundreds of file systems, list outputs can become very long. The Purity//FB CLI includes options to control the way a list output is displayed and formatted. The CLI also includes sort, filter, and limit options to control the results you see and the order in which you see them; the following descriptions about sort, filter, and limit are applicable to all commands where these options are available.

Interactive Paging

Pagination divides a large output into discrete pages. Pagination is disabled by default and is only in effect if the `--page` option is specified and the number of lines in the list output exceeds the size of the window.

To interactively move through a paginated list output:

- Press the `[Right Arrow]` key or space bar to move to the next page.
- Press the `[Left Arrow]` key to move to the previous page.

To quit interactive paging and exit the list view, press `q` .

With pagination, each page of the CLI list output begins with the column titles. To suppress the column titles, run the command with the `--notitle` option.

Using Wildcards

The asterisk (*) symbol denotes a wildcard character used in list operations to represent zero or more characters in an object name. The object name can include multiple asterisks.

When running the `purefs list` command, include the asterisk in the name argument to expand the list results. For example, the `purefs list *cat*` command displays a list of all hosts that contain "cat", such as `cat`, `catnap`, `happycats`, and `lolcat`. If the asterisks were not included, only the host named `cat` would be returned. As another example, the `purefs list *fs*` command displays a list of all file systems that contain "fs", including ones that begin or end with "fs".

```
$ purefs list *fs*
Name     Size   Used  Created                  Protocols
Myfs     100G   0.00  2016-04-11 10:18:07 PDT  http
MyFs-01  100G   0.00  2016-04-11 10:18:07 PDT  http
fs       1G     0.00  2016-04-11 11:19:19 PDT  smb
fs01     100G   0.00  2016-04-11 10:17:23 PDT  nfs
```

If Purity//FB cannot find matches for the wildcard argument specified, an error message appears.

Formatting

Format options control the way list outputs are displayed. The following format options are common to most **`list`** and **`monitor`** subcommands:

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data.

Here is a comparison between the `purefs list --space` output with formatted column titles and data, and the same output with unformatted column titles and data:

```
$ purefs list --space --page
Name     Size  Used    Data Reduction  Physical  Snapshots  Total
epic01   5T    4.03T   2.0 to 1        2.01T     0.00       2.01T
$ purefs list --space --raw --page
name   provisioned    space.virtual  space.data_reduction space.unique
 space.snapshots
epic01 5497558138880  4429186269696  2.3                 3221998891   0
space.total_physical
4339133009304
```

Sorting

When running the **`purefs list`** command, include the **`--sort`** option to sort a column of the output in ascending or descending order.

The sort option adheres to the following syntax:

`--sort SORT`

SORT represents a comma-separated list of columns to sort. Use the unformatted title name (`--raw`) of the column to represent each column listed. To sort a column in descending order, append the minus (-) sign to the column name.

For example, run the following commands to get the unformatted column titles in the `purefs list` output, and then sort the same output in descending order by the `space.virtual` (`Used`) column:

```
$ purefs list --raw --page
name      provisioned    space.virtual    created    protocols

$ purefs list --sort space.virtual-
Name Size Used    Created              Protocols
FS01 15T  2.48T   2017-06-13 09:24:44  nfs
FS06 15T  2.47T   2017-06-13 09:25:54  nfs
FS02 15T  2.25T   2017-05-13 09:24:44  nfs
FS04 10T  1.99T   2017-07-22 14:40:28  smb
FS03 50T  1.99T   2017-04-09 07:52:46  nfs
FS05 2T   1.26T   2017-05-13 09:36:33  nfs
```

If multiple columns are specified, the output is sorted by the order of the columns listed.

If a sorted column contains non-unique values and a secondary sort argument is not specified, Purity//FB automatically performs a secondary sort using the default sort criteria (typically by `Name`) .

Examples

Example 1: Display a list of file systems sorted in descending order by virtual space used.

```
$ purefs list --space --sort space.virtual-
```

Example 2: Display a list of file systems sorted in ascending order by virtual space used. If any of of the file systems are identical in virtual space used, sort those file systems in descending order by name.

```
$ purefs list --space --sort space.virtual,name-
```

Filtering

When running the **`purefs list`** command, include the **`--filter`** option to narrow the results of the output to only display the rows that meet the filter criteria.

Filtering with Operators

When filtering with operators, use the following syntax:

```
--filter "RAW_TITLE OPERATOR VALUE"
```

`RAW_TITLE` represents the unformatted title name of the column by which to filter. Run the list operation with the `--raw` option to get the unformatted title name.

OPERATOR represents the type of filter match (=, !=, <, >, <=, or >=) used to compare RAW_TITLE to VALUE.

VALUE represents the value (number, date, or string) that determines the results to be included in the list output. Literal strings must be wrapped in quotes.

Filtering supports the following operators:

Operator	Description
=	Equals
!=	Does not equal
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Filtering supports the asterisk wildcard character. For example, the value "*file*" in the `purefs list --filter "name = 'file*'"` command displays a list of all file systems that contain "fs" in the file system name.

The AND and OR operators can be used to further refine your filter. The AND operator displays only the results that meet all of the filter criteria in the command. The OR operator displays the results that meet at least one of the filter criteria in the command.

Examples

Example 1: Display a list of file systems that are greater than or equal to 5 tebibytes in provisioned size.

```
$ purefs list --filter "provisioned >= \"5T\""
```

OR

```
$ purefs list --filter "provisioned >= 5497558138880"
```

Example 2: Display a list of empty file systems.

```
$ purefs list --filter "space.virtual = '0'"
```

Example 3: Display a list of file systems that occupy a physical space greater than or equal to 3.43 tebibytes in size.

```
$ purefs list --space --filter "space.unique >= 3772099408945"
```

Example 4: Display a list of file systems that begin with `file` or are greater than 5 tebibytes in size.

```
$ purefs list --filter "name = 'file*' or provisioned > \"5T\""
```

Example 5: Display a list of file systems that begin with `file` and are greater than 5 tebibytes in size. $ purefs list --filter "name = 'file*' and provisioned > \"5T\""

Filtering with Functions

The filter option supports the CONTAINS and NOT functions.

`--filter FUNCTION(PARAMETERS)`

Function	Description
`contains(raw_title,string)`	Contains the enclosed string. Takes exactly two parameters, where `raw_title` represents the unformatted title name of the column by which to filter and `string` represents the string to search within the column. Run the list operation with the `--raw` option to get the unformatted title name.
`not(expression)`	Inverse of the enclosed expression.

Examples

Example 1: Get a list of all file systems with at least one NFS export rule set to `*(rw,no_root_squash)`.

```
$ purefs list --filter "contains(rules, '*(rw,no_root_squash)')"
```

Example 2: Get a list of file systems that do not contain "file" in the file system name.

```
$ purefs list --filter "not(contains(name, 'file'))"
```

Existence Checks

The Purity//FB CLI supports existence checks. For example, the `purefs list --filter "name"` command checks to see if the "name" column exists in the file system list output.

Examples

Example 1: Get a list of all file systems with data written to them.

```
$ purefs list --filter "space.virtual"
```

Example 2: Get a list of all file systems that do not have a provisioned size set.

```
$ purefs list --filter "not(provisioned)"
```

Setting Limits

When running the **`purefs list`** command, include the **`--limit`** option to restrict the result size to the limit specified.

The limit option adheres to the following syntax:

`--limit ROWS`

ROWS represents the maximum number of rows to return.

For example, run the following command to list only the first 5 rows of the `purefs list --space` output:

```
$ purefs list --space --limit 5
Name      Size  Used   Data Reduction  Physical  Snapshots  Total
MyFile01  5T    4.03T  2.3 to 1        4.04T     0.00       4.04T
MyFile02  5T    3.42T  2.3 to 1        3.43T     1.23T      4.66T
MyFile03  5T    3.42T  7.6 to 1        3.43T     0.00       3.43T
MyFile04  5T    3.42T  2.3 to 1        3.43T     0.00       3.43T
MyFile05  5T    3.42T  1.6 to 1        3.43T     0.00       3.43T
```

Combining Sorting, Filtering, and Limit Options

The `--sort`, `--filter`, and `--limit` options can all be combined together.

Examples

Example 1: Display the top 10 file systems that have the largest amount of pre-compressed data written to them, and include "file" in the file system name.

```
$ purefs list --sort "space.virtual-" --limit 10 --filter "name = '*file*'"
```

Example 2: Display the top 10 file systems with the largest physical space used.

```
$ purefs list --space --sort "space.unique-" --limit 10
```

CLI Login

Log in to the Purity//FB CLI to query and administer the FlashBlade. Logging in to the Purity//FB CLI requires a virtual IP address or fully-qualified domain name (FQDN) and an Purity//FB login username and password; this information is provided during the FlashBlade installation.

FlashBlade is installed with one administrative account with the username `pureuser`. The initial password for the account is `pureuser`. The password can be changed any time through the `pureadmin` CLI command.

Logging in to the Purity//FB CLI

To log in to the Purity//FB CLI, select one of the following two options:

- **UNIX.** Start a secure shell (SSH) session and connect to the array IP address provided by your Pure Storage account team. Log in to Purity//FB with the **`pureuser`** username/password combination.
- **Windows.** Start a remote terminal emulator (such as PuTTY) and connect to the array IP address provided by your Pure Storage account team. Log in to Purity//FB with the **`pureuser`** username/password combination.

Type **`exit`** or **`logout`** to log out of Purity//FB and exit the shell or terminal emulator.

CLI Command Help

CLI command help is available at both the command and subcommand levels. To use it, type the Purity//FB CLI command or subcommand followed by **`-h`** or **`--help`**. The command with the `-h` or `--help` switch displays usage information, supported syntax, and a list of subcommands for the specified command. To get help information for an Purity//FB command, type:

```
<command> -h
```

For example,

```
$ purefs -h
usage: purefs [-h] {add,create,delete,disable,enable,list,nfs,remove,setattr,snap}
…
positional arguments:
  {add,create,delete,disable,enable,list,nfs,remove,setattr,snap}
    add                 add protocols to one or more file systems
    create              create a file system
    delete              delete one or more file system(s)
    enable              enable file system attributes
    …
```

The command with the `-h` or `--help` switch displays usage information, supported syntax, and a list of options for the specified subcommand. To get help information for an Purity//FB subcommand, type:

```
<command> <subcommand> -h
```

For example,

```
$ purefs list -h
usage: purefs list [-h] [--pending | --pending-only]
                   [--space | --protocol-specific {http,nfs,smb} | --special-dir |
 --snap]
                   [--cli | --csv | --nvp] [--notitle]
                   [--total | --total-only] [--filter FILTER] [--sort SORT]
                   [--raw] [--limit LIMIT] [--page]
                   [FILE-SYSTEM …]
positional arguments:
  FILE-SYSTEM  file system name(s)
optional arguments:
  -h, --help           show this help message and exit
  --pending             list file systems including pending
  --pending-only        list pending file systems only
  --space              display space report
  --protocol-specific {http,nfs,smb}
                       display protocol specific information
  --special-dir        list status of special directories
  --snap               list snapshots
  …
```

CLI Man Pages

Man pages are available for all Purity//FB CLI commands.

To view a CLI man page, enter the **`pureman`** command followed by the name of the CLI command for that Purity//FB command.

The End License User Agreement is viewable in the CLI only by issuing the **`pureman purelicense`** command.

For comphrensive information about using the **`pureman`** command, see *pureman*.

CLI Command Tab Completion

The Purity//FB CLI supports Tab key completion of commands and most argument values that are mapped to managed objects in the system.

For example, pressing the Tab key at the end of **`puread`** completes the command name **`pureadmin`**.

For using the Tab key to complete an argument value, consider the following example:

The **`pureobj user create`** command requires that a user name is specified in the format `account/user`. If the only account that exists is acc1, pressing the Tab key after entering **`pureobj user create a`** completes the argument value with the result being, **`pureobj user create acc1/`**.

Chapter 11
Purity//FB CLI Commands

This section describes each of the Purity//FB CLI commands.

The commands listed in this chapter are used to administer and/or manage the FlashBlade, with the exception of the following information-only commands:

`purelicense`

Outlines the End User Agreement (EUA) between Pure Storage Inc. and end users of the company's products.

`pureman`

Describes the pureman command and displays a list of available Purity//FB commands.

pureadmin

pureadmin, pureadmin-create, pureadmin-delete, pureadmin-list, pureadmin-setattr, pureadmin-refresh — displays and manages the administrative account attributes

Synopsis

pureadmin create --api-token [--timeout ***PERIOD***] [***USER...***]

pureadmin delete --api-token [***USER...***]

pureadmin list [--cli|--csv|--nvp] [--notitle] [--raw] --filter ***FILTER***] [--page] [--limit ***LIMIT***] [--sort ***SORT***] [--api-token] [--expose] [--publickey] [***USER...***]

pureadmin refresh [--clear] [***USER***]

pureadmin setattr {--password | --publickey} [***USER...***]

Sub-Commands

`create`

Creates user attributes.

`delete`

Deletes user attributes.

`list`

Displays user attributes.

`setattr`

Modifies user attributes.

`refresh`

Refreshes a user group or clear directory service role cache to ensure user priviliges are up to date.

Arguments

USER

The user login name. For example, if logged in as `user1`, then specify `user1` for this argument. If logged in as `pureuser`, specifying `user1` will fail. If not specified, the login account will be the default.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--api-token`

Creates or deletes API tokens, or displays a list of users that have REST API access and the dates in which the API tokens were created.

`--clear`

Clears the directory service role cache.

`--expose`

Displays an unmasked API token.

`--timeout` ***PERIOD***

Sets the API token expiration. The value is set in the format `N[m/h/d/w}`. For example, `15m`, `1h`, `2d`, `3w`. If not set, the API token will not expire.

`--password`

Changes the password for the pureuser administrative account.

`--publickey`

Sets or displays users with SSH public keys configured.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--raw`

Displays the unformatted version of column titles and data.

Options that manage display results:

`--filter` ***FILTER***

Displays only the rows that meet the filter criteria specified.

`--limit` ***LIMIT***

Limits the size of the list output to the specified maximum number of rows.

`--page`

Turns on interactive paging.

`--sort` ***SORT***

Sorts the list output in ascending or descending order by the column specified.

Description

The current Purity//FB release comes with a single local, password-protected administrative account named **`pureuser`**.

The **`pureadmin create`** and **`pureadmin delete`** commands manage REST API tokens, which grant access to the REST API. API tokens are tied to a particular administrative account. All administrators have permission to manage their own API tokens.

The **`pureadmin setattr --password`** and **`pureadmin setattr --publickey`** commands allow you to set user authentication via password and public key.

Viewing Users

Issue the **`pureadmin list`** command to view local and remote array users.

```
pureadmin list
Name      Type
pureuser  local
```

The following information is displayed:

- **Name**: Lists the user name.
- **Type**: Displays `local` for users local to the array (pureuser), or `remote` for any other LDAP users.

Changing the User Password

Run the **`pureadmin setattr --password`** command to change the user password. The old (current) and new passwords are entered interactively. Press `Enter` after typing each password.

The CLI prompts once for the old password, and then twice for the new password. The password can be a maximum of 100 characters in length and include any character that is entered from a US keyboard.

Setting the User's Public Key for Authentication

Note: Setting other users' public keys is available to administrators and users with admin permissions only. The public key should be in the OpenSSH format, which begins with `ssh-rsa` or `ssh-ed25519`. A maximum of 100 users may have a public key set.

As an alternative to specifying user passwords, you can use public key authentication to allow users to SSH to arrays. Use the **`pureadmin setattr --publickey`** command to set the user's public key.

When running **`pureadmin setattr --publickey`** you can optional specify a user's login name. If not specified, the command sets the public key for the currently logged in user.

The CLI prompts for the public key. Enter the public key and press enter.

```
pureadmin setattr --publickey pureuser
Enter key:
Name      Type   Public Key
pureuser  local  ****
```

The **`puresetattr list --publickey`** command lists users with their public keys set.

```
pureadmin list --publickey pureuser
Name      Type   Public Key
pureuser  local  ****
```

The following information is displayed:

- **Name**: Lists the user name.
- **Type**: Displays `local` for users local to the array (pureuser), or `remote` for any other LDAP users.
- **Public Key**: The user's public key displayed as ****.

Creating an API token

The REST API requires permission to run commands. To access the server to run REST API commands, an authentication API token must be generated for the user. Each API token is unique and has an optional expiration setting. When a token expires, it can no longer be used and a new token must be generated to continue running REST API commands. It is a best practice to set an expiration on a token for the duration of time it will be used. The expiration can set by using the **`--timeout`** option with a time

indicator of `m` for minutes, `h` for hours, `d` for days, and `w` for weeks. Enter a numeric value followed by a time indicator. For example, `15m`, `1h`, `2d`, and `3w`. The indicators are not case-sensitive and only one time value is accepted.

Use the **`pureadmin create --api-token`** command to create an API token. In the following example, an API token is created for `pureuser` and expires in 2 days:

```
pureadmin create --api-token pureuser --timeout 2d
Name      Type  Role        API Token                            Created
         Expires
pureuser  local array_admin T-c3f9e02d-7ff7-44bd-b6a1-f7egg13419  2017-07-23
 09:05:21 PDT  2017-07-25 09:05:21 PDT
```

Viewing API tokens

The **`pureadmin list --api-token`** command lists all active API tokens for the array. Expired API tokens will not be shown. Use the **`--expose`** option to unmask the current user's API token.

```
 pureadmin list --api-token
Name      Type     API Token  Created                  Expires
pureuser  local    ****       2020-10-01 12:40:11 PDT  2020-10-02 12:40:11 PDT
```

- **Name**: Lists the user name.
- **Type**: Displays `local` for users local to the array (pureuser), or `remote` for any other LDAP users.
- **API Token**: Active tokens for API access displayed as ****.
- **Created**: API token creation time.
- **Expires**: Expiration time of the API token.

Refreshing User Roles

Permission groups (roles) are set using the **`pureds role`** command. If a user's permission group has been changed, the **`pureadmin refresh`** command is used to either refresh a user's group, or clear the directory service role cache to ensure the users' privileges in the array are up to date. The pureadmin refresh command cannot be used without **`--clear`** option and/or the **USER** argument.

When a user is assigned to multiple groups, for security purposes, the privileges for the more restrictive group are inherited by the user. However, because the directory service cache is cleared every 8 hours, the user may retain privileges from the original group. Use the **`pureadmin refresh USER`** command to refresh the group (and privileges) for the specified user; the privileges associated with the "new" group then take immediate effect. Use the **`pureadmin refresh --clear`** command to clear the directory service role cache for all array users.

One exception to this rule is users assigned the Ops Admin role—users assigned the Ops Admin role and any other role are denied the ability to log into the FlashBlade array. If there is an error with the

directory service configuration on the FlashBlade array, contact a user with Array Admin privileges to resolve the issue. If the LDAP server is configured incorrectly, contact your LDAP server administrator. The alert issued in the case where a user is denied access due to being assigned the Ops Admin and another role, provides additional information to help resolve the issue.

Example 1

```
pureadmin setattr --password pureuser
```

Changes the password for the pureuser administrative account. Type the old (current) password, and then press `Enter`. Type the new password, and then press `Enter`. Type the new password again, and then press `Enter`.

Example 2

```
pureadmin delete --api-token pureuser
```

Deletes the API token for the pureuser administrative account.

Example 3

```
pureadmin refresh FB_user1
```

In this example, user `FB_user1` was originally assigned to the Storage Admin group. `FB_user1` moved to a new team and was subsequently assigned to the ReadOnly group. In order to immediately remove Storage Admin privileges from `FB_user1`, the **`pureadmin refresh`** command is run.

Example 4

```
pureadmin refresh --clear
```

Clears the directory service role cache for all array users.

Example 5

```
pureadmin setattr --publickey pureuser
```

Sets the public key for the user `pureuser`, allowing `pureuser` to SSH to the array without specifying a password.

Example 6

```
pureadmin list --publickey pureuser
```

Lists the public key information for `pureuser`.

See Also

- *pureds*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purealert

purealert, purealert-flag, purealert-list, purealert-unflag, purealert-watcher-create, purealert-watcher-delete, purealert-watcher-disable, purealert-watcher-enable, purealert-watcher-list, purealert-watcher-setattr, purealert-watcher-test — manages alert history and designated alert watchers to which Purity//FB sends email notifications when significant events occur in an array

Synopsis

```
purealert flag [ ID...]

purealert list [--cli|--csv|--nvp|--report] [--closed|--closed-only]
[--notitle] [--unflagged|--unflagged-only] [--filter FILTER] [--sort SORT]
[--limit LIMIT] [--raw] [ --page] [ID...]

purealert unflag [ID...]

purealert watcher create ADDRESS...

purealert watcher delete ADDRESS...

purealert watcher disable ADDRESS...

purealert watcher enable ADDRESS...

purealert watcher list [--cli|--csv|--nvp] [--notitle] [--filter FILTER]
[--sort SORT] [--limit LIMIT] [--raw] [--page] [ADDRESS...]

purealert watcher setattr
--minimum-notification-severity {critical|info|warning} ADDRESS...

purealert watcher test [ADDRESS...]
```

Sub-Parsers

`watcher`

When specified, indicates that sub-commands are to be processed for alert watchers/

Sub-Commands

`create`

Creates an alert watcher. Must be used with the `watcher` sub-parser.

`delete`

Deletes an alert watcher. Must be used with the `watcher` sub-parser.

`disable`

Disables an alert watcher. Must be used with the `watcher` sub-parser.

`enable`

Enables an alert watcher. Must be used with the `watcher` sub-parser.

`flag`

Flags alerts.

`list`

Displays all alerts.

When used with the `watcher` sub-parser, displays all alert watchers.

`setattr`

Modifies an alert watcher.

`test`

Sends a test message to an alert watcher.

`unflag`

Unflags alerts.

Arguments

ADDRESS

Any valid email address.

ID

Unique, numeric ID assigned by the array to each alert.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--closed`

Display all alerts including closed alerts.

`--closed-only`

Display only the closed alerts.

`--minimum-notification-severity {critical|info|warning}`

Specifies the minimum alert severity (critical, info, or warning) that an alert must have in order for an email to be sent to an alert watcher.

`--unflagged`

Display alerts including unflagged alerts.

`--unflagged-only`

Display only unflagged alerts.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--filter`

Displays only the rows that meet the filter criteria specified.

`--limit`

Limits the size of the list output to the specified maximum number of rows.

`--report`

Displays extended alert information.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data.

`--sort`

Sorts the list output in ascending or descending order by the column specified.

Description

Purity//FB generates an alert whenever the health of a component degrades or a capacity threshold is reached. Alerts can also be sent as email notifications to designated alert watchers.

The **`purealert`** command displays and manages the alerts that are generated by Purity//FB.

Viewing Alerts

The **`purealert list`** command displays a list of all flagged and unclosed alerts on the FlashBlade array that have been generated by a hardware, software, or array event. Each alert in the list output includes the following information:

- **ID**: Unique number assigned by the array to the alert. ID numbers are assigned to alerts in chronological ascending order.
- **Flagged**: Alerts can be flagged as a way to call attention to certain alerts. Flagged alerts will be indicated as `True` and unflagged alerts are indicated as `False`.
- **Code**: Pure Storage alert code number that identifies the type of alert event.
- **Severity**: Alert severity, categorized as `critical`, `warning`, or `info`.

  Critical alerts are typically triggered by service interruptions, major performance issues, or risk of data loss, and require immediate attention. For example, Purity//FB triggers a critical alert if a FlashBlade has been removed from the chassis.

  Warning alerts are of low to medium severity and require attention, though not as urgently as critical alerts. For example, Purity//FB triggers a warning alert if it detects an unhealthy FlashBlade that is still processing data.

  Informational (Info) alerts inform users of a general behavior change and require no action. For example, Purity//FB triggers an informational alert if the NFS service is unhealthy.
- **State**: Current state of the alert. Possible states include: `open`, `closed`, `closing`, and `waiting to downgrade`.

  An alert goes from `open` state to `close` state when the issue is completely resolved, such as when a blade has been removed from the chassis. If the blade is then reinserted and pulled again, a new alert will be generated.

  Certain alerts are generated due to intermittent issues. Examples include temperature sensors reporting values outside of normal operating range and space usage exceeding certain capacity thresholds. These types of alerts can go from `open` state to `closing` or `waiting to downgrade` states.

  And alert goes from `open` state to `closing` state when it is no longer an issue. After the alert remains in `closing` state for 24 hours without change, it goes into `closed` state.

  An alert goes from `open` state to `waiting to downgrade` state when the issue becomes less severe. After the alert remains in `waiting to downgrade` state for 24 hours without change,

the alert severity is downgraded. For example, when array capacity reaches 100%, a critical alert is issued with an `open` state. When array capacity drops below 100%, the alert changes to `waiting to downgrade` state. After the array has remained at less than 100% but more than 90% capacity for 24 hours, the alert severity is downgraded to `warning`.

- **Created**: Date and time the alert was first generated and initial alert email notifications were sent to alert watchers.
- **Last Seen**: Most recent date and time Purity//FB saw the issue that generated the alert.
- **Subject**: Alert details.

Creating Alert Watchers

An alert watcher is an email recipient who has been designated to receive email notifications every time an alert is generated on the array. The sending account name for Purity//FB alert email notifications is the array name at the configured sender domain.

Run the **`purealert watcher create`** command to add alert watchers. The ADDRESS argument that must be included in the command represents the alert watcher's email address. Before adding an alert watcher, verify that alert notifications can reach the watcher's destination email address by sending a test message using the `purealert watcher test` command with the ADDRESS argument.

Once added, an alert watcher starts receiving alert email notifications.

To view a list of alert watchers, run the `purealert watcher list` command.

Setting the Severity Level for Notifications

When created, an alert watcher is set to receive all alert notifications, regardless of alert severity. Use the **`purealert watcher setattr`** command with the `--minimum-notification-severity` option to set the alert notifications to include info, warning, or critical-level alerts.

- `info`: If specified, the alert watcher will receive all alert notifications for all severity levels.
- `warning`: If specified, the alert watcher will receive any alert with a severity of Warning or more severe. Info-level alert notifications will not be received.
- `critical`: If specified, the alert watcher will only receive the most severe Critical-level alert notifications.

The ADDRESS argument that must be included in the command represents the alert watcher's email address.

Sending Test Messages

The **`purealert watcher test`** command tests an array 's ability to send email notifications to alert watchers, designated or not. Two types of test messages can be issued:

- Before creating an alert watcher, send a test message to the alert watcher's email address to ensure alert notifications can reach the intended destination. To do this, run **`purealert watcher test`** with the ADDRESS argument.
- At any time, send a test message to all of the enabled alert watchers to ensure alert notifications reach their intended destinations. To do this, run **`purealert watcher test`**.

Running the `purealert watcher test` command returns the following information for each alert watcher:

- **Name**: The email address of the alert watcher.
- **Accepted**: Indicates whether the test message has successfully left the array. An Accepted status of `True` indicates that the test message has successfully left the array. If the Accepted status is `False`, see the Error column for the reason why the test message failed.
- **Error**: Reason why the test message failed to leave the array.

Enabling and Disabling Alert Watchers

Alert watchers can be in enabled or disabled status.

Alert watchers who are in enabled status receive alert email notifications. When an alert watcher is created, its watcher status is automatically set to enabled status.

Alert watchers who are in disabled status do not receive alert email notifications. Disabling an alert watcher does not delete the recipient's email address - it only stops the watcher from receiving alert notifications.

Enable and disable alert watchers at any time by running the respective **`purealert watcher enable`** and **`purealert watcher disable`** commands.

To view a list of alert watchers and their enabled/disabled statuses, run the `purealert watcher list` command.

Viewing Alert Watchers

Use the **`purealert watcher list`** command to display a list of email recipients that have been designated as alert watchers. The output from command includes the following information:

- **Name**: The email address of the alert watcher.
- **Minimum Notification Severity**: The minimum alert severity that will generate an alert email for the alert watcher. Values can be info, critical, or warning.
- **Enabled**: Indicates if the alert watcher is enabled (True) and is receiving notifications, or disabled (False) and is not receiving email notifications.

Alert Flags

Alerts can be flagged or unflagged to help the administrator manage alerts. By default, alerts are flagged. To unflag an alert, set the attribute to `False` by running the **`purealert unflag`** command. To flag an alert, set the attribute to `True` by running the **`purealert flag`** command.

Deleting Alert Watchers

The **`purealert watcher delete`** command deletes an alert watcher's email address. Once an email address has been deleted, the corresponding alert watcher will no longer receive alert notifications.

Example 1

```
purealert list
```

Displays a list of all alerts on the FlashBlade array that have been generated by a hardware, software, or array event.

Example 2

```
purealert watcher test alexalert@example.com
```

Verifies that `alexalert@example.com` can receive email notifications from the FlashBlade array

Example 3

```
purealert watcher create alexalert@example.com
```

Creates a new alert watcher, `alexalert@example.com`.

Example 4

```
purealert watcher list
```

Displays a list of alert watchers that have been created on the array and the enabled/disabled status of each watcher.

Example 5

```
purealert watcher setattr --minimum-notification-severity warning
 alexalert@example.com
```

Sets the alert email notifications to only be sent to `alexalert@example.com` if the alert is at the `warning` level.

Example 6

```
purealert watcher delete alexalert@example.com
```

Removes `alexalert@example.com` as an alert watcher. Alert notifications will no longer be sent to `alexalert@example.com`.

Example 7

```
purealert watcher disable artiealert@example.com
```

Prevents Purity//FB from sending alert notifications to alert watcher `artiealert@example.com`.

Example 8

```
purealert unflag 2
```

Unflag alert ID 2 by setting the parameter to `False`.

See Also

- *purearray*
- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purearray

purearray, purearray-connect, purearray-create, purearray-disable, purearray-disconnect, purearray-enable, purearray-list, purearray-rename, purearray-setattr — displays and manages the array attributes and connections.

purearray-monitor monitors array I/O performance

Synopsis

```
purearray connect --management-address MANAGEMENT_ADDRESS
[--replication-address REPLICATION_ADDRESS]

purearray create --connection-key

purearray disable --connect --encrypted NAME

purearray disconnect NAME

purearray enable --connect --encrypted NAME

purearray list [--banner] [--cli|--csv|--nvp] [--notitle] [--raw]
[--filter FILTER] [--page] [--limit LIMIT] [--sort SORT] [--connect
|--connection-key |--ntpserver |--space |--timezone] [--file-system-only
|--object-only][--historical HISTORICAL ] [--path] [--status-detail] [ARRAY...]

purearray monitor [--csv] [--notitle] [--raw] [--filter FILTER]
[--limit LIMIT] [--sort SORT] [--historical HISTORICAL] [--interval SECONDS]
[--replication] [--file-system-only|--object-only] [--connect [ NAMES]]
[--protocol {http|nfs|smb|s3}|--protocol-specific {http|s3|nfs}] [--data]
[--file-metadata-read] [--share-metadata-read] [--file-metadata-create]
[--file-metadata-modify] [--client CLIENT] [--repeat REPEAT-COUNT] [--total]
[--total-only]

purearray rename NEW-NAME

purearray setattr
{--connect|--ntpserver NTP-SERVER|--time-zone TIME_ZONE|--banner}
[--management-address MANAGEMENT_ADDRESS]
[--replication-address REPLICATION_ADDRESS] [NAME]
```

Sub-Commands

connect

Connects two arrays.

`create`

Creates connection keys.

`disable`

Disables encryption between arrays for object replication.

`disconnect`

Terminates the connection between two arrays.

`enable`

Enables encryption between arrays for object replication.

`list`

Displays array attributes and connections.

`monitor`

Displays array performance.

`rename`

Renames an array.

`setattr`

Modifies an array's connections and configuration.

Arguments

`ARRAY...`

Name by which the array is to be known after the command executes.

`NEW-NAME`

Name by which the array is to be known after the command executes.

`NAME`

Name of the remote array.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--banner`

Sets or views the array login banner. Limit is 8,000 characters.

`--client [ CLIENT]`

Displays client-specific I/O performance information for all clients. You can optionally specify a single client or a comma-separated list of clients. Clients must be specified as an IP address and port combination. Wildcard client names are also supported. This option requires the `--protocol nfs` option.

`--connect`

When used with the `purearray enable` or `purearray disable` commands, updates an array connection.

When used with the `purearray list` command, displays array connections.

When used with the `purearray monitor` command, displays connection-specific I/O performance. You can optionally enter specific array names with the `purearray monitor` command to display I/O performance information for those arrays.

When used with the `purearray setattr` command, updates an array connection.

`--connection-key`

When used with the `purearray create` command, specifies that a connection key is created. When used with the `purearry list` command, displays array connection keys.

`--data`

Displays performance information for read and write operations.

`--encrypted`

Enables or disables array connection encryption when used with the `purearray enable` or `purearray disable` commands, respectively.

`--file-metadata-create`

Displays performance information for CREATE, LINK, MKDIR, and SYMLINK operations.

`--file-metadata-modify`

Displays performance information for REMOVE, RENAME, RMDIR, and SETATTR operations.

`--file-metadata-read`

Displays performance information for GETATTR, LOOKUP, PATHCONF, READDIR, READDIRPLUS, and READLINK operations.

`--share-metadata-read`

Displays performance information for ACCESS, FSINFO, and FSSTAT operations.

`--file-system-only`

Limits displayed space information to only file systems.

`--filter` ***FILTER***

Specifies a custom filter to display client-specific performance information. This option is allowed for use with only the `--client` option.

`--historical` ***HISTORICAL***

Displays historical data at the given resolution in the format of `N[m|h|d|w]`, for example 15m, 2h, 1d, etc. The minimum acceptable time for viewing historical data is 5 minutes for replication and 1 second for non-replication.

`--interval` ***SECONDS***

Sets the number of seconds between displays of real-time performance data. At each interval, the system displays a point-in-time view of the performance data. If omitted, the interval defaults to every 5 seconds.

`--limit` ***LIMIT***

Specifies the number of clients with client-specific performance displayed. By default 100 clients are displayed. A maximum of 100,000 is allowed. This option is allowed for use with only the `--client` option.

`--management-address` ***MANAGEMENT_ADDRESS***

Specifies the management IP address of the remote array.

`--ntpserver` ***NTP-SERVER***

Displays or sets the hostnames or IP addresses of the NTP servers used by the array to maintain reference time.

When used with **`purearray setattr`**, if specifying an IP address, for IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from `0` to 255 representing a group of 8 bits. For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`) .

`--object-only`

Limits displayed space information to only objects.

`--protocol` ***PROTOCOL***

Displays array -wide real-time and historical I/O performance information for the specified protocol, either `http`, `nfs`, `s3`, or `smb`.

`--protocol-specific` ***PROTOCOL***

Displays array -wide real-time and historical I/O performance information in the format for the specified protocol. Valid for `http`, `s3`, and `nfs`.

`--repeat` ***`REPEAT-COUNT`***

Sets the number of times to display real-time performance data. If omitted, the repeat count defaults to 1.

`--replication`

Displays replication traffic information.

`--replication-address` ***`REPLICATION_ADDRESS`***

Specifies the replication IP address of the remote array. The replication address is auto-discovered on connect if this option is not specified and Network Address Translation is not being used.

`--size`

Displays the average I/O sizes per read, write, and all operations. The information appears in the **B/op** columns of the output.

`--sort` ***`SORT`***

Specifies a custom sort be applied to client-specific performance information. This option is allowed for use with only the `--client` option.

`--space`

Displays array size and storage consumption details, including usable capacity and physical storage consumption, and array parity.

`--status-detail`

Displays status details.

`--time-zone`

When used with the `purearray list` command, displays the array time zone.

When used with the `purearray setattr` command, updates the array time zone.

`--total`

Appends a line containing aggregated data.

`--total-only`

Displays only aggregated data.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--raw`

Displays the unformatted version of column titles and data.

Common options used for display results:

`--page`

Turns on interactive paging.

Description

The **`purearray`** command displays and manages the FlashBlade array attributes, and is used to connect arrays when implementing file and object replication.

Setting the Array Login Banner

To help identify the FlashBlade system being logged into, you can set a login banner using the **`purearray setattr --banner`** command. The banner text is limited to 8,000 characters. Upon issuing the **`purearray setattr --banner`** command, you are prompted for the new banner text. Once you have entered the banner text, press **Enter** and then press **Ctrl+D**. For example:

```
purearray setattr --banner
Please enter banner followed by ^D:

Welcome to Array1 <press Enter and then press Ctrl+D>
Banner
Welcome to Array1
```

To view the array's current login banner text, issue the **`purearray list --banner`** command.

To clear the array's login banner text, issue the **`purearray setattr --banner`** command. When prompted for the banner text, press **Enter** and then press **Ctrl+D**. For example:

```
purearray setattr --banner
```

```
Please enter banner followed by ^D:

<Press Enter and then press Ctrl+D>

Banner
-
```

Viewing Array Attributes

The **`purearray list`** command displays FlashBlade array attributes, including array name, array ID, and Purity//FB version and revision numbers.

Purity//FB CLI dates and times are displayed in the time zone of the array, which is set during FlashBlade installation. The arrays maintain time synchronization by interacting with the NTP server. Include the **`--ntpserver`** option to display the Network Time Protocol (NTP) servers used by the array to maintain reference time. Run the `purearray setattr --ntpserver` command to change the NTP servers on the array.

Include the **`--space`** option to display a point-in-time view of the array size and storage consumption. Combine the **`--space`** and **`--historical`** options to display historical storage consumption over any of the following ranges of time: 3 hours, 1 day, 7 days, 30 days, or 1 year. See the list of storage consumption details:

- **Capacity**: Total usable capacity of the array.
- **Parity**: Percentage of data that is fully protected. The percentage value will drop below 100% if the data isn't fully protected, such as when a blade is pulled and the array is rebuilding the data to bring it back to full parity.
- **Data Reduction**: Ratio of the size of the written data (i.e. Used) versus the amount of space the data occupies after data compression (i.e. Physical).
- **Live**: Amount of space that the written live data occupies on the array after reduction via data compression.
- **Snapshots**: Amount of space occupied by data unique to one or more snapshots.
- **Shared Space**: Amount of space that is shared with other file systems and snapshots.
- **System**: Amount of space occupied by internal array metadata.
- **Total**: Total amount of space on the array.

Connecting Two FlashBlade Arrays

⚠ **Important note!** The management IP address required during configuration cannot be a loopback IP address.

File and object replication require connecting two FlashBlade arrays. The two arrays are designated the source and target arrays. Connecting two arrays requires performing the following steps:

1. On both the source and target arrays, use the **`purenetwork vip create --service replication`** command to create replication vips.
2. Create a connection key on the target array using the **`purearray create --connection-key`** command. Once created, the key is active for two hours. If the arrays have not been connected within that two hour period, a new connection key must be created. Run the **`purearray list --connection-key`** command to view a connection key's expiry time. Note that the key itself will not be displayed using **`purearray list --connection-key`**; the key is only visible with the **`purearray create --connection-key`** command.
3. From the target array, obtain the array's management IP address by running **`purenetwork list`**.
4. On the source array run the **`purearray connect --management-address MANAGEMENT_ADDRESS`** command, where `MANAGEMENT_ADDRESS` is the management IP address of the target array.
   - If the optional `--replication-address` argument is used, the specified replication address is set on the local array connection and the remote array connection will have no replication address set.
   - If not used, the replication address available on the target array is used, and the replication address of the local array is automatically populated on the target array connection.
5. When prompted, enter the target array's connection key.
6. If using NAT, on the target array issue the **`purearray setattr --replication address REPLICATION_ADDRESS`** command, where `REPLICATION_ADDRESS` is the replication address of the source array.

Viewing Array Connections

Use the **`purearray list ARRAY`** command with the `--connect` and `--path` options to view array connections.

Issuing **`purearray list --connect`** displays information about the connected array (to the array on which the command issued).

```
purearray list --connect
Name    ID                                    Version  Management Address ...
array2  77bb86bd-12ce-11e6-94cc-985aeb8d9f04  3.0.0    10.64.241.38       ...
```

Displayed information includes the following:

- **Name**: The name of the connected array.
- **ID**: The ID of the connected array.

- **Version**: The software version on the connected array.
- **Management Address**: The management IP address or FQDN of the connected array.
- **Replication Address**: The replication IP address of the connected array.
- **Encrypted**: Indicates if the array connection is encrypted (`True`) or not encrypted (`False`).
- **CA Certificate Group**: The name of the used certificate group. If encryption is enabled, the CA certificate group is `_default_replication_certs`.
- **Status**: The current connection status. Valid values are `connected`, `connecting`, or `partially_connected`.

Issuing **`purearray list --connect --path`** displays the connection details for the ports used in file and object replication. For file replication, the inbound and outbounds paths are displayed.

```
purearray list --connect --path
Name    Source         Destination       Status
array2  10.64.7.234    10.64.7.237:443   connected
                       10.64.7.237:8117  connected
        10.64.7.237    10.64.7.234:8117  connected
```

Displayed information includes the following:

- **Name**: The name of the target array.
- **Source**: The replication source IP address.
- **Destination**: The destination IP address. For file replication, the address is followed by `:8117`. For object replication, if the connection is encrypted, the address is followed by `:443`. If the connection is not encrypted, the address is followed by `:80`.
- **Status**: The status (`connected` or `connecting`) of the connection between the source and target arrays.

This can be useful to determine which network paths are having problems if the array connection is not healthy.

You can optionally use the `--status-detail` option to display a `Details` column, which shows detailed information about the connection health if the connection status is `connecting`.

Enabling and Disabling Encryption Between Two Arrays

Note: Encryption is only valid for object replication. Additionally, if any file replica links are configured, attempts to enable encryption will fail.

The connection between two arrays in an object replication set up can be encrypted. To enable encryption between two arrays, on the source array run the **`purearray enable --connect --encrypted NAME`** command, where NAME is the

name of the target array. To disable encryption between two arrays, on the source array run the **`purearray disable --connect --encrypted NAME`** command, where NAME is the name of the target array.

Updating the Connection Between Two Arrays

If the management and/or replication IP addresses of the source or target arrays change, the connection between two FlashBlade arrays can be lost. To update connection information, from the source array run the **`purearray setattr --connect`** command. If the replication address ofd the source array changes, from the target array run the **`purearray setattr --connect --replication-address`** command.

Terminating the Connection Between Two Arrays

The connection between two arrays can be terminated at any time, thereby terminating file and object replication. Note that terminating object replication requires removing all replica links before disconnecting. Additionally, if there is active object replication, disconnecting two arrays is not allowed. When the replication connection between arrays is terminated, the following occurs:

- For file replication, all active replication is removed.
- All in-process replication tasks are canceled.
- All transfer records are deleted.
- Policies are retained on the target array until there are no policies that reference them.

To terminate the connection between two arrays, from the source array, run the **`purearray disconnect NAME`** command, where NAME is the name of the connected target array.

Monitoring Array Performance

The **`purearray monitor`** command displays real-time and historical I/O performance information for the whole array. The output includes the following data about bandwidth, IOPS, and latency:

- **Time**: Current time.
- **B/s**: Number of bytes read/written.
- **op/s**: Number of read/write requests processed per second.
- **us/op**: Average arrival-to-completion time, measured in microseconds, for a host read/write operation.
- **B/op**: Average I/O size per read, write, and all operations. Include the `--size` option to see the **B/op** columns.

Include the **`--historical`** option to display historical performance data. Note that the minimum acceptable time for viewing historical data is 5 minutes for replication and 1 second for non-replication.

By default, the **`purearray monitor`** command displays real-time performance data at the array level. Include the **`--repeat`** option to specify the number of times to repeat the real-time update. If not specified, the repeat value defaults to 1. Include the **`--interval`** option to specify the number of seconds between each real-time update. At each interval, the system displays a point-in-time snapshot of the performance data. If not specified, the interval value defaults to every 5 seconds. The **`--interval`** and **`--repeat`** options can be combined.

With the **`--protocol`** option, `purearray monitor` displays performance throughput and IOPs information for the specified protocol.

With the **`--protocol-specific`** option, `purearray monitor` displays metadata performance information that is related only to the specified protocol.

With the `--protocol-specific http` option, the performance information reflects the array's files and directories accessed over the HTTP protocol:

- **op/s**: Number of read/write requests processed per second for this file or directory.
- **us/op**: Average arrival-to-completion time, measured in microseconds, for a host read/write operation for this file or directory.

For example:

```
purearray monitor --protocol-specific http
Time                     op/s (read dir)  op/s (write dir)  op/s (read file),,,
2020-04-07 17:52:53 MDT  44.00             0.00              50.00           ,,,
```

With the `--protocol-specific s3` option, the performance information reflects the array's buckets and objects accessed over S3:

- **op/s**: Number of read/write requests processed per second for this bucket or object.
- **us/op**: Average arrival-to-completion time, measured in microseconds, for a host read/write operation for this bucket or object.

For example:

```
purearray monitor --protocol-specific s3
Time                     op/s (read bucket)  op/s (write bucket)...
2020-04-07 16:48:28 PDT  77.00               0.00                ...
```

With the `--protocol-specific nfs` option, the performance information reflects all NFS operations, including 17 metadata operations for the entire array, grouped in the following four categories:

- File/directory level read-like metadata requests:
  - GETATTR
  - LOOKUP
  - PATHCONF
  - READDIR
  - READDIRPLUS
  - READLINK
- File/directory-level create-like metadata requests:
  - CREATE
  - LINK
  - MKDIR
  - SYMLINK
- File/directory-level modify/delete-like metadata requests:
  - REMOVE
  - RENAME
  - RMDIR
  - SETATTR
- Share-level read-like metadata requests:
  - ACCESS
  - FSINFO
  - FSSTAT

For all other operations (such as COMMIT, MKNOD, NULL, and NLM operations, and any operations supported by NFSv4.1, but not NFSv3), we continue to use an "other operations" group.

These metadata operations are measured in the units as follows:

- **op/s**: Number of a specific requests processed per file system.
- **us/op**: Average arrival-to-completion time, measured in microseconds, for a host specific operation for this file system.

By default, the `--protocol-specific nfs` option displays “read”, “ write”, “file read”, “share read”, “file create” and “file modify” and the reduced "other" group - with op/s and us/op for each operation. There are several other CLI flags you can include to view a more granular breakdown of metadata. One or more of the following flags can be included at a time:

- Include `--data` to only display performance information for read and write operations.
- Include `--file-metadata-create` to display performance information for create, link, mkdir, and symlink operations.
- Include `--file-metadata-modify` to display performance information for remove, rename, rmdir, and setattr operations.
- Include `--file-metadata-read` to display performance information for getattr, lookup, pathconf, readdir, readdirplus, and readlink operations.
- Include `--share-metadata-read` to display performance information for access, fsinfo, and fsstat operations.

Additionally, when using the `nfs` option, the following performance information can be included in the command output:

- **op/s (access)**: Average number of **access** operations processed per second.
- **us/op (access)**: Average time, measured in milliseconds, to process an **access** request.
- **op/s (create)**: Average number of **create** operations processed per second.
- **us/op (create)**: Average time, measured in milliseconds, to process a **create** request.
- **op/s(fsinfo)**: Average number of **fsinfo** operations processed per second.
- **us/op fsinfo)**: Average time, measured in milliseconds, to process a **fsinfo** request.
- **op/s (fsstat)**: Average number of **fsstat** operations processed per second.
- **us/op (fsstat)**: Average time, measured in milliseconds, to process a **fsstat** request.
- **op/s (getattr)**: Average number of **getattrs** operations processed per second.
- **us/op (getattr)** Average time, measured in milliseconds, to process a **getattr** request.
- **op/s (link)**: Average number of **link** operations processed per second.
- **us/op (link)** Average time, measured in milliseconds, to process a **link** request.
- **op/s (lookup)** Average number of **lookup** operations processed per second.
- **us/op (lookup)** Average time, measured in milliseconds, to process a **lookup** request.
- **op/s (mkdir)** Average number of **mkdir** operations processed per second.

- **us/op (mkdir)** Average time, measured in milliseconds, to process a **mkdir** request.
- **op/s (pathconf)** Average number of **pathconf** operations processed per second.
- **us/op (pathconf)** Average time, measured in milliseconds, to process a **pathconf** request.
- **op/s (read data)**: Average number of **read** operations processed per second.
- **us/op (read data)**: Average time, measured in milliseconds, to process a **read** request.
- **op/s (readdir)**: Average number of **readdir** operations processed per second.
- **us/op (readdir)**: Average time, measured in milliseconds, to process a **readdir** request.
- **op/s (readdirplus)**: Average number of **readdirplus** operations processed per second.
- **us/op (readdirplus)**: Average time, measured in milliseconds, to process a **readdirplus** request.
- **op/s (readlink)**: Average number of **readlink** operations processed per second.
- **us/op (readlink)**: Average time, measured in milliseconds, to process a **readlink** request.
- **op/s (remove)**: Average number of **remove** operations processed per second.
- **us/op (remove)**: Average time, measured in milliseconds, to process a **remove** request.
- **op/s (rename)**: Average number of **rename** operations processed per second.
- **us/op (rename)**: Average time, measured in milliseconds, to process a **rename** request.
- **op/s (rmdir)**: Average number of **rmdir** operations processed per second.
- **us/op (rmdir)**: Average time, measured in milliseconds, to process a **rmdir** request.
- **op/s (setattr)**: Average number of **setattr** operations processed per second.
- **us/op (setattr)**: Average time, measured in milliseconds, to process a **setattr** request.
- **op/s (symlink)**: Average number of **symlink** operations processed per second.
- **us/op (symlink)**: Average time, measured in milliseconds, to process a **symlink** request.
- **op/s (write data)**: Average number of **write** operations processed per second.
- **us/op (write data)**: Average time, measured in milliseconds, to process a **write** request.
- **op/s (other)**: Average number of **other** operations (commit, mknod, null, operations only enabled on nsv4 and not on nsv3-exclusive nsv4 ops) processed per second.
- **us/op (other)**: Average time, measured in milliseconds, to process an **other** request (commit, mknod, null, operations only enabled on nsv4 and not on nsv3-exclusive nsv4 ops).

- **op/s (file create)**: Average number of **file create** operations (create, link, mkdir, symlink) processed per second.
- **us/op (file create)**: Average time, measured in milliseconds, to process a **file create** request (create, link, mkdir, symlink).
- **op/s (file modify)**: Average number of **file modify** operations (remove, rename, rmdir, setattr) processed per second.
- **us/op (file modify)**: Average time, measured in milliseconds, to process a **file modify** request (remove, rename, rmdir, setattr).
- **op/s (file read)**: Average number of **file read** operations (getattr, lookup, pathconf, readdir, readdirplus, readlink) processed per second.
- **us/op (file read)**: Average time, measured in milliseconds, to process a **file read** request (getattr, lookup, pathconf, readdir, readdirplus, readlink).
- **op/s (share read)**: Average number of **share read** operations (access, fsinfo, fsstat) processed per second.
- **us/op (share read)**: Average time, measured in milliseconds, to process a **share read** request (access, fsinfo, fsstat).

Synchronizing the Array Time

Purity//FB CLI dates and times are maintained in the time zone of the array, which is set during FlashBlade installation. The arrays maintain time synchronization by interacting with the NTP server. The **`purearray setattr --ntpserver`** command sets the Network Time Protocol (NTP) servers, completely replacing any previous NTP server assignments.

Renaming the Array

The **`purearray rename`** command changes the current name of the array to the new name (`NEW-NAME`) . The name change is effective immediately and the old name is no longer recognized in CLI or GUI interactions. In the Purity//FB GUI, the new name appears upon page refresh.

Viewing Client Performance

⚠ **Important note!** The `--client` option can only be used with the `--protocol nfs` option. Additionally, the `--historical` option cannot be used with the `--client` option as I/O statistics are currently only collected instantaneously.

The **`purearray monitor --client [CLIENT]`** command is provided to sample and display client statistics. Used without specifying a specific client or list of clients, I/O performance statistics are displayed for all clients. By default, the data for 100 clients is displayed. However, the data for up to

100,000 clients can be displayed by using the `--limit LIMIT` option. Additional filtering and sorting can be applied using the `--filter` and `--sort` options.

Example 1

```
purearray create --connection-key
```

When connecting two arrays, the **`purearray create --connection-key`** command is run on the target array to generate a connection key.

Example 2

```
purearray connect --management-address 10.11.123.12
Enter the connection key of the target array:
```

After the connection key is created, the **`purearray connect --management-address MANAGEMENT_ADDRESS`** command is issued on the source array. When prompted, the connection key from the target array is entered.

Example 3

```
purearray list --connect
```

Displays the status and connection information for the connected array.

Example 4

```
purearray list --connect --path
```

Displays the connection details for the ports used in file and object replication.

Example 5

```
purearray setattr --management-address 10.13.456.77 array2
```

Updates the management IP address for the target array, `array2`.

Example 6

```
purearray disconnect array2
```

Terminates the connection between the source array (`array1`) and target array (`array2`).

Example 7

```
purearray list --csv
```

Displays a CSV output of the array attributes, including the FlashBlade array name, array ID, and Purity//FB version and revision numbers.

Example 8

```
purearray list --space
```

Displays array size and storage consumption details.

Example 9

```
purearray list --ntpserver
```

Displays the hostnames or IP addresses of the NTP servers that are currently configured on the array.

Example 10

```
purearray list --space --historical 3h
```

Displays the historical space usage of the array over the past three (3) hours.

Example 11

```
purearray monitor --historical 5m --csv
```

Displays a CSV output of historical performance data for the local array over the past five (5) minutes.

Example 12

```
purearray monitor --protocol smb
```

Displays real-time performance data for the SMB protocol only.

Example 13

```
purearray monitor --repeat 20 --interval 10
```

Displays real-time performance data for the whole array. Twenty (20) point-in-time updates are displayed, with each update taken every ten (10) seconds.

Example 14

```
purearray setattr --ntpserver MyNTPServer1.com,MyNTPServer2.com
```

Assigns NTP servers `MyNTPServer1.com` and `MyNTPServer2.com` as the array 's sources for reference time, replacing any previous NTP server assignments.

Example 15

```
purearray monitor --protocol nfs --client
```

Displays client-specific I/O performance statistics.

Example 16

```
purearray monitor --protocol-specific nfs
```

Displays array-wide real-time and historical I/O performance information for NFS protocol only.

Example 17

```
purearray monitor --protocol-specific nfs --data
```

Displays array-wide real-time and historical I/O performance information with respect to read/write data operations for NFS protocol only.

Example 18

```
purearray monitor --protocol-specific nfs --file-metadata-read --file-metadata-
write
```

Displays array-wide real-time and historical I/O performance information with respect to the file-metadata-read/write operations for NFS protocol only.

See Also

- *purealert*
- *purebucket*
- *puredns*
- *purefs*
- *purehw*
- *purenetwork*
- *pureobj*
- *puresmtp*
- *puresubnet*
- *puresupport*
- *puretarget*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

pureaudit

pureaudit, pureaudit-list — displays usage of modification commands (for example commands that create, update, and delete).

Synopsis

`pureaudit list` `[--origin] [--csv|--nvp] [--notitle] [--raw] [--filter` ***`FILTER`***`] [--page] [--limit` ***`LIMIT`***`] [--sort` ***`SORT`***`] [`***`ID...`***`]`

Sub-Commands

`list`

Displays command usage.

Arguments

`ID`

The audit ID.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--origin`

Displays the location, as well as the user interface, from where a command was issued. For example, CLI, GUI, or REST.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient

viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

`--raw`

Displays the unformatted version of column titles and data.

`--filter FILTER`

Displays only the rows that meet the filter criteria specified.

`--limit LIMIT`

Limits the size of the list output to the specified maximum number of rows.

`--page`

Turns on interactive paging.

`--sort SORT`

Sorts the list output in ascending or descending order by the column specified.

Description

The **`pureaudit`** command displays a history of commands issued on the array to make changes to the system. These include create, update, and delete commands such as `purefs create`, `purealert setattr`, and `pureobj account delete`.

Viewing Command History

Use the **`pureaudit list`** command to display all create, update, and delete commands issued on the array.

```
pureaudit list
ID     Time                       User       Command  Subcommand  Arguments
90137  2019-09-30 14:00:29 PDT  pureuser   purefs   create      ro-kb-fs2
90138  2019-09-30 14:00:29 PDT  pureuser   purefs   add         --protocol ...
90139  2019-09-30 14:00:30 PDT  pureuser   purefs   setattr     --size ...
```

The following information is displayed:

- **ID**: The audit ID.
- **Time**: The time at which the command was issued.
- **User**: The user who issued the command.

- **Command**: The base command issued.
- **Subcommand**: The subcommand(s) used with the base command.
- **Arguments**: The options and arguments used with the command.

In addition to the information listed above, issuing the **`pureaudit list`** command with the `--origin` option includes the location (IP address) and the interface information (for example, CLI, GUI, REST) about the issued command.

```
pureaudit list --origin
ID     Time                       User        Location         UI   Command   ...
90137  2019-09-30 14:00:29 PDT  pureuser    10.240.109.248  CLI  purefs    ...
90138  2019-09-30 14:00:29 PDT  pureuser    10.240.109.248  GUI  purealert ...
```

Example 1

```
pureaudit list
```

Displays information about all modification commands issued on the FlashBlade array.

Example 2

```
pureaudit list --origin
```

Displays information about all commands issued on the FlashBlade array, including the location where the command was issued, and the interface from where the command was issued.

See Also

- *purearray*
- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

pureblade

pureblade, pureblade-list — displays blade attributes

Synopsis

```
pureblade list [--csv | --nvp] [--notitle] [--target] [--total] [--sort SORT]
[--limit LIMIT] [--page] [--raw] [BLADE...]
```

Sub-Commands

`list`

Displays blade attributes.

Arguments

BLADE

Blade name.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--total`

Follows output lines with a single line containing column totals in columns where they are meaningful.

Options that control display format:

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient

viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data. The `--raw` output is used to sort and filter list results.

Options that manage display results:

`--filter` ***`FILTER`***

Displays only the rows that meet the filter criteria specified.

`--limit` ***`LIMIT`***

Limits the size of the list output to the specified maximum number of rows.

`--sort` ***`SORT`***

Sorts the list output in ascending or descending order by the column specified.

Description

Viewing blade information

Run the **`pureblade list`** command to list all of the blades on the array and their attributes, including the status of each blade. Each blade in the list output includes the following information:

- **Name**: Blade name.

- **Status**: Blade status. Possible blade statuses include:

  critical

  Component requires immediate attention. Contact Pure Storage Support at *support@purestorage.com*.

  healthy

  Component is performing as expected.

  identifying

  Component is functioning, but not yet initialized.

  unhealthy

  Component is not performing as expected.

  unknown

  Insufficient information to determine a state for this device.

  unused

  Slot is currently empty, as expected.

- **Capacity**: Provisioned size.
- **Details**: Blade creation date.

Include the `--total` option to display the total raw size of all blades.

Example 1

```
pureblade list
```

Displays the attributes of each blade component in the chassis.

Example 2

```
pureblade list CH1.FB15
```

Displays the attributes of blade `CH1.FB15`.

See Also

- *purearray*

- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purebucket

purebucket, purebucket-create, purebucket-destroy, purebucket-disable, purebucket-enable, purebucket-eradicate, purebucket-list, purebucket-monitor, purebucket-recover, purebucket-replica-link — creates, manages, and displays buckets and their contents.

Synopsis

purebucket create --account ***ACCOUNT*** ***BUCKET...***

purebucket destroy [***BUCKET...***]

purebucket disable --versioning ***BUCKET...***

purebucket enable --versioning ***BUCKET...***

purebucket eradicate ***BUCKET...***

purebucket list [--csv|--nvp] [--space][[--total] [--total-only]
[--filter ***FILTER***] [--sort ***SORT***] [--raw] [--limit ***LIMIT***] [--page] [***BUCKET...***]

purebucket monitor [--csv] [--notitle] [--raw] [--filter ***FILTER***]
[--sort ***SORT***] [--limit ***LIMIT***] [--page] --protocol {s3}
[--interval ***INTERVAL***] [--repeat ***REPEAT***] [--size] [--total] [--total-only]
[--historical {1d | 1y | 30d | 3h | 7d}] [***BUCKET...***]

purebucket recover [***BUCKET...***]

purebucket replica-link create --remote-bucket ***REMOTE_BUCKET***
--remote-credentials ***REMOTE_CREDENTIALS*** ***BUCKET***

purebucket replica-link delete --remote ***REMOTE*** --remote-bucket ***REMOTE_BUCKET***
BUCKET

purebucket replica-link list [--cli|--csv|--nvp] [--no-title] [--raw]
[--filter ***FILTER***] [--page] [--limit ***LIMIT***] [--sort ***SORT***] [--status-detail]
[--remote ***REMOTE_NAME...***] [--remote-bucket ***REMOTE_BUCKET...***]
[***LOCAL_BUCKET...***]

purebucket replica-link pause --remote ***REMOTE*** --remote-bucket ***REMOTE_BUCKET***
BUCKET

purebucket replica-link resume --remote ***REMOTE*** --remote-bucket ***REMOTE_BUCKET***
BUCKET

purebucket replica-link setattr --remote ***REMOTE***
--remote-bucket ***REMOTE_BUCKET*** --remote-credentials ***REMOTE_CREDENTIALS*** ***BUCKET***

Sub-Parsers

`replica-link`

When specified, indicates that sub-commands are to be processed for replica links.

Sub-Commands

`create`

Creates buckets and accounts.

When used with the `replica-link` sub-parser, creates a replica link.

`delete`

Deletes a replica link. Must be used with the `replica-link` sub-parser.

`destroy`

Destroys buckets.

`disable`

Disables bucket versioning.

`enable`

Enables bucket versioning.

`eradicate`

Permanently deletes destroyed buckets.

`list`

Displays bucket information.

When used with the `replica-link` sub-parser, lists replica link information.

`monitor`

Displays bucket performance.

`pause`

Pauses a replica replica link. Must be used with the `replica-link` sub-parser.

`recover`

Recovers destroyed buckets.

`resume`

Resumes a replica link. Must be used with the `replica-link` sub-parser.

`setattr`

Updates replica link attributes. Must be used with the `replica-link` sub-parser.

Arguments

BUCKET

Bucket name or names (space-separated list). Filters output by the specified bucket names when used with the **`list`** sub-command.

LOCAL_BUCKET

Bucket name or names (space-separated list). Filters output by the specified local bucket names when used with the **`replica-link list`** sub-command.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--account` ***ACCOUNT***

Specifies the bucket 's account.

`--historical {1d | 1y | 30d | 3h | 7d}`

Displays historical data at the given resolution.

`--interval` ***INTERVAL***

The number of seconds between real-time updates specified as an integer and a multiple of 30. The default is 30 seconds.

`--pending`

Lists buckets including pending eradication.

`--pending-only`

Lists only buckets pending eradication.

`--protocol-specific` ***PROTOCOL***

Adds or removes file system protocol settings when used with the **`purebucket monitor`** command. FlashBlade file systems supports the (case-sensitive) `s3` protocol.

This option is required with the **`purebucket monitor`** command. `PROTOCOL` must be specified as `s3` when used with **`purefs monitor --protocol-specific`**.

`--remote` ***REMOTE***

Specifies the name of the FlashBlade array or S3 target for the replica link.

`--remote-bucket` ***`REMOTE_BUCKET`***

Specifies the remote target bucket to which objects will be replicated.

`--remote-credentials` ***`REMOTE_CREDENTIALS`***

Specifies the remote credentials required for the replica link.

`--repeat` ***`REPEAT`***

The number of times to repeat a real-time update. The default is one.

`--size`

Displays the average I/O sizes per read, write, and all operations.

`--space`

Displays the size and storage consumption details for each bucket.

`--status-detail`

Displays status details.

`--total`

Follows output lines with a single line containing column totals in columns where they are meaningful.

`--total-only`

Displays a single line containing column totals in columns where they are meaningful.

`--versioning`

Enables or disables versioning on one or more buckets.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

`--raw`

Displays the unformatted version of column titles and data.

`--filter FILTER`

Displays only the rows that meet the filter criteria specified.

`--limit LIMIT`

Limits the size of the list output to the specified maximum number of rows.

`--page`

Turns on interactive paging.

`--sort SORT`

Sorts the list output in ascending or descending order by the column specified.

Description

The **`purebucket`** command displays and manages buckets and their contents.

Creating Buckets

To create a bucket, use the **`purebucket create --account BUCKET`** command.

Destroying Buckets

To destroy a bucket, issue the **`purebucket destroy BUCKET`** command. When a bucket is destroyed, it enters a 24-hour eradication pending period. During this time, the bucket and its contents can be recovered. When the 24-hour eradication pending period has lapsed, Purity//FB starts reclaiming storage space. During this time, the bucket can no longer be recovered.

Recovering Buckets

Destroyed buckets have 24 hours to be recovered by using the **`purebucket recover BUCKET`** command. Once 24 hours has expired, the system begins eradicating the bucket, at which point the bucket is no longer recoverable.

Eradicating Buckets

During the eradication pending period, you can manually eradicate the destroyed buckets to reclaim physical storage space by using the **`purebucket eradicate BUCKET`** commnad. Once reclamation starts, the destroyed buckets can no longer be recovered.

Viewing Buckets

Run the **`purebucket list`** command to list buckets on the array and the used space of each. The output includes the following information for each bucket listed:

- **Name**: Bucket name.
- **Account**: The account name.
- **Used**: The amount of used space.
- **Created**: The bucket 's creation date.
- **Versioning**: The bucket 's versioning state: **none**, **`enabled`**, or **`suspended`**.

Include the **`--space`** option to add the following storage consumption details for each bucket:

- **Data Reduction**: Ratio of the size of the written data (i.e. Used) versus the amount of space the data occupies after data compression (i.e. Physical).
- **Physical**: Amount of space that the written data occupies on the array after reduction via data compression.

Bucket Versioning

Bucket versioning allows multiple variants of an object in the same bucket. Buckets can be in three different versioning states: non-versioned, versioning-enabled, and versioning-suspended.

All buckets are initially created in a non-versioned state. Bucket versioning is enabled and managed using the **`purebucket enable --versioning`** and **`purebucket disable --versioning`** CLI commands. Once you enable versioning on a bucket, every object in the bucket automatically retains the entire history on every update, and that bucket can never return to the non-versioned state. However, you can suspend versioning on that bucket.

Issue the **`purebucket list`** command to display versioning status.

In a non-versioned bucket, each object name has only one version. This is always the current version. Modifying an existing object permanently replaces the existing instance of that object. Deleting an existing object name permanently destroys the object.

In a versioning-enabled bucket, an object can have multiple versions for the same object name. Each object name has one current version and either zero or multiple non-current versions. Modifying an

existing object results in that instance of the object name becoming the most current version and the previous version becoming a non-current version.

The versioning-suspended state can prevent the number of versions from growing during modify or delete operations. In the version-suspended state, the modify operation replaces a version introduced during the version-suspended state with a new version, and the delete operation replaces a previous version introduced in a version-suspended state with a delete marker.

Lifecycle rules can also prevent the number of versions from growing. Configuring a lifecycle rule, either for all objects or for objects matching a key name prefix, will cause noncurrent object versions older than the specified number of days to be permanently deleted from the bucket.

Object Replication

Objects can be replicated from a bucket on a FlashBlade (source) to a remote target bucket residing on another FlashBlade or an S3 target using replica links. A replica link is the relationship between a source bucket and target bucket, and describes the properties of that connection. During replication, all objects in the source bucket are replicated to the target bucket.

When replicating objects, object versioning must be enabled on the target bucket.

Enabling object versioning on the source bucket is not required for object replication.

Lifecycle rules can be configured in order to control space usage of noncurrent versions in versioning-enabled buckets.

The following describes how to set up object replication:

1. If you are replicating objects from a FlashBlade array to another FlashBlade array, you must first connect the source and target arrays.

   If you are replicating from a FlashBlade array to an S3 instance, you must first connect the S3 target using the **`puretarget s3 create`** command.

2. Create remote credentials for use with the replica link using the **`pureobj remote-credentials create`** command.

3. Create the replica link by running the **`purebucket replica-link create`** command. When running the command, you must specify the remote credentials created using the `pureobj` command and the target bucket to where the objects will be replicated.

4. Enable versioning on the target bucket by running the **`purebucket enable --versioning`** command. Note that if you have an S3 target, versioning is enabled using the Amazon S3 API.

5. Optionally, configure a lifecycle rule on the target bucket. See *purelifecycle*.

Object replication can be paused at anytime by issuing the **`purebucket replica-link pause`** command to pause the replica link. To resume object replication from the point where it was paused, run the **`purebucket replica-link resume`** command.

Viewing Object Replication

To view the details of your object replication set up, use the **`purebucket replica-link list`** command. The following information is displayed:

- **Name**: The name of the source bucket.
- **Direction**: The direction of replication, outbound (**→**).
- **Remote**: The name of the connected array or S3 target.
- **Remote Bucket**: The name of the remote bucket.
- **Remote Credentials**: The name of the remote credentials in the format `<remote array name or S3 target>/<remote credential name>`.
- **Status**: The current replication status.
  - **replicating**: No errors, currently attempting replication.
  - **paused**: Data transfer is currently halted due to administrator action.
  - **unhealthy**: No connectivity.
- **Lag**: The amount of time, if any, the replication target is behind the source, and is the time difference between the current time and the recovery point.
- **Recovery Point**: A point in time where all object creations, updates, and deletions that occurred after the creation of the replica link and before this point in time are guaranteed to have been replicated. Object creations, updates, and deletions that occurred before the creation of a replica link are not replicated. Those that occurred after the recovery point may or may not have been replicated.

Adding the `--status-detail` option removes the **Remote Credentials** column and replaces the **Lag** and **Recovery Point** columns with a **Details** column that displays an error if the replica link is unhealthy.

Viewing Bucket Performance

The **`purebucket monitor`** command displays real-time and historical I/O performance information for buckets. The maximum number of buckets that can be monitored at one time is 5.

By default, the **`purebucket monitor`** command displays real-time bucket performance data. Include the **`--repeat`** option to specify the number of times to repeat the real-time update. If not specified, the repeat value defaults to 1. Include the **`--interval`** option to specify the number of seconds between each real-time update. At each interval, the system displays a point-in-time snapshot of a bucket performance data. If not specified, the interval value defaults to every 30 seconds. The **`--interval`** and **`--repeat`** options can be combined.

Include the **`--size`** option to display the average performance data.

Include the **`--historical`** option to display historical bucket performance data over any of the following ranges of time: 3 hours, 1 day, 7 days, 30 days, or 1 year.

The output includes the following bucket performance data about bandwidth, IOPS, and latency:

- **Name**: Bucket name.
- **Time**: Point-in-time of the displayed bucket performance data.
- **B/s (read)**: Average number of bytes per read operation.
- **B/s (write)**: Average number of bytes per write operation.
- **op/s (read)**: Average number of read operations processed per second.
- **op/s (write)**: Average number of write operations processed per second.
- **op/s (other)**: Average number of metadata operations processed per second.
- **us/op (read)**: Average time, measured in milliseconds, to process a read request.
- **us/op (write)**: Average time, measured in milliseconds, to process a write request.
- **us/op (other)**: Average time, measured in milliseconds, to process a metadata operation request.
- **B/op (read)**: Average bytes per read operation processed.
- **B/op (write)**: Average bytes per write operation processed.
- **B/op (all)**: Average byte size per read, write, and all operations processed.

Example 1

```
purebucket list
```

Displays each bucket on the array and its details.

Example 2

```
purebucket list --space --total-only
```

Displays a one-line summary of the Used space, Data Reduction, and Physical space for all buckets on the array.

Example output:

```
Name     Account Used     Data Reduction  Physical Object Count
(total)  -       110.74T  1.7 to 1        63.37T   10
```

Example 3

```
purebucket list --pending
```

Displays all buckets, including those buckets pending eradication. If there are buckets pending eradication, the the `Time Remaining` column displays the time until the bucket is eradicated.

Example 4

```
purebucket create --account 1234 bucket1
```

Creates a bucket (`bucket1`) for account 1234.

Example 5

```
purebucket destroy bucket1
```

Destroys `bucket1`. The bucket moves into pending eradication state.

Example 6

```
purebucket recover bucket1
```

Recovers previously destroyed `bucket1` and its data.

Example 7

```
purebucket eradicate bucket1
```

Permanently removes `bucket1` and its data.

Example 8

```
purebucket monitor --interval 120 --repeat 10 bucket1
```

Display `bucket1` performance data 10 times every 120 seconds.

Example 9

```
purebucket monitor --historical 1y bucket1 bucket2
```

Display one years worth of historical performance data for `bucket1` and `bucket2`.

Example 10

```
puretarget s3 create --address 10.1.1.111 s3ds1

pureobj remote-credentials create --access-key-id --user s3ds1/s3-array 123

purebucket replica-link create --remote-credentials array1/s3-array --remote-
bucket r_bucket1 bucket1
```

Sets up the connection between the source FlashBlade, `array1`, to the remote S3 target `s3ds1`. The remote credentials, `s3ds1/s3-array`, are created and used to create the replica link between `bucket1` on `array1` and the remote bucket `r_bucket1` on the remote S3 target.

Example 11

```
purebucket replica-link pause --remote s3ds1 --remote-bucket r_bucket1 bucket1
```

In a replication set up, pauses replication between the local bucket `bucket1` and the remote bucket `r_bucket1` on a remote S3 instance `s3ds1`.

Example 12

```
purebucket replica-link resume --remote s3ds1 --remote-bucket r_bucket1 bucket1
```

Resumes previously paused replication (as shown in the previous example) between the local bucket `bucket1` and the remote bucket `r_bucket1` on the remote S3 target `s3ds1`.

See Also

- *purearray*
- *pureobj*
- *puretarget*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purecert

purecert, purecert-add, purecert-create, purecert-delete, purecert-group-add, purecert-group-create, purecert-group-delete, purecert-group-list, purecert-group-remove, purecert-list, purecert-remove, purecert-setattr — displays and manages FlashBlade SSL and CA certificates

purecert add [--group CERTIFICATE-GROUP ***CERTIFICATE...***]

purecert create [--ca-certificate] [--type ***TYPE***] ***CERTIFICATE***

purecert delete ***CERTIFICATE***

purecert group add [--certificate ***CERTIFICATE***] ***CERTIFICATE-GROUP…***

purecert group create ***CERTIFICATE-GROUP…***

purecert group delete ***CERTIFICATE-GROUP…***

purecert group list [--use] [--cli|--csv|--nvp] [--notitle] [--filter ***FILTER***] [--sort ***SORT***] [--raw] [--limit ***LIMIT***] ***CERTIFICATE-GROUP…***

purecert group remove [--certificate ***CERTIFICATE***] ***CERTIFICATE-GROUP…***

purecert list [--certificate|--intermediate-certificate|--group|--use] [--cli|--csv|--nvp] [--notitle] [--filter ***FILTER***] [--sort ***SORT***] [--raw] [--limit ***LIMIT***] [--page] ***CERTIFICATE…***

purecert remove [--group ***CERTIFICATE-GROUP***] ***CERTIFICATE…***

purecert setattr --certificate [--intermediate-certificate] [--key] [--passphrase] ***CERTIFICATE***

Sub-Parsers

`group`

When specified, indicates that sub-commands are to be processed for certificate groups.

Sub-Commands

`add`

Adds a certificate.

When used with the `group` sub-parser, adds a certificate to a certificate group.

`create`

Creates a certificate.

When used with the `group` sub-parser, creates a certificate group.

`delete`

Deletes a certificate.

When used with the `group` sub-parser, deletes a certificate group.

`list`

Displays certificates and their attributes.

When used with the `group` sub-parser displays certificate groups and their attributes.

`remove`

Removes a certificate.

When used with the `group` sub-parser, removes a certificate from a certificate group.

`setattr`

Modifies a certificate.

Arguments

`CERTIFICATE`

The SSL or CA certificate name.

`CERTIFICATE-GROUP`

The certificate group name.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--ca-certificate`

Import a Certificate Authority (CA) certificate. The FlashBlade array uses a CA certificate to verify the authenticity of configured LDAP servers to establish communication over a TLS (Transport Security Layer) protocol.

`--certificate`

Displays or imports the certificate.

`--intermediate-certificate`

Displays or imports the intermediate certificate chains.

`--group`

Displays the group membership name and associated certificates when used with `purecert list`. Adds a group membership name and certificate when used with `purecert add` and a ***`CERTIFICATE-GROUP`*** name.

`--key`

Private key of the certificate to import.

`--passphrase`

Passphrase used to decrypt the private key.

`--type`

Indicates the type of certificate to create. The types can be `array` or `external`. An `array` certificate type is used by clients to validate the array and requires a private key. For example, the FlashBlade array shows this type of certificate to an S3 client. An `external` certificate type is used when the FlashBlade array acts as a client to validate a server and no private key is required.

`--use`

Displays the how the certificate is being used. Directory services is the only type of usage available at this time.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration. The `--cli` output is not meaningful when combined with immutable attributes.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--filter FILTER`

Displays only the rows that meet the filter criteria specified.

`--limit LIMIT`

Limits the size of the list output to the specified maximum number of rows.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

`--raw`

Displays the unformatted version of column titles and data.

`--sort SORT`

Sorts the list output in ascending or descending order by the column specified.

Description

The **`purecert`** command allows you to perform tasks to manage your certificates used by Purity//FB. Purity creates a self-signed certificate and private key when you start the system for the first time. You can use the default SSL certificate or import a certificate signed by a certificate authority (CA).

LDAP over TLS (Transport Layer Security) can be optionally configured to encrypt communication between the FlashBlade array, acting as an LDAP client, and LDAP servers. In order to establish LDAP over TLS, the LDAP server and FlashBlade must have mutually supported versions of TLS. FlashBlade supports TLS versions 1.0., 1.1, and 1.2. The LDAP server must support at least one of these versions. The highest matching TLS version on the LDAP server and FlashBlade is used.

⚠ **Important note!** The LDAP over TLS configuration does not support NFS with NIS (Network Information Service).

Certificate Group

LDAP over TLS requires that each LDAP server's CA certificate be imported to the FlashBlade where they can be used individually by the FlashBlade or grouped into CA certificate groups to verify the LDAP server's identity and establish communication over TLS. CA certificate groups are used to manage multiple trusted certificates issued to different servers as a directory of certificates. A CA certificate group can contain one or more CA certificates. A CA certificate can be used in multiple CA certificate groups. CA certificates can be added, removed, or deleted without disruption to the services using them as long as the certificates for any configured servers remain in the group. While a CA certificate group is in use, it cannot be deleted, nor can you remove or delete all CA certificates in that group.

⚠ **Important note!** Adding multiple CA certificates in a certificate group which secure the same hostname can produce undefined behavior --there is no guarantee which certificate will be used.

Import a Certificate

To import a new certificate with new attributes, run the **`purecert setattr`** command. Certificate attributes include organization-specific information, such as country, state, locality, organization, organizational unit, common name, and email address.

Certificate Administration

Run **`purecert list`** to list the attributes of current certificate.

To export a certificate or an intermediate certificate, such as for backup purposes, run **`purecert list --certificate`** or **`purecert list --intermediate-certificate`**, respectively.

These columns are displayed when issuing the `purecert list` command:

- **Name**: Name of the certificate or ca certificate.
- **Type**: Certificate type can be `array` or `external`.
- **Status**: It can be self-signed.
- **Key Size**: Bit size of the certificate key.
- **Issued To**: Certificate to whom it was issued.
- **Valid From**: The certificate's creation date.
- **Valid To**: The certificate's expiration date.
- **Country**: Country name. Two-letter ISO code for the country where the organization is located.
- **State/Province**: Full name of the state or province where the organization is located.
- **Locality**: The city where the organization is located.
- **Organization**: Full and exact name in which the organization is legally registered. Organization name should not be abbreviated and should include suffixes such as Inc, Corp, or LLC.
- **Organizational Unit**: Name of the department within the organization that is managing the certificate.
- **Email**: Email address used to contact the organization.

Example 1

```
purecert setattr --certificate --key --intermediate-certificate PURE
```

Imports the intermediate certificate and private key for the certificate named PURE.

Example 2

```
purecert setattr --certificate --key PURE
```

Imports the certificate and private key for the certificate named PURE.

Example 3

```
purecert group create Team_Pure
```

Creates a certificate group named `Team_Pure`. Certificates can be added to the `Team_Pure` certificate group.

Example 4

```
purecert group add --certificate PURE Team_Pure
```

Adds the certificate PURE to the certificate group `Team_Pure`.

See Also

- *pureadmin*
- *purearray*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

pureconfig

pureconfig displays the commands required to reproduce an array 's current object and system configuration

Synopsis

```
pureconfig list [--object|--system]
```

Sub-Commands

`list`

Displays the current configuration.

Options

`--object`

Displays object configuration.

`--system`

Displays system configuration.

Description

Displays an array 's current configuration including array parameters, alert watchers, files systems, network settings, and support configurations in the form of CLI commands that would be required to reproduce the configuration on a newly-installed or unconfigured array. The output does not contain delete or destroy subcommands and does not contain snapshot or other senstive information such as passwords and access keys. Additionally, the output does not contain data that is not createable from the pureuser access level such as special Pure Storage support settings or buckets created from the Pure Storage FlashBlade Object Store REST API.

Run the **`pureconfig list`** command to list all object and system configurations. The output of the command can be as used a script to configure a previously unconfigured array so that it is identical to the array on which the command is executed.

Running the `pureconfig list` command is roughly equivalent to running the following commands in sequence:

```
purefs list --cli
pureobjaccount list --cli
pureobjuser list --cli
pureobjuser access-key list --cli
purealert list --watcher --cli
purearray list --cli
```

```
purearray list --ntpserver --cli
puresmtp list --cli
pureds list --cli
pureds global list --cli
puredns list --cli
purehw connector list --cli
purelag list --cli
puresubnet list --cli
purenetwork list --cli
puresupport list --cli
```

Include the **`--object`** option to list only file system configurations.

Include the **`--system`** option to list only the array, network, alert watcher, and support configurations.

See Also

- *purealert*
- *purearray*
- *puredns*
- *purefs*
- *purenetwork*
- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

puredns

puredns, puredns-list, puredns-setattr — displays and manages the DNS attributes of the array

Synopsis

puredns `list [--cli|--csv|--nvp] [--notitle]`

puredns `setattr [--domain` ***DOMAIN***`] [--nameservers` ***NAMESERVERS***`]`

Sub-Commands

`list`

Displays the array's DNS attributes.

`setattr`

Modifies the array's DNS attributes.

Arguments

None.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--domain` ***DOMAIN***

Domain suffix to be appended by the array when performing DNS lookups. To remove the domain suffix from Purity//FB DNS queries, set to an empty string (`""`) .

`--nameservers` ***NAMESERVERS***

Comma-separated list of up to three DNS server IP addresses. To clear the DNS server IP addresses, set to an empty string (`""`) . Once the DNS server IP addresses have been cleared, Purity//FB stops making DNS queries.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

Description

The **`puredns`** command displays and manages the DNS attributes for an array 's administrative network.

Viewing DNS Attributes

The **`puredns list`** command displays the current DNS parameters, including domain suffix and name servers, of the array. Include the `--cli` option to display the CLI command line that would reproduce the array 's current DNS configuration. This can, for example, be copied and pasted to create an identical DNS configuration in another array, or saved as a backup.

Setting DNS Parameters

The **`puredns setattr`** command sets the DNS parameters of the array. The **`--domain`** option sets the domain suffix to be appended to DNS queries. The **`--nameservers`** option sets the list of DNS name server IP addresses. The new list of DNS name servers replaces any name servers that are currently configured on the array. Purity//FB queries the DNS servers in the order in which the name server IP addresses are listed in this option.

Example 1

```
puredns list
```

Displays the domain suffix and name servers that are configured on the array.

Example 2

```
puredns setattr --domain mydomain.com --nameservers 192.168.0.125,192.168.1.125
```

Sets the domain suffix `mydomain.com` for DNS searches. Also sets the IP addresses of two DNS servers for Purity//FB to use to resolve hostnames to IP addresses.

See Also

- *purearray*
- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

pureds

pureds, pureds-disable, pureds-enable, pureds-list, pureds-role-list, pureds-role-setattr, pureds-setattr, pureds-test — manages FlashBlade integration with a directory service.

pureds-role-list pureds-role-setattr manages FlashBlade directory service roles

Synopsis

pureds disable ***SERVICE-NAME…***

pureds enable ***SERVICE-NAME…***

pureds list [--cli]] [--management-specific] [--nfs-specific]
[--service-specific] [--smb-specific] [--raw] ***[SERVICE-NAME...]***

pureds role list [--cli] [***ROLE…***]

pureds role setattr [--group-base ***GROUP-BASE***] [--group ***GROUP***] ***ROLE...***

pureds setattr [--base-dn ***BASE-DN***] [--bind-password]
[--bind-user ***BIND-USER***] [--ca-certificate ***CA_CERTIFICATE***]
[--ca-certificate-group ***CA_CERTIFICATE_GROUP***] [--join-ou ***JOIN_OU***]
[--uri ***URI***] [--nis-domain ***NIS_DOMAINS***] [--nis-server ***NIS_SERVERS***]
[--user-login-attribute ***USER_LOGIN_ATTRIBUTE***]
[--user-object-class ***USER_OBJECT_CLASS***] ***SERVICE-NAME…***

pureds test [--base-dn ***BASE-DN***] [--bind-password]
[--bind-user ***BIND-USER***] [--ca-certificate ***CA_CERTIFICATE***]
[--ca-certificate-group ***CA_CERTIFICATE_GROUP***] [--join-ou ***JOIN_OU***]
[--uri ***URI***] [--nis-domain ***NIS_DOMAINS***] [--nis-server ***NIS_SERVERS***]
[--filter ***FILTER***] [--sort ***SORT***] [--raw]
[--user-login-attribute ***USER_LOGIN_ATTRIBUTE***]
[--user-object-class ***USER_OBJECT_CLASS***] ***SERVICE-NAME…***

Sub-Parsers

`role`

When specified, indicates that sub-commands are to be processed for directory service roles.

Sub-Commands

`disable`

Disables directory service support.

`enable`

Enables directory service support.

`list`

Displays directory service attributes.

When used with the `role` sub-parser, displays directory service role attributes.

`setattr`

Sets directory service configuration.

When used with the `role` sub-parser, sets directory service role configuration.

`test`

Tests directory service configuration.

Arguments

ROLE

Name of the role. The role can be `readonly`, `storage_admin`, `array_admin`, or `ops_admin`.

SERVICE-NAME

Name of the service. Values can be `management`, `nfs`, or `smb`.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--base-dn` ***BASE-DN***

Base of the Distinguished Name (DN) of the directory service groups. For example, `DC=example,DC=com`.

`--bind-password`

Displays a prompt from which the password of the **`bind-user`** account is entered interactively. The `--bind-user` option is required.

`--bind-user` ***BIND-USER***

Username used to bind to and query the directory.

For Active Directory, enter the username - often referred to as the sAMAccountName or user login name - of the account that is used to perform directory lookups. The username cannot contain the characters `" [] : ; | = + * ? <> / \`, and cannot exceed 320 characters in length.

For OpenLDAP, enter the full DN of the user. For example, `CN=John,OU=Users,DC=example,DC=com`.

`--ca-certificate` ***`CA_CERTIFICATE`***

Name of the CA certificate used to validate server authenticity of configured LDAP servers for communication over TLS (Transport Layer Security).

`--ca-certificate-group` ***`CA_CERTIFICATE_GROUP`***

Name of the CA certificate group containing CA certificates. Using a CA certificate group offers the array administrator the option of using multiple CA certificates to validate server authenticity of multiple configured LDAP servers for communication over TLS (Transport Layer Security).

`--join-ou` ***`JOIN_OU`***

The relative DN of the organizational unit (OU) within your domain where the system machine account is created when joining the domain for SMB. For example, `OU=Arrays,OU=Storage,OU=ServiceMachines`.

`--group` ***`GROUP`***

Specifies the common name (CN) of the directory group being configured.

`--group-base` ***`GROUP-BASE`***

Specifies the organizational unit (OU) path to the configured group.

`--management-specific`

Displays additional directory service attributes that are specific to management.

`--nis-domain` ***`NIS_DOMAINS`***

The NIS Domain. For example, `yp-example-domain`.

`--nis-server` ***`NIS_SERVERS`***

The comma-separated NIS server hostnames to NIS server Fully Distinguished Qualified Names (FDQN) or IP addresses. For example, `server1-example.com,188.121.25.4`.

`--nfs-specific`

Displays additional directory service attributes that are specific to NFS.

`--service-specific`

Lists additional directory service attributes that are specific to individual services.

`--smb-specific`

Displays additional directory service attributes that are specific to SMB.

`--uri` ***`URI-LIST`***

Comma-separated list of up to 30 Uniform Resource Identifiers (URIs) of the directory servers.

For file sharing over SMB, this is optional. If not specified, the base DN (`--base-dn`) of the SMB directory service is used to try to discover domain controllers within an Active Directory domain. If URIs are configured for SMB, they will be treated as the preferences for the servers with which FlashBlade will communicate when joining the domain.

Each URI must include the scheme `ldap://` or `ldaps://` (for LDAP over SSL), a hostname, and a domain name or IP address. For example, `ldap://ad.company.com` configures the directory service with the hostname "ad" in the domain "company.com" while specifying the unencrypted LDAP protocol.

We highly recommend either configuring StartTLS by enabling a certificate or certificate group, or configuring URIs by using the `ldaps://` scheme to use LDAP over SSL, unless there is no need for secure communication in your environment.

If specifying a domain name, it should be resolvable by the configured DNS servers. See *puredns* for more information.

For SMB, only one domain controller (DC) is supported and its preferences cannot be set when configuring the URI.

If specifying an IP address, for IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from `0` to 255 representing a group of 8 bits.

For IPv6, specify the IP address in the form `[xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Enclose the entire address in square brackets (`[]`) . Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`) .

If the scheme of the URIs is `ldaps://`, SSL is enabled. SSL is either enabled or disabled globally, so the scheme of all supplied URIs must be the same. They must also all have the same domain.

If base DN is not configured and a URI is provided, the base DN will automatically default to the domain components of the URIs.

Optionally specify a port. If a port number is specified, append it to the end of the address. Default ports are 389 for ldap, and 636 for ldaps. Non-standard ports can be specified in the URI if they are in use.

`--user-login-attribute` ***`USER_LOGIN_ATTRIBUTE`***

Specifies the user login attribute in your LDAP structure. This is typically the field that holds the user's unique login name. For active directory, this defaults to sAMAccountName. For other server types, this defaults to the uid.

`--user-object-class` ***`USER_OBJECT_CLASS`***

Specifies the object class of a management LDAP user. For active directory, this defaults to User. For OpenLDAP, this defaults to posixAccount or shadowAccount depending on the server's schema. For posix-compliant servers, this defaults to posixAccount. For all other server types, this defaults to person.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--raw`

Displays the unformatted version of column titles and data.

Options that manage display results:

`--filter`

Displays only the rows that meet the filter criteria specified.

`--sort`

Sorts the list output in ascending or descending order by the column specified.

Description

The **`pureds`** command manages the directory service settings of the FlashBlade array.

To export a file system over NFS where a user might belong to more than 16 groups, the FlashBlade array must be integrated with a directory service, such as Active Directory or OpenLDAP. To export a file system over SMB, the FlashBlade array must be integrated with an Active Directory service.

The **`pureds list`** command displays the current base configuration of the directory service. To view service-specific output, issue the **`pureds list`** command with the `--service-specific` option. To view the subsections of the service specific output, use the `--management-specific`, `--nfs-specific`, and `--smb-specific` options.

The **`pureds setattr`** command configures the directory service settings, including the URIs, base DN, OU, bind user, bind password, certificate, certificate group, NIS domain, and NIS server.

For file sharing over SMB, the base DN (`--base-dn`) of the directory service is used in place of the URI (`--uri`) to represent the LDAP URL.

A "bind" account must be configured to allow the array to read the directory. The bind account should be tied to a service account (rather than an individual), and the account password should never expire or be required to change periodically. The account must have read privileges for users and groups and write permissions to add computer objects in Active Directory.

For Active Directory, users with disabled accounts will not be able to access the file systems.

You can optionally specify an OU for your FlashBlade to join within your active directory domain for SMB using the **`pureds setattr --join-ou smb`** command. By default, FlashBlade creates its machine account under `Computers` when joining an active directory domain. By configuring the Join OU, the machine account is created within the configured OU rather than within `Computers`. This allows use of a less privilged service account, without the create and delete permissions in `Computers`, to be used as the bind user in your directory services configuration. This also provides

flexibility in grouping different storage array machine accounts into specific OUs, as well as flexibility with associating security policies to specific OUs. If you wish to clear a previously set Join OU entry, issue the **`pureds setattr --join-ou ""`** command.

Note: If any non-Active Directory servers are configured for SMB, the binding test will fail.

The **`pureds test`** command tests the current directory service configuration. After configuring the directory service settings, test the configuration to verify that the URIs can be resolved and the FlashBlade array can bind and query the tree using the bind user credentials. Make sure the directory server is accessible through a management interface before testing the directory service configuration. For file sharing over SMB, make sure the Active Directory server is accessible through a management interface before testing the directory service configuration.

The **`pureds enable`** command enables the directory service, allowing users in the directory to access the file systems. Enable the directory service after it has passed the directory service test.

The **`pureds disable`** command disables the directory service. This will stop any users in the directory from accessing the file system.

Directory Services

The FlashBlade is delivered with a single local user, named `pureuser`, with array -wide (Array Admin) permissions. Users can be added to the array through Lightweight Directory Access Protocol (LDAP) by integrating the array with a directory service such as Active Directory or Open LDAP.

To integrate the FlashBlade array with a directory service,

1. Configure the FlashBlade directory service.
2. Test the directory service configuration.
3. Enable the directory service.

When configuring the directory service for array management or SMB or NFS protocol integration with LDAP, specify the URIs, base DN, and bind username and password of the directory service. Note that for file sharing over SMB, the base DN of the directory service is used to represent the domain that is joined, and the URIs, if specified, control what servers are communicated with when joining the domain.

⚠ **Important note!** Anonymous binding is supported for NFS and Management. If configuring anonymous binding for NFS or Management, neither the bind user nor the bind password should be specified.

A "bind" account must be configured to allow the array to read the directory. The bind account should be tied to a service account (rather than an individual), and the account password should never expire or be required to change periodically. The account must have read privileges for users and groups. Bind account privileges should be configured at the Organizational Unit.

For Active Directory, users with disabled accounts are not able to access the file systems.

After the directory service has been configured, test the configuration to verify that the URIs can be resolved and the FlashBlade array can bind and query the tree using the bind user credentials.

Enable the directory service after it has passed the directory service test. The directory service can be disabled at any time. Disabling a directory service halts all features that rely on it. In the case of NFS, effects include netgroups ceasing to function, users not receive extended group permissions, etc. In the case of SMB, users will be unable to access the file system. In the case of management, disabling directory service prevents users from logging into the array.

Configuring the Directory Service for Protocol Support

To export a file system over NFS where a user might belong to more than 16 groups, the FlashBlade array must be integrated with a directory service such as Active Directory, OpenLDAP, or Network Information Service (NIS). If configuring the NIS directory service for NFS, clear any existing LDAP configurations before specifying the NIS servers and NIS domain. Note that NIS is only supported for NFS. To export a file system over SMB, the FlashBlade array must be integrated with an Active Directory service. The SMB account must have read privileges for users and groups, including write permissions to add computer objects in Active Directory.

Configuring the directory service for SMB or NFS is a one-time process and is only required if you are integrating the FlashBlade array with a directory service for protocol support.

Before you configure the directory service, verify that the DNS settings can be used to resolve the directory server domain and server (`puredns list`) . For file sharing over SMB, also verify that the FlashBlade array can access the directory service through a management interface (`purenetwork list`) .

For example, run the following commands to configure the directory service for file sharing over SMB:

```
pureds setattr --base-dn DC=mycompany,DC=com
               --bind-user CN=John,OU=Users,DC=mycompany,DC=com
               --bind-password
pureds test smb
pureds enable smb
```

Configuring Roles

For LDAP users, role-based access control (RBAC) is achieved by configuring groups in the directory that correspond to the following permission groups (roles) on the array.

Role	Description
readonly	Read Only users have read-only privileges to run commands that convey the state of the array. Read Only users cannot alter the state of the array

Role	Description
storage_admin	Storage Admin users have all the privileges of Read Only users, plus the ability to run commands related to storage operations, such as administering file systems and snapshots, as well as object store accounts, users, and access keys. Storage Admin users cannot perform operations that deal with global and system configurations.
array_admin	Array Admin users can perform all FlashBlade operations.
ops_admin	Ops Admin users have all the privileges of Read Only users, plus the ability to run commands related to support operations, such as managing remote assist sessions, phone home, and proxy settings. Ops Admin users cannot perform operations that deal with storage or non-support global and system configurations.

When a user connects to the FlashBlade, the array confirms the user's identity from the directory service. The response from the directory service includes the user's group, which Purity//FB maps to a role on the array, granting access accordingly.

If users are assigned to more than one role, an alert is issued. For security purposes, a user belonging to more than one FlashBlade role will inherit the permissions of the most restrictive role. For example, a user who has been assigned both the Storage Admin and Read Only roles will inherit the privileges of the more restrictive Read Only role. Note that the user will continue to have privileges from the original role until the directory service role cache is cleared (cache is automatically cleared every 8 hours). To make new permissions take immediate effect, run the **`pureadmin refresh`** command.

One exception to this rule is users assigned the Ops Admin role—users assigned the Ops Admin role and any other role are denied the ability to log into the FlashBlade. If there is an error with the directory service configuration on the FlashBlade array, contact a user with Array Admin privileges to resolve the issue. If the LDAP server is configured incorrectly, contact your LDAP server administrator. The alert issued in the case where a user is denied access due to being assigned the Ops Admin and another role, provides additional information to help resolve the issue.

As a best practice, after changing a user's role or management directory service configuration settings, use the **`pureadmin refresh`** command to refresh the user's role or clear directory service role cache to ensure all user privileges are up to date.

For directory service-enabled accounts, user passwords to the FlashBlade are managed through the directory service while API tokens are configured through Purity//FB. The password can be a maximum of 100 characters in length and include any character that is entered from a US keyboard.

Unauthorized CLI commands per the user's configured role will not appear.

Example 1

```
pureds setattr --uri ldaps://[2001:0db8:85a3::ae26:8a2e:0370:7334]
```

```
               --base-dn DC=mycompany,DC=com --bind-username ldapreader smb
```

Sets the URI to IPv6 address `2001:0db8:85a3::ae26:8a2e:0370:7334` and the scheme to `ldaps://` to enable SSL. Also sets the base DN to be the correct Domain Components and sets the bind username as the username used to bind to and query the directory.

Example 2

```
pureds setattr --uri ldaps://ad1.mycompany.com,ldaps://ad2.mycompany.com
               --base-dn DC=mycompany,DC=com
               --bind-user CN=John,OU=Users,DC=mycompany,DC=com smb
```

Sets the URI to both `ad1.mycompany.com` and `ad2.mycompany.com` and the scheme to `ldaps://` to enable SSL. Also sets the base DN to be the correct Domain Components and sets the bind username as the username used to bind to and query the directory.

Example 3

```
pureds setattr --nis-domain yp-example-domain --nis-server yp-
example.com,188.121.25.4 nfs
Name  Join OU  NIS Domain            NIS Server
nfs   -        yp-example-domain     yp-example.com
                                     188.121.25.4
```

Sets the NIS domain to `yp-example-domain` and NIS servers to `yp-example.com` and `188.121.25.4` for the NFS protocol.

Example 4

```
pureds list --nfs-specific
Name        NIS Domain  NIS Server
management  -           -
nfs         -           -
smb         -           -
```

Displays the NFS configuration.

Example 5

```
pureds setattr --bind-user CN=John,OU=Users,DC=mycompany,DC=com --bind-password
 smb
Enter bind password:
Retype bind password:
```

Displays the interactive prompt for entering a password for the bind user account. The password is not shown while typing, so a confirmation prompt is presented. If the passwords do not match, no change is made.

Example 6

```
pureds test management
Name         Enabled  Component Name  Component Address  Destination          …
management   True     fm1             10.64.240.191      ldap://ir-ldap…
                      fm2             10.64.240.192      ldap://ir-ldap…
```

Displays the output of testing the current management configuration.

Example 7

```
pureds role list array_admin readonly
Name           Group          Group Base
array_admin    pureadmins     OU=posix
readonly       readergroup    CN=nextlevel_p,OU=posix
```

Displays the associated group and group base for roles `array_admin` and `readonly`.

Example 8

```
pureds role setattr array_admin --group pureadmins --group-base OU=posix
Name           Group          Group Base
array_admin    pureadmins     OU=posix
```

Sets the role `array_admin` as belonging to the group `pureadmins` in the group base `OU=posix`.

Example 9

```
pureds setattr --join-ou OU=machines,OU=lowsecurity,OU=storage,OU=service smb
Name         Join OU
smb          OU=machines,OU=lowsecurity,OU=storage,OU=service
```

Sets the join OU parameters for the active directory domain for SMB.

Example 10

```
pureds setattr nfs --ca-certificate MyCert
```

Configures the nfs directory service to trust servers with certificates that were signed by the CA certificate named MyCert, which was uploaded to FlashBlade, when establishing TLS communication.

See Also

- *pureadmin*
- *puredns*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purefs

purefs, purefs-add, purefs-cancel, pureds-copy, purefs-create, purefs-demote, purefs-destroy, purefs-disable, purefs-enable, purefs-eradicate, purefs-list, purefs-monitor, purefs-nfs, purefs-promote, purefs-recover, purefs-remove, purefs-replica-link, purefs-setattr, purefs-smb, purefs-snap — displays and manages the file systems and snapshots.

Synopsis

purefs add (--protocol {http,nfsv3,nfsv4.1,smb}|--policy ***POLICY***)
[--remote ***REMOTE***] ***NAME...***

purefs cancel --transfer [--remote ***REMOTE***] ***FILE-SYSTEM-SNAPSHOT...***

purefs copy [--overwrite] [--discard-non-snapshotted-data ***SOURCE***] ***TARGET...***

purefs create [--size ***SIZE***] ***FILE-SYSTEM...***

purefs demote [--discard-non-snapshotted-data] ***FILE-SYSTEM...***

purefs destroy [-delete-link-on-eradication] ***FILE-SYSTEM...***

purefs disable [--fast-remove-dir] [--snapshot-dir] [--hard-limit]
[--ignore-usage] [--safeguard-acls] [--writable] ***FILE-SYSTEM...***

purefs enable [--fast-remove-dir] [--snapshot-dir] [--hard-limit]
[--ignore-usage] [--safeguard-acls] [--writable] ***FILE-SYSTEM...***

purefs eradicate ***FILE-SYSTEM...***

purefs list [--pending|--pending-only] [--space|
--protocol-specific {http,nfs,smb}|--multi-protocol|--special-dir|
--snap|--quota] [--cli|--csv|--nvp] [--notitle] [--total|--total-only]
[--filter ***FILTER***] [--sort ***SORT***] [--raw] [--limit ***LIMIT***] [--page] [--policy]
[--transfer] [***FILE-SYSTEM...***]

purefs monitor [--csv] [--notitle] [--raw] [--filter ***FILTER***] [--sort ***SORT***]
[--limit ***LIMIT***] [--page] [--interval ***INTERVAL***] [--repeat ***REPEAT***] [--size]
[--total] [--total-only] [--historical ***HISTORICAL***] [***FILE-SYSTEM...***]

purefs nfs setattr [--add-rules ***ADD_RULES***] [--after ***AFTER***]
[--remove-rules ***REMOVE_RULES***] [--rules ***RULES***] ***FILE-SYSTEM...***

purefs promote ***FILE-SYSTEM...***

purefs recover ***FILE-SYSTEM...***

purefs remove [--protocol ***PROTOCOL***|--policy ***POLICY***] [--remote ***REMOTE***] ***NAME...***

```
purefs replica-link create --remote REMOTE
[--policy POLICY][--remote-fs REMOTE_FILE_SYSTEM] LOCAL_FILE-SYSTEM...

purefs replica-link list [--cli|--csv|--nvp] [--notitle] [--filter FILTER]
[--status-detail] [--sort SORT] [--raw] [--remote REMOTE] [--limit LIMIT]
[--page] [LOCAL_FILE_SYSTEM...]

purefs setattr [--ignore-usage] [--default-group-quota DEFAULT_GROUP_QUOTA]
[--default-user-quota DEFAULT_USER_QUOTA] [--discard-detailed-persmissions]
[--size SIZE]
[--access-control-style {independent,mode-bits,nfs,smb,shared}]
FILE-SYSTEM...

purefs snap [--send] [--suffix SUFFIX] [--target TARGET] FILE-SYSTEM...
```

Sub-Parsers

`nfs`

When specified, indicates that sub-commands are to be processed for NFS.

`replica-link`

When specified, indicates that sub-commands are to be processed for replica links.

Sub-Commands

`add`

Adds protocols to file systems, or adds policies to files systems or snapshots.

`cancel`

Cancels snapshot replication.

`copy`

Copies a snapshot to one or more file systems.

`create`

Creates a file system.

When used with the `replica-link` sub-parser, creates file sytem replica links.

`demote`

Demotes file systems.

`destroy`

Destroys file systems or snapshots.

`disable`

Disables file system attributes.

`enable`

Enables file system attributes.

`eradicate`

Permanently deletes destroyed file systems or snapshots.

`list`

Displays file systems.

When used with the `replica-link` sub-parser, displays file system replica links.

`monitor`

Displays file system performance information.

`promote`

Promotes file systems.

`recover`

Recovers previously destroyed file systems or snapshots.

`remove`

Removes protocols from file systems, or removes plicies from file systems or snapshots.

`setattr`

Modifies file systems.

When used with the `nfs` sub-parser, modifies NFS attributes.

`snap`

Creates a snapshot of a file system.

Arguments

`FILE-SYSTEM`

The file system name.

`FILE-SYSTEM-SNAPSHOT`

The file system snapshot name.

LOCAL-FILE-SYSTEM

The local file system name.

NAME

The file system or snapshot name.

SOURCE

The source file system snapshot name.

TARGET

The target file system name(s).

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--access-control-style` ***independent,mode-bits,nfs,smb,shared***

Selects the access control style for client actions such as setting file and directory ACLs. Valid values are `shared`, `nfs`, `smb`, `independent`, or `mode-bits`.

Access control style interactions are as follows:

- `shared` - NFS and SMB clients are both able to set permissions on, and access files and directories over any protocol.
- `nfs` - Only NFS clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- `smb` - Only SMB clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- `independent` - NFS and SMB clients are both able to set permissions on files and directories and can access files and directories created over any protocol. Permissions set by SMB clients will not affect NFS clients and vice versa. NFS clients will be restricted to only using mode bits to set permissions.
- `mode-bits` - NFS and SMB clients are both able to set permissions on files and directories, but permissions are enforced in mode bits-only format. When SMB clients set an ACL it will be converted to mode bits.

`--add-rules` ***ADD_RULES***

Specifies the rules that will be added to the existing Network File System (NFS) export rules for the file system.

`--after` ***AFTER***

This option can be used with the `--add-rules` and `--remove-rules` options.

If used with the `--add-rules` option, the rules specified in the `--add-rules` string will be added after the first rule specified with the `--after` string.

If used with the `--remove-rules` option, the rules specifed in the first occurrence of the `--remove-rules` string will be removed after the first rule specified with the `--after` string.

If used with both the `--add-rules` and `--remove-rules` options, the `--remove-rules` string will be processed before the `--add-rules` string.

`--delete-link-on-eradication`

Destroys the file system with links.

`--discard-detailed-permissions`

Discards more granular access control lists set within the file system. This option is required when changing to an access control style where permissions will only be enforced at the granularity of NFS mode bits (changing from `nfs`, `shared`, or `smb` to `independent` or `mode-bits`).

`--discard-non-snapshotted-data`

Discards data since the last snapshot.

`--fast-remove-dir`

Enables or disables the `.fast-remove` directory for one or more file system(s).

`--hard-limit`

Enables or disables the hard limit on the provisioned size of a file system.

`--historical HISTORICAL`

Displays historical data at the given resolution in the format of `N [m | h | d | w]`, for example 15m, 2h, 1d, etc.

`--interval`

The number of seconds between real-time updates specified as an integer and a multiple of 30. The default is 30 seconds.

`--ignore-usage`

Ignores the current file system usage when setting the file system 's virtual size. When specified with the `enable` subcommand, the - `-hard-limit` option is required. When specified with the `setattr` subcommand, the `--size` option is required.

`--default-group-quota` ***DEFAULT_GROUP_QUOTA***

Sets the default quota size for all groups in the file system. DEFAULT_GROUP_QUOTA must be an integer followed by a capacity suffix: K,M,G,T,P, or E.

`--default-user-quota` ***`DEFAULT_USER_QUOTA`***

Sets the default quota size for all users in the file system. DEFAULT_USER_QUOTA must be an integer followed by a capacity suffix: K,M,G,T,P, or E.

`--multi-protocol`

Displays attributes pertaining to the interactions of different protocols.

The following behaviors occur for the access control style as set with the `--access-control-style` option:

- `shared` - NFS and SMB clients are both able to set permissions on, and access files and directories over any protocol.
- `nfs` - Only NFS clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- `smb` - Only SMB clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- `independent` - NFS and SMB clients are both able to set permissions on files and directories and can access files and directories created over any protocol. Permissions set by SMB clients will not affect NFS clients and vice versa. NFS clients will be restricted to only using mode bits to set permissions.
- `mode-bits` - NFS and SMB clients are both able to set permissions on files and directories, but permissions are enforced in mode bits-only format. When SMB clients set an ACL it will be converted to mode bits.

`--overwrite`

Allows an existing file system to be overwritten.

`--policy` ***`POLICY`***

Adds or removes a policy or policies from file systems or adds a policy to a replica link. Enter multiple policies as a comma-separated list.

`--protocol` ***`PROTOCOL`***

Adds or removes file system protocol settings when used with the **`purefs add`** or **`purefs remove`** command. FlashBlade file systems support the (case-sensitive) `http`, `nfsv3`, `nfsv4.1`, and `smb` protocols. Enter multiple protocols as comma-separated values. Adding `http` enables both `http` and `https`.

This option is required with the **`purefs monitor`** command. `PROTOCOL` must be specified as `nfs` (`nfs` is used to represent both NFSv3 and NFSv4.1) when used with **`purefs monitor`**.

`--protocol-specific` ***`http,nfs,smb`***

Displays protocol-specific attributes. FlashBlade file systems support the (case-sensitive) `http`, `nfs`, and `smb` protocols. Note that `nfs` is specified to view attributes for both NFSv3 and NFSv4.1.

`--remote` ***REMOTE***

Specifies the remote array name.

`--remote-fs` ***REMOTE_FILE_SYSTEM***

Specifies the remote file system name when creating a replica link. If this option is not used, a default name is (same as the local file system name) is used when creating a replica link.

`--remove-rules` ***REMOVE_RULES***

Specifies the rules that will be removed from the existing Network File System (NFS) export rules for the file system. Only the first occurrence of the `--remove-rules` option will be removed.

`--rules` ***RULES***

Defines the Network File System (NFS) export rules for the given file system, completely replacing the default set of rules. Rules can be applied to an individual client or a range of clients. Valid export options are `ro`, `rw`, `fileid_32bit`, `no_fileid_32bit`, `anonuid`, `anongid`, `root_squash`, `no_root_squash`, `all_squash`, `no_all_squash`, `secure`, `insecure`, `atime`, and `noatime`.

.

By default, the export options are: `ro,root_squash`.

⚠ **Important note!** To prevent the shell from interpreting the following options as wildcards and other globs, and because rule strings can contain spaces, RULES should always be protected with single quotes. For example:

```
--rules '*(rw) -ro 10.20.255.1(no_root_squash)'
```

The examples below are partial rule strings, so we omit the quotation marks.

Client specifications have multiple formats, where d denotes a base 10 decimal digit, and x denotes a hexadecimal number:

- IPv4 IP address: `ddd.ddd.ddd.ddd`
  IPv4 Netmask: `ddd.ddd.ddd.ddd/dd`
- IPv6 IP address: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`
  IPv6 Netmask: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/ddd`
- Represent all clients with an asterisk: *

To apply options to a single client specification, append a comma-separated list of options within parentheses. For example:

```
10.20.255.2(rw,no_root_squash)
```

```
*(rw,no_root_squash)
```

To apply rules to multiple client specifications, include a dash (-) followed by a comma-separated list of options: `-options host`.... For example,

```
-rw,no_root_squash 10.20.255.1 10.20.255.2 -ro 10.20.255.3
```

Options specified in this manner apply to all client specifications that appear to their right in the rule string, so the first two IP addresses in the example will get read-write, no-root-squash access. The `-ro` does not modify root-squash options, so 10.20.255.3's permissions will be `ro,no_root_squash`.

`--safeguard-acls`

Available if the applied access control style is `nfs`, `smb`, or `shared`. Enables ACL safeguarding for one or more file systems. Controls how a mode bit change impacts the ACL on the file or directory. When enabled, modification of mode bits will result in the modification of the existing ACL to merge the permissions from the mode bits with permissions in the ACL. If disabled, modification of mode bits results in the removal of any existing ACL.

- If the access control style was set to `smb`, ACL safeguarding is automatically enabled. Disabling ACL safeguarding is not allowed.
- ACL safeguarding is not available for `independent` or `mode-bits` access control styles.

`--size` ***SIZE***

Sets the provisioned size of the file system. If not set, the provisioned size is unlimited. SIZE must be an integer followed by a capacity suffix: K,M,G,T,P, or E.

`--send`

Manually replicates the snapshot to the target.

`--snap`

Lists file system snapshots.

`--snapshot-dir`

Enables or disables the `.snapshot` directory for one or more file system(s).

`--space`

Displays size and storage consumption details for each file system.

`--special-dir`

Displays the fast remove and snapshot directory status for each file system.

`--status-detail`

Displays status details.

`--suffix` ***SUFFIX***

Assigns a suffix name to a file system snapshot. If not set, a snapshot suffix name will be assigned.

`--repeat` ***REPEAT***

The number of times to repeat a real-time update. The default is one.

`--target` ***TARGET***

Specifies the replication target.

`--total`

Follows output lines with a single line containing column totals in columns where they are meaningful.

`--total-only`

Displays a single line containing column totals in columns where they are meaningful.

`--transfer`

When used with the `cancel` subcommand, cancels snapshot transfer between arrays. When used with the `list` subcommand and `--snap` option, displays transfer records.

`--writable`

When used with the `disable` subcommand, makes the file system read only. When used with the `enable` subcommand, makes the file system writable.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data. The `--raw` output is used to sort and filter list results.

Options that manage display results:

`--filter` ***FILTER***

Displays only the rows that meet the filter criteria specified.

`--limit` ***LIMIT***

Limits the size of the list output to the specified maximum number of rows.

`--pending`

Displays all file systems including destroyed file systems in the eradication pending state awaiting to be eradicated. Include the **`--snap`** option to display snapshots pending eradication.

`--pending-only`

Displays only destroyed file systems in the eradicate pending state awaiting to be eradicated. Include the **`--snap`** option to display snapshots pending eradication.

`--policy`

Lists policies added to file systems or snapshots.

`--sort` ***SORT***

Sorts the list output in ascending or descending order by the column specified.

`--quota`

Displays the default quota.

For more information on how to use the CLI options to control display formatting and manage display results, refer to *pureman*

Description

The **`purefs`** command displays and manages the file systems for export.

Viewing File Systems

The **`purefs list`** command lists all of the file systems that have been created on the FlashBlade array and the export rules that accompany each file system. Each file system in the list output includes the following information:

- **Name**: File system name.
- **Size**: Provisioned size of the file system. If not set, the provisioned size is unlimited.
- **Used**: Total amount of pre-compressed data written to the file system.
- **Created**: File system creation date.

- **Protocols**: One or more protocols configured for the file system. A dash (-) means the file system has no protocols assigned.

Include the **`--snap`** option to display the size and storage consumption details for each file system snapshot:

- **Name**: File system snapshot name.
- **Source**: Snapshot created from the source file system.
- **Created**: File system snapshot creation date.

Include the **`--space`** option to display the following size and storage consumption details for each file system:

- **Name**: File system name.
- **Size**: Provisioned size of the file system. If not set, the provisioned size is unlimited.
- **Used**: Total amount of data written to the file system.
- **Data Reduction**: Ratio of the size of the written data (i.e. Used) versus the amount of space the data occupies after data compression (i.e. Physical).
- **Physical**: Amount of space that the written data occupies on the array after reduction via data compression.
- **Snapshots**: Amount of space written to the snapshots of the file system.
- **Total**: Total combined physical and snapshot space used.

Include the **`--space`** and **`--raw`** options to display output in bytes:

- **name**: File system name.
- **provisioned**: The usable amount of space assigned to a file system. If not set, the provisioned size is unlimited.
- **space.virtual**: The space written to the array that can be read, including file systems, buckets, and snapshots.
- **space.data_reduction**: Ratio of the size of the written data (i.e. Used) versus the amount of space the data occupies after data compression (i.e. Physical).
- **space.unique**: Amount of space used by file systems and buckets but does not include snapshots.
- **space.snapshots**: Amount of space written to the snapshots of the file system.
- **space.total_physical**: Total amount of space used by a file system.

Include the **`--total`** option to display the `purefs list` output plus the totals for the following columns of the output:

- **Name**: File system name.
- **Size**: Provisioned size of the file system. If not set, the provisioned size is unlimited.
- **Used**: Total amount of pre-compressed data written to all file systems.
- **Created**: File system creation date.
- **Protocols**: One or more protocols configured for the file system. A dash (-) means the file system has no protocols assigned.

Include the **`--pending`** option to display all file systems and eradication pending file systems.

Include the **`--pending-only`** option to only display eradication pending file systems.

Include the **`--special-dir`** option to display the status of the `.fast-remove` and `.snapshot` directories for each file system. Directory status is displayed as `True` for enabled and `False` for disabled.

Include the **`--total-only`** option to display a single line containing column totals for the `purefs list` output.

Include the **`--protocol-specific`** option to display only the file systems with the specified protocol enabled. For example, run the `purefs list --protocol-specific nfs` command to display only the file systems with the NFS protocol enabled.

With the ability to create and manage hundreds of file systems, list outputs can become very long. The Purity//FB CLI includes options to control the way a list output is displayed and formatted. The CLI also includes sort, filter, and limit options to control the results you see and the order in which you see them. For information about CLI display and format options, refer to *pureman*.

About Creating and Exporting File Systems

Creating and exporting a file system involves creating the file system and enabling protocol support for the file system. FlashBlade file systems support the Hypertext Transfer Protocol (HTTP), the Network File system (NFS), and Server Message Block (SMB) protocols.

Note: Exporting a file system requires at least one data virtual IP address (data vip). The data vip can be created before or after creating the file system. Data vips are managed through the `purenetwork` command. For more information about data vips, refer to *purenetwork*.

When creating a file system, consider which protocol(s) you want to configure:

- **NFS**. File sharing over NFSv3 or NFSv4.1 requires that you configure the NFS export rules and then enable the protocol. Per NFS limitations, each user can belong to a maximum of 16 groups. If users

belong to more than 16 groups and a directory service contains the group information, configure file sharing over NFS with a directory service.

Follow these steps to set up file sharing over NFS without a directory service:

1. Create the data vip. Data vips are managed through the `purenetwork` command.
2. Create the file system. Optionally set a provisioned size for the file system.
3. Define the NFS export rules.
4. Enable the NFS protocol.

- **NFS with a directory service**. For file sharing over NFS where a user might belong to more than 16 groups, configure the FlashBlade directory service to integrate with a directory server. This assumes that you already have a directory service, such as OpenLDAP or Active Directory set up.

  Follow these steps to set up file sharing over NFS with a directory service:

  1. Configure the directory service for protocol support. The directory service is managed through the `pureds` command. For more information about the directory service, refer to *pureds*.
  2. Create the data vip. Data vips are managed through the `purenetwork` command.
  3. Create the file system. Optionally set a provisioned size for the file system.
  4. Define the NFS export rules.
  5. Enable the NFS protocol.

- **SMB with Active Directory**. FlashBlade offers the following SMB authentication modes:

  - **AD RFC2307.** Recommended for customers where the same data needs to be accessed over SMB and NFS, and where access control is required. For more information, refer to the RFC 2307 specifications set by the Internet Engineering Task Force.
  - **AD Auto.** Recommended for customers with SMB-only clients or when SMB and NFS clients never access the same file systems.

  The default SMB authentication mode is AD RFC2307. To configure file sharing over SMB in other authentication modes, such as AD Auto mode, contact Pure Storage Support.

  Before you enable the SMB protocol adapter, consider the following limitations:

  - Due to the nature of the SMB protocol, disruptions occur when a blade is added or removed, an upgrade is performed, or the system encounters network connectivity issues. Therefore, your applications must be able to tolerate I/O errors and disruptions.
  - File system permissions default to traditional UNIX-like permissions with an option to select the `native` ACL mode to include support for Windows ACLs. The SMB Access Control List (ACL) is

restricted to the following three entries: the file **owner**, the primary **group** to which the file owner belongs, and **others**.

- For file sharing over SMB, the FlashBlade array must be integrated with an Active Directory service. Before creating the file system, configure the directory service. For more information about the directory service, refer to *pureds*.

For more information about these and other limitations related to the SMB protocol, refer to the **FlashBlade SMB - Limitations in Purity//FB 2.x** KB article at: *https://support.purestorage.com/FlashBlade/Purity////FB*

Follow these steps to set up file sharing over SMB in AD RFC2307 mode:

1. Prepare Active Directory for multi-protocol support by setting the `gidNumber` and `uidNumber` attributes for each domain user and group that will be accessing the SMB shares. The `gidNumber` and `uidNumber` attributes can be set through the Microsoft Identity Management for UNIX service or PowerShell.
2. Configure the directory service for protocol support. The FlashBlade array must be integrated with an Active Directory service. The directory service is managed through the `pureds` command. For more information about the directory service, refer to *pureds*.
3. Create the data vip.
   Data vips are managed through the `purenetwork` command.
4. Create the file system. Optionally set a provisioned size for the file system.
5. Enable the embedded SMB protocol adapter.

- **HTTP access**. Enabling the HTTP protocol turns on HTTP and HTTPS access to a file system. Before you enable the HTTP protocol, consider the following access control limitations:
  - All HTTP APIs by default are executed on behalf of a root user.
  - Certain HTTP APIs provide the ability to specify a user by specifying the UID as part of the request.
  - Access checking in HTTP is based solely on the client's crafted request. Any user can read, modify, and delete any files on a file system when HTTP is enabled.
  - The use of HTTP connections, as opposed to HTTPS connections, potentially gives unauthenticated users access to the file system. Anyone with network access to the data vip has access to the file system.

  Follow these steps to set up file sharing over HTTP access:

  1. Create the data vip. Data vips are managed through the **Settings** > **Network** page.
  2. Create the file system. Optionally set a provisioned size for the file system.
  3. Enable the HTTP protocol.

  ⚠ **Important note!** The use of HTTP connections, as opposed to HTTPS connections, potentially gives unauthenticated users access to the file system. Anyone with network access to the data vip has access to the file system.

NFS and File Locking

Enabling the NFS protocol enables the Network Lock Manager (NLM) protocol. The NLM protocol works with the NFS protocol to ensure file locks are visible across all NFS clients for I/O coordination and to help clients coordinate access to files. The NLM protocol allows all NFS clients mounting the same NFS shared file system to see file locks set by other clients.

NLM locks are considered advisory locks in that an NFS client application must check for the existence of a lock in order to coordinate access; the NFS service itself does not automatically prevent a client from accessing a file if it has the security permission to do so, and the service does not check for the existence of or try to obtain a lock on a file ahead of time.

The NLM service is enabled by default with the NFS protocol service and cannot be disabled.

Creating File Systems

The **`purefs create`** command creates file systems as distributed file shares for export.

The `--size` option represents the provisioned size allocated to a file system. The size is a quota of space that is set to help gauge the fullness of a file system.

When the amount of data written to the file system reaches a certain percentage of its quota, alerts are generated notifying administrators of the threshold reached. For example, Purity//FB generates a `WARNING` alert when the amount of data written to the file system reaches 90% and then 100% of its provisioned size. This behavior differs slightly if a hard limit is set on the file system. When the file system usage reaches 90% of its provisioned size, the system generates a `WARNING` alert. When the file system usage reaches 100% of its provisioned size, the system generates a `CRITICAL` alert.

The file system size can be blank or set to any value between `0` and `8191` petabytes. If the field is not set or set to `0`, the provisioned size will default to an unlimited size.

Purity//FB does not enforce any restrictions nor does it take any further action when space consumption reaches or exceeds the provisioned size. To address the issue, decrease the percentage of storage space used by either deleting files or increasing the size of the file system. In contrast, if a hard limit is set for the file system, all writes are halted once the provisioned size is reached.

After a file system has been created, add protocols to the file system to make it accessible to the data vips. By default, a newly created file system has no protocols assigned.

Creating File System Snapshots

File system snapshots are immutable, point-in-time images of the contents of one or more file systems. Snapshots can be taken of a file system at any time using the **`purefs snap`** command. Snapshots are named after the file system it is copying with an assigned suffix to differentiate it from the source file system. A suffix is automatically assigned to a snapshot if one is not given at the time of creation. Use the **`--suffix`** option to assign a specific suffix to a snapshot. Suffix names must be between 1 and 63 characters in length (alphanumeric and the '-' character) and must begin and end with a letter or number. The suffix must include at least one letter.

In the following example, a snapshot is created for file system `MyFiles` and the suffix `snap01` is added:

```
purefs snap --suffix snap01 MyFiles
Name             Source     Created                  Policy
MyFiles.snap01   MyFiles    2017-06-28 09:43:34 PDT -
```

Defining NFS Export Rules

The **`purefs nfs`** command changes the NFS attributes of a file system.

The NFS export rules define the access rights and privileges that an NFS client has to the file system. Include the **`--rules`** option to define the NFS export rules for a file system, completely replacing the existing set of rules. Any NFS export rule changes will be synchronously applied to all currently mounted clients. Note that NFS rules apply to both NFSv3 and NFSv4.1. NFS export rules of a file system can be changed at any time.

Note that User and Group IDs that are zero (0) have root privilege and IDs that are non-zero do not have root privilege.

The default export rules will be applied to NFS clients if none are applied:

- `ro`
- `root_squash`
- `no_all_squash`
- `no_fileid_32bit`
- `anonuid=65534`
- `anongid=65534`

The RULES argument represents the export rules that apply to the file system. Options can be applied to a single or multiple IP addresses, Netmasks, or LDAP group. See the syntax examples:

- Options applied to an single client: `'host(options)'`
- Options applied to multiple clients: `'-options host1 host2…'`
- Represent all clients with an asterisk: *

The `host` parameter must belong to one of the following categories:

- IPv4 IP address: `ddd.ddd.ddd.ddd`
  IPv4 Netmask: `ddd.ddd.ddd.ddd/dd`
- IPv6 IP address: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`
  IPv6 Netmask: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/ddd`

Valid export rules are listed below:

Export Rule	Rules Description
`ro`	Read-Only grants an NFS client and their users or groups the privilege to read the file system.
`rw`	Read-Write grants an NFS client and their users or groups the privilege to read and write to the file system.
`fileid_32bit`	Allows 32-bit inode support for clients.
`nofileid_32bit`	Disables 32-bit inode support for clients.
`root_squash`	Prevents client users and groups with root privilege from mapping their rootprivilege to a file system. All users with UID 0 will have their UID mapped to `anonuid`. All users with GID 0 will have their GID mapped to `anongid`.
`no_root_squash`	Allows root users and groups to access the file system with root privilege.

Export Rule	Rules Description
`all_squash`	Maps all UIDs (including root) to anonuid and all GIDs (including root) to anongid.
`no_all_squash`	Prevents the remapping of user and group IDs to `anonuid 65534` or `anongid 65534`. All users and groups will retain their IDs unless `root_squash` is also specified.
`anonuid`	Any user whose UID is affected by `root_squash` or `all_squash` will have their UID mapped to `anonuid`. The default `anonuid` is `65534`.
`anongid`	Any user whose GID is affected by `root_squash` or `all_squash` will have their GID mapped to `anongid`. The default `anongid` is `65534`.
`atime`	After a read operation has occurred, the inode access time is updated only if any of the following conditions is true: the previous access time is less than the inode modify time, the previous access time is less than the inode change time, or the previous access time is more than 24 hours ago.
`noatime`	Disables the update of inode access times after read operations.
`secure`	Prevents NFS access to client connections coming from non-reserved ports. Applies to NFSv3, NFSv4.1, and auxiliary protocols MOUNT and NLM.
`insecure`	Allows NFS access to client connections coming from non-reserved ports. Applies to NFSv3, NFSv4.1, and auxiliary protocols MOUNT and NLM.

Enclose the entire rule set with single quotes. For example,

```
purefs nfs setattr --rules '10.20.225.2(rw,root_squash)' MyFiles
```

If multiple rules are specified, separate each rule with a space, enclosing the entire rule set with single quotes. For example,

```
purefs nfs setattr --rules '10.20.225.2(rw,root_squash) 1.2.0.0(ro)' MyFiles
```

For each client request, the file system will check every export rule, find the matching IP filter, and apply the options from the rule. If the IP address of a client does not match any rules in a given export, the client will not be able to mount the server.

The following guidelines apply when creating NFS export rules:

- If rules are not defined before enabling the NFS protocol, then all clients will have `rw` access and `no_root_squash` privileges to the file system. For example,

  ```
  purefs nfs setattr MyFiles
  ```

  This is equivalent to specifying export rule `'*(rw,no_root_squash)'`.

- Omitting the `(options)` portion of a rule is equivalent to specifying `(ro,root_squash)`. In the following example, the rule `'*'` will grant all clients `read-only`, `root_squash` access to the `MyFiles`file system.

  ```
  purefs nfs setattr --rules '*' MyFiles
  ```

  This is equivalent to specifying export rule `'*(ro,root_squash)'`.

- If the rule is defined but one or more export options are not specified, then the undefined option(s) will default to `ro` access and `root_squash` privileges. For example, an export rule that is defined as `'*(rw)'` will have `root_squash` privileges. Likewise, an export rule that is defined as `'*(no_root_squash)'` will have `ro` access.

- A rule cannot include conflicting options. For example, `ro` and `rw` cannot be included in the same rule. Likewise, `root_squash` and `no_root_squash` cannot be included in the same rule.

- If a client IP address matches multiple rules, Purity//FB uses the first rule it encounters in priority based on rule category. IP address rules have the highest priority. Specifically, the priority of matching rules is as follows: IP address > Netmask > Wildcard.

- If an address matches multiple rules in the same category, then the first (leftmost) rule applies.

- By default, the FlashBladefile systems use 64-bit inode numbers. If your environment contains clients that can only support 32-bit inode numbers, add a third export option named `fileid_32bit` to the NFS export rule of the file system that requires 32-bit inode support. For example, `*(rw,no_root_squash,fileid_32bit)`

- If you are applying multiple rules using options specified in parentheses or dashes, the options in parentheses will have priority over the default options that are not implicitly applied. For example,

  ```
  purefs nfs setattr --rule ‘-rw,no_root_squash 10.20.255.1
  10.20.255.2 10.20.255.3’ 10.20.255.4(ro) MyFiles
  ```

  The `(ro)` option for 10.20.255.4 will be applied, but the `-no_root_squash` option will overwrite the default `root_squash` option.

- If you are applying multiple rules with a dash (-), observe the priority overwrite of default options. An option specified with a dash (-) will overwrite any options to the right if not implicitly defined. For example,

```
purefs nfs setattr --rule '-rw 10.20.255.1 10.20.255.2 -no_root_squash
 10.20.255.3 -ro 10.20.255.4' MyFiles
```

  - The `rw` and `root_squash` options are applied to 10.20.255.1 and 10.20.255.2. The `root_squash` option was not overwritten and is retained as the default option.
  - The `rw` and `no_root_squash` options are applied to 10.20.255.3. The `ro` default was overwritten by the `rw` option with the dash (-).
  - The `ro` and `no_root_squash` options are applied to 10.20.255.4. The `root_squash` default was overwritten by the `no_root_squash` option with the dash (-).

Here is an example of the `purefs list --protocol-specific nfs` output with various rules set for each file system created:

```
purefs list --protocol-specific nfs
Name    Protocols Rules
File1   nfs       *
File2   nfs       *()
File3   nfs       -
File4   nfs       10.20.225.2(rw)
File5   nfs       2620::/16(no_root_squash)
File6   nfs       *(rw,no_root_squash)
File7   nfs       *(rw,root_squash,fileid_32bit)
File8   nfs       10.20.30.40(root_squash,anonuid=7777,anongid=8888)
```

In the list output above, the NFS export rules are described as follows:

Export Rules	Rules Description
`*`	All clients have default `ro` access and `root_squash` privileges to the file system.
`*()`	All clients have default `ro` access and `root_squash` privileges to the file system.
`-`	Export rules have been deleted, rendering the file system inaccessible.
`10.20.225.2(rw)`	Client `10.20.225.2` has `rw` access and `root_squash` (default) privileges.
`2620::/16(no_root_squash)`	Client `2620::/16` has `ro` (default) access and `no_root_squash` privileges.

Export Rules	Rules Description
`*(rw,no_root_squash)`	All clients have `rw` access and `no_root_squash` privileges. This is the default rule used when creating a file system.
`*(rw,root_squash,fileid_32bit)`	All clients have `rw` access and `root_squash` privileges. The file system also supports clients that require 32-bit inode support.

`10.20.30.40`
 	
`(root_squash,anonuid=7777,anongid=8888)`
	Client 10.20.30.40 has `rw` access. Any access attempts with UID 0 will be mapped to UID 7777. Any access attempts with GID 0 will be mapped to GID 8888.

Adding and Removing Export Rules

After export rules have been defined, you can easily add additional rules and remove rules using the `purefs nfs setattr` command with the `--add-rules` and `--remove-rules` options. When these options are specifed together, the `--remove-rules` operation is processed first. Specifying the `--after` option indicates where rules are added or removed.

Consider the following example:

1 All clients have read-write (`rw`) access and `no_root_squash` privileges for file system `fs2`

```
$ purefs list --protocol-specific nfs fs2
Name  Protocols  Rules
fs2   -          *(rw,no_root_squash)
```

2 A new read-only rule is added for client `10.20.255.2` using the `--add-rules` option.

```
$ purefs nfs setattr --add-rules '-ro 10.20.255.2' --after
 '*(rw,no_root_squash)' fs2
Name  Protocols  Rules
fs2   -          *(rw,no_root_squash) -ro 10.20.255.2
```

- The new rule is added after the original rules (`'*(rw,no_root_squash)'`) by specifying `--after`.
- The read-write rule is overwritten by specifying the `ro` rule with a dash (`-`).

3 The read-only rule is then removed from client `10.20.255.2` using the `--remove-rules` option, and the read-only rule is added to client `10.20.255.4` using the `--add-rules` option.

```
$ purefs nfs setattr --add-rules '10.20.255.4' --remove-rules '10.20.255.2' --
after '-ro' fs2
Name  Protocols  Rules
```

```
fs2   -          *(rw,no_root_squash) -ro 10.20.255.4
```

- The `--after '-ro'` option specifies that the `--remove-rules` operation is processed before `--add-rules` operation.
- The `--after '-ro'` option indiciates that the `--add-rules` string (`--add-rules 10.20.255.4`) is added after the read-only rule, indicating that the read-only rule applies to client `10.20.255.4`.

4 Alternatively, after example 2 above, the added read-only rule for client `10.20.255.2` is removed completely.

```
$ purefs nfs setattr --remove-rules '-ro 10.20.255.2' --after
 '*(rw,no_root_squash)' fs2
Name  Protocols  Rules
fs2   -          *(rw,no_root_squash)
```

- The `--remove-rules '-ro 10.20.255.2'` option indicates that the read-only rule for client 10.20.255.2 is removed.
- The `--after '*(rw,no_root_squash)'` option indicates that any rule(s) specified by `--remove-rules` after the original read-write and no_root_squash rules is removed.

Adding and Removing Policies for File Systems

Policies are rules that govern snapshot replication and retention, which are applied to file systems. Snapshot policies are created using the **`purepolicy`** command or the GUI. To add a snapshot policy to a file system, use the **`purefs add --policy POLICY FILE-SYSTEM`** command. To remove a snapshot policy from a file system, use the **`purefs remove --policy POLICY FILE-SYSTEM`** command.

Enabling Protocols for File System Export

Protocols determine the access rights and privileges a client has to a distributed file system. FlashBlade file systems support the following protocols:

- Hypertext Transfer Protocol (HTTP)
- Network File System (NFS)
- Server Message Block (SMB)

The **`purefs add --protocol`** command enables protocols on a file system.

A file system must have at least one protocol (**`nfsv3`**, **`nfsv4.1`**, **`smb`**, or **`http`**) enabled in order for it to be accessible. By default, protocols are not configured for a file system, as indicated by the dash (-) symbol in the `purefs list` output.

More than one protocol can be assigned to a file system. Enter multiple protocols as comma-separated values.

Before you enable the NFS or SMB protocol, perform the following checks:

- If you are integrating the FlashBlade array with a directory service, verify that the directory service has been properly configured and enabled. For more information about the directory service, refer to *pureds*.
- If you are enabling the NFS protocol, verify the NFS export rules have been defined so the file system is accessible to the appropriate clients. If the NFS export rules are not defined, the file system uses the default `*(rw,no_root_squash)` rule, which means that all clients have `read` and `write` access and `no_root_squash` privileges to the file system. Run the `purefs setattr --rules` command to define the NFS export rules.
- If you are enabling the SMB protocol in AD RFC2307 (default) mode, prepare Active Directory for multi-protocol support by setting the `gidNumber` and `uidNumber` attributes for each domain user and group that will be accessing the SMB shares. The `gidNumber` and `uidNumber` attributes can be set through the Microsoft Identity Management for UNIX service or PowerShell.

Once a file system has been created and its protocols have been enabled, it is visible through any of the data vips configured on the subnet, ready to be mounted on the clients.

The **`purefs list --protocol-specific`** command is used to view file system attributes that are specific to particular protocols. Running `purefs list --protocol-specific nfs` displays NFS file systems and their configured rules.

```
$ purefs list --protocol-specific nfs
Name  Protocols  Rules
fs1   nfsv3      *(rw,no_root_squash)
fs2   smb        *(rw,no_root_squash)
fs3   http       *(rw,no_root_squash)
fs4   http       *(rw,no_root_squash)
      nfsv3
      smb
```

In the output above:

- **Name** - The name of the file system.
- **Protocols** - The protocols applied to the file system.
- **Rules** - The export rules applied to the file system. Note that this column is only displayed if `nfs` is specified.

Setting the Access Control Style

An access control list (ACL) is a list of security permissions associated with a directory or file. Specific rules can be added to the ACL to define who can access the file and what actions they can perform. When an ACL is applied to a directory or file, then that directory or file will reference its list of rules to

determine whether to grant or deny access, including which actions are allowable such as: read, write, and execute.

For implementations where more than one file system export protocol is enabled, you can set the file system's access control style to dictate how the protocols will interact.

The following access control styles are available:

- **Shared** - NFS and SMB clients are both able to set permissions on, and access files and directories over any protocol.
- **NFS** - Only NFS clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- **SMB** - Only SMB clients can set permissions on files and directories; however, files and directories can be accessed over any protocol.
- **Independent** - NFS and SMB clients are both able to set permissions on files and directories and can access files and directories created over any protocol. Permissions set by SMB clients will not affect NFS clients and vice versa. NFS clients will be restricted to only using mode bits to set permissions.
- **Mode Bits** - NFS and SMB clients are both able to set permissions on files and directories, but permissions are enforced in mode bits-only format. When SMB clients set an ACL it will be converted to mode bits.

To set the access control style, issue the **`purefs setattr --access-control-style`** command and specify `independent`, `mode bits`, `nfs`, `smb`, or `shared`.

Note the following behaviors and restrictions:

- If you set the access control style to `smb` and then disable the SMB protocol, the access control style is automatically reset to `shared`.
- If you set the access control style to `nfs` and then disable both NFSv3 and NFSv4.1, the access control style is automatically reset to `shared`.
- If neither the NFSv3 or NFSv4.1 protocols are enabled, the access control style cannot be set to `nfs`.
- If the SMB protocol is not enabled, the access control style cannot be set to `smb` or `independent`.

Use the **`purefs setattr --protocol`** command to change the access control style for a file system. If you are changing the access control style from `nfs`, `shared`, or `smb` to `mode-bits` or `independent`, you can specify the `--discard-detailed-permissions` option to discard more detailed access control lists currently set within the file system; NFS permission controls will only be enforced at the granularity level of NFS mode bits.

In the following example, the access control style for file system `fs1` is set to `smb`.

```
purefs setattr --access-control-style smb fs1
Name     Protocols  Access Control Style  Safeguard ACLs
fs1      smb        smb                   True
```

The access control style is then changed to `mode-bits`.

```
purefs setattr --access-control-style mode-bits --discard-detailed-permissions fs1
Name     Protocols  Access Control Style  Safeguard ACLs
fs1      smb        mode-bits             False
```

ACL Safeguarding

ACL safeguarding controls how a mode bit change will impact the ACL on the file or directory. When disabled, modification of mode bits will result in the removal of any existing ACL. When enabled, modification of mode bits will result in the modfication of the existing ACL to merge the permissions from the mode bits with the permissions in the ACL.

- If the access control style was set to `smb`, ACL safeguarding is automatically enabled. Disabling ACL safeguarding is not allowed.
- If the access control style was set to `nfs` or `shared`, consider enabling ACL safeguarding.
- ACL safeguarding is not available for `independent` or `mode-bits` access control styles.

ACL safeguarding is enabled and disabled by issuing the **`purefs enable --safeguard-acls`** and **`purefs disable --safeguard-acls`** commands, respectively. When enabling or disabling ACL safeguarding, you must specify the file system(s) to which it applies.

To view set access control styles and the status of ACLs safeguarding, run the **`purefs list --multi-protocol`** command.

```
$ purefs list --multi-protocol
Name  Protocols    Access Control Style  Safeguard ACLs
fs1   nfsv3        nfs                   True
      nfsv4.1
      smb
fs2   nfsv4.1      shared                False
      smb
fs3   smb          smb                   False
```

In the output above:

- **Name** - The name of the file system.
- **Protocols** - The protocols applied to the file system.
- **Access Control Style** - The access control style applied to the file system.
- **Safeguard ACLs** - Whether safeguard ACLs is enabled (`True`) or disabled (`False`).

Disabling Protocols on File Systems

The **`purefs remove --protocol`** command removes protocols from a file system.

Removing a protocol renders the file system inaccessible by the clients.

Changing the Provisioned Size of the File System

Run the **`purefs setattr --size`** option to change the provisioned size of the file system.

Managing Special Directories

Special directories are features in Purity//FB that perform specific functions. The fast remove feature and snapshot feature can be enabled or disabled. The fast remove feature manages the process of deleting a large number of file systems efficiently. When enabled, a `.fast-remove` directory will appear in the root directory. The snapshot feature provides a virtual repository for snapshots of a specified file system. When enabled (by default at file system creation), a `.snapshot` directory will appear in the root directory that is accessible at each directory.

The following rules apply to special directories:

- Special directories can be enabled or disabled.
- Special directories cannot be renamed, deleted, or moved into other directories.
- Special directory names are reserved. You cannot create a regular directory with the `.fast-remove` or `.snapshot` name, regardless of the enabled/disabled status.
- Fast remove and snapshot directories can only be removed by disabling it.

Enabling and Disabling Fast Remove

When a directory and its contents are no longer required, they can be removed to reclaim space.

Removing the contents of a directory recursively by running the `rm -r` command causes the client to delete files one at a time. This can be a time consuming and expensive operation.

The fast remove feature allows you to quickly remove large directories by offloading this work onto the server. When the fast remove feature is enabled, a special pseudo-directory named `.fast-remove` is created in the root directory of the NFS mount. To remove a directory and its contents, run the `mv` command to move the directory into the `.fast-remove` directory.

In the following example, from the root directory of the NFS mount, a directory called `big_dir` is being moved into the `.fast-remove` directory.

```
mv big_dir/.fast-remove/
```

Once a directory has been moved into the `.fast-remove` directory, it is immediately destroyed, and through background operations, the space it occupied will be freed.

The fast remove feature can be enabled or disabled at any time. Fast remove is disabled by default.

Run the **`purefs enable --fast-remove-dir`** command to enable the fast remove feature. Enabling fast remove creates a `.fast-remove` directory in the root directory of the NFS mount.

For example, run the following commands to 1) enable the fast remove feature on file system MyFiles, and 2) view the contents of the root directory of the NFS mount to verify that the `.fast-remove` directory has been created.

```
//Run on the array to enable the fast remove feature.
purefs enable --fast-remove-dir MyFiles
```

```
//Run on the NFS mount to view the content of the root directory.
ls -al
drwxrwxrwx   1  root  root      0 Oct 30 10:46 .
drwxrwxrwx  25  root  root  12288 Jun 30 16:44 ..
drwxr-xr-x   1  root  root      0 Oct 30 11:27 .fast-remove
…
```

You cannot move individual files to the `.fast-remove` directory. To remove a file, either use the `rm` command or move the directory enclosing the file to the `.fast-remove` directory.

The `.fast-remove` directory can only be removed by disabling the fast remove feature. To disable the fast remove feature, run **`purefs disable --fast-remove-dir`**.

The fast remove feature is powerful and comes with the following cautionary notes:

- Once a directory has been moved into the `.fast-remove` directory, it is no longer recoverable.
- With fast remove, users can remove any files that are contained in a directory that they can rename, even if they do not have read, write, or execute permissions on the child directories. This breaks from the standard UNIX permissions model. To restrict access to the `.fast-remove` directory, set the directory permissions.

By default, the `.fast-remove` directory permissions are set to `drwxr-xr-x`. These permissions can be changed to grant or restrict access to the `.fast-remove` directory. Disabling and then re-enabling the fast remove feature will reset the settings to the default permissions.

Enabling and Disabling a Snapshot Directory

Each file system has a special directory for snapshots that can be enabled or disabled to provide access to stored snapshots. The snapshot feature provides a virtual repository for snapshots of a specified file system. By enabling a file system 's snapshot directory, a `.snapshot` directory will be created in the root directory of that file system. All snapshots for that file system will be accessible in the directory, including the snapshots created prior to enabling the directory. You can enable the snapshot directory using the **`purefs enable --snapshot-dir`** command. This example will enable the snapshot directory for the file system named `MyFiles`:

```
purefs enable --snapshot-dir MyFiles
Name     Fast Remove   Snapshot
MyFiles  False         True
```

The `.snapshot` directory can be removed from the root directory by disabling the directory. Once disabled, historical snapshots can be viewed but will not be accessible until the directory is enabled. Disabling the the directory will not destroy any snapshots, they must be actively destroyed using the **`purefs destroy`** command.

Setting File System Write Limits

⚠ **Important note!** File system hard limits can only be set for a file system with sizes greater than 0.

To help manage resource usage, a hard limit can be enabled on the provisioned size of a file system. The **`purefs enable --hard-limit`** command is used to limit writes of the file system to its provisioned size.

When the system detects the file system has exceeded its provisioned size, all future writes are halted until usage decreases below the provisioned size. This occurs within minutes of the provisioned size being reached. Note that the provisioned size can be exceeded during this detection period when a hard limit is enabled. For example, a 5G write is in-process for a file system whose provisioned size is 10G and current usage is 9G. When the system halts the write, the file system ’s provisioned size will be slightly exceeded.

A hard limit cannot be enabled if the file system usage has already exceeded its provisioned size unless forced. You can force the hard limit by issusing the **`purefs enable --hard-limit --ignore-usage`** command.

Additionally, if a hard limit is enabled, the file system 's size cannot be reduced to a value less than its current space usage. You can force shrink the file system 's size by issuing the **`purefs setattr --size --ignore-usage`** command.

The `--ignore-usage` option described above provides the storage admin greater control over the file system ’s space by storage users. For example, if a file system has been provisioned with a hard limit of 10TB and 8TB are quickly used, the admin has the option to reset the provisioned size to 5TB to control usage. In this case, writes will stop once 5TB is reached; users who wish to continue writing to this file system will need to manage (delete or truncate) their data to under 5TB.

You can disable a hard limit issuing the **`purefs disable --hard-limit`** command.

Setting User and Group Write Limits

Write limits (quotas) can also be set for users and groups on a per- file system basis to help manage file system resource usage. The **`purefs setattr --default-user-quota`** and **`setattr --default-group-quota`** commands are used to set quotas for all file system users or groups (default quotas). If you wish to set quotas for specific users or groups (explicit quotas) use the **`purequota user create`** or **`purequota group create`** command.

Similar to file system write limits, when the system detects that a user or group has exceeded its quota, all future writes are halted until usage decreases below the set quota size via deletion or truncation of the users' or groups' data from the file system. Note that a quota can be exceeded during this detection

period. For example, if a user is writing a large file that causes them to exceed the quota, their write will fail, but their total usage may have slightly exceed their set quota.

Destroying and Recovering File Systems

When a file system or snapshot is no longer required, it can be **destroyed** using the **`purefs destroy`** command. Destroying a file system implicitly destroys all of its snapshots. Similarly, when a file system is recovered, snapshots that were implicitly destroyed with the source file system will be recovered as well.

Destroying an individual snapshot only reclaims space that is uniquely charged to the snapshot. Space shared with older snapshots or with the source file system remains allocated.

As a precaution, protocols must be removed from a file system before it can be destroyed. Use the **`purefs remove`** command to remove all protocols from a file system.

⚠ **CAUTION:** If Pure File SafeMode is enabled, contact Pure Support to eradicate file systems and file system snapshots.

Destruction of file systems and snapshots is not immediate. The **`purefs destroy`** command places file systems and snapshots in the **eradication pending** state, making them immediately inaccessible, while leaving their data intact for 24 hours (known as the **eradication pending period**) . At any time during the eradication pending period, an administrator can use the **`purefs recover`** command to recover the contents of a destroyed file system or destroyed snapshot.

When a destroyed file system's or snapshot's 24-hour eradication pending period has lapsed (or if an administrator eradicates the file system or snapshot during the eradication pending period), the array reclaims the physical storage occupied by its data.

If the physical storage occupied by a destroyed file system 's or destroyed snapshot's data is required immediately (for example, if a large amount of new data is to be written), an administrator can use the **`purefs eradicate`** command to terminate the eradication pending period and initiate immediate storage reclamation. The **`purefs eradicate`** command only acts on previously destroyed file systems and snapshots.

Once reclamation starts, whether because an eradication pending period has lapsed or because a **`purefs eradicate`** command was executed, a destroyed file system's or destroyed snapshot's data can no longer be recovered. Names of file systems can be reused only after a file system has been eradicated.

File System Replication

File systems can be replicated from one FlashBlade array (local array) to another FlashBlade array (target array). File system replication requires two connected arrays and a replica link between the source file system and a target file system. A replica link is the connection between a local file system and target file system.

To connect two FlashBlade arrays, use the **`purenetwork vip create`** command to create replication vips on the source and target arrays, and then connect the arrays using the **`purearray`** command.

Note that port 8117 is used for file replication.

Creating a Replica Link

Once the two FlashBlades are connected, you must create a replica link. The replica link completes the environment set up for replication. Create a replica link by running the **`purefs replica-link create`** command on the local array. You must specify the target array name with the `--remote` option and provide the name of the local file system to be replicated. If the name of the file system already exists on the target array, you must specify a unique file system target name using the `--remote-fs` option.

You can optionally apply a policy to the replica link to schedule when file system snapshots are replicated from the source file systems to the remote file system by issuing the **`purefs replica-link create`** command with the `--policy` option. Note that you must have previously created a policy (using the **`purepolicy create`** command).

Manual File Replication

If you did not attach a policy to your replica link, you can manually perform file replication. If you did attach a policy to your replica link, you can manually perform replication outside of the policy's schedule. To manually take a snapshot of a file system on the local array and replicate it to the target file system on the remote array, use the **`purefs snap --send --target`** command and specify the file system being replicated.

Viewing File System Replication

To view the details of your replication set up as well as replication status, use the **`purefs replica-link list`** command. The following information is displayed:

- **Name**: The name of the source file system.
- **Direction**: The direction of replication, either inbound (←) or outbound (→).
- **Remote**: The name of the connected array.
- **Remote File System**: The name of the remote file system.
- **Policy**: The name of the applied replication policy.
- **Status**: The current replication status.
  - **idle**: No errors, no current replication.
  - **replicating**: No errors, currently attempting replication.
  - **paused**: Data transfer is currently halted due to administrator action.
  - **unhealthy**: Replication is in an unhealthy state.

- **Lag**: The amount of time, if any, the replication target is behind the source, and is the time difference between the current time and the recovery point.
- **Recovery Point**: The creation time of the last snapshot that was completely replicated.

Adding the `--status-detail` option replaces the **Lag** and **Recovery Point** columns with a **Details** column that displays an error if the replica link is unhealthy.

File Replication Failover, Reprotect, and Failback

In file system replication setups you can stop writes to the local file system and redirect writes to the target file system (failover). Failover can be used if there is an issue with the local file system. During failover, replication stops until you reprotect.

⚠ **CAUTION:** If Pure File SafeMode is enabled, contact Pure Support to promote and demote file systems.

Failover requires promotion of the target file system and redirection of all clients to the target array to return to operation. To reprotect, demote the (former) source file system. Once the file system is demoted, replication policies will start replicating in the new direction.

Note that when demoting the local file system, any writes performed since the last replication will be discarded. If a local snapshot was taken since last replication, demoting the file system requires that snapshot be destroyed and eradicated. You must remove any data since the last snapshot of the file system was taken. The file system must either have no snapshots, or the most recent snapshot must be the last replicated snapshot. Additionally, it is suggested that any policies attached to the local file system and replica link be removed or disabled in order to avoid taking new snapshots.

Once the local file system is demoted, re-enable or re-add the policies that were disabled before demoting the file system. Replication resumes as scheduled between the promoted remote file system and the demoted local file system. Data is written to the promoted remote file system, which is then replicated to the demoted local file system.

To resume the original replica link direction (failback), manually stop writes on the remote file system and send the remaining data to the local file system. Once data is sent to the local file system, promote the local file system and demote the remote file system. You can change the file system to not writeable or stop clients while failing back to avoid having any writes that need to be discarded when demoting the target.

For example:

1 On the target array, array2, promote the target file system:

```
purefs promote fs1-R
```

2 Manually redirect clients to the the target array, array 2.

3 On the local array, array1, remove the policy `fs_snap_pol1` that attached to the local file system `fs1`.

```
purepolicy remove --fs fs1 fs_snap_pol1
```

4 Remove the policy from the replica link using the **`purepolicy remove --remote`** command, where `--remote` specifies the remote (target) array. In the following example the policy `dailyBackup` is removed from the replica link:

```
# purepolicy remove --remote array2 dailyBackup
```

5 On the local array, array1, demote the local file system :

```
purefs demote --discard-non-snapshotted-data fs1
```

6 Add the policy back to the replica link using the **`purepolicy add --remote`** command, where `--remote` specifies the demoted array. In the following example, the policy `dailyBackup` is added back to the replica link:

```
# purepolicy add --remote array1 rl_dailyBackup
```

7 On the local array, verify that replication is disabled and that the target file system is promoted:

```
purefs replica-link list
Name  Direction  Remote  Remote File System  Policy       Status     Recov...
fs1   <--        array2  fs1-R               dailyBackup  unhealthy  2019-...

purefs replica-link list --status-details
Name  Direction  Remote  Remote File System  Policy       Status     Details
fs1   <--        array2  fs1-R               dailyBackup  unhealthy  The
 target file system is promoted.
```

Since the local file system `fs1` is demoted, writes to `fs1` are stopped. Instead, because the target file system `fs1-R` was promoted and all clients redirected to the target file system, writes are redirected to `fs1-R`.

Once the local file system comes back up, replication resumes as scheduled between the newly promoted target file system and demoted local file system because the policy was re-added to the replica link. Because clients have been redirected to the target, data is written to the promoted remote file system, which is then replicated to the demoted local file system.

To resume the original replica link direction and avoid data loss in the process, you can manually stop writes on the remote file system and send the remaining data to the local file system by issuing the **`purefs disable --writable`** and **`purefs snap --send`** commands on the remote array. Once data is sent to the local file system, promote the local file system using the **`purefs promote`** command and demote the remote file system with the **`purefs demote`** command.

In the following example:

- The local array is `array2`
- The local file system is `fs1-R`
- The target array is `array1`
- The target file system is `fs1`
- The policy re-attached to the local file system `fs1` is `fs_snap_pol1`.
- The policy attached to the replica link is `rl_pol1`.

1. On the local array, `array2`, stop all writes on the local file system `fs1-R` and then transfer all remaining data to the target file system `fs1`:

```
purefs disable --writable fs1-R
purefs snap --send --target array1 fs1
```

2. Create a snapshot of the file system and send it to the target file system `fs1` to ensure that data since the last snapshot resides on the to-be-promoted file system (`fs1`). Skip this step if you do not need data written since the last snapshot.

3. On the target array, `array1`, promote the target file system `fs1`

```
purefs promote fs1
```

4. Add back any previously removed file system policy. In the following example, the `policy fs_snap_pol1` is added back to file system `fs1`.

```
# purepolicy add fs_snap_pol1
```

5. Remove the policy from the replica link using the **`purepolicy remove --remote`** command, where `--remote` specifies the remote (target) array. In the following example the policy `dailyBackup` is removed from the replica link:

```
# purepolicy remove --remote array1 dailyBackup
```

6. Manually direct all clients back to the local file system.

7. On the local array, `array2`, demote the local file system `fs1-R`

```
purefs demote --discard-non-snapshotted-data fs1-R
```

8. Add the policy back to the replica link using the **`purepolicy add --remote`** command, where `--remote` specifies the demoted array In the following exmaple, the policy `dailyBackup` is added back to the replica link:

```
# purepolicy add --remote array2 rl_pol1
```

9 On the (now) local array `array1`, verify that the replica link has returned to its original direction (from file system `fs1` to the target file system `fs1-R`):

```
purefs replica-link list
Name  Direction  Remote  Remote File System  Policy       Status   Lag
 Recovery P...
fs1   -->        array2  fs1-R               dailyBackup  healthy  10m
 2019-11-11 12...
```

Restoring a File System from a Snapshot

CAUTION: If Pure File SafeMode is enabled, contact Pure Support to restore a file system from a snapshot.

Note: Restoring a file system from a snapshot can only be performed on snapshots created with Purity version 3.0.0 and later.

A file system can be restored, or *rolled back*, from the most recent snapshot of that file system using the **`purefs copy`** command. To restore a file system from a previous snapshot, you must include the name of the source snapshot that is the restore point (SOURCE) as well as the name of the target file system (TARGET). You must include the `--overwrite` option to indicate that the file system is being restored from the specified snapshot and discard the data on the existing existing target file system (but not the existing data in the specified source snapshot) using the `--discard-non-snapshotted-data` option.

Use the **`purefs list --snap`** command to identify the source snapshot from which you wish to restore the target file system. Once identified, issue the **`purefs copy SOURCE TARGET`** command to restore the target file system from the snapshot. For example:

1 On the local array, issue the **`purefs list --snap`** command.

```
purefs list --snap
Name                          Source      Created                    Policy
fs1.2019_11_20_02_48          fs1         2019-11-20 02:48:51 PST  5m
fs1.2019_11_20_02_33          fs1         2019-11-20 02:33:51 PST  5m
fs1.2019_10_30_13_29          fs1         2019-10-30 13:29:45 PDT  -
```

2 Issue the **`purefs copy`** command to restore the file system fs1 from the latest snapshot `fs1.2019_11_20_02_48`.

```
purefs copy --discard-non-snapshotted-data --overwrite fs1.2019_11_20_02_48
 fs1
```

Restoring from an earlier snapshot is possible, however you must first destroy and eradicate all more recent snapshots. Using the **`purefs list --snap`** output example above, if you wish to restore from the snapshot `fs1.2019_10_30_13_29`, the more recent snapshots `fs1.2019_11_20_02_48` and `fs1.2019_11_20_02_33` must first be destroyed and eradicated.

1 Destroy and then eradicate snapshots `fs1.2019_11_20_02_48` and `fs1.2019_11_20_02_33` using the **`purefs destroy`** command.

```
purefs destroy fs1.2019_11_20_02_48 fs1.2019_11_20_02_33
purefs eradicate fs1.2019_11_20_02_48 fs1.2019_11_20_02_33
```

2 Issue the **`purefs copy`** command to restore the file system fs1 from the snapshot `fs1.2019_10_30_13_29`.

```
purefs copy --discard-non-snapshotted-data --overwrite fs1.2019_10_30_13_29
 fs1
```

Accessing Read-only Replicas

To access read-only replicas, export the replica file system as read-only by running the **`purefs add --protocol`** command on the remote file system. You must specify NFSv3 as the protocol as well as the file system name. For example, **`purefs add --protocol nfsv3 fs1`**. The export displays only the `.snapshot` directories with read-only access to the data.

Viewing File System Performance

You can view real-time file system-specific I/O performance by issuing the **`purefs monitor --protocol nfs FILE-SYSTEM…`** command. Performance statistics are displayed for the specified file system or file systems. Note that currently, per file system performance monitoring is available for NFS only. If you do not have NFS enabled, all output data will appear as 0s. Performance statistics are limited to displaying a maximum of five file systems at any given time. If your array has more than five file systems, an error message will appear if your query parameters do not limit the results to five or fewer file systems. In addition to common formatting and filtering options, the `--interval`, `--repeat`, `--size`, `--total`, and `--total-only` options can be used to control the frequency and content of the real time update. Additionally, historical data can be viewed by using the `--historical` option. The smallest window of time that can be specified for `--historical` is three hours (`3hr`) .

Example 1

```
purefs list
```

Displays a list of file systems that have been created on the FlashBlade array for export, and the protocols assigned to each file system.

Example 2

```
purefs create --size 10T MyFiles
```

Creates file system `MyFiles` with a provisioned size of 10 terabytes.

Example 3

```
purefs nfs setattr --rules '10.20.225.2() 1.2.0.0/16(rw,no_root_squash)' MyFiles
```

Sets the NFS export rules for file system`MyFiles`, completely replacing the existing set of rules. The first export rule grants `ro` access and `root_squash` privileges to clients with IP address `10.20.225.2`. The second export rule grants `rw` access and `no_root_squash` privileges to clients with IP address `1.2.0.0.*`.

Example 4

```
purefs nfs setattr --rules '-rw,root_squash 10.20.225.1 10.20.225.2 10.20.225.3'
 MyFiles
```

Sets the NFS export rules for file system`MyFiles`, completely replacing the existing set of rules. The export rule grants `rw` access and `root_squash` privileges to clients 10.20.255.1, 10.20.255.2, and 10.20.255.3.

Example 5

```
purefs nfs setattr --rules '-ro,root_squash 10.20.255.1 10.20.255.2 10.20.255.3
-rw,no_root_squash 10.20.255.4 10.20.255.5 10.20.255.6' MyFiles
```

Sets the NFS export rules for file system`MyFiles`, completely replacing the existing set of rules. The export rules grant `ro` access and `root_squash` privileges for clients 10.20.255.1, 10.20.255.2, 10.20.255.3 and `rw` access and `no_root_squash` privileges for clients 10.20.255.4, 10.20.255.5, 10.20.255.6.

Example 6

```
purefs nfs setattr --rules '*(ro,root_squash)' MyFiles
```

or

```
purefs nfs setattr --rules '*' MyFiles
```

or

```
purefs nfs setattr --rules '*()' MyFiles
```

Sets the export rule for file system `MyFiles` to `ro` access and `root_squash` privileges for all clients, replacing any existing rules.

Example 7

```
purefs nfs setattr --rules "" MyFiles
```

Deletes all export rules from file system`MyFiles`, rendering the file system inaccessible to any client.

Example 8

```
purefs add --protocol nfsv3,smb MyFiles
```

Enables the NFSv3 and SMB protocols on file system`MyFiles`.

Example 9

```
purefs setattr --access-control-style shared MyFiles
```

Sets the access control style of the file system `MyFiles` to `shared`.

Example 10

```
purefs enable --safeguard-acls MyFiles
```

Enables ACLs safeguarding for file system MyFiles.

Example 11

```
purefs disable --fast-remove-dir MyFiles
```

Disables the fast remove directory for file system`MyFiles`.

Example 12

```
purefs destroy MyFiles
```

Destroys file system `MyFiles` and places it in the eradication pending state until the time expires or is recovered.

Example 13

```
purefs recover MyFiles
```

Recovers file system `MyFiles` if performed before the eradication pending period expires.

Example 14

```
purefs eradicate MyFiles
```

Eradicates file system `MyFiles` manually before the eradication pending period expires and Purity//FB reclaims the physical storage occupied by its data.

Example 15

```
purefs enable --hard-limit --ignore-usage fs1
Name  Size  Used    Hard Limit  Created                  Protocols
fs1   1M    15.51M  True        2018-04-20 21:16:18 PDT  nfsv4.1
```

Enables a hard limit on file system fs1's provisioned size of 1M.

Example 16

```
purefs disable --hard-limit fs1
Name  Size  Used    Hard Limit  Created                  Protocols
fs1   1M    15.51M  False       2018-04-20 21:16:18 PDT  nfsv4.1
```

Disables the hard limit on file system fs1.

Example 17

```
purefs list fs2
Name  Size  Used   Hard Limit  Created                  Protocols
fs2   121G  90G    True        2018-04-20 10:15:01 PDT  -
purefs setattr --size 70G --ignore-usage fs2
Name  Size  Used   Hard Limit  Created                  Protocols
fs2   100G  90G    True        2018-04-20 21:16:25 PDT  -
```

In this example, 90G of file system fs2's provisioned size of 121G has been used. In order to control usage, the provisioned size is reset to 70G.

Example 18

```
purefs monitor --protocol nfs --interval 60 --repeat 3 fs1
```

Real-time performance statistics are displayed for file system fs1. Peformance data is queried and displayed three times, once every 60 seconds.

Example 19

```
purefs add --policy pol1 fs1
```

In this example, snapshot policy `pol1` is added to file system`fs1`.

Example 20

```
purefs setattr --default-user-quota 2G fs1
```

Sets the default user quota for all users in file system fs1 to 2G.

Example 21

```
purefs add fs3 --protocol http,nfsv3,nfsv4.1
        Name  Size   Used    Hard Limit   Created                  Protocols
        fs3   -      81.91G  False        2018-08-01 20:21:12 PDT  http
                                                                   nfsv3
                                                                   nfsv4.1
```

Adds HTTP, NFSv3, and NFSv4.1 protocols to file system fs3.

Example 22

```
purefs nfs setattr --rules '*(rw,no_root_squash)' fs1
Name  Protocols  Rules
fs1   -          *(rw,no_root_squash)
```

Adds read-write export rules with root_squash privileges to `fs1` for all clients.

Example 23

```
purefs nfs setattr --add-rules '-ro 10.20.255.2' --after '*(rw,no_root_squash)'
 fs1
Name  Protocols  Rules
fs1   -          *(rw,no_root_squash) -ro 10.20.255.2
```

Adds a new read-only export rule after the original read-write, no_root_squash rules that applies only to client `10.20.255.2`

Example 24

```
purefs nfs setattr --remove-rules '-ro 10.20.255.2' --after '*(rw,no_root_squash)'
 fs1
Name  Protocols  Rules
fs1   -          *(rw,no_root_squash)
```

Removes the read-only rule from client `10.20.255.2`.

Example 25

```
purefs list fs3 --protocol-specific nfs
        Name         Protocols  Rules
        fs3          http       *(rw,no_root_squash)
                     nfsv3
                     nfsv4.1
```

Displays NFS attributes for file system fs3.

Example 26

```
purefs replica-link create --remote array2 --remote-fs fs1-R --policy pol_weekly
 fs1
```

Creates a replica link between the file system `fs1` on the local array `array1`, and the remote file system `fs1-R` (also created with this command) on the remote array `array2` Note that if the file system `fs1-r` already existed on the remote array, this command would fail. Replication occurs per the schedule defined in the policy `pol_weekly`.

Example 27

```
purefs snap --send --target array1 fs1-R
```

(Continuing from example 22) Manually creates a snapshot of file system `fs1` and replicates it to the target array `array2`.

Example 28

```
purefs replica-link list
```

Displays replication details between connected arrays.

Example 29

```
purefs demote --discard-non-snapshotted-data fs1
purefs promote fs1-R
```

For failover, demotes the local file system `fs1` and promotes the target file system `fs1-R`. Replication is stopped. Once clients are directed to the target file system then data writes will occur on the target file system `fs1-R`. The direction of the replica link is reversed—when the local file system `fs1` comes back up, the target file system `fs1-R` is replicated to the local file system `fs1`.

Example 30

```
On array2:
purefs disable --writable fs1-R
purefs snap --send --target array1 fs1-R
purefs demote --discard-non-snapshotted-data fs1-R

On array1:
purefs promote fs1
```

To avoid data loss during failback, once the local file system `fs1` on array 1 comes back up, writes are stopped on the target array `fs1-R` and a snapshot of all recent data is sent to the local file system `fs1`. The target array `fs1-R` is then demoted and the local array `fs1` is promoted. The direction of the replica link is returned to its original direction—the local file system `fs1` is replicated to the target file system `fs1-R`.

See Also

- *purealert*
- *purearray*
- *purenetwork*
- *purequota*
- *puresubnet*
- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purehw

purehw, purehw-connector-list, purehw-connector-setattr, purehw-list, purehw-setattr — displays and modifies the attributes of the array 's hardware components.

Synopsis

```
purehw connector list [--cli] [--filter FILTER] [--csv | --nvp] [--limit LIMIT]
[--notitle] [--page] [--sort SORT] [--raw] [CONNECTOR...]

purehw connector setattr [--lane-speed LANE_SPEED] [--port-count {1|4}]
[CONNECTOR]

purehw list [--csv | --nvp] [--filter FILTER] [--limit LIMIT] [--notitle]
[--page] [--raw] [--sort SORT] [--spec] [--type] [COMPONENT-TYPE]
[COMPONENT...]

purehw setattr [--identify {off|on}] [COMPONENT]
```

Sub-Parsers

`connector`

When specified, indicates that sub-commands are processed for connectors.

Sub-Commands

`list`

Displays hardware components.

When used with the `connector` sub-parser, displays connector attributes.

`setattr`

Sets hardware attributes

When used with the `connector` sub-parser, sets connector attributes.

Arguments

COMPONENT

Hardware component whose information is to be displayed or whose attribute is to be set to the specified value.

CONNECTOR

The name of the connection interface.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--identify {off|on}`

Illuminates the component's LED. When this option is specified as `on`, the LED for the specified component is illuminated.

`--lane-speed` ***LANE_SPEED***

The lane (sub-port) speed for the specified hardware connector, specified in bps, kbps, Mbps, or Gbps (case-insensitive).

`--port-count {1|4}`

The number of lanes (sub-ports), either 1 or 4, for the specified hardware connector.

`--spec`

Displays the model and serial numbers for each of the hardware components.

`--type` ***COMPONENT-TYPE***

Type of component for which information is to be displayed. The component type can be specified as `ch` (chassis), `eth` (Ethernet port), `fb` (flash blade), `fm` (fabric module), `pwr` (power supply), or `xfm`. When this option is specified, information is displayed for all components of the specified type.

Options that control display format:

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--filter` ***FILTER***

Displays only the rows that meet the filter criteria specified.

`--limit` ***LIMIT***

Limits the size of the list output to the specified maximum number of rows.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data.

`--sort` ***SORT***

Sorts the list output in ascending or descending order by the column specified.

Description

In the FlashBlade array, the chassis contains the blades and the bays that host the fabric modules. The FlashBlade chassis name has the form `CHx` where `x` can be `1` through `5`. The chassis status is generated from a combination of software conditions and the status of its components.

The names of the other components within the chassis have the form `CHx.XXm` or `CHx.XXm.YYYn` where:

XX

Denotes the type of component within the chassis. Possible components include: `FB` (Blade), `FM` (Fabric Module), and `PWR` (Power Supply Unit).

m

Identifies the specific component by its index (its relative position within the chassis, starting at `1`).

YYY

Denotes the type of component within the fabric module. Possible components include `ETH` (Ethernet Port) and `FAN` (Fan).

n

Identifies the specific port by its index (its relative position within the fabric module, starting at `1`).

The following table lists the FlashBlade array components that report status:

Component Name	Identify Light	Component Type
`CHx`	Chassis	Chassis. For multi-chassis FlashBlade configurations, chassis can be numbered CH1 through CH5.

Component Name	Identify Light	Component Type
`CHx.FBm`	Flash blade	Flash blade server.
`CHx.FMm`	Fabric module	I/O modules (IOMs) within the chassis.
`CHx.PWRm`	Power module	Power supplies within the chassis.
`CHx.FMm.ETHn`	Ethernet port	Ethernet ports to the blades.
`CHx.FMm.FANn`	Fan	Fabric module cooling fan.

For multi-chassis configurations, the external fabric modules have the form `XFMx`, where `x` can be 1 or 2.

The names of the other components within the chassis have the form `XFMx.XXm` where:

XX

Denotes the type of component within the external fabric module. Possible components include `ETH` (Ethernet Port), `PWR` (Power Supply Unit) and `FAN` (Fan).

m

Identifies the specific component by its index (its relative position within the chassis, starting at 1).

n

Identifies the specific port by its index (its relative position within the fabric module, starting at 1).

The following table lists the FlashBlade external fabric module components that report status:

Component Name	Identify Light	Component Type
`XFMx`	External Fabric Module	External fabric module, modules can be numbered FM1 and FM2
`XFMx.ETHn`	External Fabric Module	Ethernet port.
`XFMx.FANm`	External Fabric Module	External fabric module cooling fan.
`XFMx.PWRm`	Power module	External fabric module power supplies.

Displaying Hardware Components

The **`purehw list`** command displays information about array hardware components that are capable of reporting their status. The display is primarily useful for diagnosing hardware-related problems.

The `purehw list` output includes the following columns:

- **Status**: Component status. Possible hardware statuses include:

  critical

  Component requires immediate attention. Contact Pure Storage Support at *support@purestorage.com*.

  healthy

  Component is performing as expected.

  identifying

  Component is functioning, but not yet initialized.

  unhealthy

  Component is not performing as expected

  unknown

  Insufficient information to determine a state for this device.

  unused

  Slot is currently empty, as expected.

- **Identify**: State (on or off) of an LED used to visually identify the component. (Relevant only for controllers, NVRAM bays, storage bays, and storage shelves.)
- **Slot**: Slot number occupied by the PCI Express card that hosts the component. (Relevant only for ports hosted by PCI Express cards in controller chassis.)
- **Index**: Number that identifies the relative position of a hardware component within the array.
- **Speed**: The maximum speed (in Gb/s) of the component.
- **Temperature**: Temperature reported by the component. (Relevant only for temperature sensors.)

Include the **`--spec`** option to display the model and serial numbers for each of the hardware components.

The **`purehw connector list`** command displays information about all connectors. The display is useful for showing connector information that can be used with the **`purehw connector setattr`** command.

The **`purehw connector list`** command output includes the following columns:

- **Name**: The name of the connector.
- **Connector Type**: The form factor of the interface. Possible values include:
  - QSFP
  - SFP
  - RJ-45
- **Port Count**: Number of ports.
- **Lane Speed**: Speed (in Gb/s) at which the component is operating.
- **Transceiver Type**: Details about the transceiver that is plugged into the connector port. If nothing is plugged into a QSFP port, the value displayed is `Unused`. If a transceiver is plugged in, but the type has not been explicitly specified and the type cannot be auto-detected, the value displayed is `Unknown`. Transceiver Type is not applicable for RJ-45 connectors; the value displayed is `-`.

Example 1

```
purehw list --csv
```

Displays hardware component information for all components in the FlashBlade array. The output is displayed in CSV format.

Example 2

```
purehw list --spec
```

Displays the model and serial numbers for each of the hardware components.

Example 3

```
purehw list CH1.FB4
```

Displays hardware component information for the blade in slot 4 of the chassis.

Identifying Hardware Components

The **`purehw setattr`** command identifies hardware by illuminiating the hardware's LED. This is useful for diagnosing hardware-related problems.

Example 4

```
purehw setattr --identify on CH1.FB1
```

Illuminates the LED for flash blade FB1 in chassis CH1.

Setting Connector Attributes

The **`purehw connector setattr`** command is used to set the attributes for a connection interface.

Example 5

```
purehw connector setattr --port-count 4 --lane-speed 10Gbps CH1.FM1.ETH1
```

The QSFP connector FM1.ETH1 is set to 4 ports with 10Gb/s lane speed.

See Also

- *purealert*
- *purearray*
- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purekeytab

purekeytab, purekeytab-create, purekeytab-delete, purekeytab-list — uploads, displays, and manages keytabs on the FlashBlade array.

Synopsis

purekeytab create --prefix ***PREFIX*** --keytab-file

purekeytab delete ***KEYTAB_NAME...***

purekeytab list [--cli|--csv|--nvp] [--no-title] [--raw] [--filter ***FILTER***] [--page] [--limit ***LIMIT***] [--sort ***SORT***] [--security| --keytab-file] [***KEYTAB_NAME...***]

Sub-Commands

`create`

Creates a keytab.

`delete`

Deletes keytabs.

`list`

Displays keytabs.

Arguments

KEYTAB_NAME

The name of the keytab.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--keytab-file`

When used with the `create` subcommand, imports a Base64 MIME-formatted keytab file.

When used with the `list` subcommand, displays the keytab as a Base64 MIME-formatted file.

`--prefix` ***PREFIX***

Specifies the prefix used for naming keytab entries.

`--security`

Displays security-related fields.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

`--raw`

Displays the unformatted version of column titles and data.

`--filter FILTER`

Displays only the rows that meet the filter criteria specified.

`--limit LIMIT`

Limits the size of the list output to the specified maximum number of rows.

`--page`

Turns on interactive paging.

`--sort SORT`

Sorts the list output in ascending or descending order by the column specified.

Description

The **purekeytab** command allows you to upload and manage keytab files on a FlashBlade array. This allows you to configure files systems to require that NFSv4.1 clients authenticate with a Kerberos client ticket before they can access the file system.

Uploading Keytab Files

Keytab files can be uploaded to a FlashBlade array using the **purekeytab create** command. You must specify a file entry prefix using the `--prefix` option. Specifying a file entry prefix is required because a single keytab file can contain multiple keytab entries in multiple slots. By providing a prefix, you can view attributes of, delete, or export individual keytab file entries without the overhead of managing all entries from the file at once. You must also specify the `--keytab-file` option, followed by supplying a Base64-MIME-formattefd file via an interactive prompt. Note that before uploading a keytab file using the CLI, the file must be encoded into Base64—Purity only accepts MIME-type Base64.

Viewing Keytab Files

To view uploaded keytabs, issue the **purekeytab list** command. The following information is displayed:

- **Name**: The name of the keytabs.
- **Principal**: The service principal name.
- **FQDN**: The fully qualified domain name that the client will mount the file system over for NFS. This can be any of the FQDNs that have been registered for data VIPs on the FlashBlade.
- **Realm**: The Kerberos realm that issued the keytab.

Issuing the **purekeytab list --keytab-file** command returns the keytab data as a Base64 MIME-formatted file.

Issuing the **purekeytab list --security** command returns the following information:

- **Name**: The name of the keytab entry.
- **Encryption Type**: The suite of encryption ciphers used for kerberized communication with clients whose Kerberos tickets are issued against this keytab.
- **KVNO**: The version number of the key used to issue the keytab.

Deleting Keytab Files

Keytabs can be removed at any time using the **purekeytab delete** command. When issuing the command, you must specify the name of the keytab being deleted.

Example 1

```
purekeytab create --prefix file1 --keytab-file
```

Uploads a MIME-formatted keytab file to the FlashBlade array and inserts a prefix of file1 for the keytab file name.

Example 2

```
purekeytab list
```

Displays all keytab entries uploaded to the FlashBlade array.

Example 3

```
purekeytab delete file1.1
```

Deletes the keytab file `file1.1` from the FlashBlade array.

Example 4

```
purekeytab list file1.1 file1.3 --keytab-file
```

Displays a Base64 MIME-formatted keytab file comprised of the keytab entries `file1.1` and `file1.3` from the FlashBlade array.

See Also

- *purearray*
- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purelag

purelag, purelag-add, purelag-create, purelag-delete, purelag-list, purelag-remove — displays and manages the link aggregation group (LAG) on the array.

Synopsis

```
purelag add --port PORT  NAME

purelag create --port PORT  NAME

purelag delete  NAME…

purelag list [--cli | --csv | --nvp] [--notitle] [--filter FILTER]
[--sort SORT] [--limit LIMIT] [--raw] [--page] [NAME..]

purelag remove --port PORT  NAME

purelag setattr --port PORT  NAME
```

Sub-Commands

`add`

Adds ports to a link aggregation group.

`create`

Creates a link aggregation group.

`delete`

Deletes a link aggregation group.

`list`

Displays link aggregation groups.

`remove`

Removes ports from link aggregation groups.

`setattr`

Sets ports for link aggregation groups.

Arguments

NAME

The link aggregation group name.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--port` ***PORT***

Adds ports to or removes ports from a LAG. Enter multiple ports as comma-separated values.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data.

Options that manage display results:

`--filter` ***FILTER***

Displays only the rows that meet the filter criteria specified.

`--limit` ***LIMIT***

Limits the size of the list output to the specified maximum number of rows.

`--sort` ***SORT***

Sorts the list output in ascending or descending order by the column specified.

Description

The **`purelag`** command configures a link aggregation group (LAG) of Ethernet ports on the array. This concept is also known as a port channel, or a bond group. Multiple LAGs can be used to connect to separate physical networks. By default, the array is preconfigured with all ports in a single LAG named `uplink` which cannot be deleted. Ports can be removed from the `uplink` LAG and added to another LAG.

Viewing a LAG

The **`purelag list`** command displays the details of all LAG configurations. Each LAG listed displays the following information:

- **Name**: Assigned name of the LAG.
- **Status**: Condition of the LAG. If the status is `healthy` then all ports in the LAG are functioning as expected.
- **Ports**: Port names associated with a LAG.
- **Port Speed**: Maximum speed of each port.
- **LAG Speed**: Combined speed of all ports in the LAG.
- **MAC**: Unique media access control (MAC) address assigned to the network interface. This field cannot be modified.

The following example displays the LAG attributes for `uplink` LAG. Eight ports with a speed of 10 Gb/s each are associated with the LAG. The combined speed of all ports in the `uplink` LAG is 80 Gb/s. Note that the `uplink` LAG is preconfigured and cannot be deleted.

```
purelag list
Name     Status   Ports           Port Speed  LAG Speed   MAC
uplink   healthy  CH1.FM1.ETH1.1  10.00 Gb/s  80.00 Gb/s  24:a9:37:11:9c:ce
                  CH1.FM1.ETH1.2
                  CH1.FM1.ETH1.3
                  CH1.FM1.ETH1.4
                  CH1.FM2.ETH1.1
                  CH1.FM2.ETH1.2
                  CH1.FM2.ETH1.3
                  CH1.FM2.ETH1.4
```

Creating a LAG

The **`purelag create --port`** command creates a new LAG configuration. Enter multiple ports as comma-separated values and assign a name for the LAG. Follow these guidelines when creating a LAG:

- All ports in a LAG must be of the same speed.
- Ports must be created in pairs, mirroring FM1 and FM2. For example, ETH1.1 from FM1 and ETH1.1 from FM2 are a pair of mirrored ports from each FM.
- Ports can only be assigned to one LAG. Ports must be removed from one LAG before being added to another LAG.

Adding Ports to a LAG

The **`purelag add --port`** command adds additional ports to an existing LAG. Enter multiple ports as comma-separated values and assign them to an existing LAG configuration. Follow these guidelines when adding ports to a LAG:

- All ports in a LAG must be of the same speed.
- Ports must be added to a LAG in pairs, mirroring FM1 and FM2. For example, ETH1.1 from FM1 and ETH1.1 from FM2 are a pair of mirrored ports from each FM.
- Ports can only be assigned to one LAG. Ports must be removed from one LAG before being added to another LAG.

Removing Ports from a LAG

The **`purelag remove --port`** command removes ports from a LAG configuration. Ports must be removed from a LAG in pairs, mirroring FM1 and FM2. For example, ETH1.1 from FM1 and ETH1.1 from FM2 are a pair of mirrored ports from each FM.

Deleting a LAG

The **`purelag delete`** command followed by the name of the LAG will delete the entire LAG configuration and unbinds the ports.

Example 1

```
purelag create --port CH1.FM1.ETH1.1,CH1.FM2.ETH1.1 LAG01
```

Creates a new LAG named `LAG01` that consists of ports `CH1.FM1.ETH1.1` and `CH1.FM2.ETH1.1`.

Example 2

```
purelag delete LAG01
```

Deletes the `LAG01` LAG and unbinds ports `CH1.FM1.ETH1.1` and `CH1.FM2.ETH1.1`.

See Also

- *purehw*
- *puresubnet*
- *purenetwork*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purelifecycle

purelifecycle, purelifecycle-rule-create, purelifecycle-rule-delete, purelifecycle-rule-disable, purelifecycle-rule-enable, purelifecycle-rule-list, purelifecycle-rule-setattr—manages object lifecycle configuration rules.

Synopsis

```
purelifecycle rule create --bucket BUCKET [--rule-id RULE_ID]
--keep-previous-version-for KEEP_PREVIOUS_VERSION_FOR [--prefix PREFIX]

purelifecycle rule delete [--bucket BUCKET] [--rule-id RULE_ID] [RULE...]

purelifecycle rule disable [--bucket BUCKET] [--rule-id RULE_ID] [RULE...]

purelifecycle rule enable [--bucket BUCKET] [--rule-id RULE_ID] [RULE...]

purelifecycle rule list [--cli|--csv|--nvp] [--notitle] [--raw]
[--filter FILTER] [--page] [--limit LIMIT] [--sort SORT] [--bucket BUCKET]
[--rule RULE_ID] [--rule-id RULE_ID] [RULE...]

purelifecycle rule setattr
[--keep-previous-version-for KEEP_PREVIOUS_VERSION_FOR] [--prefix PREFIX]
[--bucket BUCKET] [--rule-id RULE_ID] [RULE...]
```

Sub-Parsers

`rule`

When specified, indicates that sub-commands are to be processed for object lifecycle rule operations.

Sub-Commands

`create`

Creates a lifecycle rule.

`delete`

Deletes lifecycle rules.

`disable`

Disables lifecycle rules.

`enable`

Enables lifecycle rules.

`list`

Displays lifecycle rules.

`setattr`

Sets lifecycle rules attributes.

Arguments

RULE

The rule name in the format `<bucket_name>/<rule_id>`.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--bucket` ***BUCKET***

The bucket to which the rule is applied.

`--keep-previous-version-for` ***KEEP_PREVIOUS_VERSION_FOR***

The time duration in minutes, hours, days, or weeks, after which old object versions are marked as expired, specified as `N[m/h/d/w]`. The minimum valid retention period is 1 day. Note that the value entered must be in a multiple of 1 day increments. For example: 24h, 1440m, etc.

`--prefix` ***PREFIX***

Specifies a prefix to filter objects to which the rule applies. The prefix cannot exceed 1,024 characters.

`--rule-id` ***RULE_ID***

Specifies the rule ID (unique for the bucket). The ID cannot exceed 255 characters.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--raw`

Displays the unformatted version of column titles and data.

Options that manage display results:

`--filter`

Displays only the rows that meet the filter criteria specified.

`--limit` ***`LIMIT`***

Specifies the number of clients with client-specific performance displayed. By default 100 clients are displayed. A maximum of 100,000 is allowed. This option is allowed for use with only the `--client` option.

`--page`

Turns on interactive paging.

`--sort SORT`

Sorts the list output in ascending or descending order by the column specified.

Description

The **`purelifecycle`** command allows you to create and manage rules governing object lifecycles.

Note: If bucket versioning was previously enabled, but currently suspended, the versions created while versioning was enabled are still subject to any lifecycle rules created after versioning was suspended.

When bucket versioning is enabled (using, for example, the **`purebucket enable --versioning`** command), multiple variants of an object in the same bucket are created that retain the object's entire history on every update. In order to help manage space usage, the **`purelifecycle`** command allows you to create rules to automatically delete old, unneeded versions of the object. Without an object lifecycle rule, an object's versions are retained indefinitely until manually deleted.

Note: When viewing, deleting, disabling, enabling, or modifying an object lifecycle rule, you have the option to enter the `--bucket` and `--rule-id` arguments together, or the `RULE` argument (specifying the rule name in the `<bucket_name>/<rule_id>` format)—these are equivalent. For example, specifying `--bucket bucket1 --rule-id rule1` is equivalent to specifying `bucket1/rule1` as the rule name.

Creating an Object Lifecycle Rule

When an object lifecycle rule is created, that rule manages the amount of time that the 'non-current' version of object is retained. For example, you can specify that a non-current version be retained for 10 days. When the 10 day retention period is reached, that version will be deleted from the system within 24 hours.

When a lifecycle policy is applied, then for all objects in the bucket or, if specified, for those objects matching the key prefix, all versions that have been noncurrent for more than the specified retention period are subject to immediate deletion.

Up to 1,000 object lifecycle rules can be created per bucket.

To create an object lifecycle rule, issue the **`purelifecycle rule create`** command. You must specify the name of the bucket in which the object resides using the `--bucket` option, as well as the retention period using the `--keep-previous-version-for` option. Note that the value entered must be in a multiple of 1 day increments. For example: 24h, 1440m, etc. You can optionally specify a unique rule ID (`--rule-id`—if not used the rule ID will default to `fbRuleIdX`, where X is the next available rule ID number in the `fbRuleIdX` format). Note that the rule ID cannot exceed 255 characters in length.

In the following example, an object lifecycle rule with the ID `rule1` is created for the bucket `logbucket`. Non-current object versions are retained for 30 days (`30d`):

```
purelifecycle rule create --bucket logbucket --rule-id rule1 --keep-previous-
version-for 30d
```

When creating object lifecycle rules, you can also optionally specify a prefix to filter objects within the bucket to which the rule is applied. For example, if you specify `--prefix /tmp/`, the created rule will apply to all objects with the prefix `/tmp/`. Note that the prefix cannot exceed 1,024 characters.

In the following example, an object lifecycle rule with the ID `rule2` is created for the bucket `logbucket`. In addition to specifying a one week retention period for non-current object versions, the prefix `/logs/` is also specified resulting in the new object lifecycle rule applying to only objects with the prefix `/logs/`:

```
purelifecycle rule create --bucket logbucket --rule-id rule2 --keep-previous-
version-for 1w --prefix /logs/
```

Once an object lifecycle rule is created, it is automatically enabled.

Viewing Object Lifecycle Rules

Issue the **`purelifecycle rule list`** command to view lifecycle rules.

Note: Wildcards are not supported in names used in CLI or REST because lifecycle rule IDs support special characters.

Note: A maximum of 10 buckets containing rules can be viewed. Use the **`purelifecycle rule list --bucket`** command to view rules for the specified bucket. If you wish to change this limit, contact Pure Support.

In the following example, object lifecycle rules for bucket `dup3` are listed:

```
purelifecycle rule list --bucket dup3
Bucket Name       Rule ID  Enabled    Prefix    Keep Previous Version For
dup3              rule1    True       /log/     1d
```

The following information is displayed:

- `Bucket Name` - The name of the bucket on which the rule resides.
- `Rule ID` - The rule ID. This ID is either created by the system or created by the user. If system-assigned, the ID is in the format `fbRuleIdX`, where X increases by one with each rule.
- `Enabled` - If the rule is enabled (`True`) or disabled (`False`).
- `Prefix` - If specified during creation, the object prefix. When specified, the object lifecycle rule is applied to only those objects with the prefix.
- `Keep Previous Version For` - The duration that the non-current object version is retained.

Editing an Object Lifecycle Rule

An object lifecycle rule can be edited using the **`purelifecycle rule setattr`** command. All configuration attributes that are available at creation can be edited.

In the following example, the object version retention period for the rule with the ID `rule1` is updated to 1 week (`1w`):

```
purelifecycle rule setattr --rule-id rule1 --bucket logbucket --keep-previous-
version-for 1w
```

Disabling and Enabling an Object Lifecycle Rule

To disable an object lifecycle rule, issue the **`purelifecycle rule disable`** command. You must specify either the `--rule-id` and/or `--bucket` option, or the rule name.

In the following example, the object lifecycle rule `logbucket/rule1` is disabled:

```
purelifecyle rule disable logbucket/rule1
```

Similarly, to enable an object lifecycle rule, issue the **`purelifecycle rule enable`** command to enable an object lifecycle rule. You must specify either the `--rule-id` and/or `--bucket` option, or the rule name.

In the following example, the rule `logbucket/rule1` is enabled:

```
purelifecycle rule enable logbucket/rule1.
```

Deleting an Object Lifecycle Rule

To delete an object lifecycle rule, issue the **`purelifecycle rule delete`** command. You must specify either the `--rule-id` and/or `--bucket` option, or the rule name.

In the following example, the object lifecycle rule `logbucket/rule1` is deleted:

```
purelifecyle rule delete logbucket/rule1
```

Example 1

```
purelifecycle rule create --bucket bucket1 --rule-id rule1 --keep-previous-
version-for 2d
```

Creates a new object lifecycle rule with the ID `rule1` and retains the non-current object version for 2 days.

Example 2

```
purelifecycle rule setattr --keep-previous-version-for 3d --prefix /tmp/ bucket1/
rule1
```

Previously the object lifecycle rule `bucket1/rule1` was created with an object version retention period of 2 days and applied to all objects. This command modifies the retention period to three days and is applied to only objects with the prefix `/tmp/`.

Example 3

```
purelifecycle rule list --bucket bucket1
```

Displays the object lifecycle rules for the bucket `bucket1`.

Example 4

```
purelifecycle rule disable bucket1/rule1
```

Disables the object lifecycle rule `bucket1/rule1`.

Example 5

```
purelifecycle rule enable bucket1/rule1
```

Enables the object lifecycle rule `bucket1/rule1`.

Example 6

```
purelifecycle rule delete bucket1/rule1
```

Deletes the object lifecycle rule `bucket1/rule1`.

See Also

- *purebucket*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purelog

purelog, purelog-create, purelog-delete, purelog-global, purelog-list, purelog-setattr, purelog-test — creates, manages, and displays log server attributes and configurations.

Synopsis

```
purelog create --uri URI NAME

purelog delete NAME...

purelog global list [--cli|--csv|--nvp] [--no-title] [--raw]
[--filter FILTER] [--page]] [--limit LIMIT] [--sort SORT]

purelog global setattr [--ca-certificate CERTIFICATE]
[--ca-certificate-group CERTIFICATE-GROUP]

purelog list [--cli|--csv|--nvp] [--no-title] [--raw] [--filter FILTER]
[--page]] [--limit LIMIT] [--sort SORT] NAME...

purelog setattr [--uri URI] NAME

purelog test
```

Sub-Parsers

`global`

When specified, indicates that sub-commands are to be processed to manage global log server configuration attributes.

Subcommands

`create`

Creates a log server configuration.

`delete`

Deletes a log server configuration.

`list`

Displays log server configurations.

When used with the `global` sub-parser, displays global log server configuration.

`setattr`

Sets log server configuration attributes.

When used with the `global` sub-parser, sets global log server configuration attributes.

`test`

Tests log server configurations.

Arguments

NAME

The name of the log server configuration.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--ca-certificate` ***CA-CERTIFICATE***

Specifies the CA certificate to be imported. This is the name of CA certificate imported using the **`purecert`** command or GUI.

`--ca-certificate-group` ***CA-CERTIFICATE-GROUP***

Specifies the CA certificate group to be imported.

`--uri` ***URI***

The Uniform Resource Identifier (URI) of the remote log server. The URI must be specified in the format `PROTOCOL://HOSTNAME:PORT`, where:

- `PROTOCOL` is TCP, TLS, or UDP. Note that this is not case-sensitive.
- `HOSTNAME` is the syslog server's hostname or IP address. If specifying an IP address, for IPv4, specify the IP address in the form ddd.ddd.ddd.ddd, where ddd is a number ranging from 0 to 255 representing a group of 8 bits.
  For IPv6, specify the IP address in the form [xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx], where xxxx is a hexadecimal number representing a group of 16 bits. Enclose the entire address in square brackets ([]). Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (::).
- `PORT` is the port at which the server is listening. Append the port number after the end of the entire address. This is optional if the standard port number 514 for UDP or TCP, or the standard port number 6514 for TLS, is used.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

`--raw`

Displays the unformatted version of column titles and data.

`--filter FILTER`

Displays only the rows that meet the filter criteria specified.

`--limit LIMIT`

Limits the size of the list output to the specified maximum number of rows.

`--page`

Turns on interactive paging.

`--sort SORT`

Sorts the list output in ascending or descending order by the column specified.

Description

The syslog server feature enables you to forward syslog messages to remote servers. The **`purelog`** command is used to configure, manage, and view remote syslog server configurations.

Creating a Syslog Server Connection

The **`purelog create`** command is used to create a remote syslog server connection. You can connect up to thee remote syslog servers.

To create a new remote syslog connection, issue the **`purelog create`** command and specify the remote syslog server's URI using the `--uri` option as well as the name of the syslog server configuration. The URI must be specified in the format `PROTOCOL://HOSTNAME:PORT`, where:

- `PROTOCOL` is `TCP`, `TLS`, or `UDP`.
  Note that when establishing TLS communication, the syslog server must be configured with either a CA certificate or CA certificate group (for multiple remote syslog servers). Additionally, your firewall must be set to allow in-bound and out-bound traffic.
- `HOSTNAME` is the syslog server's hostname or IP address. If specifying an IP address, for IPv4, specify the IP address in the form ddd.ddd.ddd.ddd, where ddd is a number ranging from 0 to 255 representing a group of 8 bits.
  For IPv6, specify the IP address in the form [xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx], where xxxx is a hexadecimal number representing a group of 16 bits. Enclose the entire address in square brackets ([]). Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (::).
- `PORT` is the port at which the server is listening. Append the port number after the end of the entire address. This is optional if the standard port number 514 for UDP or TCP, or the standard port number 6514 for TLS, is used.

In the following example, a connection is created for the syslog server `LOGSERVER1` running on the host `myhost.com` using `TCP` at the port `614`.

```
purelog create --uri TCP://myhost.com:614 LOGSERVER1
```

Selecting CA Certificates for Syslog Server Connections

If you are using the TLS protocol, each remote syslog servers' certificate must have been signed by the configured CA certifcate, or by one of the CA certificates in the configured CA certificate group. As a best practice, it is recommended that you create a separate CA certificate group for syslog certificates.

Use the **`purelog global setattr`** command with either the `--ca-certificate` or `--ca-certificate-group` option to apply a CA certificate or CA certificate group for your syslog server connection. If clearing a previous CA certificate or CA certificate group entry, specify `""`.

In the following example, the CA certificate MyCert is applied to the syslog server connection configuration for TLS communication:

```
purelog global setattr --ca MyCert
```

Testing Remote Syslog Server Connections

Once you have configured the remote syslog servers, test their connections using the **`purelog test`** command. When issued, a test message is sent to configured syslog servers. If successful, the command's output will display `True` under the `Success` column.

Note: A test message is sent to the remote syslog server. You must manually check the remote syslog server to confirm that the message was logged at the destination. Issues with firewall settings, IPv6

connection, URI typos, invalid certificate configurations, etc., can result in the test message delivery failure.

```
purelog test
Name             Component Name  Component Address  ...  Success  ...
NagiosLogServer  fm1             10.242.11.103      ...  True     ...
NagiosLogServer  fm2             10.242.11.104      ...  True     ...
syslog-ng        fm1             10.242.11.103      ...  True     ...
syslog-ng        fm2             10.242.11.104      ...  True     ...
```

Viewing Remote Syslog Server Connection Configurations

The **`purelog list`** command shows the configured URI for each connected remote syslog server. The following information is displayed:

- **Name**: The name of the syslog server.
- **URI**: The URI of the syslog server.

Issuing the **`purelog global list`** command displays the CA certificates or CA certificate group currently in use.

- **CA Certificate**: The name of the CA certificate. If no CA certificate is in use, `--` is displayed.
- **CA Certificate Group**: The name of the CA certificate group. If no CA certificate group is in use, `--` is displayed.

Editing Syslog Server Connections

A remote syslog server's URI information, as well as CA certificate information, can be changed at any time.

Issue the **`purelog setattr`** command to edit a remote syslog server's URI information. You must specify the new URI and the syslog server's name.

In the following example, the URI informaiton for syslog server `LOGSERVER1` is changed from `TCP://myhost.com:614` (from the previous creation example) to `TCP://mynewhost.com:614` :

```
purelog setattr --uri TCP://mynewhost.com:614 LOGSERVER1
```

Similarly, for TLS connections, you can edit the remote syslog server's CA certificate (or CA certficate group) information by using the **`purelog global setattr`** command.

In the following example, the previously CA certificate MyCert configured for the remote syslog server is changed to the new CA certificate MyNewCert:

```
purelog global setattr --ca-certificate MyNNewCert
```

Deleting Remote Syslog Server Connections

The **`purelog delete`** command deletes remote syslog server connections. Once a connection is deleted, all future syslog messages to that server are stopped and that server's configuration is removed from the system. You must specify the name of the syslog server whose connection you wish to delete. Multiple server names can be specified using a comma-separated list.

In the following example, the connection to the syslog server `LOGSERVER1` is deleted:

```
purelog delete LOGSERVER1
```

Example 1

```
purelog create --uri TCP://myhost.com:614 LOGSERVER1
```

Configures a remote syslog server connection named `LOGSERVER1` running on host `myhost.com`, using `TCP` and at the port `614`.

This immediately enables transmission of all future syslog messages to `LOGSERVER1`

Example 2

```
purelog setattr --uri TCP://mynewhost.com:614 LOGSERVER1
```

Changes the configured host for remote syslog server connection `LOGSERVER1` to `mynewhost.com`.

Example 3

```
purelog create --uri TLS://myhost.com:800 LOGSERVER2

purelog global seattr --ca-certificate MySysLogCert
```

Configures a remote syslog server named connection `LOGSERVER2` running on host `myhost.com`, using `TLS` and at the port `800`.

Because TLS protocol was specified for secure communication, the CA certificate `MySysLogCert` is applied to the configuration.

Example 4

```
purelog test
```

Tests all configured remote syslog server connections.

Example 5

```
purelog list
```

Displays all connected remote syslog servers and their URI configuration information.

Example 6

```
purelog delete LOGSERVER1
```

Stops transmission of future syslog messages to `LOGSERVER1` and deletes the configured syslog server connection.

See Also

- *purecert*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

pureman

pureman — displays man pages for Purity//FB commands.

Synopsis

`pureman` { `pureadmin` | `purealert` | `purearray` | `pureblade` | `purebucket` | `purecert` | `pureconfig` | `puredns` | `pureds` | `purefs` | `purehw` | `purekeytab` | `purelag` | `purelicense` | `purelog` | `purenetwork` | `pureobj` | `purepolicy` | `purequota` | `puresmtp` | `puresnmp` | `puresubnet` | `puresupport` | `puretarget` }

Description

Enter the command **`pureman`** with one of the arguments listed in the synopsis to display the man page for that Purity//FB command.

Help on Purity//FB subcommands is also supported.

Enter the subcommand preceded with a hyphen. For example:

```
pureman purearray-list
```

Purity//FB Conventions

Purity//FB follows certain naming and numbering conventions.

Object Names

Valid characters are letters (A-Z and a-z), digits (0-9), and the hyphen (-) character. File system names can also include the underscore (_) character. The first and last characters of a name must be alphanumeric, and a name must contain at least one letter.

Most objects in Purity//FB that can be named, including file systems, data vips, and subnets can be 1-63 characters in length.

Array names can be 1-56 characters in length. The array name length is limited to 56 characters so that the names of the individual controllers, which are assigned by Purity//FB based on the array name, do not exceed the maximum allowed by DNS.

Array and File System Sizes

Array and file system sizes are specified as integers followed by one of the suffix letters `K`, `M`, `G`, `T`, `P`, representing `KiB`, `MiB`, `GiB`, `TiB`, and `PiB`, respectively, where " `Ki` " denotes 2^10, " `Mi` " denotes 2^20, and so on.

Time Zones

Dates and times appear throughout the Purity//FB GUI and CLI to mark when an event occurred or when a change was made to the array.

Purity//FB GUI dates and times are displayed in the user's local time zone, which is determined by the browser's local time. The user's local time zone appears at the bottom of the navigation pane of the Purity//FB GUI.

Purity//FB CLI dates and times are displayed in the time zone of the array, which is set during FlashBlade installation. The array time zone appears and can be edited in the **Time** panel of the **System** > **Settings** page.

CLI Command Syntax

The Purity//FB CLI is comprised of built-in commands specific to the Purity//FB operating environment.

Purity//FB CLI commands have the general form:

```
command subcommand --options OBJECT-LIST
```

The parts of a command are:

COMMAND

> Type of FlashBlade object to be acted upon, prefixed by "pure". For example, the **`purefs`** command acts on Purity//FB file systems. Run the **`purehelp`** command to see a list of Purity//FB CLI commands.

SUBCOMMAND

> Action to be performed on the specified object. Most CLI subcommands are common to some or all object types. For example, **`puresubnet list`** lists all subnets on the array.

OPTIONS

> Options that specify attribute values or modify the action performed by the subcommand.
>
> For example, in the following command, the **`--size`** option sets the provisioned size of file system `MyFiles` to 100 gigabytes.
>
> ```
> purefs setattr --size 100G MyFiles
> ```

OBJECT-LIST

Object or list of objects upon which the command is to be operated. If a subcommand changes the object state, then at least one object must be specified. Examples of subcommands that change the object state include **`create`**, **`delete`**, and **`setattr`**. For example, **`purefs create MyFiles`** creates file system `MyFiles`. In the command synopses, OBJECT specifications that are not enclosed in square brackets (for example, ***NAME*** in `purenetwork` commands) represent ones that are required.

Passive subcommands, such as **`list`**, which do not change object state, do not require object specification. Leaving out the object is equivalent to specifying all objects of the type. For example, **`purefs list`** with no file systems specified displays information about all file systems in an array. In the command synopses, OBJECT specifications enclosed in square brackets (for example, ***[NAME]***) represent ones that are optional.

Most subcommands act on a single object. For example, in the following command, the **`create`** subcommand can only be run on a single file system to change the attributes of that file system.

```
purefs setattr --size 10T MyFiles
```

Certain subcommands can operate on multiple objects. For example, the following command creates two alert watchers.

```
purealert create watcher wendywatcher@example.com,walterwatcher@example.com
```

In the command synopses, OBJECT specifications that take in a list of objects are appended with ellipses (for example, ***ADDRESS...***) .

The following list describes the conventions used in CLI documentation:

- Text in fixed-width (Courier) font must be entered exactly as shown. For example, **`puresubnet list`**.
- Text not enclosed in brackets represents mandatory text.
- Text inside square brackets (“[]”) represents optional items. Do not type the brackets.
- Text inside curly braces (“{ }”) represents text, where one (and only one) item must be specified. Do not type the braces.
- The vertical bar (|) separates mutually exclusive items.
- Uppercase italic text represents entered text whose value is based on the nature of the subcommand or option. For example, **`--size SIZE`** , where SIZE represents the value to be entered, such as `10T`.

CLI Output

Various Purity//FB CLI subcommands, such as `list`, generate list outputs. With the ability to create and manage hundreds of file systems, list outputs can become very long. The Purity//FB CLI includes options

to control the way a list output is displayed and formatted. The CLI also includes sort, filter, and limit options to control the results you see and the order in which you see them.

Interactive Paging

Pagination divides a large output into discrete pages. Pagination is disabled by default and is only in effect if the `--page` option is specified and the number of lines in the list output exceeds the size of the window.

To interactively move through a paginated list output:

- Press the `[Right Arrow]` key or space bar to move to the next page.
- Press the `[Left Arrow]` key to move to the previous page.

To quit interactive paging and exit the list view, press `q` .

With pagination, each page of the CLI list output begins with the column titles. To suppress the column titles, run the command with the `--notitle` option.

Using Wildcards

The asterisk (*) symbol denotes a wildcard character used in list operations to represent zero or more characters in an object name. The object name can include multiple asterisks.

When running the `purefs list` command, include the asterisk in the name argument to expand the list results. For example, the `purefs list *cat*` command displays a list of all hosts that contain "cat", such as `cat`, `catnap`, `happycats`, and `lolcat`. If the asterisks were not included, only the host named `cat` would be returned. As another example, the `purefs list *fs*` command displays a list of all file systems that contain "fs", including ones that begin or end with "fs".

```
$ purefs list *fs*
Name     Size   Used  Created                  Protocols
Myfs     100G   0.00  2016-04-11 10:18:07 PDT  http
MyFs-01  100G   0.00  2016-04-11 10:18:07 PDT  http
fs       1G     0.00  2016-04-11 11:19:19 PDT  smb
fs01     100G   0.00  2016-04-11 10:17:23 PDT  nfs
```

If Purity//FB cannot find matches for the wildcard argument specified, an error message appears.

Formatting

Format options control the way list outputs are displayed. The following format options are common to most **`list`** and **`monitor`** subcommands:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration. The `--cli` output is not meaningful when combined with immutable attributes.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data.

Here is a comparison between the `purefs list --space` output with formatted column titles and data, and the same output with unformatted column titles and data:

```
$ purefs list --space --page
Name     Size  Used    Data Reduction  Physical  Snapshots  Total
epic01   5T    4.03T   2.0 to 1        2.01T      0.00       2.01T
$ purefs list --space --raw --page
name   provisioned    space.virtual  space.data_reduction space.unique
 space.snapshots
epic01 5497558138880  4429186269696  2.3                 3221998891   0
space.total_physical
42339133009304
```

Sorting

When running the `purefs list` command, include the **`--sort`** option to sort a column of the output in ascending or descending order.

The sort option adheres to the following syntax:

`--sort SORT`

SORT represents a comma-separated list of columns to sort. Use the unformatted title name (`--raw`) of the column to represent each column listed. To sort a column in descending order, append the minus (-) sign to the column name.

For example, run the following commands to get the unformatted column titles in the `purefs list` output, and then sort the same output in descending order by the `space.virtual` (`Used`) column:

```
$ purefs list --raw --page
name      provisioned    space.virtual    created    protocols
$ purefs list --sort space.virtual-
Name Size Used    Created              Protocols
FS01 15T  2.48T   2017-06-13 09:24:44  nfs
FS06 15T  2.47T   2017-06-13 09:25:54  nfs
FS02 15T  2.25T   2017-05-13 09:24:44  nfs
FS04 10T  1.99T   2017-07-22 14:40:28  smb
FS03 50T  1.99T   2017-04-09 07:52:46  nfs
FS05 2T   1.26T   2017-05-13 09:36:33  nfs
```

If multiple columns are specified, the output is sorted by the order of the columns listed.

If a sorted column contains non-unique values and a secondary sort argument is not specified, Purity//FB automatically performs a secondary sort using the default sort criteria (typically by `Name`) .

Examples

Example 1: Display a list of file systems sorted in descending order by virtual space used.

```
$ purefs list --space --sort space.virtual-
```

Example 2: Display a list of file systems sorted in ascending order by virtual space used. If any of of the file systems are identical in virtual space used, sort those file systems in descending order by name.

```
$ purefs list --space --sort space.virtual,name-
```

Filtering

When running the `purefs list` command, include the **`--filter`** option to narrow the results of the output to only display the rows that meet the filter criteria.

Filtering with Operators

When filtering with operators, use the following syntax:

```
--filter "RAW_TITLE OPERATOR VALUE"
```

RAW_TITLE represents the unformatted title name of the column by which to filter. Run the list operation with the `--raw` option to get the unformatted title name.

OPERATOR represents the type of filter match (=, !=, <, >, <=, or >=) used to compare RAW_TITLE to VALUE.

VALUE represents the value (number, date, or string) that determines the results to be included in the list output. Literal strings must be wrapped in quotes.

Filtering supports the following operators:

Operator	Description
=	Equals
!=	Does not equal
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Filtering supports the asterisk wildcard character. For example, the value "*fs*" in the `purefs list --filter "name = '*fs*'"` command displays a list of all file systems that contain "fs" in the file system name.

The AND and OR operators can be used to further refine your filter. The AND operator displays only the results that meet all of the filter criteria in the command. The OR operator displays the results that meet at least one of the filter criteria in the command.

Examples

Example 1: Display a list of file systems that are greater than or equal to 5 tebibytes in provisioned size.

```
$ purefs list --filter "provisioned >= \"5T\""
```

OR

```
$ purefs list --filter "provisioned >= 5497558138880"
```

Example 2: Display a list of empty file systems.

```
$ purefs list --filter "space.virtual = '0'"
```

Example 3: Display a list of file systems that occupy a physical space greater than or equal to 3.43 tebibytes in size, not including snapshot specific data.

```
$ purefs list --space --filter "space.unique >= 3772099408945"
```

Example 4: Display a list of file systems that begin with `file` or are greater than 5 tebibytes in size.

```
$ purefs list --filter "name = 'file*' or provisioned > \"5T\""
```

Example 5: Display a list of file systems that begin with `file` and are greater than 5 tebibytes in size.

```
$ purefs list --filter "name = 'file*' and provisioned > \"5T\""
```

Filtering with Functions

The filter option supports the CONTAINS and NOT functions.

`--filter FUNCTION(PARAMETERS)`

Function	Description
`contains(raw_title,string)`	Contains the enclosed string. Takes exactly two parameters, where `raw_title` represents the unformatted title name of the column by which to filter and `string` represents the string to search within the column. Run the list operation with the `--raw` option to get the unformatted title name.
`not(expression)`	Inverse of the enclosed expression.

Examples

Example 1: Get a list of all file systems with at least one NFS export rule set to `*(rw,no_root_squash)`.

```
$ purefs list --filter "contains(rules, '*(rw,no_root_squash)')"
```

Example 2: Get a list of file systems that do not contain "file" in the file system name.

```
$ purefs list --filter "not(contains(name, 'file'))"
```

Existence Checks

The Purity//FB CLI supports existence checks. For example, the `purefs list --filter "name"` command checks to see if the "name" column exists in the file system list output.

Examples

Example 1: Get a list of all file systems with data written to them.

```
$ purefs list --filter "space.virtual"
```

Example 2: Get a list of all file systems that do not have a provisioned size set.

```
$ purefs list --filter "not(provisioned)"
```

Setting Limits

When running the `purefs list` command, include the **`--limit`** option to restrict the result size to the limit specified.

The limit option adheres to the following syntax:

`--limit ROWS`

`ROWS` represents the maximum number of rows to return.

For example, run the following command to list only the first 5 rows of the `purefs list --space` output:

```
$ purefs list --space --limit 5
Name      Size  Used   Data Reduction  Physical  Snapshots  Total
MyFile01  5T    4.03T  2.3 to 1        4.04T     0.00       4.04T
MyFile02  5T    3.42T  2.3 to 1        3.43T     1.23T      4.66T
MyFile03  5T    3.42T  7.6 to 1        3.43T     0.00       3.43T
MyFile04  5T    3.42T  2.3 to 1        3.43T     0.00       3.43T
MyFile05  5T    3.42T  1.6 to 1        3.43T     0.00       3.43T
```

Combining Sorting, Filtering, and Limit Options

The `--sort`, `--filter`, and `--limit` options can all be combined together.

Examples

Example 1: Display the top 10 file systems that have the largest amount of pre-compressed data written to them, and include "file" in the file system name.

```
$ purefs list --sort "space.virtual-" --limit 10 --filter "name = '*file*'"
```

Example 2: Display the top 10 file systems with the largest physical space used.

```
$ purefs list --space --sort "space.unique-" --limit 10
```

CLI Command Help

CLI command help is available at both the command and subcommand levels. To use it, type the Purity//FB CLI command or subcommand followed by **`-h`** or **`--help`**. The command with the `-h` or `--help` switch displays usage information, supported syntax, and a list of subcommands for the specified command. To get help information for an Purity//FB command, type:

```
<command> -h
```

For example,

```
$ purefs -h
usage: purefs [-h] {add,create,delete,disable,enable,list,nfs,remove,setattr,snap}
…
positional arguments:
  {add,create,delete,disable,enable,list,nfs,remove,setattr,snap}
    add                 add protocols to one or more file systems
    create              create a file system
    delete              delete one or more file system(s)
    enable              enable file system attributes
    …
```

The command with the `-h` or `--help` switch displays usage information, supported syntax, and a list of options for the specified subcommand. To get help information for an Purity//FB subcommand, type:

```
<command> <subcommand> -h
```

For example,

```
$ purefs list -h
usage: purefs list [-h] [--pending | --pending-only]
                   [--space | --protocol-specific {http,nfs,smb} | --special-dir |
 --snap]
                   [--cli | --csv | --nvp] [--notitle]
                   [--total | --total-only] [--filter FILTER] [--sort SORT]
                   [--raw] [--limit LIMIT] [--page]
                   [FILE-SYSTEM …]
positional arguments:
  FILE-SYSTEM  file system name(s)
optional arguments:
  -h, --help           show this help message and exit
  --pending             list file systems including pending
  --pending-only        list pending file systems only
  --space              display space report
  --protocol-specific {http,nfs,smb}
                       display protocol specific information
  --special-dir        list status of special directories
  --snap               list snapshots
```

```
...
```

Viewing the EULA

To view the End User License Agreement, issue the **`pureman purelicense`** command.

Troubleshooting

Help! Who do I contact about problems with the FlashBlade Array?

Contact a member of your Pure Storage account team, visit us at *http://support.purestorage.com*, or email Pure Storage Support at *support@purestorage.com*.

If you are contacting Pure Storage Support about a technical issue, they may ask you to open an RA session so that they can help you diagnose the issue.

Error Messages

The following error messages may appear when you perform operations through the Purity//FB GUI or CLI:

Service Temporarily Unavailable

> An internal service is currently unavailable. This is a temporary issue. Wait a few minutes and then try the Purity//FB GUI or CLI operation again. If you still get the error message, contact Pure Storage Support at *support@purestorage.com*.

Unexpected Error

> The Purity//FB GUI or CLI operation that you performed generated an unexpected error. Contact Pure Storage Support at *support@purestorage.com*.

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purenetwork

purenetwork, purenetwork-create, purenetwork-delete, purenetwork-list, purenetwork-setattr — manages the network interfaces used to connect a FlashBlade system to a network.

Synopsis

```
purenetwork vip create --address ADDRESS --servicelist SERVICELIST NAME

purenetwork delete NAME…

purenetwork list [--cli|--csv|--nvp] [--notitle] [--filter FILTER]
[--sort SORT] [--limit LIMIT] [--raw] [--page] [--service SERVICE]
[--vlan VLAN]

purenetwork setattr [--address ADDRESS] NAME
```

Sub-Parsers

`vip`

When specified, indicates that sub-commands are to be processed for virtual interfaces.

Sub-Commands

`create`

Creates a virtual interface. Must be used with the `vip` sub-parser.

`delete`

Deletes a network interface.

`list`

Displays network interface attributes.

`setattr`

Sets network interface attributes.

Arguments

NAME

Data virtual IP address (data vip). In CLI commands, data vip names are case-sensitive.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--address` ***`ADDRESS`***

IP address to be associated with the specified data vip.

For IPv4, enter the address in CIDR notation `ddd.ddd.ddd.ddd/dd`. For example, `10.20.20.210/24`.

For IPv6, enter the address and prefix length in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`. For example, `2001:0db8:85a3::ae26:8a2e:0370:7334/64`. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`) .

`--service` ***`SERVICE`***

Lists only the interfaces configured with the specified service. The supported service for data vips is `data` and the supported service for replication vips is `replication`.

`--servicelist` ***`SERVICELIST`***

Assigns the specified service to the data vip. The supported service for data vips is `data`.

`--vlan`

Lists only the interfaces configured with the specified VLAN ID.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data.

Options that manage display results:

`--filter` ***`FILTER`***

Displays only the rows that meet the filter criteria specified.

`--limit` ***`LIMIT`***

Limits the size of the list output to the specified maximum number of rows.

`--sort` ***`SORT`***

Sorts the list output in ascending or descending order by the column specified.

Description

The **`purenetwork`** command displays and manages the data virtual IP addresses (data vips) on the FlashBlade array.

Viewing Data Vips

The **`purenetwork list`** command lists all the network interfaces on the array. Include the **`--service data`** option to display all of the data vips that have been created on the array. Include the **`--vlan`** option to display only the interfaces that are configured with the specified VLAN ID.

Creating Data Vips

Exporting a file system requires a data vip which, in turn, must be attached to a subnet. To create a data vip:

1. Verify that a subnet with the correct network prefix, VLAN ID, and gateway has been created. To view subnet details, run `puresubnet list`. For more information about subnets, refer to *puresubnet*.
2. Create the data vip and attach it to the subnet. To create a data vip, run `purenetwork create vip`.

The **`purenetwork create vip`** command creates a data vip and attaches it to a subnet. Include the required **`--address`** option to create a reachable data vip. Specify the IP address of the client, making sure it matches the network prefix of the subnet. Include the required **`--servicelist data`** option to assign the `data` service to the data vip. Once the data vip is created, it inherits the gateway, MTU and VLAN ID from the subnet. Likewise, the subnet inherits the `data` service type from the data vip.

After a data vip has been created, the next step is to export and mount the file systems. For more information about file systems, refer to *purefs*.

Configuring Data Vip Attributes

The **`purenetwork setattr`** command changes the attributes of the specified data vip, completely replacing any existing configurations. Include the **`--address`** option to change the IP address of the data vip.

Deleting Data Vips

The **`purenetwork delete`** command deletes a data vip. Once a data vip has been deleted, any clients connected through the data vip will lose their connection to the file system.

Example 1

```
purenetwork list --service data
```

Displays a list of data vips created for file system export.

Example 2

```
purenetwork create vip --address 10.10.200.10 --servicelist data DataVip01
```

Creates data vip `DataVip01` using address IP `10.10.200.10` to export the file system.

Example 3

```
purenetwork setattr --address 10.255.9.101 DataVip03
```

Changes the IP address of `DataVip03` from `10.255.9.100` to `10.255.9.101`. Note that both addresses belong to the same subnet.

Example 4

```
purenetwork delete DataVip01 DataVip02
```

Deletes data vips `DataVip01` and `DataVip02` from the FlashBlade array.

See Also

- *purearray*
- *purefs*
- *puresubnet*
- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

pureobj

pureobj, pureobj-account, pureobj-remote-credentials, pureobj-user — creates, manages, and displays object store objects.

Synopsis

pureobj account create ***ACCOUNT...***

pureobj account delete ***ACCOUNT...***

pureobj account list [--cli | --csv | --nvp] [--no-title] [--raw] [--filter ***FILTER***] [--page]] [--limit ***LIMIT***] [--sort ***SORT***] [--space] [--total | --total-only] [***ACCOUNT...***]

pureobj remote-credentials create --access-key-id ***KEY-ID*** ***REMOTE-CREDENTIALS***

pureobj remote-credentials delete ***REMOTE-CREDENTIALS***

pureobj remote-credentials list [--cli | --csv | --nvp] [--no-title] [--raw] [--filter ***FILTER***] [--page] [--limit ***LIMIT***] [--sort ***SORT***]

pureobj remote-credentials rename ***OLD-NAME*** ***NEW-NAME***

pureobj remote-credentials setattr [--access-key-id ***KEY-ID***] ***REMOTE-CREDENTIALS***

pureobj user access-key create --user ***USER*** [***ACCESS-KEY-ID***]

pureobj user access-key delete ***ACCESS-KEY-ID...***

pureobj user access-key disable ***ACCESS-KEY-ID...***

pureobj user access-key enable ***ACCESS-KEY-ID***

pureobj user access-key list [--cli | --csv | --nvp] [--no-title] [--raw] [--filter ***FILTER***] [--page] [--limit ***LIMIT***] [--sort ***SORT***] ***ACCESS-KEY-ID...***

pureobj user create ***USER...***

pureobj user delete ***USER...***

pureobj user list [--cli | --csv | --nvp] [--no-title] [--raw] [--filter ***FILTER***] [--page] [--limit ***LIMIT***] [--sort ***SORT***] ***USER...***

Sub-Parsers

`access-key`

When specified, indicates that sub-commands are to be processed for access keys. This sub-parser can only be used with the `user` sub-parser.

`account`

When specified, indicates that sub-commands are to be processed for object store accounts.

`remote-credentials`

When specified, indicates that sub-commands are to be processed for object store remote credentials.

`user`

When specified, indicates that sub-commands are to be processed for object store users.

Subcommands

`create`

When used with the `account` sub-parser, creates an object store account.

When used with the `remote-credentials` sub-parser, creates remote credentials.

When used with the `user` sub-parser, creates an object store user.

When used with the `user` and `access-key` sub-parsers, creates or imports an access key.

`delete`

When used with the `account` sub-parser, deletes an object store account.

When used with the `remote-credentials` sub-parser, deletes remote credentials.

When used with the `user` sub-parser, deletes an object store user.

When used with the `user` and `access-key` sub-parsers, deletes an access key.

`disable`

When used with the `user` and `access-key` sub-parsers, disables an access key.

`enable`

When used with the `user` and `access-key` sub-parsers, enables an access key.

`list`

When used with the `account` sub-parser, displays object store accounts.

When used with the `remote-credentials` sub-parser, displays remote credentials.

When used with the `user` sub-parser, displays object store users.

When used with the `user` and `access-key` sub-parsers, displays access keys.

`rename`

Renames remote credentials. Must be used with the `remote-credentials` sub-parser.

`setattr`

Sets remote credential attributes. Must be used with the `remote-credentials` sub-parser.

Arguments

ACCESS-KEY-ID

The access key. Can be used with the `access-key` sub-parser followed by the `create`, `delete`, `disable`, `enable`, or `list` subcommands.

When used with the `create` subcommand, specifies the access key to be imported.

ACCOUNT

The account name. Must be used with the `account` sub-parser.

OLD-NAME

The target's original name. Must be used with the `remote-credentials` sub-parser.

NEW-NAME

The target's new name. Must be used with the `remote-credentials` sub-parser.

REMOTE-CREDENTIALS

The remote credentials in the format `<remote-name>/<remote-credentials-name>`, for example `fb1/credentials1`. Must be used with the `remote-credentials` sub-parser.

USER

The user name in the format `<account-name>/<user-name>`, for example `account1/warren`. This must be used with the `user` sub-parser followed by the `create`, `delete`, or `list` subcommands.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--access-key-id` ***KEY-ID***

Sets the access key ID.

`--user` ***USER***

When used with the `user` and `access-key` sub-parsers and the `create` sub command, specifies the user to which the access key belongs.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

`--raw`

Displays the unformatted version of column titles and data.

`--filter FILTER`

Displays only the rows that meet the filter criteria specified.

`--limit LIMIT`

Limits the size of the list output to the specified maximum number of rows.

`--page`

Turns on interactive paging.

`--sort SORT`

Sorts the list output in ascending or descending order by the column specified.

Description

The **`pureobj`** command is used to create, manage, and display object store objects such as users, accounts, access keys, and credentials.

Creating and Deleting Accounts

The **`pureobj account`** command is used to create and manage all object store accounts. Purity//FB supports multitenancy object store accounts. More than one account can exist on the array and is logically separated, meaning each account can be administered independently from other accounts.

To create a new object store account, run the **`pureobj account create`** command and specify the new account name. Multiple accounts can be created by specifying the account names in a space-separated list, for example, `pureobj account create pureaccount1 pureaccount2 pureaccount3`. Accounts must be created before an object store user can be created.

Accounts are deleted using the **`pureobj account delete`** command. Similar to account creation, a single account, or multiple accounts can be specified for deletion.

Viewing an Account

The **`pureobjaccount list`** command displays the creation date for all object store accounts. Each account listed displays the following information:

- **Name**: The name of the object store account.
- **Created**: The time the account was created.

Issuing the **`pureobjaccount list --space`** command provides additional details about the object stores' space consumption.

- **Used**: The amount of used space.
- **Data Reduction**: Ratio of the size of the written data (i.e. Used) versus the amount of space the data occupies after data compression (i.e. Physical).
- **Physical**: Amount of space that the written data occupies after reduction via data compression.
- **Object Count**: The amount (in GB) of objects.

Creating and Deleting Users

The **`pureobj user create`** command creates users for an object store account. Users are created to administer the object storage for a specified object store account. A user name must be unique and prefixed with the account name it is assigned, thus requiring an account to be created before creating a user for that account.

To create a user `user1` for an object store account named `pureaccount`, add `pureaccount` as a prefix followed by a forward slash. For example, `pureaccount/user1`.

```
pureobj user create pureaccount/user1
```

Issue the **`pureobj user delete`** command to delete a user. The user name must be specified in the format `<objectstore account>/<user>`. For example, `pureaccount/user1`.

Creating an Access Key

An access key allows the user to administer the object store. A maximum of two sets of keys can be created for each user. A set of keys consists of an access key ID and Secret Access Key.

The Secret Access Key will only be displayed as unencrypted once during the creation of a new access key. Copy and paste the Secret Access Key if necessary and place in a secure location.

Use the **`pureobj user access-key create --user`** command followed by the user name, including the account prefix. For example:

```
pureobj user access-key create --user pureaccount/user1
```

An access key ID and secret access key is created for the user named `pureaccount/pureuser`.

Importing Access Keys

Note: FlashBlade does not validate if the access key being imported from another FlashBlade was a previously deleted access key during or after import.

Credentials generated on one FlashBlade can be imported on another FlashBlade for a single account, with data on both arrays being accessible using the same credentials. This can be useful in FlashBlade replication setups, or in environments where multiple FlashBlades are used in multi-site deployments of Splunk SmartStore. When importing access keys, the following should be noted:

- Only new access keys created after Purity//FB 3.0.0 can be imported on another FlashBlade.
- The access key you are importing cannot already exist on the FlashBlade to which it is being imported.
- If there are two access keys present for a user, you cannot import a third access key.
- Credentials are not synchronized or validated against the FlashBlade from which they were imported; deleting or disabling a set of credentials on the source cluster will not delete or disable the credentials on any other clusters where they were imported.

As a best practice, Pure Storage recommends that customers assign a unique access key for each individual user and each application, and avoid sharing an access key across two FlashBlade arrays except when specifically required by your application configuration (for example, Splunk SmartStore configurations on FlashBlade that use a second FlashBlade for disaster recovery).

To import an access key, on the FlashBlade to which you are importing the key, issue the **`pureobj user access-key create`** command with the `--user` USER and ACCESS-KEY-ID

arguments, where USER is the user to which the access key is being imported and ACCESS-KEY-ID is the access key that is being imported. For example:

```
pureobj user access-key create pureaccount/user1
 PSFBSAZRPAKJMOKLFDENLDKBPAJBHEKGMNAFEXAMPLE
```

In the example above, PSFBSAZRPAKJMOKLFDENLDKBPAJBHEKGMNAFEXAMPLE is the access key being imported to the user `pureaccount/user1`.

Once entered, you are prompted to enter the secret access key.

Deleting, Disabling, and Enabling Access Keys

An access key for a user is enabled by default. Keys can be disabled for temporary purposes and re-enabled at a later time. Additionally, keys can be deleted and keys must be generated again for the user to gain access to the object store. Use the following commands for each action:

- **`pureobj user access-key disable`** followed by the access key ID to disable a user's access key.
- **`pureobj user access-key enable`** followed by the access key ID to re-enable a user's access key.
- **`pureobj user access-key delete`** followed by the access key ID to delete a user's access key.

Viewing Users and Access Keys

The **`pureobj user list`** command displays the details of all user configurations. Each user listed displays the following information:

- **Name**: Assigned name of the user.
- **Access Key ID**: The access key ID created for the specific user.
- **Created**: The time the user identity was created.

The **`pureobj user access-key list`** command displays all users with access key IDs.

- **Access Key ID**: The access key ID created for the specific user.
- **Enabled**: The status of the access key can be set to True or False.
- **Secret Access Key**: The secret access key displayed as `****`.
- **User**: The name of the user.
- **Created**: The time the access key was created for the user.

Creating Remote Credentials

Remote credentials are required to set up replica links between FlashBlade arrays and S3 targets for object replication. To create remote credentials, you must have previously created an access key ID by issuing the **`pureobj user access-key create`** command on the connected array, or have the access key/secret key from your Amazon S3 account.

Create remote credentials by issuing the **`pureobj remote-credentials create`** command. In addition to specifying the access key, you must specify a unique remote credential name prefixed with the remote target it is assigned. For example, `s3r1/s3c1`. Note that once the command issued, you are prompted to enter the secret access key.

```
pureobj remote-credentials create --access-key-id ABCDE12345EXAMPLE s3r1/s3c1
```

Editing Remote Credentials

Remote credentials can be renamed using the **`pureobj remote-credentials rename`** command. You must specify the original name of the credentials (`OLD-NAME`) and the new name (`NEW-NAME`). For example:

```
pureobj remote-credentials rename s3remote1/purecredential s3remote1/s3c1
```

The access key associated with the remote credential can also be changed. Issue the **`pureobj remote-credentials setattr`** command with the `--access-key-id` option. You must specify the remote credentials to which the access key change will apply. For example, to change the credential `s3r1/s3c1` access key from `ABCDE12345EXAMPLE` to `EFGHI56789EXAMPLE` :

```
pureobj remote-credentials setattr --access-key-id EFGHI56789EXAMPLE s3r1/s3c1
```

Note that you will be prompted to re-enter the secret access key.

Deleting Remote Credentials

To delete a remote credential, issue the **`pureobj remote-credentials delete`** command. The remote credentials that you are deleting must be specified in the format`<remote-name>/<remote-credentials-name>`, for example, `s3r1/s3c1`. Multiple remote credentials can be deleted by specifying the remote credentials in a space-separated list, for example, `pureobj remote-credentials delete s3r1/s3c1 s3r2/s3c2 s3r3/s3c3`.

Example 1

```
pureobj account create pureaccount1 pureaccount2 pureaccount3
```

Creates three object store accounts, `pureaccount1`, `pureaccount2`, and `pureaccount3`.

Example 2

```
pureobj user create pureaccount1/user1
```

Creates the user `pureaccount1/user1` for the account `pureaccount1`.

Example 3

```
pureobj user access-key create --user pureaccount1/user1
```

Creates a secret access key pair for the user `pureaccount1/user1`.

Example 4

```
pureobj remote-credentials create --access-key-id ABCDE12345EXAMPLE s3r1/s3c1
```

Creates the remote credential `s3r1/s3c1`.

Example 5

```
pureobj user access-key create pureaccount/user1
 PSFBSAZRPAKJMOKLFDENLDKBPAJBHEKGMNAFEXAMPLE
Enter Secret Access Key: ED4F1D950Dc6a47190/a2d85AED7AEB84abEXAMPLE
Retype Secret Access Key: ED4F1D950Dc6a47190/a2d85AED7AEB84abEXAMPLE
Access Key ID                              Enabled  Secret Access Key  ...
PSFBSAZRPAKJMOKLFDENLDKBPAJBHEKGMNAFEXAMPLE  True     ****               ...
```

Imports the access key PSFBSAZRPAKJMOKLFDENLDKBPAJBHEKGMNAEXAMPLE for use by the user `pureaccount1/user1` on the FlashBlade from which the command is issued.

See Also

- *purearray*
- *purebucket*
- *puretarget*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purepolicy

purepolicy, purepolicy-add, purepolicy-create, purepolicy-delete, purepolicy-disable, purepolicy-enable, purepolicy-list, purepolicy-remove, purepolicy-rule-add, purepolicy-rule-remove, purepolicy-rule-list — creates, manages, and displays array snapshot policies.

Synopsis

purepolicy add --fs ***FILE-SYSTEM*** [--remote ***REMOTE***] ***POLICY...***

purepolicy create ***POLICY...***

purepolicy delete ***POLICY***

purepolicy disable ***POLICY***

purepolicy enable ***POLICY***

purepolicy list [--cli|--csv|--nvp] [--notitle] [--member|--fs|--fs-snap] [--filter ***FILTER***] [--sort ***SORT***] [--raw] [--limit ***LIMIT***] [--page] [***POLICY...***]

purepolicy remove {--fs ***FILE-SYSTEM***|--fs-snap ***FILE-SYSTEM-SNAPSHOT***} [--remote ***REMOTE***] ***POLICY...***

purepolicy rule add --keep-for ***KEEP_FOR*** --every ***EVERY*** [--at ***AT***] [--time-zone ***TIME_ZONE***] ***POLICY...***

purepolicy rule remove --keep-for ***KEEP_FOR POLICY...***

purepolicy rule list [[--cli|--csv|--nvp] [--notitle] [--filter ***FILTER***] [--sort ***SORT***] [--raw] [--limit ***LIMIT***] [--page] [***POLICY...***]

Sub-Parsers

`rule`

When specified, indicates that sub-commands are to be processed for policy rules.

Sub-Commands

`add`

Adds file systems or replica links to policies.

When used with the `rule` sub-parser, adds policy rules.

`create`

Creates policies.

`delete`

Deletes policies.

`disable`

Disables policies.

`enable`

Enables policies.

`list`

Displays policies.

When used with the `rule` sub-parser, displays policy rules.

`remove`

Removes policies.

When used with the `rule` sub-parser, removes policy rules.

Arguments

`POLICY`

The name of the policy.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--at` ***`AT`***

The time of day a snapshot is taken, specified in the format `HH[:MM][am|pm]`; for example `23`, `5am`, `2:15am`, `14:45`. If entering a minute (`MM`) value, it must be a two digit number between 0 and 59; for example `01`, `15`. The AT time can only be set if the snapshot interval is a measured in days. If the time was previously set, specify `""` to clear the time.

`--every` ***`EVERY`***

The snapshot interval, specified in the format `N[m|h|d|w]`; for example `15m`, `1h`, `2d`, `3w`.

`--fs` ***FILE-SYSTEM***

A file system or a comma-separated list of file systems to which a policy or policies are added or removed.

`--fs-snap` ***FILE-SYSTEM-SNAPSHOT***

A file system snapshot or a comma-separated list of file system snapshots to which a policy or policies are removed.

`--keep-for` ***KEEP_FOR***

The period in which snapshots are retained before they are eradicated, specified in the format `N[m|h|d|w]`; for example `15m`, `1h`, `2d`, `3w`.

`--remote` ***REMOTE***

Specifies the remote array name.

`--time-zone` ***TIME_ZONE***

Specifies the time zone for the value specified for the `--at` option. By default, the array's time zone is used.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--fs`

Lists file systems added to policies.

`--fs-snap`

Lists file system snapshots added to policies.

`--member`

Lists file systems and file system snapshots added to policies.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient

viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data.

Options that manage display results.

`--filter` ***`FILTER`***

Displays only the rows that meet the filter criteria specified.

`--sort` ***`SORT`***

Sorts the list output in ascending or descending order by the column specified.

`--limit` ***`LIMIT`***

Limits the size of the list output to the specified maximum number of rows.

Description

Policies are rules that govern snapshot creation and retention, which are applied to file systems. In a typical workflow:

1. Create a policy.
2. Add rules to the policy.
3. Add the policy to a file system.

Creating Snapshot Policies

The **`purepolicy create POLICY`** command is used to create one or more snapshot policies. If creating multiple policies, the new policies' names, as specified by the **`POLICY`** argument, must be specified in a space-separated list. For example, **`purepolicy create pol1 pol2 pol3`**. Once a snapshot policy has been created, by default it becomes enabled.

Adding Snapshot Policy Rules

After creating a snapshot policy, add snapshot creation and retention rules using the **`purepolicy rule add`** command. Snapshot rules include:

- the snapshot interval (**`--every`**)

- the time of day snapshots are taken (**`--at`**)
- when snapshots are eradicated (**`--keep-for`**)

Once the snapshot creation and retention rules are created for the snapshot policy, apply the policy to a file system or replica link by issuing the **`purepolicy add`** command. You must specify the snapshot to which the policy is applied using the `--fs` option. For example:

```
purepolicy add --fs fs1 replicationpolicy
```

For environments where file system snapshots are replicated from a local array to a target array, use the `purepolicy add` command to apply the snapshot policy to the replica link to govern the file system replication schedule. In addition to specifying the file system, you must also use the `--remote` option and specify the target array's name. For example:

```
purepolicy add --fs fs1 --remote array2 replicationpolicy
```

Multiple retention rules can also be added to policies. Issuing **`purepolicy rule add --every 1h --keep-for 1d p1`** sets the rule for policy p1 that creates an hourly snapshot and retains that snapshot for one day. Issuing the **`purepolicy rule add --every 1d --keep-for 1w p1`** sets another rule for policy p1 that saves a snapshot (created from the first rule) each day and retains that snapshot for one week. Note that multiple policies can be specified in a space-separated list. For example:

```
purepolicy rule add --every 1h --keep-for 1d p1
Name  Every  At     Time Zone            Keep For
p1    1h     -      -                    1d

purepolicy rule add --every 2d --keep-for 1w p1
Name  Every  At     Time Zone            Keep For
p1    1h     -      -                    1d
      1d     -      -                    2w
```

Rule 1 results in 24 snapshots being created in a 24-hour period. After 24 hours, the first snapshot that was created is destroyed and another snapshot is created; this cycle is repeated for each snapshot resulting in 24 snapshots each day. Per Rule 2, since one snapshot from Rule 1 is saved each day and retained for a week, six additional snapshots are retained for an additional six days. After one week, a total of 30 snapshots will be retained.

Disabling Snapshot Policies

To disable a snapshot policy, issue the **`purepolicy disable POLICY`** command. Once the snapshot policy, as specified with the **`POLICY`** argument, is disabled the snapshot creation and retention rules are suspended for any file systems to which the policy was added. Multiple policies can be disabled at once by specifying a space-separated list of policy names.

Viewing Snapshot Policies

To view all snapshot policies, run the **`purepolicy list`** command. You can apply filters to display only policies applied to file systems, snapshots as a result of applied policies, or both file systems and snapshots using the `--fs`, `--fs-snap`, or `--member` options, respectively.

You can also view the details of all policies and their rules by issuing the **`purepolicy rule list`** command.

Modifying Snapshot Policy Rules

To modify the rules of an existing snapshot policy, you must first remove the policy's original rules using the **`purepolicy rule remove --keep-for KEEP_FOR POLICY`** command. Note that when removing rules you must specify the same snapshot retention before eradication period originally set for the policy. For example, if the policy's snapshot retention before eradication period was originally set to 2h, when removing the rule, you must specify a `--keep-for` value of 2h.

Policies with multiple rules can be removed by specifying multiple `--keep-for` values in a comma-separated list. If you wish to remove rules from multiple policies, the polices should be specified in a space-separated list. For example, if policies p1 and p2 both have multiple snapshot retention rules of 1h and 2d, issue the **`purepolicy rule remove --keep-for 1h,2d p1 p2`** command:

```
purepolicy rule remove --keep-for 1h,2d p1 p2
Name  Every  At  Time Zone  Keep For
p1    -      -   -          -
p2    -      -   -          -
```

Once the original rules are removed, issue the **`purepolicy rule add`** command and specify new of snapshot interval, creation time, and period-before-eradication parameters.

Note: Existing snapshots managed by a modified policy will be destroyed if they do not comply with the policy's new snapshot schedule rules.If you modified a policy and snapshots were destroyed, you can restore the snapshots under the policy by using the `purepolicy rule add` command to add back the removed rule. This must be performed within the 24 hour eradication period.

Removing and Deleting Snapshot Policies

Snapshot policies can be removed and deleted. For example, if a policy no longer meets your current needs, remove it from its associated file system using the **`purepolicy remove --fs FILE-SYSTEM POLICY`** command and then delete the policy using the **`purepolicy delete POLICY`** command.

Example 1

```
purepolicy create pol1 pol2 pol3
```

Creates snapshot policies `pol1`, `pol2`, and `pol3`.

Example 2

```
purepolicy rule add --every 1d --at 6am --keep-for 5d pol1
```

Sets snapshot rules for snapshot policy `pol1` so that one snapshot is taken at 6am every day and then eradicated after five days.

Example 3

```
purepolicy add --fs fs1 pol1
```

Adds snapshot policy `pol1` to file system`fs1`.

Example 4

```
purepolicy add --fs fs1 --remote array2 pol1
```

Adds snapshot policy `pol1` to the replica link for the local array and target `array2`.

Example 5

```
purepolicy disable pol1
```

Disables snapshot policy `pol1`.

See Also

- *purealert*
- *purearray*
- *purefs*
- *puresupport*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

purequota

purequota, purequota-group-create, purequota-group-delete, purequota-group-list, purequota-group-setattr, purequota-notification-disable, purequota-notification-enable, purequota-notification-list, purequota-notification-setattr, purequota-user, purequota-user-create, purequota-user-delete, purequota-user-list, purequota-user-setattr — Sets and displays user and group quotas.

Synopsis

```
purequota group create --fs FILE-SYSTEM --quota QUOTA [--gid GID|
--group-name GROUP_NAME]

purequota group delete --fs FILE-SYSTEM [--gid GID|--group-name GROUP_NAME]

purequota group list [--cli|--csv|--nvp] [--notitle] [--filter FILTER]
[--sort SORT] [--raw] [--limit LIMIT] [--page] --fs FILE-SYSTEM [--gid GID|
--group-name GROUP_NAME] [--all-usage]

purequota group setattr --fs FILE-SYSTEM [--gid GID| --group-name GROUP_NAME]
--quota QUOTA

purequota notification disable

purequota notification enable

purequota notification list [--cli|--csv|--nvp] [--notitle] [--raw]

purequota notification setattr --contact CONTACT

purequota user create --fs FILE-SYSTEM --quota QUOTA [--uid UID|
--user-name USER_NAME]

purequota user delete --fs FILE-SYSTEM [--uid UID| --user-name USER_NAME]

purequota user list [--cli|--csv|--nvp] [--notitle] [--filter FILTER]
[--sort SORT] [--raw] [--limit LIMIT] [--page] --fs FILE-SYSTEM [--all-usage]
[--uid UID| --user-name USER_NAME]

purequota user setattr --fs FILE-SYSTEM [--uid UID| --user-name USER_NAME]
- -quota QUOTA
```

Sub-Parsers

`group`

When specified, indicates that sub-commands are to be processed for group quotas.

`notification`

When specified, indicates that sub-commands are to be processed for quota notifications.

`user`

When specified, indicates that sub-commands are to be processed for user quotas.

Sub-Commands

`create`

When used with the `group` sub-parser, creates group quotas.

When used with the `user` sub-parser, creates user quotas.

`delete`

When used with the `group` sub-parser, deletes group quotas.

When used with the `user` sub-parser, deletes user quotas.

`disable`

Disables quota mail notifications. Must be used with the `notification` sub-parser.

`enable`

Enabled quota mail notifications. Must be used with the `notification` sub-parser.

`list`

When used with the `group` sub-parser, displays group quotas.

When used with the `notification` sub-parser, displays email notification information.

When used with the `user` sub-parser, displays user quotas.

`setattr`

When used with the `group` sub-parser, modifies group quotas.

When used with the `notification` sub-parser, modifies email notification contact information.

When used with the `user` sub-parser, modifies user quotas.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--all-usage`

Lists all user usage or group usage, with or without set quotas.

`--contact` ***CONTACT***

The contact information of a FlashBlade administrator included in a direct email notification to users or groups approaching or exceeding their file system quota. The contact information is a free-form string of a name, email, or phone number enclosed in quotes. For example, `"John Doe, Storage Admin Team. jdoe@example.com (555-867-5309)"`.

The contact information for the individual or organization who will receive the quota notification email alerts. The contact information is a free-form entry for names, emails, or phone numbers. The contact field cannot exceed 256 characters in length.

`--fs` ***FILE-SYSTEM***

A file system to which a group or user quota is added or removed.

`--gid` ***GID***

The group ID.

`--group-name` ***GROUP_NAME***

The group name. Note that the group name maps to the sAMAccountName field of a group in Active Directory servers, and the Common Name (CN) of a group in all other types of directory servers.

`--quota` ***QUOTA***

The quota size for the group or user. The size must be an integer followed by a capacity suffix: K,M,G,T,P, or E.

`--uid` ***UID***

The user ID. Note that the user name maps to the sAMAccountName field of a user in Active Directory servers, and to the UID field of a user for all other types of directory servers.

`--user-name` ***USER_NAME***

The user name.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient

viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data.

`--filter` ***FILTER***

Displays only the rows that meet the filter criteria specified.

`--limit` ***LIMIT***

Limits the size of the list output to the specified maximum number of rows.

`--sort` ***SORT***

Sorts the list output in ascending or descending order by the column specified.

Description

Quotas can be set for users and groups on a per- file system basis to help manage file system space usage by those users and groups.

When the array detects that a user or group has exceeded its quota, further writes to the file system by that user or group are halted until usage decreases below the set quota size via deletion or truncation of that user or group's data from the file system. Note that a quota can be exceeded during this detection period. For example, if you are writing a large file that exceeds the quota, the write will fail, but the total usage may slightly exceed the quota.

Creating User Quotas and Group Quotas

Use the **`purequota group create`** and **`purequota user create`** commands to set user and group quotas for each file system group ID and user ID (quotas can be set explicitly). The group or user ID must be specified using the **`--gid`** or **`--uid`** options, as well as the file system (**`--fs`**) and quota size (**`--quota`**) . Alternatively, you can specify the group or user name rather than the group or user ID using the **`--group-name`** or **`--user-name`** options. If a quota already exists for a user or group, the **`create`** command will fail.

If you wish to create a default user or group quota that applies to all users and groups on a file system, use the **`purefs setattr --defaut-user-quota`** or **`purefs setattr --default-group-quota`** command. Default quotas apply to all users and groups that do not have explicit quotas.

Modifying User Quotas and Group Quotas

Existing user and group quotas can be modified at any time using the **`purequota group setattr`** and **`purequota user setattr`** commands. Note that if you modify a user's or group's quota to a lower value and that user or group's usage has already exceeded the new value, writes will automatically halt until usage decreases below the new quota setting.

Deleting User Quotas and Group Quotas

User and group quotas can be deleted. For example, if a user's or group's quota is no longer applicable for your needs, delete it from its associated file system using the **`purequota group delete`** or **`purequota user delete`** command. As with creating quotas, you must specify either the user ID (**`--uid`**) , user name (**`--user-name`**, group ID (**`--gid`**) , or group name (**`--group-name`**) , as well as the file system (**`--fs`**) . If a default quota was set for the file system 's users or groups, the default quota will take effect when the explicit quota (quota per user or group ID) is deleted.

Viewing User Quota and Group Quota Reports

To view the quotas and current usage of users and groups whose quotas were set using the **`purequota create`** command, run the **`purequota user list`** and **`purequota group list`** commands. In addition to standard filtering options, you can also view quota and usage information for specific users or groups using the **`--uid`**, **`--user-name`**, **`--gid`**, **`--group-name`** options.
To view all file system user and group usage, regardless of if a specific quota was set, run the **`purequota user list --all-usage`** and **`purequota group list --all-usage`** commands.

Enabling Quota Notifications

Direct email notifications can optionally be enabled to send emails to users and groups when they approach or exceed their file system quota. An email with a warning message is sent to a user or group when their quota reaches 80% capacity and again at 90% capacity. When a quota is at 100% capacity, a critical email is sent. Emails will be sent every 24-hours until the quota falls below 80% capacity threshold. Use the **`purequota notification enable`** command to enable notifications or **`purequota notification disable`** to disable notifications.

A direct email notification can optionally be configured to include the administrator's contact information which the user and group can use to inquire about their quota and manageability options, such as requesting an increase of their individual space quota.
The FlashBlade administrator can include a free-form string enclosed by quotation marks. The contact information can include a name, phone number, or email. For example **`"John Doe, Storage Admin Team. jdoe@example.com (555-867-5309)"`**. Use the **`purequota notification setattr --contact`** command to enter contact information.

Note: Quota usage alerts will be downgraded from critical or warning to info-level when notifications are enabled. This will prevent the administrators and watchers from constantly receiving direct email notifications.

Example 1

```
purequota user create --fs fs1 --uid 1001,1002 --quota 100G
```

Creates a quota of 100G for users with user IDs 1001 and 1002 on file system fs1.

Example 2

```
purequota user setattr --fs fs1 --uid 1001 --quota 500G
```

Modifies the quota (from 100G as shown in Example 1 to 500G) for user ID 1001 on file system fs1.

Example 3

```
purequota user list --fs fs1 --all-usage
```

Lists the quotas and current usage for all users on file system fs1.

Example 4

```
purequota user list --fs fs1
```

Lists the current usage for all users with set quotas on file system fs1.

Example 5

```
purequota user delete --fs fs1 --uid 1002
```

Deletes the quota from user ID 1002.

Example 6

```
purequota notification enable
```

Enables the FlashBlade array to send email notifications to any user or group approaching or exceeding their quota.

Example 7

```
purequota notification setattr --contact "John Doe, Storage Admin Team.
 jdoe@example.com (555-867-5309)"
```

Adds the contact information of John Doe. Email notifications will include John Doe's contact information should users or groups need to inquire about their quota.

See Also

- *purealert*
- *purefs*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

puresmtp

puresmtp, puresmtp-list, puresmtp-setattr — displays and sets SMTP attributes.

puresmtp `list [--cli|--csv|--nvp] [--notitle]`

puresmtp `setattr [--relayhost` ***RELAYHOST***`] [--senderdomain` ***SENDERDOMAIN***`]`

Sub-Commands

`list`

Displays SMTP attributes.

`setattr`

Sets SMTP attributes.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--relayhost` ***RELAYHOST***

Displays or sets the hostname or IP address of the email relay server currently being used as a forwarding point for alert email notifications generated by the array. If specifying an IP address, enter the IPv4 or IPv6 address.

For IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from `0` to 255 representing a group of 8 bits. If a port number is also specified, append it to the end of the address in the form `ddd.ddd.ddd.ddd:PORT`, where `PORT` represents the port number.

For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`) . If a port number is also specified, enclose the entire address in square brackets (`[]`) and append the port number to the end of the address. For example, `[xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]:PORT`, where `PORT` represents the port number.

`--senderdomain` ***SENDERDOMAIN***

Displays or sets the domain name. The domain name determines how logs are parsed and treated by Pure Storage Support and Escalations. The domain name is also used in the "from" address of outgoing alert email notifications. The domain name must be set to your company's domain name. For example, `mydomain.com`

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

Description

The **`puresmtp`** command allows you to view and set your array 's SMTP attributes.

Viewing SMTP Attributes

The **`puresmtp list`** command allows you to view the configured relay host and sender domain for your array.

Configuring the SMTP Relay Host

The **`puresmtp setattr --relayhost`** command sets the hostname or IP address of the email relay server that is to be used as a forwarding point for alert email notifications generated by the array. To clear the configured hostname or IP address, set `--relayhost` to an empty value (`""`) . Once cleared, Purity//FB sends all alert email notifications directly to the recipient addresses rather than route them via the relay (mail forwarding) server.

Configuring the Sender Domain

The **`puresmtp setattr --senderdomain`** command sets the domain name.

Set the domain name to your company's domain name. For example, `mydomain.com`.

The domain name determines how logs are parsed and treated by Pure Storage Support and Escalations. The domain name is also used in the "from" address of outgoing alert email notifications. For example, `purearray-c01-1@mydomain.com`.

Example 1

```
puresmtp list
```

Displays the array 's current SMTP attributes.

Example 2

```
puresmtp setattr --relayhost fake.relay.purestorage.com
```

Sets the email relay server to host `fake.relay.purestorage.com`, which will be used as a forwarding point for alert email notifications generated by the array.

Example 3

```
puresmtp setattr --senderdomain mydomain.com
```

Sets the sender domain `mydomain.com` to be used as the forwarding point for alert email notifications generated by the array.

Example 4

```
puresmtp setattr --relayhost relay.purestorage.com --senderdomain mydomain.com
```

The `--relayhost` and `--senderdomain` options can be issued together.

In this example, the the email relay server is set to host `relay.purestorage.com` as the forwarding point for alert email notifications, and the sender domain `mydomain.com` is set as the forwarding point for alert email notifications generated by the array.

See Also

- *pureadmin*
- *purearray*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

puresnmp

puresnmp, puresnmp-agent puresnmp-create, puresnmp-delete, puresnmp-list, puresnmp-rename, puresnmp-setattr, puresnmp-test — creates, manages, and displays SNMP settings.

Synopsis

```
puresnmp agent list [--cli|--csv|--nvp] [--no-title] [--raw] [--mib]

puresnmp agent setattr [--auth-passphrase] [--auth-protocol AUTH_PROTOCOL]
[--community] [--privacy-passphrase] [--privacy-protocol PRIVACY_PROTOCOL]
[--user USER] [--version VERSION]

puresnmp create [--auth-passphrase] [--auth-protocol AUTH_PROTOCOL]
[--community] --host HOST [--notification inform|trap]
[--privacy-passphrase] [--privacy-protocol PRIVACY_PROTOCOL] [--user USER]
--version VERSION MANAGER...

puresnmp delete MANAGER...

puresnmp list [--cli|--csv|--nvp] [--no-title] [--raw] [--filter FILTER]
[--page] [--limit LIMIT] [--sort SORT] [MANAGER...]

puresnmp rename OLD-NAME NEW-NAME

puresnmp setattr [--auth-passphrase] [--auth-protocol AUTH_PROTOCOL]
[--community] [--host HOST] [--notification inform|trap]
[--privacy-passphrase] [--privacy-protocol PRIVACY_PROTOCOL] [--user USER]
[--version VERSION] MANAGER...

puresnmp test MANAGER...
```

Sub-Parsers

`agent`

When specified, indicates that sub-commands are to be processed for SNMP agents.

Subcommands

`create`

Creates an SNMP manager.

`delete`

Deletes one or more SNMP managers.

`list`

Displays SNMP manager information.

When used with the `agent` sub-parser, displays SNMP agent information.

`rename`

Renames an SNMP manager.

`setattr`

Sets SNMP manager attributes.

When used with the `agent` sub-parser, sets SNMP agent attributes.

`test`

Sends a test trap to an SNMP manager.

Arguments

MANAGER

The name of the SNMP manager.

OLD-NAME

The SNMP manager's original name when renaming an SNMP manager.

NEW-NAME

The SNMP manager's new name when renaming an SNMP manager.

Options

`--auth-passphrase`

SNMPv3 only. Passphrase used by Purity to authenticate the array with the specified managers. Purity prompts for the passphrase. The passphrase must be at least 8 characters in length. Allowed characters are [A-Z], [a-z], [0-9], _ (underscore), and - (hyphen). To clear the passphrase, press Enter at the prompts. Note that the authentication protocol must be configured in order to set the authentication passphrase.

`--auth-protocol` ***AUTH_PROTOCOL***

SNMPv3 only. Hash algorithm used to validate the authentication passphrase. Valid values are MD5, SHA, or null (set to ""). Values are case sensitive.

`--community`

SNMPv2c only. Manager community string under which Purity is to communicate with the specified managers. Purity prompts twice for the community string. Allowed characters are [A-Z], [a-z], [0-9], _ (underscore), - (hyphen). To remove the array from the community, press Enter at the prompts

`--host` ***HOST***

DNS hostname or IP address of a computer that hosts an SNMP manager to which Purity is to send messages when it generates alerts. If specifying an IP address, enter the IPv4 or IPv6 address.

For IPv4, specify the IP address in the form ddd.ddd.ddd.ddd, where ddd is a number ranging from 0 to 255 representing a group of 8 bits. If a port number is also specified, append it to the end of the address in the format ddd.ddd.ddd.ddd:PORT, where PORT represents the port number.

For IPv6, specify the IP address in the form xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx, where xxxx is a hexadecimal number representing a group of 16 bits. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (::). If a port number is also specified, enclose the entire address in square brackets ([]) and append the port number to the end of the address. For example, [xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]:PORT, where PORT represents the port number.

Note: This host option refers to SNMP managers and is not related to host options in other CLI commands.

`--mib`

Displays SNMP MIB text.

`--notification trap|inform`

Notification type that determines whether the recipient (remote host) acknowledges receipt of SNMP messages. Valid options are trap (default) and inform. An SNMP trap is an unacknowledged (asynchronous) SNMP message, meaning the recipient does not acknowledge receipt of the message. An SNMP inform request is an acknowledged trap. Values are case sensitive.

`--privacy-passphrase`

SNMPv3 only. Passphrase used to encrypt SNMP messages. Purity prompts for the passphrase, which must be at least 8 non-space ASCII characters. To clear the passphrase, press Enter at the prompts. Note that the privacy protocol must be configured in order to set the privacy passphrase.

`--privacy-protocol` ***PRIVACY_PROTOCOL***

SNMPv3 only. Encryption protocol for SNMP messages. Valid values are AES, DES, or null (set to ""). Values are case sensitive. Note that the authentication settings must be configured in order to set the privacy options.

`--user` ***USER***

Required for SNMPv3. User ID that Purity uses in communications with SNMP managers. Allowed characters are [A-Z], [a-z], [0-9], _ (underscore), and - (hyphen).

`--version` ***VERSION***

Version of the SNMP protocol to be used by Purity in communications with the specified manager(s). Valid values are v2c and v3. Values are case sensitive.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

`--raw`

Displays the unformatted version of column titles and data.

`--filter FILTER`

Displays only the rows that meet the filter criteria specified.

`--limit LIMIT`

Limits the size of the list output to the specified maximum number of rows.

`--page`

Turns on interactive paging.

`--sort SORT`

Sorts the list output in ascending or descending order by the column specified.

Description

The **`puresnmp`** command is used to create, manage, and display SNMP managers and agents.

The Simple Network Management Protocol (SNMP) is used by SNMP agents and SNMP managers to send and retrieve information. FlashBlade supports SNMP versions v2c and v3.

FlashBlade's built-in SNMP agent has local knowledge of the array. The agent collects and organizes this array information and translates it via SNMP to or from the SNMP managers. The agent cannot be deleted. The managers are defined by creating SNMP manager objects on the array. The managers communicate with the agent via the standard TCP port 161, and they receive notifications on port 162.

The SNMP agent has two functions, namely, responding to GET-type SNMP requests and transmitting alert messages.

The agent responds to GET-type SNMP requests made by the SNMP managers, and return values for an information block, such as purePerformance, or individual variables within the block, depending on the type of request issued. The following variables are supported:

Table 11.8 Supported Variables

Variable	Description
pureArrayReadBandwidth	Array read bandwidth (bytes/s)
pureArayWriteBandwidth	Array write bandwidth (bytes/s)
pureArrayReadIOPS	Array read IOPS (op/s)
pureArrayWriteIOPS	Array write IOPS (op/s)
pureArrayReadLatency	Array read latency (us/op)
pureArrayWriteLatency	Array write latency (us/op)
pureArrayAverageOperationSize	Array average operation size (bytes)
pureArrayAverageReadSize	Array average read size per read operation (bytes)
pureArrayAverageWriteSize	Array average write size per write operation (bytes)
pureArrayOtherIOPS	Array other operations process per second (op/s)
pureArrayOtherLatency	Array other latency (us/op)

The FlashBlade Management Information Base (MIB) describes the purePerformance variables and can be downloaded from the array to your local machine using the GUI or displayed using the CLI or REST API.

Note: MIB-2 object identifiers (OIDs) are not supported.

The SNMP agent generates and transmits messages to the SNMP manager as traps or inform requests (informs), depending on the notification type that is configured on the manager. An SNMP trap is an unacknowledged SNMP message, meaning the SNMP manager does not acknowledge receipt of the message. An SNMP inform is an acknowledged trap.

To configure the notification type, include the `--notification` option when creating or modifying the attributes of the SNMP manager. If the SNMP manager notification type is set to `trap`, the agent sends the SNMP message (trap) without expecting a response. If the SNMP manager is set to `inform`, the agent sends the SNMP message (inform) and waits for a reply from the manager confirming message retrieval. If the agent does not receive a response within a certain timeframe, it will retry until the inform has passed through successfully. If the notification type is not set, the manager defaults to `trap`.

Creating and Deleting an SNMP Manager Object

The **`puresnmp create`** command is used to create an SNMP manager object.

Include the required `--host` and `--version` options when creating an SNMP manager object. Include the `--host` option to specify the DNS hostname or IP address of the computer that hosts the SNMP manager. Include the `--version` option to specify the version of the SNMP protocol. Valid versions are `v2c` and `v3`.

SNMPv2 uses a type of password called a community string to authenticate the messages that are passed between the agent and manager. The community string is sent in clear text, which is considered an unsecured form of communication. SNMPv3 supports secure communication between the agent and manager through the use of authentication and privacy encryption methods. As such, SNMPv2c and SNMPv3 have different security attributes.

To configure the SNMPv2c agent and managers, include the `--community` option to set the community string under which the agent is to communicate with the managers. The agent and manager must belong to the same community; otherwise, the agent will not accept requests from the manager. When setting the community, Purity prompts twice for the community string. To remove the agent or manager from the community, press Enter at the community prompts.

To configure the SNMPv3 agent and managers, include the `--user` option to specify the user ID that Purity uses to communicate with the SNMP manager. Also set the authentication and privacy encryption security levels for the agent and managers. SNMPv3 supports the following security levels:

- **noAuthNoPriv**: Authentication and privacy encryption is not set. Similar to SNMPv2c, communication between the SNMP agent and managers is not authenticated and not encrypted. noAuthNoPriv security requires no configuration.
- **authNoPriv**: Authentication is set, but privacy encryption is not set. Communication between the SNMP agent and managers is authenticated but not encrypted. Password authentication is based on MD5 or SHA hash authentication. To configure authNoPriv security, set the `--auth-protocol` and `--auth-passphrase` attributes.
- **authPriv**. Communication between the SNMP agent and managers is authenticated and encrypted. Password authentication is based on MD5 or SHA hash authentication. Traffic between the FlashBlade and SNMP manager is encrypted using encryption protocol AES or DES. To configure authPriv security, set the `--auth-protocol`, `--auth-passphrase`, `--privacy-protocol`, and `--privacy-passphrase` attributes.

  Note that privacy cannot be configured without authentication

Once an SNMP manager is created, the FlashBlade immediately starts transmitting SNMP messages and alerts to the manager.

The **`puresnmp delete`** command stops communication with the specified SNMP manager and deletes the SNMP manager object from Purity.

Viewing SNMP Manager Information

The **`puresnmp list`** command displays communication and security attributes of the SNMP managers. Each manager listed displays the following information:

- **Name**: The SNMP manager name that Purity uses in communications with SNMP managers.
- **Host**: The DNS hostname or IP address of a computer that hosts an SNMP manager to which Purity is to send messages when it generates alerts.
- **Version**: The SNMP version, either `v2c` or `v3`.
- **Community**: SNMPv2c only. The manager community string under which Purity is to communicate with the specified managers.
- **User**: Required for SNMPv3. The user ID that Purity uses in communications with SNMP managers
- **Auth Protocol**: SNMPv3 only. The hash algorithm used to validate the authentication passphrase.
- **Auth Passphrase**: SNMPv3 only. The passphrase used by Purity to authenticate the array with the specified managers.
- **Privacy Protocol**: SNMPv3 only. The encryption protocol for SNMP messages. Valid values are AES, DES, or null (set to "").
- **Privacy Passphrase**: SNMPv3 only. The passphrase used to encrypt SNMP messages.
- **Notification**: The notification type, either `inform` or `trap`.

To view configuration information for a specific SNMP manager, issue the **`puresnmp list command`** with the SNMP manager name specified.

Viewing SNMP Agent Information

The **`puresnmp agent list`** command displays attributes of the SNMP agent.

- **Version**: The SNMP version, either `v2c` or `v3`.
- **Community**: SNMPv2c only. The manager community string under which Purity is to communicate with the specified managers.
- **User**: Required for SNMPv3. The user ID that Purity uses in communications with SNMP managers

- **Auth Protocol**: SNMPv3 only. The hash algorithm used to validate the authentication passphrase.
- **Auth Passphrase**: SNMPv3 only. The passphrase used by Purity to authenticate the array with the specified managers.
- **Privacy Protocol**: SNMPv3 only. The encryption protocol for SNMP messages. Valid values are AES, DES, or null (set to "").
- **Privacy Passphrase**: SNMPv3 only. The passphrase used to encrypt SNMP messages.
- **Engine ID**: The unique SNMP engine ID.

Renaming an SNMP Manager

SNMP manager objects can be renamed using the `puresnmp rename` command. You must supply the current name of the SNMP manager (`OLD_NAME`) and the new name you wish to give the SNMP manager (`NEW_NAME`).

SNMP manager names are used in Purity administrative commands, and have no external significance. The name change is effective immediately.

Editing an SNMP Manager

After an SNMP manager is created, in addition to the ability to later rename the manager, you can also edit its settings. Use the `puresnmp setattr` command to edit the SNMP manager. You can change the hostname, IP address, notification, or SNMP version, corresponding protocol and security attributes, and notification type of the specified SNMP manager.

Changing the SNMP version clears all of the previous version settings.

Editing an SNMP Agent

Use the `puresnmp agent setattr` command to edit the SNMP agent. You can change the SNMP version, community string, protocol and security attributes, and user.

Changing the SNMP version clears all of the previous version settings.

Sending a Test Trap

Use the `puresnmp test` command to send a test SNMP message (trap or inform) to an SNMP manager. You must include the manager name. For SNMP traps, successful transmission is verified at the destination. For SNMP informs, the recipient sends a response, acknowledging receipt of the SNMP message.

Example 1

```
puresnmp create --host mysnmp.example.com --version v2c --community MySNMPManager
Enter community:
Retype community:
```

Creates the SNMP manager object named `mySNMPManager` for the SNMP manager running in host `mysnmp.example.com`. Purity prompts for the new community string. Purity uses SNMPv2c to communicate in the new SNMP community.

Example 2

```
puresnmp delete MySNMPManager
```

Stops transmission of any future messages to `MySNMPManager` and deletes the SNMP manager. If `MySNMPManager` is recreated at a later time, its attributes must be re-defined.

Example 3

```
puresnmp list MySNMPManager
```

Displays the protocol and security attributes for `MySNMPManager`.

Example 4

```
puresnmp rename MySNMPManager MySNMPManager1
```

Changes the name of `MySNMPManager` to `MySNMPManager1`. None of the manager object's protocol or security attributes are changed.

Example 5

```
puresnmp setattr --host 172.169.0.12 MySNMPManager
```

Changes the IPv4 address associated with `MySNMPManager` to `172.169.0.12`.

Example 6

```
puresnmp setattr --host [2001:db8:85a3::8a2e:370:7334] MySNMPManager
```

Changes the IPv6 address associated with `MySNMPManager` to `2001:db8:85a3::8a2e:370:7334`.

Example 7

```
puresnmp setattr --auth-passphrase MySNMPManager
Enter auth passphrase:
Retype auth passphrase:
Name    Host               Version Community User  Auth ...
MyV3Mgr mysnmp.example.com v3      -         User1 SHA  ...
```

For an existing SNMPv3 manager named `MySNMPManager`, sets the authentication passphrase to a new value. Purity prompts for the new authentication passphrase value.

Example 8

```
puresnmp setattr --auth-protocol MD5 MySNMPManager
Name          Host               Version Community User  Auth ...
MySNMPManager mysnmp.example.com v3      -         User1 MD5  ...
```

For an existing SNMPv3 manager named `MySNMPManager`, sets the authentication protocol to `MD5`.

Example 9

```
puresnmp setattr --auth-protocol SHA --auth-passphrase MySNMPManager
Enter auth passphrase:
Retype auth passphrase:
Name          Host               Version Community User Auth  ...
MySNMPManager mysnmp.example.com v3      -         User1 SHA  ...
```

For an existing SNMPv3 manager named `MySNMPManager`, sets the authentication protocol attribute to `SHA` and sets the authentication passphrase to a new value. Purity prompts for the new authentication passphrase value.

Example 10

```
puresnmp setattr --community MySNMPManager
Enter community:
Retype community:
Name          Host               Version Community User Auth ...
MySNMPManager mysnmp.example.com v2c     ****      -    -    ...
```

For an existing SNMPv2c manager named `MySNMPManager`, sets the community string to a new value. Purity prompts for the new community string.

Example 11

```
puresnmp agent setattr --version v3 --user User1 localhost
Name      Host      Version Community User   Auth Protocol ...
```

```
localhost localhost v3      -         User1  -            ...
```

Sets the existing localhost agent to SNMPv3, with a user named `User1`. SNMP Managers can now communicate with the built-in SNMP agent using the SNMPv3 protocol.

Example 12

```
puresnmp setattr --notification inform MySNMPManager
Name          Host               Version Community User Auth Protocol ...
MySNMPManager mysnmp.example.com v2c     ****      -    -             ...
```

Sets the notification type for SNMP manager `MySNMPManager` to `inform`. SNMP messages sent will be acknowledged.

See Also

- *pureadmin*
- *purearray*
- *purenetwork*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

puresubnet

puresubnet, puresubnet-create, puresubnet-delete, puresubnet-list, puresubnet-setattr — displays and manages the subnets on the FlashBlade array.

Synopsis

```
puresubnet create [--gateway GATEWAY] [--lag LAG] [--mtu MTU]
[--prefix PREFIX] [--vlan VLAN] SUBNET

puresubnet delete SUBNET

puresubnet list [--service SERVICE] [-cli|--csv|--nvp] [--notitle]
[--service SERVICE] [--filter FILTER] [--sort SORT] [--limit LIMIT] [--raw]
[--page] [SUBNET…]

puresubnet setattr [--gateway GATEWAY] [--lag LAG] [--mtu MTU]
[--prefix PREFIX] [--vlan VLAN] SUBNET
```

Sub-Commands

`create`

Creates a subnet.

`delete`

Deletes a subnet.

`list`

Displays subnet attributes.

`setattr`

Sets subnet attributes.

Arguments

SUBNET

Subnet name.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--gateway` ***GATEWAY***

IP address of the gateway through which the specified interface is to communicate with the network. For IPv4, specify the gateway IP address in the form `ddd.ddd.ddd.ddd`. For IPv6, specify the gateway IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`) .

To remove the gateway specification, set to a null value (`""`) . The interfaces in the subnet inherit the gateway address.

`--lag` ***LAG***

Sets the link aggregate group (LAG) that will be associated with the subnet.

`--mtu` ***MTU***

Maximum transmission unit (MTU) for the data vips, in bytes. The data vips in the subnet inherit the MTU value.

`--prefix` ***PREFIX***

IP address of the network prefix and prefix length of the subnet. For IPv4, specify the subnet prefix in the form `ddd.ddd.ddd.ddd/dd`. For IPv6, specify the subnet prefix in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`) .

`--service` ***SERVICE***

Lists only the subnets configured with the specified service. For FlashBlade, the single supported service is `data`.

`--vlan` ***VLAN***

VLAN ID number. The data vips in the subnet inherit the VLAN ID. Valid VLAN ID numbers are between 1 and 4094.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--page`

Turns on interactive paging.

`--raw`

Displays the unformatted version of column titles and data.

Options that manage display results:

--filter ***FILTER***

Displays only the rows that meet the filter criteria specified.

--limit ***LIMIT***

Limits the size of the list output to the specified maximum number of rows.

--sort ***SORT***

Sorts the list output in ascending or descending order by the column specified.

Description

The **`puresubnet`** command displays and manages the subnets on the FlashBlade array.

In the FlashBlade array, data virtual IP addresses (data vips) are organized into subnetworks (subnets). Exporting a file system requires a data vip which, in turn, must be attached to a subnet. Data vips that share the same network prefix and gateway are grouped into the same subnet.

Before creating a data vip, verify the subnet with the correct network prefix, VLAN ID, and gateway appears in the list output. For more information about creating data vips, refer to *purenetwork*.

Viewing Subnets

The **`puresubnet list`** command displays a list of subnets on the array and the attributes of each subnet. For example,

```
puresubnet list
Name   Enabled  Prefix         VLAN  Gateway      MTU   LAG     Services
 Interfaces
SUB01  True     10.10.200.0/24 100   10.10.200.1  1500  uplink  -         -
```

If the desired subnet does not appear in the list output, run the `puresubnet create` command to create a new subnet.

The **`puresubnet list --service data`** command displays a list of subnets on the array to which data vips are attached for file system export. For example,

```
puresubnet list --service data
Name   Enabled  Prefix         VLAN  Gateway      MTU   LAG     Services
 Interfaces
SUB01  True     10.10.200.0/24 100   10.10.200.1  1500  uplink  data
 DataVip01
```

Note that the subnet inherits the `data` service type from the data vip.

Creating Subnets

The **`puresubnet create`** command creates subnets.

Creating a subnet requires a prefix and VLAN interface. Include the **`--prefix`** option to define the subnet prefix. Specify the IP address of the subnet prefix and prefix length in the form `ddd.ddd.ddd.ddd/dd` for IPv4, or `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx` for IPv6. Include the **`--vlan`** option to specify the VLAN ID to which the subnet is configured. Valid VLAN ID numbers are between 1 and 4094.

Optionally include the **`--gateway`** option to define the gateway IP address through which the data vip is to communicate with the network.

Optionally specify a LAG name using the **`--lag`** option if creating a subnet associated to a specific LAG. For more information about creating LAGs, refer to *purelag*.

Optionally include the **`--mtu`** option to specify the maximum transmission unit (MTU) of the data vip. If the MTU is not specified during subnet creation, the value defaults to `1500`.

When a data vip is attached to the subnet, it inherits the gateway, MTU and VLAN ID of the subnet. Likewise, the subnet inherits the `data` service type from the data vip.

Modifying Subnets

The **`puresubnet setattr`** command modifies the attributes of a subnet.

Deleting Subnets

The **`puresubnet delete`** command deletes subnets that are no longer needed. Note that exporting a file system requires at least one valid data vip, which must be attached to a subnet. Therefore, there must always be at least one subnet (for service type `data`) on the array.

Example 1

```
puresubnet list
```

Displays a list of all subnets on the FlashBlade array and the attributes of each subnet.

Example 2

```
puresubnet list SUB01
```

Displays the attributes of subnet `SUB01`.

Example 3

```
puresubnet list --service data
```

Displays a list of subnets to which data vips are attached for file system export.

Example 4

```
puresubnet create --prefix 192.168.1.0/24 --vlan 100 SUB01
```

Creates subnet `SUB01` with prefix `192.168.1.0/24` and VLAN ID `100`. The MTU defaults to `1500`.

Example 5

```
puresubnet delete SUB01
```

Deletes subnet `SUB01`. A subnet can only be deleted if it does not contain data vips.

Example 6

```
puresubnet setattr --vlan 50 SUB01
```

Sets subnet `SUB01` to VLAN ID `50`.

See Also

- *purearray*
- *purefs*

- *purenetwork*
- *puresupport*
- *purelag*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

puresupport

puresupport, puresupport-connect, puresupport-disable, puresupport-disconnect, puresupport-enable, puresupport-list, puresupport-setattr, puresupport-test — manage the phone home facility and remote assistance sessions

Synopsis

puresupport connect

puresupport disable --phonehome

puresupport disconnect

puresupport enable phonehome

puresupport list [--connect] [--path] [--cli|--csv|--nvp] [--notitle]

puresupport setattr {--proxy ***HTTPS-PROXY***}

puresupport test {--connect| --phonehome} [--filter ***FILTER***] [--sort ***SORT***] [--raw]

Sub-Commands

`connect`

Connects a remote assist session.

`disable`

Disables the transmission of diagnostic logs.

`disconnect`

Disconnects a remote assist session.

`enable`

Enables the transmission of diagnostic logs.

`list`

Displays infomration about support connections

`setattr`

Sets support connection attributes.

`test`

Tests the connection to CloudAssist.

Arguments

None.

Options

`-h | --help`

Can be used with any command or subcommand to display a brief syntax description.

`--connect`

For `puresupport list`, displays remote assistance connection details. For `puresupport test` used by Pure Storage Support to troubleshoot remote assistance issues.

`--path`

Displays the information about the path of the remote assistance session on each fabric module. This option requires the `--connect` option.

`--phonehome`

Used by Pure Storage Support to troubleshoot phone home issues.

`--proxy` ***HTTPS-PROXY***

Server to be used as the HTTP or HTTPS proxy. Specify the server name, including the scheme and proxy port number. For example, *http://proxy.example.com:8080*. To clear the proxy setting, set to an empty string (`""`) .

Options that control display format:

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--notitle`

Lists information without column titles.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--raw`

Displays the unformatted version of column titles and data.

Options that manage display results:

`--filter`

Displays only the rows that meet the filter criteria specified.

`--sort`

Sorts the list output in ascending or descending order by the column specified.

Description

The **`puresupport`** command manages the phone home facility, remote assistance (RA) sessions, and proxy settings.

Phone Home Facility

The phone home facility allows the array to transmit log and diagnostic information to Pure Storage Support via a secure network connection. Optionally configure the proxy host.

The phone home facility can be enabled and disabled at any time.

To enable the phone home facility, run the **`puresupport enable phonehome`** command. Once enabled, the phone home facility immediately begins to transmit logs to Pure Storage Support. The logs are transmitted on a regular basis until the phone home facility is disabled.

To disable the phone home facility and thereby stop log transmission, run the **`puresupport disable phonehome`** command.

To view the phone status (enabled or disabled), run the `puresupport list` command.

Remote Assistance (RA) Sessions

In some cases, the most efficient way for Pure Storage Support to service a FlashBlade array or diagnose problems is through direct access to the array. A remote assistance (RA) session grants Pure Storage Support direct and secure access to the array through a reverse tunnel which you, the administrator, open. This is a two-way communication. Optionally configure the proxy host.

RA sessions can be opened and closed at any time.

The **`puresupport connect`** command opens an RA session, giving Pure Storage Support the ability to log into the array, effectively establishing an administrative session. Once the RA session is successfully established, the array returns connection details, including the date and time when the session was opened, the date and time when the session expires, and the proxy status (true, if configured). To view the details of an open RA session, run the `puresupport list --connect` command.

After the Pure Storage Support team has performed all of the necessary diagnostic or maintenance functions, run the **`puresupport disconnect`** command to terminate the RA session.

An open RA session automatically terminates (closes) after two days have lapsed.

To view the connection status of the RA session, run the `puresupport list` command.

Proxy Hostnames

The **`puresupport setattr --proxy`** command configures the proxy hostname. The format for the proxy host name is `http(s)://hostname:port`, where `hostname` is the name of the proxy host, and `port` is the TCP/IP port number used by the proxy host.

To view the current proxy settings, run the `puresupport list` command.

Displaying Support Connection Details

The **`puresupport list`** command displays the following support connection details:

- Remote assistance session status as open (RA Active is True) or closed (RA Active is False). Include the **`--connect`** option to view the details of an open RA session.
- Proxy host setting.
- Phonehome status as enabled or disable.

You can also display the following information about the remote assistance connection path by issuing the **`puresupport list --connect --path`** command:

- The name of the fabric module.
- Per-fabric module session status.

Troubleshooting

The **`puresupport test`** command displays information to help Pure Storage Support troubleshoot connection issues.

If you are experiencing connection issues, contact a member of the Pure Storage account team or email Pure Storage Support at *support@purestorage.com*.

Example 1

```
puresupport list
RA Active  Proxy  Phonehome
False      -      enabled
```

Displays the remote assistance (RA), proxy, and phone home settings for the array.

Example 2

```
puresupport list --connect
Name        Status        Opened                    Expires
Ar1         disconnected  2017-11-27 15:39:27 PST   2017-11-29 15:39:27 PST
```

Displays the connection details and status of a remote assistance (RA) session.

Example 3

```
puresupport list --connect --path
Name                      Status
FM1                       connected
FM2                       unknown
```

Displays the connection status of a remote assistance (RA) session per fabric module.

Example 4

```
puresupport connect
Name   Status                Opened                      Expires
Ar1    partially connected   2017-11-27 15:39:27 PST     2017-11-29 15:39:27 PST
```

Opens a remote assistance (RA) session between the current array and Pure Storage Support, giving Pure Storage Support the ability to establish an administrative session.

Example 5

```
puresupport disconnect
Name          Status                Opened              Expires
Ar1           disconnected          -                   -
```

Closes a remote assistance (RA) session between the current array and Pure Storage Support.

Example 6

```
puresupport enable phonehome
```

Enables the automatic transmission of array logs and diagnostic information to Pure Storage Support.

Example 7

```
puresupport setattr --proxy http://proxy.example.com:8080
```

Sets the proxy host for HTTPS communication between the array and Pure Storage Support.

Example 8

```
puresupport test
Component Name  Test           Success  Details
FM1             phonehome      True     -
                remote-assist  True     -
FM2             phonehome      True     -
                remote-assist  True     -
```

Displays the current remote and phone home status for each fabric module.

See Also

- *purealert*
- *purearray*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

puretarget

puretarget, puretarget-list, puretarget-monitor, puretarget-s3 — creates, manages, and displays targets.

Synopsis

```
puretarget list [--cli|--csv|--nvp] [--no-title] [--raw] [--filter FILTER]
[--page] [--limit LIMIT] [--sort SORT] [--status-detail] [TARGET...]

puretarget monitor [--csv|--nvp] [--notitle] [--raw] [--filter FILTER]
[--limit LIMIT] [--sort SORT] [--total] [--total-only]] [--interval INTERVAL]
[--historical HISTORICAL][TARGET...]

puretarget s3 create --address ADDRESS TARGET

puretarget s3 delete TARGET...

puretarget s3 list [--cli|--csv|--nvp] [--no-title] [--raw]
[--filter FILTER] [--page] [--limit LIMIT] [--sort SORT] [--status-detail]
[TARGET...]

puretarget s3 rename OLD-NAME NEW-NAME

puretarget s3 setattr --address ADDRESS TARGET
```

Sub-Parsers

`s3`

When specified, indicates that sub-commands are to be processed for s3 targets.

Sub-Commands

`create`

Creates S3 targets. Must be used with the s3 sub-parser.

`delete`

Deletes S3 targets. Must be used with the s3 sub-parser.

`list`

Displays all targets.

When used with the s3 sub-parser, displays all S3 targets.

`monitor`

Diplays replication bandwidth monitoring statistics.

`rename`

Renames S3 targets. Must be used with the `s3` sub-parser.

`setattr`

Sets S3 target attributes. Must be used with the `s3` sub-parser.

Arguments

TARGET

The target name.

OLD-NAME

The target's original name. Must be used with the `rename` subcommand.

NEW-NAME

The target's new name. Must be used with the `rename` subcommand.

Options

`--address` ***ADDRESS***

Species the address of the target.

`--historical HISTORICAL`

Displays historical data at the given resolution in the format of `N [m | h | d | w]`, for example 15m, 2h, 1d, etc. The minimum acceptable time for viewing historical data is 5 minutes for replication and 1 second for non-replication.

`--interval` ***INTERVAL***

The number of seconds between real-time updates specified as an integer and a multiple of 30. The default is 30 seconds.

`--status-detail`

Displays status details.

Options that control display format:

`--cli`

Displays output in the form of CLI commands that can be issued to reproduce the current configuration.

`--csv`

Lists information in comma-separated value (CSV) format. The **`--csv`** output can be used for scripting purposes and imported into spreadsheet programs.

`--nvp`

Lists information in name-value pair (NVP) format, in the form ITEMNAME=VALUE. Argument names and information items are displayed flush left. The `--nvp` output is designed both for convenient viewing of what might otherwise be wide listings, and for parsing individual items for scripting purposes.

`--notitle`

Lists information without column titles.

`--raw`

Displays the unformatted version of column titles and data.

`--filter FILTER`

Displays only the rows that meet the filter criteria specified.

`--limit LIMIT`

Limits the size of the list output to the specified maximum number of rows.

`--page`

Turns on interactive paging.

`--sort SORT`

Sorts the list output in ascending or descending order by the column specified.

Description

The **`puretarget`** command is used to create, maintain, and display the connection between a FlashBlade array and S3 target for object replication.

Creating and Maintaining Target Connections

Object replication requires that a FlashBlade array (local source) and S3 target be connected by using the **`puretarget s3 create`** command. When using this command, the address and name of the S3 target must be specified. Once the connection is created, it allows the creation of a replica link which enables object replication.

Targets can be renamed at any time using the **`puretarget s3 rename`** command. The original (`OLD-NAME`) and new (`NEW-NAME`) names must be specified.

If the S3 target has moved, update the attributes of the target by issuing the **`puretarget s3 setattr`** command and providing the updated target address with the `--address` option.

Deleting a Target Connection

A specific target connection can be deleted using the `puretarget s3 delete` command. Note that a target connection cannot be deleted until the replica link being used in the replication set up is deleted. Once the replica link is deleted, replication is terminated.

Viewing Targets

To view all targets and their connection status, issue the **`puretarget list`** command. To view only S3 targets, issue the **`puretarget s3 list`** command. The following information is displayed:

- **Name**: Displays the target name.
- **Address**: Displays the target address.
- **CA Certificate Group**: Displays the CA certificate group in use. Currently the only valid CA certificate group is `_default_replication_certs`.
- **Status**: Displays if the target is connected.

If the `--status-detail` option is used, detailed information is displayed if the connection is unhealthy. If the connection is healthy, `--` is displayed.

Viewing Target Performance

By default, the **`puretarget monitor`** command displays real-time target performance data. Include the **`--interval`** option to specify the number of seconds between each real-time update. At each interval, the system displays a point-in-time snapshot of a target's performance data.

Include the **`--historical`** option to display historical target performance data over a specified resolution in the format of `N [m|h|d|w]`. For example, if `--historical 5d` is specified, the historical performance data over the last 5 days is displayed. Note that the minimum acceptable time for viewing historical data is 5 minutes for replication and 1 second for non-replication.

The output includes the following information:

- **Name**: Target name.
- **Time**: Point-in-time of the displayed target performance data.
- **B/s (transmitted)**: Average number of bytes per send operation.
- **B/s (received)**: Average number of bytes per receive operation.

Example 1

```
puretarget s3 create --address s3.amazonaws.com s3target
```

Creates a connection between the source FlashBlade array (where this command is issued) and the S3 target, `s3target`.

Example 2

```
puretarget s3 rename s3target s3newname
```

Renames the S3 target from `s3target` to `s3newname`.

Example 3

```
puretarget s3 list
```

Lists all S3 targets and their connection status.

See Also

- *purearray*
- *purebucket*
- *pureobj*

Author

Pure Storage Inc. *documentfeedback@purestorage.com*

Appendix A
Quick Reference Guide

This section provides step-by-step instructions on how to perform various tasks to administer the FlashBlade array. For detailed information about each step, refer to the Purity//FB CLI and GUI sections of the FlashBlade User Guide.

Exporting a File System

FlashBlade file systems support the HTTP, NFS, and SMB protocols. Each file system can support multiple protocols.

When creating a file system, consider which protocol(s) you want to configure:

- **File sharing over NFS**

  File sharing over NFS requires that you configure the NFS export rules and then enable the protocol. Per NFS limitations, each user can belong to a maximum of 16 groups. If users belong to more than 16 groups and a directory service contains the group information, configure file sharing over NFS with a directory service.

  Follow these steps to set up file sharing over NFS without a directory service:

  1. Create the data vip.
  2. Create the file system.
  3. Define the NFS export rules.
  4. Enable the NFS protocol.

- **File sharing over NFS with a directory service**

  For file sharing over NFS where a user might belong to more than 16 groups, configure the FlashBlade directory service to integrate with a directory server. This assumes that you already have a directory service, such as OpenLDAP or Active Directory set up.

  Follow these steps to set up file sharing over NFS with a directory service:

  1. Configure the directory service for protocol support.
  2. Create the data vip.
  3. Create the file system.
  4. Define the NFS export rules.
  5. Enable the NFS protocol.

- **File sharing over SMB with Active Directory**
  FlashBlade offers the following SMB authentication modes:

  - **AD RFC2307.** Recommended for customers where the same data needs to be accessed over SMB and NFS, and where access control is required. For more information, refer to the RFC 2307 specifications set by the Internet Engineering Task Force.
  - **AD Auto.** Recommended for customers with SMB-only clients or when SMB and NFS clients never access the same file systems.

  The default SMB authentication mode is AD RFC2307. To configure file sharing over SMB in other authentication modes, such as AD Auto mode, contact Pure Storage Support.
  Follow these steps to set up file sharing over SMB in AD RFC2307 mode:

  1. Configure the directory service for protocol support.
  2. Create the data vip.
  3. Create the file system.
  4. Enable the embedded SMB protocol adapter.

- **File system access over HTTP**
  Note that any user can read, modify, and delete any files on a filesystem when HTTP is enabled. The use of HTTP connections, as opposed to HTTPS connections, potentially gives unauthenticated users access to the file system.

  1. Create the data vip.
  2. Create the file system.
  3. Enable HTTP access.

The following sections walk you through the steps on how to export a file system via the Purity//FB CLI and GUI.

Exporting a File System Via the CLI

Perform the following Purity//FB CLI commands to export a file system:

Configuring the Directory Service for Protocol Support

Configuring the directory service is a one-time process and is only required if you are integrating the FlashBlade array with a directory service.

To configure the directory service for protocol support:

1. Verify that the directory server is accessible through a management interface. For file sharing over SMB, verify that the Active Directory server is accessible through a management interface.

2 Verify that the DNS settings can be used to resolve the directory server domain and server.

```
puredns list
```

3 Configure the directory service.

```
pureds setattr [--uri URI-LIST] [--base-dn BASE-DN]
[--bind-user BIND-USER] [--bind-password] SERVICE-NAME
```

For file sharing over SMB, the base DN (`--base-dn`) of the directory service is used in place of the URI (`--uri`) to represent the LDAP URL. `SERVICE-NAME` is specified as SMB or NFS.

4 Test the directory service configuration to verify that the URI can be resolved and that the directory service can successfully bind and query the tree using the bind user credentials.

```
pureds test SERVICE-NAME
```

5 Enable the directory service.

```
pureds enable SERVICE-NAME
```

Creating the Data VIP

Exporting a file system requires a data vip, which in turn must be attached to a subnet. It is through the data vip that clients connect to the file systems. Data vips that share the same network prefix and gateway are grouped into the same subnet.

To create the data vip:

1 Find the subnet to which the data vip will be added.

```
puresubnet list
```

If the desired subnet does not appear in the list output, create it.

```
puresubnet create [--gateway GATEWAY] [--mtu MTU]
--prefix PREFIX [--vlan VLAN] SUBNET
```

2 Add the data virtual IP address (data vip) to the subnet, making sure it falls within the subnet mask. The data vip will be used to export the file system.

```
purenetwork create vip --address ADDRESS --servicelist
data NAME
```

Creating and Exporting the File System

Creating and exporting a file system involves creating the file system and configuring protocol support for the file system. A file system can support multiple protocols.

Before you export a file system, verify that a data vip has been created. Clients connect to the file system via the data vip.

If you are integrating the FlashBlade array with a directory service, also verify that the directory service has been properly configured and enabled.

If you are enabling the SMB protocol in AD RFC2307 mode, prepare Active Directory for multi-protocol support by setting the `gidNumber` and `uidNumber` attributes for each domain user and group that will be accessing the SMB shares. The `gidNumber` and `uidNumber` attributes can be set through the Microsoft Identity Management for UNIX service or PowerShell.

To create and export the file system:

1 Create the file system.

```
purefs create [--size SIZE] FILE-SYSTEM
```

2 Perform the following protocol-specific configurations for NFS and SMB:

- **NFS**. Define the NFS export rules so the file system is accessible to the appropriate clients.

  ```
  purefs setattr --rules RULES FILE-SYSTEM
  ```

- **SMB**. By default, file sharing over SMB is configured in AD RFC2307 mode for mixed Windows/ UNIX environments.

3 Enable the protocols. Enter multiple protocols as comma-separated values.

```
purefs add --protocol PROTOCOL FILE-SYSTEM
```

The file system is now visible through any of the data vips configured on the subnet, ready to be mounted on the clients.

Example

In the following example, a series of CLI commands are performed on a FlashBlade array to export an NFS file system for the very first time. The CLI commands will perform the following tasks:

- Configure the directory service to integrate the FlashBlade array with Active Directory.
- Create the data vip so that clients can connect to the `MyFiles` file system.

- Create and export the `MyFiles` file system. NFS client `192.168.1.0/24` should be able to access the exported files with `read-only access` and `root_squash` privileges. NFS client `1.2.0.0/16` should be able to access the exported files with `read-write` access and `root` privileges.
- Mount the exported `MyFiles` file system from server `10.10.200.10` to the `/mnt` directory of the local host.

On the server:

```
puredns list
      //Verify the DNS settings can be used to resolve the Active Directory domain
 and server.
pureds setattr --uri ldaps://ad1.mycompany.com,ldaps://ad2.mycompany.com
               --base-dn DC=mycompany,DC=com
               --bind-user CN=John,OU=Users,DC=mycompany,DC=com
               --bind-password
pureds enable
puresubnet list
      //Verify the subnet with the correct configurations exists.
purenetwork create vip --address 10.10.200.10 --servicelist data DataVip01
purefs create --size 10T  MyFiles
purefs setattr --rules '192.168.1.0/24(ro) 1.2.0.0/16(rw,no_root_squash)' MyFiles
purefs add --protocol nfs MyFiles
```

On the client...

```
mkdir -p ~/mnt
    mount 10.10.200.10:/MyFiles ~/mnt
```

Exporting a File System Via the GUI

Perform the following Purity//FB GUI commands to export a file system:

Configuring the Directory Service for Protocol Support

Configuring the directory service is a one-time process and is only required if you are integrating the FlashBlade array with a directory service.

To configure the directory service for protocol support:

1. Verify that the directory server is accessible through a management interface. For file sharing over SMB, verify that the Active Directory server is accessible through a management interface.

2 Verify that the DNS settings can be used to resolve the directory server domain and server.

   a Select **Settings** > **Network**

   b In the **DNS Settings** panel, verify that the DNS settings be used to resolve to the directory server domain and server.

3 Configure the directory service.

   a Select **Settings** > **Users**

   b In the Directory Service panel, click the Edit icon. The Edit Directory Service Configuration pop-up window appears.

   c In the URIs field, type the comma-separated list of up to 30 URIs of the directory servers. For file sharing over SMB this field is optional. If specified, the base DN of the directory service is used in place of the URI to represent the LDAP URL. Each URI must include the scheme `ldap://` or `ldaps://` (for LDAP over SSL), a hostname, and a domain name or IP address. For example, `ldap://ad.company.com` configures the directory service with the hostname "ad" in the domain "company.com" while specifying the unencrypted LDAP protocol. If specifying a domain name, it should be resolvable by the configured DNS servers. If specifying an IP address, enter the IPv4 or IPv6 address. For IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from `0` to 255 representing a group of 8 bits. For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). If the scheme of the URIs is `ldaps://`, SSL is enabled. SSL is either enabled or disabled globally, so the scheme of all supplied URIs must be the same. They must also all have the same domain. If base DN is not configured and a URI is provided, the base DN will automatically default to the domain components of the URIs. Optionally specify a port. Append the port number after the end of the entire address. Default ports are 389 for ldap, and 636 for ldaps. Non-standard ports can be specified in the URI if they are in use.

   d In the Base DN field, type the base distinguished name (DN) of the directory service. The Base DN is built from the domain. For example, for `ldap://ad.storage.company.com`, the Base DN would be: `DC=storage,DC=company,DC=com`.

   e In the Bind User field, type the username used to bind to and query the directory. For Active Directory, enter the username - often referred to as the sAMAccountName or user login name - of the account that is used to perform directory lookups. The username cannot contain the characters `" [] : ; | = + * ? < > / \`, and cannot exceed 20 characters in length. For OpenLDAP, enter the full DN of the user. For example, `CN=John,OU=Users,DC=example,DC=com"`.

   f In the Bind Password field, type the password for the bind user account.

   g (For Array Management) In the Group Base field, enter the organizational unit (OU) to the configured groups in the directory tree. The Group Base consists of OUs that, when combined with the base DN attribute and the configured group CNs, complete the full Distinguished Name of each groups. The group base should specify "OU=" for each OU and multiple OUs

should be separated by commas. The order of OUs should get larger in scope from left to right. In the following example, SANManagers contains the sub-organizational unit PureGroups: "OU=PureGroups,OU=SANManagers"

h (For Array Management) In the Array Admin Group field, enter the Common Name (CN) of the directory service group containing administrators with full privileges to manage the FlashArray. Array Admin Group administrators have the same privileges as pureuser. The name should be the Common Name of the group without the "CN=" specifier. If the configured groups are not in the same OU, also specify the OU. For example, "pureadmins,OU=PureStorage", where pureadmins is the common name of the directory service group.

i Click **Save**.

j (For Array Management) The **User Login Attribute** defaults to the sAMAccountName for Active Directory, or UID for other directory services.

k (For Array Management) The **User Object Class** defaults to User for AD, posixAccount or shadowAccount for OpenLDAP, posixAccount for posix compliant servers, or person for all others.

4 Test the directory service configuration to verify that the URI can be resolved and that the directory service can successfully bind and query the tree using the bind user credentials.

- From the **Directory Service** panel, click **Test**.

5 Enable the directory service. From the **Directory Service** panel, click the toggle button to enable (blue) the directory service.

Creating the Data VIP

Exporting a file system requires a data vip, which in turn must be attached to a subnet. It is through the data vip that clients connect to the file systems. Data vips that share the same network prefix and gateway are grouped into the same subnet.

To create the data vip:

1 Find the subnet to which the data vip will be added.

- Select **Settings** > **Network**.

  The subnet appears in the Subnets list.

  If the desired subnet does not appear in the list output, create it by clicking the Add button in the Subnets title bar.

2 Add the data virtual IP address (data vip) to the subnet, making sure it falls within the subnet mask. The data vip will be used to export the file system.

   a Select **Settings** > **Network**.

   b In the Subnets list, find the subnet with the correct network prefix, VLAN ID, and gateway. The data vip will be attached to this subnet for file export purposes. Create a subnet if none of the existing ones meet your requirements.

   c Click the Add interface button (under the Interfaces column) belonging to the subnet to which the data vip will be attached. The Create Network Interface pop-up window appears.

   d In the Name field, type the name of the data vip.

   e In the Address field, type the IP address to be associated with the data vip.

   f In the Services field, leave the service type as `data`.

   g In the Subnet field, leave the subnet name as the default.

   h Click **Create**.

Creating and Exporting the File System

Creating and exporting a file system involves creating the file system and configuring protocol support for the file system. A file system can support multiple protocols.

Before you export a file system, verify that a data vip has been created. Clients connect to the file system via the data vip.

If you are integrating the FlashBlade array with a directory service, also verify that the directory service has been properly configured and enabled.

If you are enabling the SMB protocol in AD RFC2307 mode, prepare Active Directory for multi-protocol support by setting the `gidNumber` and `uidNumber` attributes for each domain user and group that will be accessing the SMB shares. The `gidNumber` and `uidNumber` attributes can be set through the Microsoft Identity Management for UNIX service or PowerShell.

To create and export the file system:

1 Create the file system.

   a From the **Storage** page, click **Add (+)** in the heading of the File Systems list. The Create File System pop-up window appears.

   b In the Name field, type the name of the directory to be exported.

   c In the Provisioned Size field, specify the provisioned size allocated to the file system. The size is a quota of space that helps gauge the fullness of the file system. If left blank, the provisioned

size will default to an unlimited size. For more information about the provisioned size, refer to the **File System Size** section.

d To enable the ability to quickly remove large directories using the fast remove feature, click the Fast Remove toggle button to enable (blue) the feature. For more information about the fast remove feature, refer to the **Fast Remove** section.

2 Perform the following protocol-specific configurations:

- **NFS**. Define the NFS export rules so the file system is accessible to the appropriate clients.

  a From the Create File system pop-up window, click **NFS** in the Protocols section.

  b Click the Enabled toggle button to enable (blue) NFS.

  c In the Export Rules field, configure the NFS export rules to define the access rights and privileges a client has to the file system.

  If the rule is defined but one or more export options are not specified, then the undefined option(s) will default to `ro` access and `root_squash` privileges. For example, an export rule that is defined as `*(rw)` will have `root_squash` privileges. Likewise, an export rule that is defined as `*(no_root_squash)` will have `ro` access.

  d Click **Create**.

- **SMB**. By default, file sharing over SMB is configured for a mixed Windows/UNIX (RFC2307) environment.

  a From the Create File system pop-up window, click **SMB** in the Protocols section.

  b Click the SMB Adapter Enabled toggle button to enable (blue) the SMB protocol adapter.

- **HTTP**.

  a From the Create File system pop-up window, click **HTTP** in the Protocols section.

  b Click the HTTP toggle button to enable (blue) HTTP access.

3 Click **Create**.

The file system is now visible through any of the data vips configured on the subnet, ready to be mounted on the clients.

Adding an Alert Watcher

1 Send a test message to an email address to ensure alert notifications can reach the intended destination.

```
purealert test watcher ADDRESS
```

2 Verify that the test message reached the destination email address.

3 Create an alert watcher to which alert email notifications are sent.

```
purealert create watcher ADDRESS
```

Newly added alerts watchers are by default enabled to receive alert email notifications.

4 Verify that the new email address has been added to the alert watcher list.

```
purealert list --watcher
```

Optionally run `purealert test` to send a test message to the email addresses of all enabled alert watchers - including the newly added watcher - to ensure alert notifications reach the intended destinations.

The file system is now visible through any of the data vips configured on the subnet, ready to be mounted on the clients.

For example, run the following commands to:

- Send a test message to Aartie Alert's email address aartiealert@example.com.
- After verifying that Aartie Alert received the email notification, add Aartie Alert as an alert watcher.
- Verify Aartie Alert's email address has been added to the list of alert watchers.

```
purealert test watcher aartiealert@example.com
purealert create watcher aartiealert@example.com
purealert list --watcher
```

Appendix B
SMB Port Assignments

This section provides the minimum ports required on each VLAN for SMB to function successfully. This information should be provided to your network team who configures the firewall.

Management Network Ports

The following table provides the ports required on the management network. Without these ports opened, SMB and LDAP will not be able to authenticate. These ports must be open between the FlashBlade management VIP and the LDAP/Active Directory (AD) server(s).

Port	TCP	UDP	Purpose
53	Yes	Yes	DNS used to locate the AD hosts and services.
123	Yes	Yes	NTP to avoid time skews, which needs to be limited for authentication.
389	Yes	Yes	Used to manage array integration with a directory service using LDAP. Active Directory requires both TCP and UDP. LDAP only requires TCP.
636	Yes	Yes	Used to manage array integration with a directory service using LDAPS (LDAP over TLS/SSL).
445	Yes	No	Used by SMB join/unjoin AD and perform all authentication/ authorization via the management network.

Firewall Ports

The following table lists the required firewall ports for FlashBlade access. These ports are outgoing traffic from the FlashBlade to the public network.

Port	TCP	UDP	Purpose
443	Yes	No	Used for both HTTPS and SSHover HTTPSFor IPv4. Use the following IP block and hostnames: • IP block 52.40.255.224/27 • Hostname: cloud-support.purestorage.com Static IPs are not supported for IPv6. Use the hostname: .cloud-support.purestorage.com Add all items above to your whitelist.

Data Network Ports

The following table describes the required ports on the data network. These ports must be open between the FlashBlade data VIP and SMB hosts.

Port	TCP	UDP	Purpose
53	Yes	Yes	Share name resolution.
445	Yes	No	SMB data traffic. SMB 2.1 is supported; therefore port 139 is not needed.

Appendix C
About NFSv4.1 Support

In the current implementation of the NFSv4.1 protocol, FlashBlade supports the following subset of core NFSv4.1 protocol features, which are compliant with Linux clients:

- Native byte-range locking semantics required by Linux clients
- All NFS traffic through single secure port (2049)
- Top level root mountable with file systems as directories
- Exportable to NFSv3 and NFSv4.1 clients simultaneously
- Client qualification: CentOS/RedHat OS
- Management support using the FlashBlade GUI, CLI, and REST API
- Kerberos

The following features of the NFSv4.1 protocol are currently unavailable:

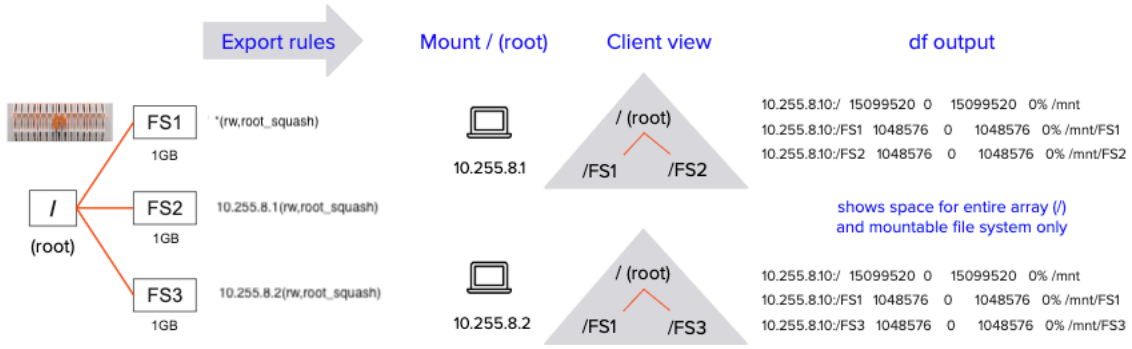
- pNFS
- Delegation
- Referrals
- Shared reservation

⚠ **Important note!** NFSv4.0 is not supported.

Top Level Mounting

In the current implementation of NFSv4.1, hosts can mount at the top level root (`/`) with file systems as directories. File systems can be exported to different clients with different export rules and permissions. Note that issuing `ls` on `/` displays the file system that is mountable to that host, and issuing `df` on `/` displays space information for `/` and any mountable, actively used (accessed within the last 60 seconds) file systems.

Figure 14.1 Top Level Mounting



Appendix D
Replication and Recovery

Replication

File systems can be replicated from one FlashBlade array to another FlashBlade array. File system replication requires two connected arrays and a replica link between the source file system and a target file system. A replica link is the connection between a local file system and target file system.

Objects can be replicated from a bucket on a FlashBlade (source) to a remote target bucket residing on another FlashBlade or an S3 target using replica links. A replica link is the relationship between a source bucket and target bucket, and describes the properties of that connection. During replication, all objects in the source bucket are replicated to the target bucket.

File Replication Prerequisites

- Replication traffic requires a replication vip using port 8117; before arrays can replicate, they must first be connected over the management vip/port 443.
- Ports 8117 and 80 are used for non-encrypted file replication.

Object Replication Prerequisites

- Replication traffic requires a replication vip using port 8117; before arrays can replicate, they must first be connected over the management vip/port 443.
- Ports 8117 and 443 are used for encrypted object replication.
- Versioning must be enabled on the target bucket for object replication to proceed, and a lifecycle rule may need to be configured to limit space used by noncurrent versions.

Recovery

⚠ **CAUTION:** If Pure File SafeMode is enabled, contact Pure Technical Services to promote and demote file systems.

In file system replication setups you can stop writes to the local file system and redirect writes to the target file system (failover). Failover can be used if there is an issue with the local file system. During failover, replication stops until you reprotect.

Failover requires promotion of the target file system and redirection of all clients to the target array to return to operation. To reprotect, demote the (former) source file system. Once the file system is demoted, replication policies will start replicating in the new direction.

Note that when demoting the local file system, any writes performed since the last replication will be discarded. If a local snapshot was taken since last replication, demoting the file system requires that

snapshot be destroyed and eradicated. You must remove any data since the last snapshot of the file system was taken. The file system must either have no snapshots, or the most recent snapshot must be the last replicated snapshot. Additionally, it is suggested that any policies attached to the local file system and replica link be removed or disabled in order to avoid taking new snapshots.

Once the local file system is demoted, re-enable or re-add the policies that were disabled before demoting the file system. Replication resumes as scheduled between the promoted remote file system and the demoted local file system. Data is written to the promoted remote file system, which is then replicated to the demoted local file system.

To resume the original replica link direction (failback), manually stop writes on the remote file system and send the remaining data to the local file system. Once data is sent to the local file system, promote the local file system and demote the remote file system. You can change the file system to not writeable or stop clients while failing back to avoid having any writes that need to be discarded when demoting the target.

Getting Started

Replication setup and recovery can be performed using either the Purity//FB GUI or CLI.

The following sections provide an overview of the tasks required for file and object replication setup and for performing recovery. Complete instructions using the GUI and CLI are provided for each in subsequent topics.

Replication Setup

To set up file replication, the following tasks must be completed:

1. Create replication network interfaces on the source and target arrays.
2. Create a connection key on the target array.
3. Connect the source and target arrays.
4. Create a policy to govern snapshot creation and retention.
5. Create a file replica link on the local array.

Complete instructions for completing each of these tasks are provided in:

- *Setting Up File Replication Using the GUI*
- *Setting Up File Replication Using the CLI*

To set up object replication from a bucket on a source FlashBlade array to a remote target bucket on another FlashBlade array, the following tasks must be completed:

1. Create replication network interfaces on the source and target arrays.

2 Create a connection key on the target array.

3 Connect the source and target arrays.

4 Create remote credentials for use with the replica link.

5 Create an object replica link on the local array.

6 Enable versioning on the target bucket.

7 Optionally configure a lifecycle rule on the target bucket to control space usage of non-concurrent versions.

Complete instructions for completing each of these tasks are provided in:

- *Setting Up Array to Array Object Replication Using the GUI*
- *Setting Up Array to Array Object Replication Using the CLI*

To set up object replication from a bucket on a source FlashBlade array to a remote target bucket on an Amazon S3 target, the following tasks must be completed:

1 Connect the source FlashBlade array to the S3 target.

2 Create remote credentials for use with the replica link.

3 Create an object replica link on the local array.

4 Enable versioning using the AWS Management Console or the Amazon S3 API.

5 Optionally enable a lifecycle rule using the AWS Management Console or the Amazon S3 API.

Complete instructions for completing each of these tasks are provided in:

- *Setting Up FlashBlade to Amazon S3 Object Replication Using the GUI*
- *Setting Up FlashBlade to Amazon S3 Object Replication Using the CLI*

File Replication Failover

To perform file replication failover, the following tasks must be completed:

1 Promote the remote file system.

2 Manually direct clients to the remote array.

3 Remove any policies from the local file system.

4 Demote the local file system.

Complete instructions for completing each of these are provided in:

- *Performing File Replication Failover and Reprotect Using the GUI*
- *Performing Failover Using the CLI*

File Replication Failback

To perform file replication failback, the following tasks must be completed:

1. Disable writes on the remote file system.
2. Promote the local file system.
3. Re-add any policies to the local file system.
4. Manually redirect clients back to the local array.
5. Demote the target file system.

Complete instructions for completing each of these tasks are provided in:

- *Performing Failback Using the GUI*
- *Performing Failback Using the CLI*

Setting Up File Replication Using the GUI

The following instructions describe how to set up file replication from one FlashBlade array to another FlashBlade array using the Purity//FB GUI. These instructions should be followed in the order that they are presented.

Creating Replication Network Interfaces

Perform the following on both your source and target FlashBlade arrays:

1. Select **Settings** > **Network**.
2. In the **Subnets** list, locate the subnet to which you wish to add a network interface and click the **Add interface** (+) button. The **Create Network Interface** pop-up window appears.
3. In the **Name** field, enter the interface name.
4. In the address field, enter the network address.
5. From the **Services** list, select `replication`.
6. The subnet defaults to the subnet in which you are creating the interface. Click **Create**.

Creating a Connection Key

⚠ **Important note!** The connection key is active for two hours. If the source and target arrays have not been connected within that two hour period, a new connection key must be created.

Once you have created replication network interfaces, you must create a connection key on the target array.

1. On the target array, select **Storage** > **Array** page.
2. In the **FlashBlade Array Connections** pane, click **More Options** > **Create Connection Key**. The **Connection Key** pop-up window appears displaying the new connection key.
3. Copy the new connection key.

Connecting the Source and Target Arrays

With the creation of replication network interfaces on the source and target arrays, and the creation of a connection key on the target array, you can now connect the source and target arrays.

Note: Encryption is only valid for object replication.

⚠ **Important note!** When connecting the source array to the target array, a secure connection is required even if encryption is not enabled. Certificates with small key sizes, weak signature algorithms, or that only support weak TLS algorithms are not supported.

⚠ **Important note!** The management IP address required during configuration cannot be a loopback IP address.

1. On the source array, select **Storage** > **Array** page, click the add (+) button in the **FlashBlade Array Connections** panel. The **Connect FlashBlade Array** pop-up window appears.
2. Enter the target array's hostname or management address in the **Management Address** field. The address can be located from the **Subnets** table by navigating to the **Settings** > **Network** page.
3. Paste the copied connection key for the target array in the **Connection Key** field.
4. The replication address is auto-discovered, unless using NAT. If using NAT, enter the target array's replication network address in the **Replication Address** field.
5. (Optional) Enable encryption by setting the **Encrypted** toggle to on. If set, the `_default_replication_certs` CA certificate group is applied. Before toggling, make sure to import the CA certificate to the source array if the target array's array-certificate is signed by a CA and add it to the certificate group. Note that encryption is set on the source array only. If enabled, the encryption setting displays as enabled on the target array.

   Warning: Use of CA certificates to establish trust and secure communication for object replication is the recommended configuration. If the target array's array-certificate is self-signed, directly

importing the target array certificate to the source and adding it to the certificate group will not provide as strict validation around aspects of the peer, such as certificate expiry.

6 Click **Connect**.

7 If using NAT, on the target array, enter the source array's replication address.

   a On the target array, in the **FlashBlade Array Connections** panel on the **Storage** > **Array** page, click the edit button at the end of the row displaying the connected source array. The **Edit Connected Array** pop-up window appears.

   b Enter the replication address for the source array in the **Replication Address** field and click **Save**.

Creating a Policy

Policies set the rules that govern the frequency of replication and the duration that snapshots are retained.

1 From the source array, select**Protection** > **Polices**, and click the add (+) button in the heading of the **Policies** list. The **Create Policy** pop-up window appears.

2 Enter a policy name.

3 In the **Create or replicate 1 snapshot every** field, enter the frequency at which snapshots are created or replicated. The value must be entered in the format n{m|h|d|w}. For example, 15m, 3h, 2d, 1w, etc.

4 In the **At** field, enter the time of day the snapshot is created. The value must be entered in the format n[am|pm], where n is a value of 1-12. If entered without am or pm, n must be a value of 0-23. This field is only editable if the value specified in the **Create or replicate 1 snapshot every** field is in days. For example, 24h, 48h, 72h, 1d, 2d, 3d, 1w, etc.

5 In the **And keep for** field, enter the retention period for the snapshot. The value must be entered in the format n{m|h|d|w}. For example, 30m, 1h, 2d, 1w, etc. The retention period cannot be less than the snapshot interval.

6 Click **Create**.

Creating a File Replica Link

Once the source and target arrays are connected, you must create a file replica link on the source array. The file replica link contains the details and specifies the rules of the replication relationship between the local and remote file systems.

1 From the **Protection** > **File Replica Links** page, click the add (+) button in the heading of the **File Replica Links** list. The **Create File Replica Link** pop-up window appears.

2 From the **Local File System** list, select the local (source) file system to be replicated.

3 From the **Remote Connection** list, select the remote (target) array.

4 (Optional) Enter a name for the remote file system to which data from the local file system will be replicated in the **Remote File System** field.

5 From the **Policy** list, select a replication policy to attach to the replica link.

6 Click **Create**.

Setting Up File Replication Using the CLI

The following instructions describe how to set up file replication from one FlashBlade array to another FlashBlade array using the Purity//FB CLI. These instructions should be followed in the order that they are presented.

Creating Replication Network Interfaces

On both the source and target arrays, issue the **`purenetwork vip create --address ADDRESS --servicelist replication`** command, where ADDRESS is the IP address associated with the replication vip.

```
# purenetwork vip create --address 123.123.123.123 --servicelist replication
```

Creating a Connection Key

⚠ **Important note!** The connection key is active for two hours. If the source and target arrays have not been connected within that two hour period, a new connection key must be created.

Once you have created replication network interfaces on the source and target arrays, you must create a connection key on the target array.

1 Issue the purearray **`create --connection-key`** command.

```
# purearray create --connection-key
Connection Key                          Created                  Expires
T-71a63c01-80aa-4173-842c-8ec9a1285d5b  2020-01-09 14:49:08 PST  2020-01-09
 16:49:08 PST
```

2 Copy the connection key.

Connecting the Source and Target Arrays

With the creation of replication network interfaces on the source and target arrays, and the creation of a connection key on the target array, you can now connect the source and target arrays.

⚠ **Important note!** When connecting the source array to the target array, a secure connection is required even if encryption is not enabled. Certificates with small key sizes, weak signature algorithms, or that only support weak TLS algorithms are not supported.

⚠ **Important note!** The management IP address required during configuration cannot be a loopback IP address.

1 On the target array, issue the **`purenetwork list`** command and make note of the array's management IP address.

```
# purenetwork list
Name          Enabled  Subnet Address       ...  Services
vir1          True     admin  10.11.123.12  ...  management
```

2 On the source array, issue the **`purearray connection --management-address MANAGEMENT_ADDRESS`** command, where `MANAGEMENT_ADDRESS` is the mangement IP address of the target array.

- You can optionally specify the `--replication-address` argument, which will set the specified replication address on the source array and the remote array connection wil have no replication address set.
- If the `--replication-address` argument is not used, the available replication address on the target array is used and the replication address of the source array is automatically populated on the target array connection.

```
# purearray connect --management-address 10.11.123.12
Enter the connection key of the target array:
```

3 When prompted, enter (paste) the target array's connection key.

4 If using NAT, on the target array issue the **`purearray setattr --replication address REPLICATION_ADDRESS`** command, where `REPLICATION_ADDRESS` is the replication address of the source array.

5 The connection between two arrays in an object replication set up can be encrypted. To enable encryption between two arrays, on the source array run the **`purearray enable --connect --encrypted NAME`** command, where NAME is the name of the target array.

Creating a Policy

Policies set the rules that govern the frequency of replication and the duration that snapshots are retained.

1 From source array, issue the **`purepolicy create POLICY`** command, where `POLICY` is the name of the snapshot policy.

```
# purepolicy create snap_pol1
```

2 Add rules to the new policy by issuing the **`purepolicy rule add`** command. You must specify when the snapshots are created, the interval at which they are created, and the duration they are retained using the `--at`, `--every`, and `--keep-for` arguments. In the following example, snapshots are created at 12 AM each day and retained for one day for the policy `snap_pol1`.

```
# purepolicy rule add --at 12am --every 1d --keep-for 1d snap_pol1
```

Next, create the file replica link.

Creating a File Replica Link

Once the source and target arrays are connected, you must create a file replica link on the source array. The file replica link contains the details and specifies the rules of the replication relationship between the local and remote file systems.

From the source array, issue the **`purefs replica-link create --policy LOCAL_FILE-SYSTEM`** command, where `LOCAL_FILE-SYSTEM` is the name of the source file system. Additionally, specify the remote array as well as the remote file system to which the source will be replicating, using the `--remote` and `--remote-fs` arguments.

```
# purefs replica-link create --remote Array2 --remote-fs fs1-R --policy pol_pol1
fs1
```

In this example, the replica link is created between the file system `fs1` on the local array and the remote file system `fs1-R` (also created with this command) on the remote array `Array2`. Replication occurs per the schedule defined in the policy `pol_pol1`. Note that if the file system `fs1-R` already existed on the remote array, this command would fail.

Setting Up Array to Array Object Replication Using the GUI

The following instructions describe how to set up object replication from one FlashBlade array to another FlashBlade array using the Purity//FB GUI. These instructions should be followed in the order that they are presented.

Creating Replication Network Interfaces

Perform the following on both your source and target FlashBlade arrays:

1. Select **Settings** > **Network**.
2. In the **Subnets** list, locate the subnet to which you wish to add a network interface and click the **Add interface** (+) button. The **Create Network Interface** pop-up window appears.
3. In the **Name** field, enter the interface name.
4. In the address field, enter the network address.
5. From the **Services** list, select `replication`.
6. The subnet defaults to the subnet in which you are creating the interface. Click **Create**.

Creating a Connection Key

⚠ **Important note!** The connection key is active for two hours. If the source and target arrays have not been connected within that two hour period, a new connection key must be created.

Once you have created replication network interfaces, you must create a connection key on the target array.

1. On the target array, select **Storage** > **Array** page.
2. In the **FlashBlade Array Connections** pane, click **More Options** > **Create Connection Key**. The **Connection Key** pop-up window appears displaying the new connection key.
3. Copy the new connection key.

Connecting the Source and Target Arrays

With the creation of replication network interfaces on the source and target arrays, and the creation of a connection key on the target array, you can now connect the source and target arrays.

Note: Encryption is only valid for object replication.

⚠ **Important note!** When connecting the source array to the target array, a secure connection is required even if encryption is not enabled. Certificates with small key sizes, weak signature algorithms, or that only support weak TLS algorithms are not supported.

⚠ **Important note!** The management IP address required during configuration cannot be a loopback IP address.

1. On the source array, select **Storage** > **Array** page, click the add (+) button in the **FlashBlade Array Connections** panel. The **Connect FlashBlade Array** pop-up window appears.
2. Enter the target array's hostname or management address in the **Management Address** field. The address can be located from the **Subnets** table by navigating to the **Settings** > **Network** page.
3. Paste the copied connection key for the target array in the **Connection Key** field.
4. The replication address is auto-discovered, unless using NAT. If using NAT, enter the target array's replication network address in the **Replication Address** field.
5. (Optional) Enable encryption by setting the **Encrypted** toggle to on. If set, the `_default_replication_certs` CA certificate group is applied. Before toggling, make sure to import the CA certificate to the source array if the target array's array-certificate is signed by a CA and add it to the certificate group. Note that encryption is set on the source array only. If enabled, the encryption setting displays as enabled on the target array.

   Warning: Use of CA certificates to establish trust and secure communication for object replication is the recommended configuration. If the target array's array-certificate is self-signed, directly importing the target array certificate to the source and adding it to the certificate group will not provide as strict validation around aspects of the peer, such as certificate expiry.
6. Click **Connect**.
7. If using NAT, on the target array, enter the source array's replication address.
   a. On the target array, in the **FlashBlade Array Connections** panel on the **Storage** > **Array** page, click the edit button at the end of the row displaying the connected source array. The **Edit Connected Array** pop-up window appears.
   b. Enter the replication address for the source array in the **Replication Address** field and click **Save**.

Creating Remote Credentials

Object replica links, which allow object data to be replicated between arrays and S3 targets, require remote credentials. In order to create a remote credential, you must have previously created an access key ID and secret access key (see *Creating a User* for instructions).

1. From the **Protection** > **Object Replica Links** page, click the add (+) button in the heading of the **Remote Credentials for Object Replication** list. The **Create Remote Credentials** pop-up window appears.
2. From the **Remote Array or Target** list, select the connected remote (target) array.
3. In the **Name** field, enter the credential name.
4. Enter the user access key from the target array in the **Access Key ID** field.
5. Enter the user secret access key from the target array in the **Secret Access Key** field.
6. Click **Create**.

Creating an Object Replica Link

Once the source and target are connected, you must create an object replica link on the source array. The object replica link contains the details about the source array and bucket from which objects are replicated, and the target bucket to which objects are replicated.

1. From the **Protection** > **Object Replica Links** page, click the add (+) button in the heading of the **Object Replica Links** list. The **Create Object Replica Link** pop-up window appears.
2. From the **Local Bucket Name** list, select the local (source) bucket from which data will be replicated.
3. From the **Remote Array or Target** list, select the remote (target) array.
4. From the **Remote Bucket** field, enter the name of the remote (target) bucket to which data will be replicated.
5. From the **Remote Credential** list, select the remote credential that was previously created to allow replication.
6. Click **Create**.

Enabling Bucket Versioning

Bucket versioning allows multiple variants of an object in the same bucket.

Object replication requires versioning to be enabled on the target bucket. Versioning is not required on the source bucket. On buckets with versioning enabled, you can use lifecycle rules to control the space used by non-current object versions.

In a versioning-enabled bucket, an object can have multiple versions for the same object name. Each object name has one current version and either zero or multiple non-current versions. Modifying an existing object results in that instance of the object name becoming the most current version and the previous version becoming a non-current version.

1. From the **Storage** > **Object Store** page, locate the bucket for which you wish to enable versioning in the **Buckets** panel.
2. Click the more options button at the end of the row in which the bucket appears and select **Enable versioning**.
3. Click **Enable** when prompted for confirmation.

Setting Up Array to Array Object Replication Using the CLI

The following instructions describe how to set up object replication from one FlashBlade array to another FlashBlade array using the Purity//FB CLI. These instructions should be followed in the order that they are presented.

Creating Replication Network Interfaces

On both the source and target arrays, issue the **`purenetwork vip create --address ADDRESS --servicelist replication`** command, where ADDRESS is the IP address associated with the replication vip.

```
# purenetwork vip create --address 123.123.123.123 --servicelist replication
```

Creating a Connection Key

⚠ **Important note!** The connection key is active for two hours. If the source and target arrays have not been connected within that two hour period, a new connection key must be created.

Once you have created replication network interfaces on the source and target arrays, you must create a connection key on the target array.

1. Issue the purearray **`create --connection-key`** command.

```
# purearray create --connection-key
Connection Key                          Created                  Expires
T-71a63c01-80aa-4173-842c-8ec9a1285d5b  2020-01-09 14:49:08 PST  2020-01-09
 16:49:08 PST
```

2. Copy the connection key.

Connecting the Source and Target Arrays

With the creation of replication network interfaces on the source and target arrays, and the creation of a connection key on the target array, you can now connect the source and target arrays.

⚠ **Important note!** When connecting the source array to the target array, a secure connection is required even if encryption is not enabled. Certificates with small key sizes, weak signature algorithms, or that only support weak TLS algorithms are not supported.

⚠ **Important note!** The management IP address required during configuration cannot be a loopback IP address.

1 On the target array, issue the **`purenetwork list`** command and make note of the array's management IP address.

```
# purenetwork list
Name          Enabled  Subnet Address       ...  Services
vir1          True     admin  10.11.123.12  ...  management
```

2 On the source array, issue the **`purearray connection --management-address MANAGEMENT_ADDRESS`** command, where `MANAGEMENT_ADDRESS` is the mangement IP address of the target array.

- You can optionally specify the `--replication-address` argument, which will set the specified replication address on the source array and the remote array connection wil have no replication address set.
- If the `--replication-address` argument is not used, the available replication address on the target array is used and the replication address of the source array is automatically populated on the target array connection.

```
# purearray connect --management-address 10.11.123.12
Enter the connection key of the target array:
```

3 When prompted, enter (paste) the target array's connection key.

4 If using NAT, on the target array issue the **`purearray setattr --replication address REPLICATION_ADDRESS`** command, where `REPLICATION_ADDRESS` is the replication address of the source array.

5 The connection between two arrays in an object replication set up can be encrypted. To enable encryption between two arrays, on the source array run the **`purearray enable --connect --encrypted NAME`** command, where NAME is the name of the target array.

Creating Remote Credentials

Object replica links, which allow object data to be replicated between arrays, require remote credentials. In order to create a remote credential, you must have previously created an access key ID and secret access key using the **`pureobj user access-key create --user`** command.

To create remote credentials, issue the **`pureobj remote-credentials create`** command. You must specify the `--access-key-id` argument to set the access key ID and also specify the name of the remote credentials. The remote credentials name must be specified in the format of `<remote_array_name>/<remote_credential_name>`.

In the following example, the credentials `array1/credential1` are created.

```
# pureobj remote-credentials create --access-key-id ABCDE12345 array1/credential1
```

Creating an Object Replica Link

Once the source and target are connected, and you have created remote credentials, you must create an object replica link on the source array. The object replica link contains the details about the source array and bucket from which objects are replicated, and the target bucket to which objects are replicated.

From the source array, issue the **`purebucket replica-link create BUCKET`** command, where BUCKET is the name of the local bucket from which object data is replicated. You must also specify the name of the remote bucket and the remote credentials using the `--remote-bucket` and `--remote-credentials` arguments.

```
# purebucket replica-link create --remote-bucket r_bucket1 --remote-credentials
 array1/credential1 bucket1
```

In this example, the replica link is created between the local bucket `bucket1` on the local array and the remote bucket `r_bucket1`.

Enabling Bucket Versioning

Bucket versioning allows multiple variants of an object in the same bucket.

Object replication requires versioning to be enabled on the target bucket. Versioning is not required on the source bucket. On buckets with versioning enabled, you can use lifecycle rules to control the space used by non-current object versions.

In a versioning-enabled bucket, an object can have multiple versions for the same object name. Each object name has one current version and either zero or multiple non-current versions. Modifying an existing object results in that instance of the object name becoming the most current version and the previous version becoming a non-current version.

Issue the **`purebucket enable --versioning BUCKET`** command, where bucket is the name of the remote bucket.

```
# purebucket enable --versioning r_bucket1
```

In this example, versioning is enabled on the remote bucket `r_bucket1`.

Setting Up FlashBlade to Amazon S3 Object Replication Using the GUI

The following instructions describe how to set up object replication from one FlashBlade array to an Amazon S3 target using the Purity//FB GUI. These instructions should be followed in the order that they are presented.

Creating a Replication Network Interface

Perform the following on the source FlashBlade array:

1. Select **Settings** > **Network**.
2. In the **Subnets** list, locate the subnet to which you wish to add a network interface and click the **Add interface** (+) button. The **Create Network Interface** pop-up window appears.
3. In the **Name** field, enter the interface name.
4. In the address field, enter the network address.
5. From the **Services** list, select `replication`.
6. The subnet defaults to the subnet in which you are creating the interface. Click **Create**.

Connecting the FlashBlade Array to the S3 Target

With the creation of replication network interfaces on the source array, you can now connect the source array and S3 target.

1. On the source array, select **Storage** > **Array** page, click the add (+) button in the **S3 Target Connections** panel. The **Connect S3 Target** pop-up window appears.
2. Enter the S3 target name in the **Name** field.
3. Enter the S3 targets's address in the **Address** field. For example, `s3.amazonaws.com`.

4 Click **Connect**.

Creating Remote Credentials

Object replica links, which allow object data to be replicated between arrays and S3 targets, require remote credentials. In order to create a remote credential, you must have previously created an access key ID and secret access key (see *Creating a User* for instructions).

1 From the **Protection** > **Object Replica Links** page, click the add (+) button in the heading of the **Remote Credentials for Object Replication** list. The **Create Remote Credentials** pop-up window appears.

2 From the **Remote Array or Target** list, select the connected remote (target) array.

3 In the **Name** field, enter the credential name.

4 Enter the user access key from the target array in the **Access Key ID** field.

5 Enter the user secret access key from the target array in the **Secret Access Key** field.

6 Click **Create**.

Creating an Object Replica Link

Once the source and target are connected, you must create an object replica link on the source array. The object replica link contains the details about the source array and bucket from which objects are replicated, and the target bucket to which objects are replicated.

1 From the **Protection** > **Object Replica Links** page, click the add (+) button in the heading of the **Object Replica Links** list. The **Create Object Replica Link** pop-up window appears.

2 From the **Local Bucket Name** list, select the local (source) bucket from which data will be replicated.

3 From the **Remote Array or Target** list, select the remote (target) array.

4 From the **Remote Bucket** field, enter the name of the remote (target) bucket to which data will be replicated.

5 From the **Remote Credential** list, select the remote credential that was previously created to allow replication.

6 Click **Create**.

Enabling Bucket Versioning

Bucket versioning allows multiple variants of an object in the same bucket.

Object replication requires versioning to be enabled on the target bucket. Versioning is not required on the source bucket. On buckets with versioning enabled, you can use lifecycle rules to control the space used by non-current object versions.

In a versioning-enabled bucket, an object can have multiple versions for the same object name. Each object name has one current version and either zero or multiple non-current versions. Modifying an existing object results in that instance of the object name becoming the most current version and the previous version becoming a non-current version.

Bucket versioning can be enabled on an Amazon S3 bucket via the AWS Management Console, the Amazon S3 API, or the AWS CLI.

Optionally configure a lifecycle rule on the target bucket to control space usage of non-concurrent versions.

Setting Up FlashBlade to Amazon S3 Object Replication Using the CLI

The following instructions describe how to set up object replication from one FlashBlade array to an Amazon S3 target using the Purity//FB CLI. These instructions should be followed in the order that they are presented.

Creating a Replication Network Interface

On the source array, issue the **`purenetwork vip create --address ADDRESS --servicelist replication`** command, where ADDRESS is the IP address associated with the replication vip.

```
# purenetwork vip create --address 123.123.123.123 --servicelist replication
```

Connecting the FlashBlade Array to the S3 Target

On the FlashBlade array, issue the **`puretarget s3 create TARGET`** command, where TARGET is the name of the S3 target. You must use the `--address` argument to specify the address of the S3 target.

```
# puretarget s3 create --address s3.amazonaws.com s3r1
```

In this example, the S3 target name is `s3r1` and its address is `s3.amazonaws.com`.

Creating Remote Credentials

Object replica links, which allow object data to be replicated between arrays, require remote credentials. In order to create a remote credential, you must have previously created an access key ID and secret access key using the **`pureobj user access-key create --user`** command.

To create remote credentials, issue the **`pureobj remote-credentials create`** command. You must specify the `--access-key-id` argument to set the access key ID and also specify the name of the remote credentials. The remote credentials name must be specified in the format of `<remote_array_name>/<remote_credential_name>`.

In the following example, the credentials `array1/credential1` are created.

```
# pureobj remote-credentials create --access-key-id ABCDE12345 array1/credential1
```

Creating an Object Replica Link

Once the source and target are connected, and you have created remote credentials, you must create an object replica link on the source array. The object replica link contains the details about the source array and bucket from which objects are replicated, and the target bucket to which objects are replicated.

From the source array, issue the **`purebucket replica-link create BUCKET`** command, where BUCKET is the name of the local bucket from which object data is replicated. You must also specify the name of the remote bucket and the remote credentials using the `--remote-bucket` and `--remote-credentials` arguments.

```
# purebucket replica-link create --remote-bucket r_bucket1 --remote-credentials
 array1/credential1 bucket1
```

In this example, the replica link is created between the local bucket `bucket1` on the local array and the remote bucket `r_bucket1`.

Enabling Bucket Versioning

Bucket versioning allows multiple variants of an object in the same bucket.

Object replication requires versioning to be enabled on the target bucket. Versioning is not required on the source bucket. On buckets with versioning enabled, you can use lifecycle rules to control the space used by non-current object versions.

In a versioning-enabled bucket, an object can have multiple versions for the same object name. Each object name has one current version and either zero or multiple non-current versions. Modifying an existing object results in that instance of the object name becoming the most current version and the previous version becoming a non-current version.

Bucket versioning can be enabled on an Amazon S3 bucket via the AWS Management Console, the Amazon S3 API, or the AWS CLI.

Optionally configure a lifecycle rule on the target bucket to control space usage of non-concurrent versions.

Performing File Replication Failover and Reprotect Using the GUI

CAUTION: If Pure File SafeMode is enabled, contact Pure Technical Services to promote and demote file systems.

To initiate failover in a file replication setup:

- The target file system must first be promoted.
- All clients must then be directed to the target array.
- The local file system is then demoted. Note that when a file system is demoted, any writes performed since the last replication will be discarded.

To initiate failover, perform the following:

1. On the target array, from the **Storage** > **File Systems** page, locate the remote file system that you wish to promote from the **File Systems** panel.
2. Click **More Options** > **Promote** at the end of the row in which the remote file system appears.
3. Click **Promote** when prompted for confirmation.
4. Manually redirect clients to the target array.

   To reprotect, you must remove snapshots since the last replicated snapshot. This can be difficult if the policy engine is creating new snapshots. You can stop the policy engine from creating new snapshots by removing the policies form the file system and replica link.

5 On the local array, remove policies attached to the to-be-demoted local file system and replica link.

To remove the policy from the local file system:

a From the **Storage** > **File Systems** page, locate and click local file system that you wish to demote from the **File Systems** panel. The file system's details page appears.

b In the **Policies** panel, click the remove button (**X**) in the same row as the policy you wish to remove.

c Click **Remove** when prompted for confirmation.

To remove the policy from the replica link:

a From the **Protection** > **File Replica Links** page, click **More Options** > **Remove Policies** at the end of the row of the replica link you wish to edit. The **Remove Policies** pop-up window appears.

b Select the policy or policies you wish to remove from the replica link from the **Available Policies** list.

c Click **Remove**.

6 Navigate back to the **Storage** > **File Systems** page and locate the local file system that you wish to demote from the **File Systems** panel.

7 Click **More Options** > **Demote** at the end of row in which the local file system appears.

8 Click **Demote** when prompted for confirmation.
Writes to the local file system are stopped and are redirected to the promoted remote file system.

9 If you removed policies in step 5, add the removed policy or policies back to the file system and replica link.

To add the policy to the file system:

a From the **Storage > File Systems** page, locate and click the file system from the **File Systems** panel. The file system's details page appears.

b In the **Policies** panel, click the add button (+). The **Add Policies** pop-up window appears.

c Select the policy or policies you wish to add to the file system from the **Available Policies** list.

d Click **Add**.

To add the policy to the replica link:

a From the **Protection > File Replica Links** page, click **More Options > Add Policies** at the end of the row of the replica link you wish to edit. The **Add Policies** pop-up window appears.

b Select the policy or policies you wish to add back to the replica link from the **Available Policies** list.

c Click **Add**.

Once the local file system comes back up, replication resumes as scheduled between the newly promoted target file system and demoted local file system because the policy was re-added to the replica link. Because clients have been redirected to the target, data is written to the promoted remote file system, which is then replicated to the demoted local file system.

Performing Failover Using the CLI

⚠ **CAUTION:** If Pure File SafeMode is enabled, contact Pure Technical Services to promote and demote file systems.

In a replication setup, failover requires promoting the file system on target (remote) array and then demoting the file system on the local (source) array.

Note that when demoting the local file system, any writes performed since the last replication will be discarded. If a local snapshot was taken since last replication, demoting the file system requires that snapshot be destroyed and eradicated. You must use the `--discard-non-snapshotted-data` option to remove any data since the last snapshot of the file system was taken. The file system must either have no snapshots, or the most recent snapshot must be a replication snapshot.

In the procedure below:

- The local array is `array1`

- The local file system is `fs1`
- The target array is `array2`
- The target file system is `fs1-R`
- The policy attached to the local file system `fs1` is `fs_snap_pol1`.
- The policy attached to the replica link is `rl_pol1`.

1 On the target array, `array2`, promote the target file system. In the following example, the target file system `fs1-R` is promoted:

```
# purefs promote fs1-R
```

2 Manually redirect clients to the the target array, `array2`.

3 On the local array, `array1`, remove the policy `fs_snap_pol1` that attached to the local file system `fs1`

```
# purepolicy remove fs_snap_pol1
```

4 Remove the policy from the replica link using the **`purepolicy remove --remote`** command, where `--remote` specifies the remote (target) array. In the following example the policy `rl_pol1` is removed from the replica link:

```
# purepolicy remove --remote array2 rl_pol1
```

5 On the local array, `array1`, demote the local file system :

```
# purefs demote --discard-non-snapshotted-data fs1
```

6 Add the policy back to the replica link using the **`purepolicy add --remote`** command, where `--remote` specifies the demoted array In the following exmaple, the policy `rl_pol1` is added back to the replica link:

```
# purepolicy add --remote array1 rl_pol1
```

7 On the local array, verify that replication is disabled and that the target file system is promoted:

```
# purefs replica-link list
Name  Direction  Remote  Remote File System  Policy       Status     Recov...
fs1   <--        array2  fs1-R               rl_pol1      unhealthy  2019-...

purefs replica-link list --status-details
Name  Direction  Remote  Remote File System  Policy       Status     Details
fs1   <--        array2  fs1-R               rl_pol1      unhealthy  The
 target file system is promoted.
```

Once the local file system comes back up, replication resumes as scheduled between the newly promoted target file system and demoted local file system because the policy was re-added to the replica link. Because clients have been redirected to the target, data is written to the promoted remote file system, which is then replicated to the demoted local file system.

Performing Failback Using the GUI

CAUTION: If Pure File SafeMode is enabled, contact Pure Technical Services to promote and demote file systems.

Note: In this procedure, the remote file system was promoted and the local file system was demoted during the failover procedure.

As noted in the failover procedure, if the replication schedule policy was re-added to the replica link in the replication setup, once the local file system comes back up, replication resumes as scheduled between the promoted target file system and demoted local file system. Performing failback restores the original local/target relationship between arrays and file systems in a replication setup. During failback, once the remote file system is demoted and the local file system is promoted and clients redirected to the local file system, the original replication setup resumes; data is written to the local file system and then replicated to the remote file system.

To initiate failback in a file replication setup:

- Stop writes on the promoted remote file system.
- Optionally manually create a snapshot of the file system and send it to the demoted local file system to ensure that data written since the last snapshot resides on the to-be-promoted file system.
- Promote the previously demoted local file system.
- Add back any removed policy to the local file system and replica link.
- Demote the remote file system.

To initiate failback, perform the following:

1. On the target array, from the **Storage > File Systems** page, locate the remote file system that is to be demoted from the **File Systems** panel.
2. Click **More Options > Disable Writes** at the end of row in which the remote file system appears.
3. Click **Disable** when prompted for confirmation.
4. Ensure all writes are sent to the source file system by creating a snapshot on the target and send it to the source.

5 On the local array, from the **Storage** > **File Systems** page, locate the local file system that you wish to promote from the **File Systems** panel.

6 Click **More Options** > **Promote** at the end of row in which the local file system appears.

7 Manually redirect clients back to the to the local array.

8 Remove policies attached to the to-be-demoted remote file system and replica link.

To remove the policy from the file system:

a From the **Storage** > **File Systems** page, locate and click local file system that you wish to demote from the **File Systems** panel. The file system's details page appears.

b In the **Policies** panel, click the remove button (**X**) in the same row as the policy you wish to remove.

c Click **Remove** when prompted for confirmation.

To remove the policy from the replica link:

a From the **Protection** > **File Replica Links** page, click **More Options** > **Remove Policies** at the end of the row of the replica link you wish to edit. The **Remove Policies** pop-up window appears.

b Select the policy or policies you wish to remove from the replica link from the **Available Policies** list.

c Click **Remove**.

9 On the target array, from the **Storage** > **File Systems** page, locate the remote file system that you wish to demote from the **File Systems** panel.

10 Click **More Options** > **Demote** at the end of row in which the remote file system appears.

11 Click **Demote** when prompted for confirmation.
Writes to the remote file system are stopped and are redirected to the promoted local file system.

12 If you removed policies in step 8, add the removed policy or policies back to the file system and replica link.

To add the policy to the file system:

a From the **Storage > File Systems** page, locate and click the file system from the **File Systems** panel. The file system's details page appears.

b In the **Policies** panel, click the add button (+). The **Add Policies** pop-up window appears.

c Select the policy or policies you wish to add to the file system from the **Available Policies** list.

d Click **Add**.

To add the policy to the replica link:

a From the **Protection > File Replica Links** page, click **More Options > Add Policies** at the end of the row of the replica link you wish to edit. The **Add Policies** pop-up window appears.

b Select the policy or policies you wish to add back to the replica link from the **Available Policies** list.

c Click **Add**.

The original replication setup has been restored.

Performing Failback Using the CLI

CAUTION: If Pure File SafeMode is enabled, contact Pure Technical Services to promote and demote file systems.

As noted in the failover procedure, if the replication schedule policy was re-added to the replica link in the replication setup, once the local file system comes back up, replication resumes as scheduled between the promoted target file system and demoted local file system. Performing failback restores the original local/target relationship between arrays and file systems in a replication setup. During failback, once the remote file system is demoted and the local file system is promoted and clients redirected to the local file system, the original replication setup resumes; data is written to the local file system and then replicated to the remote file system.

In the procedure below:

- The local array is `array2`
- The local file system is `fs1-R`
- The target array is `array1`
- The target file system is `fs1`

- The policy re-attached to the local file system `fs1` is `fs_snap_pol1`.
- The policy attached to the replica link is `rl_pol1`.

1 On the local array, `array2`, stop all writes on the local file system `fs1-R` and then transfer all remaining data to the target file system `fs1`:

```
# purefs disable --writable fs1-R
# purefs snap --send --target array1 fs1
```

2 Create a snapshot of the file system and send it to the target file system `fs1` to ensure that data since the last snapshot resides on the to-be-promoted file system (`fs1`). Skip this step if you do not need data written since the last snapshot.

3 On the target array, `array1`, promote the target file system `fs1`:

```
# purefs promote fs1
```

4 Add back any previously removed file system policy. In the following example, the `policy fs_snap_pol1` is added back to file system `fs1`.

```
# purepolicy add fs_snap_pol1
```

5 Remove the policy from the replica link using the **`purepolicy remove --remote`** command, where `--remote` specifies the remote (target) array. In the following example the policy `rl_pol1` is removed from the replica link:

```
# purepolicy remove --remote array1 rl_pol1
```

6 Manually direct all clients back to the promoted file system `fs1`.

7 On the local array, `array2`, demote the local file system `fs1-R`:

```
# purefs demote --discard-non-snapshotted-data fs1-R
```

8 Add the policy back to the replica link using the **`purepolicy add --remote`** command, where `--remote` specifies the demoted array In the following exmaple, the policy `rl_pol1` is added back to the replica link:

```
# purepolicy add --remote array2 rl_pol1
```

9 On the (now) local array `array1`, verify that the replica link has returned to its original direction (from file system `fs1` to the target file system `fs1-R`):

```
# purefs replica-link list
Name  Direction  Remote  Remote File System  Policy       Status   Lag
 Recovery P...
```

```
fs1   -->        array2  fs1-R              rl_pol1      healthy  10m
 2019-11-11 12...
```

The original replication setup has been restored.

Pure Storage, Inc.
Twitter: @purestorage

650 Castro Street, Suite 400
Mountain View, CA 94041
T: 650-290-6088
F: 650-625-9667

Sales: sales@purestorage.com
Support: support@purestorage.com
Media: pr@purestorage.com
General: info@purestorage.com