FlashBlade User Guide 2.2.3

purestorage.com

Contents

About this Guide v
Organization of this Guide v
A Note on Format and Content v

Part 1: Purity//FB Overview

Chapter 1: Overview 9
Purity//FB Conventions 9
IP Addresses 10
Troubleshooting 10

Part 2: Using the Purity//FB GUI to Administer a FlashBlade Array

Chapter 2: Purity//FB GUI Overview 13
Purity//FB GUI Navigation 14
Purity//FB GUI Login 17

Chapter 3: Dashboard 19
Hardware Health 20
Capacity 21
Recent Alerts 21
Performance 22

Chapter 4: Storage 25
File Systems 25
Object Store 41

Chapter 5: Health 47
Hardware 47
Alerts 48

Chapter 6: Analysis 53
Performance 53
Capacity 55

Chapter 7: Settings 57
System 57
Network 64
Users 67

Part 3: Using the Purity//FB CLI to Administer a FlashBlade Array

Chapter 8: Purity//FB CLI Overview 75
CLI Command Syntax 75
CLI Output 76
CLI Login 82
CLI Command Help 82

Part 4: Appendices

Appendix A: Quick Reference Guide 191
Exporting a File System 191
Adding an Alert Watcher 198

Appendix B: Documentation 201
Related Documentation 201
Contact Us 201

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 into the following major sections:

Part 1, "Purity//FB Overview" [7] - Defines FlashBlade array naming and numbering conventions.

Part 2, "Using the Purity//FB GUI to Administer a FlashBlade Array " [11] - Describes the use of the browser-based Purity//FB graphical user interface (GUI) to administer a FlashBlade array.

Part 3, "Using the Purity//FB CLI to Administer a FlashBlade Array" [73] - Describes the Purity//FB command line interface (CLI) and its use in FlashBlade array administration.

Part 4, "Appendices" [189] - 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.

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 Purity//FB 2.2.3 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.

Part 1:
Purity//FB Overview

Chapter 1. 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 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
```

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>**.

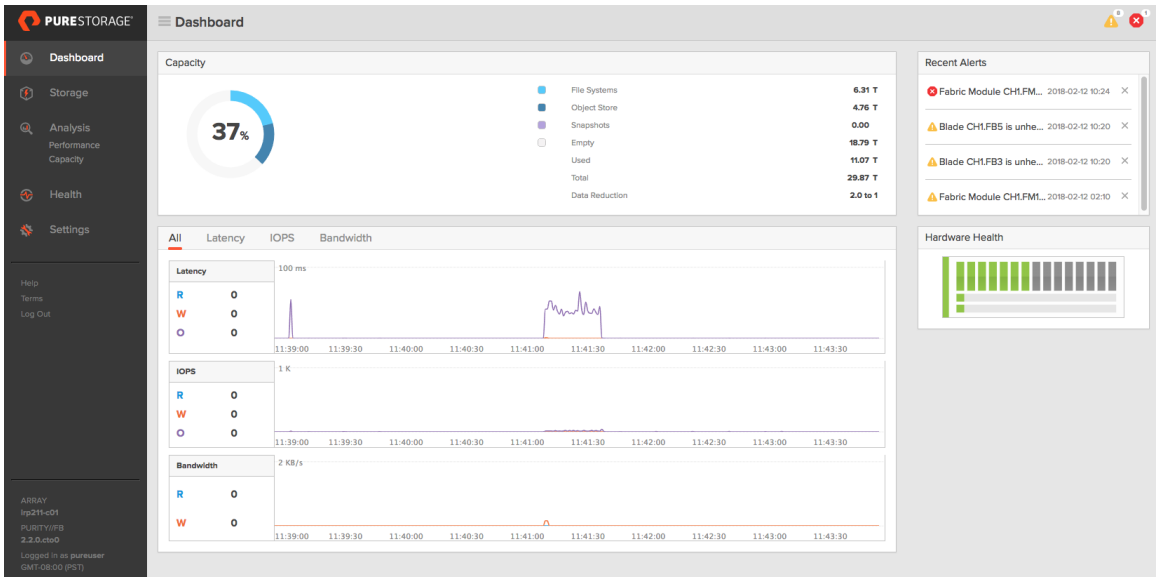
Part 2:
Using the Purity//FB GUI to Administer a FlashBlade Array

Chapter 2. 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 2.1: 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, and input/output (I/O) performance metrics.

Storage
Displays a list of file systems on the array and its attributes, including provisioned size, capacity usage details, and export rules, as well as object store accounts, buckets, and users.

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

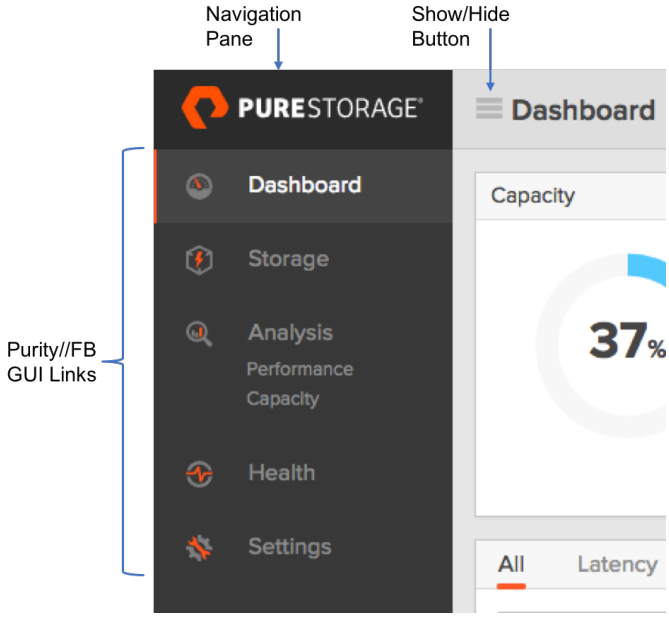
Analysis
Displays historical array information, including space and I/O performance metrics, from various viewpoints.

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 2.2: 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.

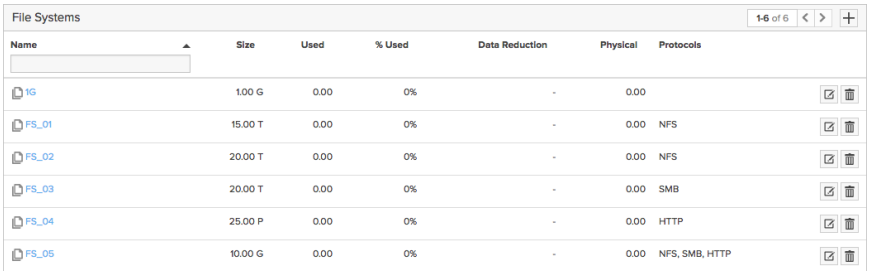
The page to the right of the navigation pane displays the information and configuration options for the selected Purity//FB GUI link. The information on each page is organized into panels, charts, and lists.

Figure 2.3: 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.

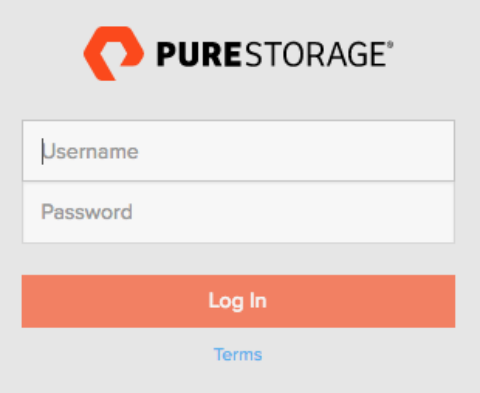
FlashBlade is installed with one administrative account with the username **`pureuser`**. The initial password for the account is **`pureuser`**. The password can be changed any time through the **`pureadmin`** CLI command.

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 2.4: 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.

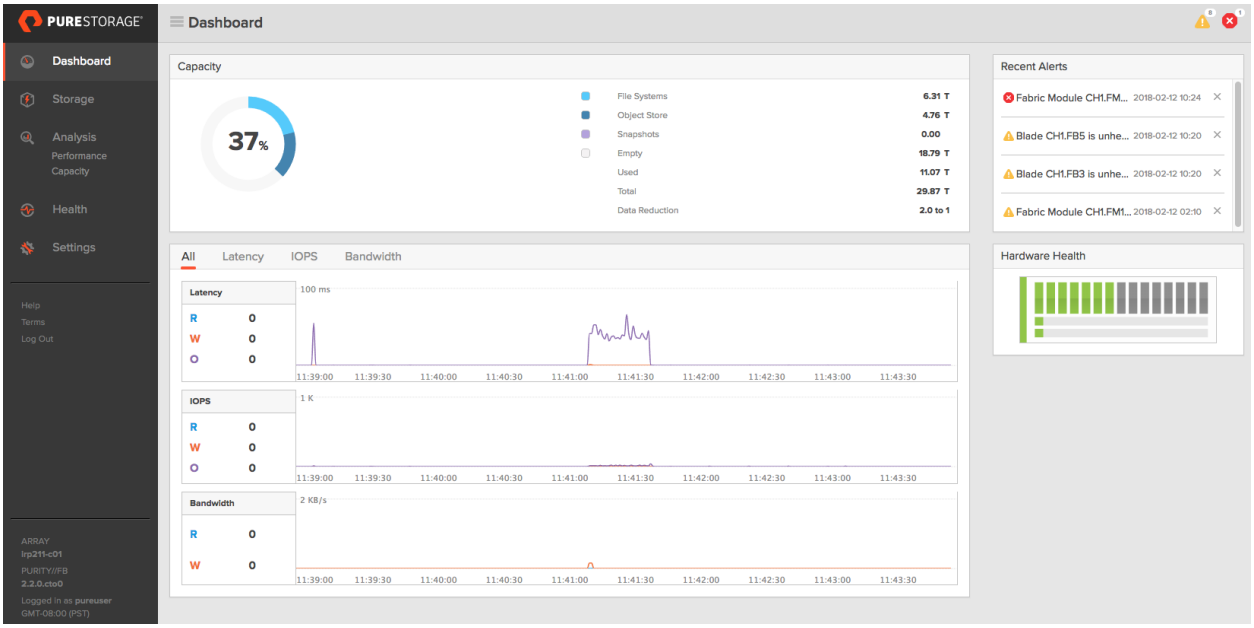
To log in to the Purity//FB GUI:

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 3. Dashboard

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

Figure 3.1: Dashboard



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

- Hardware Health
- Capacity
- Recent Alerts
- Performance Charts

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 3.2: Hardware Health Panel - Single chassis

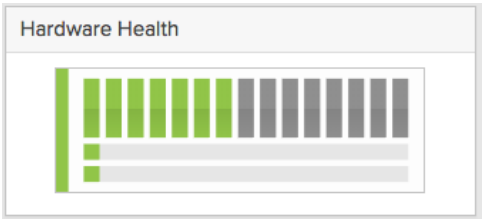
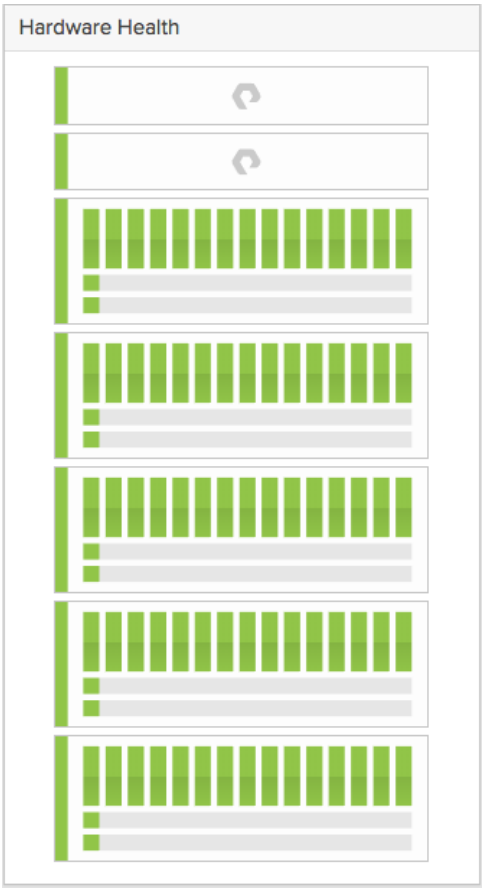


Figure 3.3: Hardware Health Panel - Multi-chassis



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

Capacity

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

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).

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**.

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

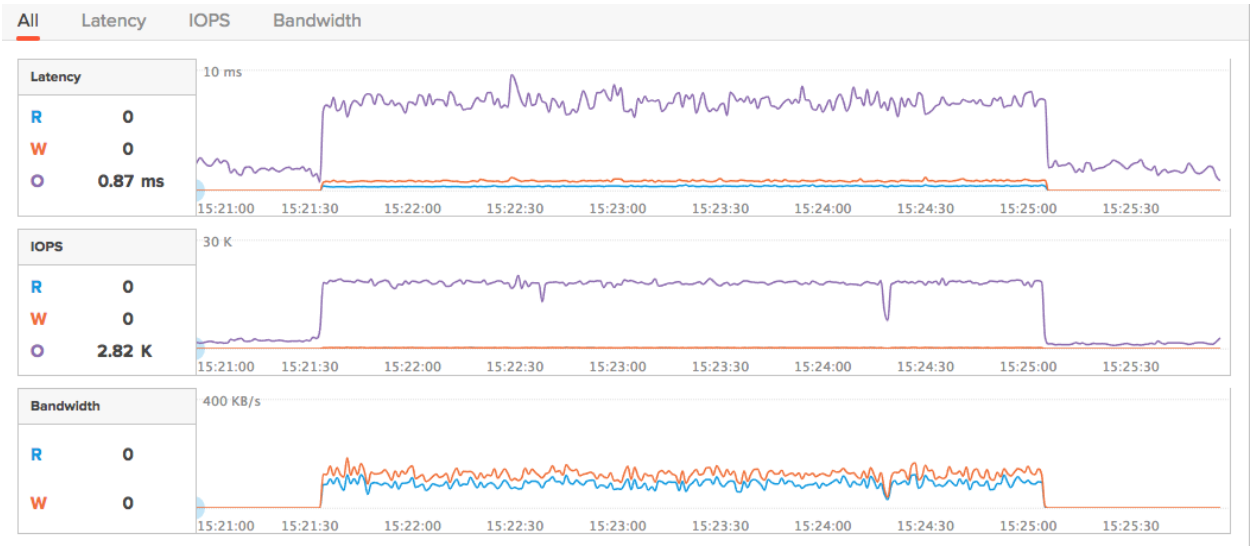
To remove an alert from the **Recent Alerts** panel, click the clear flag (**X**) button. The alert will no longer be displayed in the **Recent Alerts** panel, but will still appear on the **Health** page.

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

Performance

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

Figure 3.4: 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:

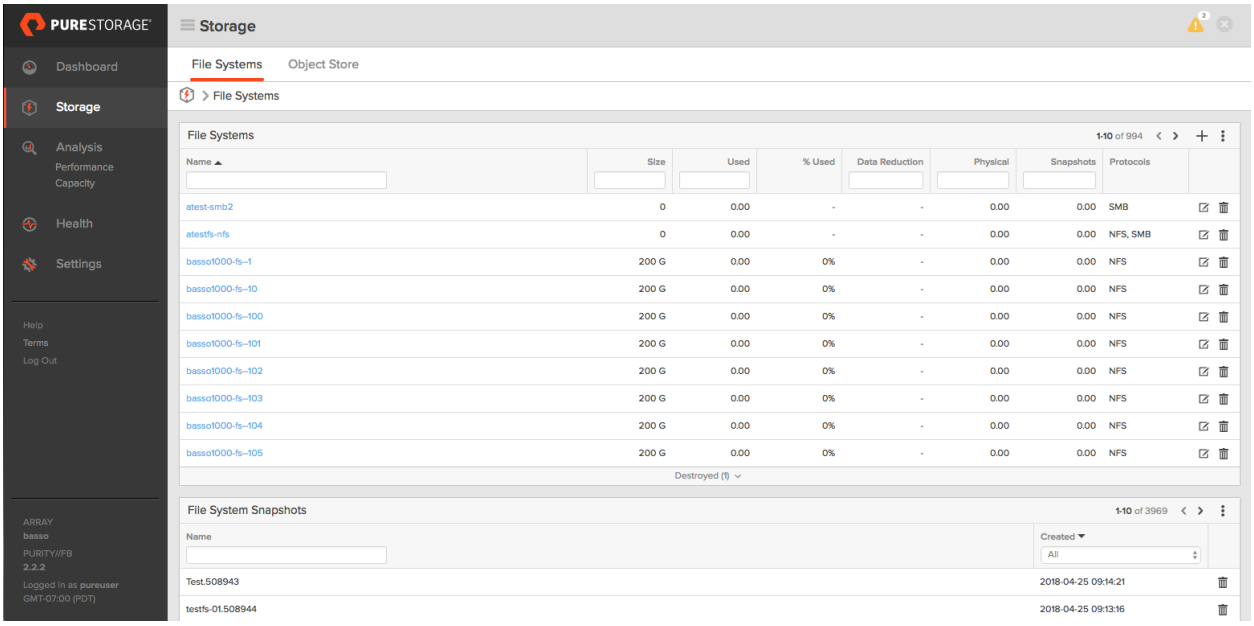
- At its shortest range (5m), the performance charts are updated once every second. As the range increases, the update frequency (and resolution) decreases.
- The performance charts display up to 1 year of historical data.
- The performance charts can be filtered to display metrics by protocol.

Chapter 4. Storage

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 4.1: File System Storage



File Systems

The **File Systems** page displays information for file systems and snapshots. The **File Systems** and **File Systems Snapshots** panels display configured file system and snapshot information while the **Destroyed File Systems** and **Destroyed Snapshots** panels display destroyed file systems and snapshots that are pending eradication.

File Systems Panel

Each file system in the list 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.
- **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.
- **Protocols**: One or more protocols configured for the file system. A dash (-) means the file system has no protocols assigned.

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

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

File System Snapshots Panel

The file system snapshot list includes the following information:

- **Name**: The snapshot name.
- **Created**: The snapshot creation date and time.

You can search the file system snapshots list by entering search criteria in the search boxes under the **Name** and **Policy** columns.

Destroyed File Systems and Destroyed Snapshots Panels

The panels display a list of destroyed file systems and snapshots and are located directly beneath the **File Systems** and **File System Snapshots** panels. By default, they are minimized. Clicking the arrow beside the **Destroyed** heading expands the panel. Once a file system or snapshot has been destroyed, they are moved to the Destroyed panels making them 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 and snapshots 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 and snapshots can be manually eradicated.

- **Name**: File system or snapshot name.
- **Time Remaining**: Indicates the file systems or snapshots expiration period in the eradication pending period. Time is represented in hours (**h**) and minutes (**m**). When time expires the file systems and snapshots 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. FlashBlade file 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 Chapter 7, *Settings* [57].

For more information about using directory services with file system protocols, refer to Chapter 7, *Settings* [57].

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 Chapter 7, *Settings* [57].

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 Chapter 7, *Settings* [57].
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 an **`INFO`** alert when the amount of data written to the file system exceeds 100% of its provisioned size. 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.

File system sizes can be changed at any time.

File System Usage Limits

Important note! File system hard limits cannot be set 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.

NFS Export Rules

NFS export rules define the access rights and privileges an NFS client has to the file system.

Rules are in the form **`host(options)`**.

`host` represents one of the following categories:

- IP address in the form **`ddd.ddd.ddd.ddd`** for IPv4, or **`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`** for IPv6.
- Netmask in the form **`ddd.ddd.ddd.ddd/dd`** for IPv4, or **`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`** for IPv6.
- Wildcard in the form **`*`** to represent all clients

`options` in parenthesis represents a comma-separated list of NFS export options.

Valid export options are **`rw`**, **`ro`**, **`root_squash`**, **`no_root_squash`**, and **`fileid_32bit`**.

If changes are being made to existing NFS export rules, the changes will be synchronously applied to all currently mounted clients.

If multiple rules are specified, separate 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:

```
Name   Size Used    % Used ...  Protocols Rules                          Fast Remove
File1  5T   4.03T   81%    ...  nfs       *                             True
File2  5T   3.42T   68%    ...  nfs       *()                           True
File3  5T   3.42T   68%    ...  nfs       10.20.225.2(rw)               True
File4  5T   3.42T   68%    ...  nfs       2620::/16(no_root_squash)     True
File5  5T   3.42T   68%    ...  nfs       *(rw,no_root_squash)          True
File6  5T   3.42T   68%    ...  nfs       *(rw,root_squash,fileid_32bit) True
```

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

- ***** - 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.
- **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.

The NFS export rules of a file system can be changed at any time.

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.

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.

SMB ACL Mode

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.

The SMB protocol supports two ACL modes, **`native`** and **`shared`**.

The **`shared`** ACL mode is set as the default and shares UNIX-like ACL permissions with the NFS protocol. In **`shared`** mode, both protocol ACL permissions are required to match. When one protocol creates files or modifies permissions, they must comply with the permission settings of the other protocol. For example, when an administrator attempts to change the SMB ACL permissions of a file to read and write, the attempt will prompt an action to perform a match on the file's NFS ACL permissions. If the NFS ACL permissions for that file is only allowed for read, then the attempt will fail.

The **`native`** ACL mode supports UNIX-like ACLs and Windows ACLs. In native mode, because SMB natively supports both ACLs while NFS only supports UNIX ACLs, ACLs sharing will be restricted between SMB and NFS. See the list of ACL sharing restrictions in **`native`** mode:

- Permissions created or changed in SMB will not be applied to NFS permissions. For example, if User X access permissions are removed for File Y using SMB, then User X will still have access to File Y through NFS.
- Permissions changed in NFS will not affect SMB applied ACLs.

- NFS permissions do not apply to directories and files created by SMB or permissions set by SMB.
- If an SMB directory or file does not have an ACL applied to it, one will be applied to it from the NFS permissions.
- If an NFS directory or file does not have an ACL applied to it, SMB ACLs will not be applied to that NFS directory/file.
- UNIX and Windows ACLs grant full permissions by default.

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.

Process Steps for File System

This section describes the steps on how to create and manage file systems through the **Storage** page of the Purity//FB GUI.

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.
      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, set the **Fast Remove** toggle button to enable (blue).
      For more information about the fast remove feature, refer to the *Fast Remove* section.

e. To enable the ability to access stored snapshots in a snapshot directory, set the **Snapshots** toggle button to enable (blue).

f. 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).
Important note! File system hard limits cannot be set for file systems with unlimited sizes.

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.
  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 Adapter Enabled** toggle button to enable (blue) the SMB protocol adapter.
  Important note! Selecting Native SMB Rich ACLs may lead to divergence from NFS access control.

- **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 Enabled** toggle button to enable (blue) the HTTP protocol.

3. Click **Create**.

Changing the SMB ACL Mode

To change the SMB protocol ACL mode:

1. From the **Storage > File Systems** page, click the 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 **SMB** in the **Protocols** section.
3. From the **Access Control** section, select **Shared access control for SMB and NFS** or **Native SMB ACLs**.
   Important note! Selecting Native SMB Rich ACLs may lead to variation from NFS access control.
4. Click **Save**.

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`**, and **`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**.

Enabling and disabling the hard limit of a file system

Important note! File system hard limits cannot be set 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.

Destroying a file system snapshot

To destroy a snapshot:

1. Select **Storage > File Systems**.
2. In the **File Systems Snapshots** panel, click the trash icon in the row of the snapshot you want to destroy. The **Destroy Snapshot** pop-up window appears.
   If you wish to destroy multiple snapshots, click **More Options > Destroy**. The **Destroy File System Snapshots** pop-up window appears. From the **Existing File System Snapshots** list, select the snapshots you want 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 checkbox next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
3. Click **Destroy**. The destroyed snapshot appears in the **Destroyed Snapshots** panel of the **File Systems** page and begins its 24-hour eradication pending period.

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

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

Recovering a file system snapshot

When a 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.

Destroyed snapshots cannot be recovered if its source file system is also destroyed. Recover the source file system before recovering any of its snapshots.

To recover a destroyed snapshot:

1. Select **Storage > File Systems**.
2. In the **Destroyed Snapshots** panel under **FIle System Snapshots**, click the clock icon in the row of the snapshot you want to restore. The **Recover Snapshot** pop-up window appears.
   If you wish to recover mutliple snapshots, click **More Options > Recover** in the **Destroyed Snapshots** panel. The **Recover File System Snapshots** pop-up window appears. 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 checkbox next to the search box. All selected snapshots appear in the **Selected File System Snapshots** list.
3. Click **Recover**. The recovered snapshot will now appear in the source file system's snapshot page.

Eradicating a file system snapshot

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.

To eradicate a destroyed snapshot:

1. Select **Storage > File Systems**.
2. In the **Destroyed Snapshots** panel under **File System Snapshots**, click the trash icon in the row of the snapshot you want to eradicate. The **Eradicate Snapshot** pop-up window appears.
   If you wish to eradicate mutliple snapshots, click **More Options > Eradicate** in the **Destroyed Snapshots** panel. The **Eradicate File System Snapshots** pop-up window appears. 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.
3. Click **Eradicate**. The array will begin reclaiming the eradicated space of the snapshot.

Enabling and disabling fast remove

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.

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.

To eradicate a destroyed file system:

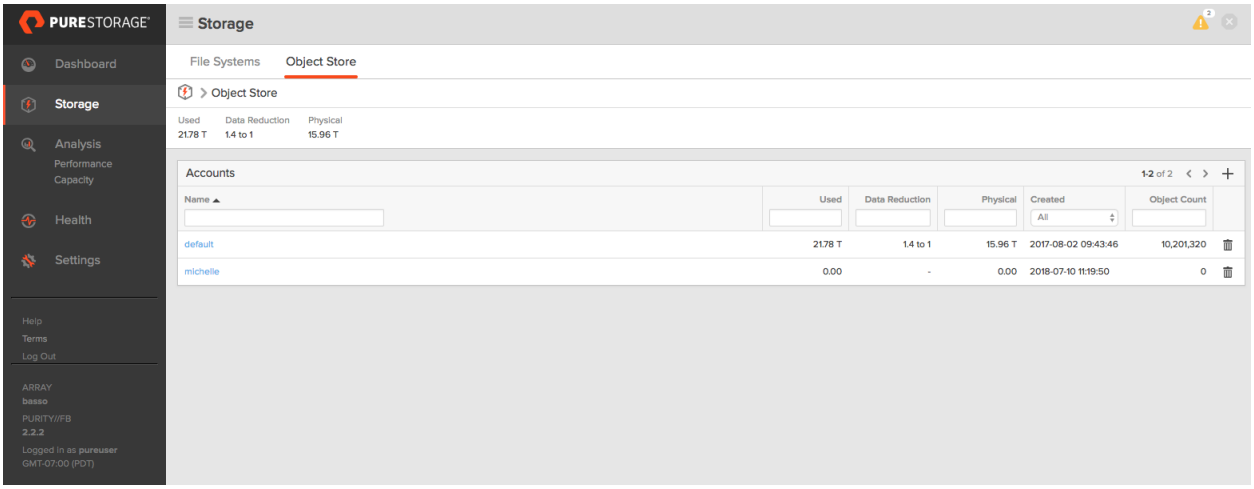
1. Select **Storage > File Systems**.
2. In the **Destroyed File Systems** panel, click the trash icon at the end of the row of the file system you want to eradicate. The **Eradicate File System** pop-up window appears.

   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.

Object Store

The **Object Store** tab is used to manage the array's object stores.

Figure 4.2: Object Store



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.

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

Buckets

Buckets and their contents can be created and managed with the FlashBlade 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.

You can also use the **`purebucket`** CLI command to list the array's buckets and the buckets' space usage. See purebucket(1) [106].

Process Steps

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.

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. Click **Create**.

Once you have created a new user, create an access key.

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.

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 `pureobjuser` CLI command to disable access keys. See pureobjuser(1) [173].

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 `pureobjuser` CLI command to enable access keys. See pureobjuser(1) [173].

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** section, click the add (+) button. The **Create Bucket** pop-up window appears.
3. Enter a name for the bucket.
4. Click **Create**.

Destroying a bucket

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** section, 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 mutliple 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

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 mutliple 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

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 5. Health

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 5.1: Health - Hardware Page (single chassis)

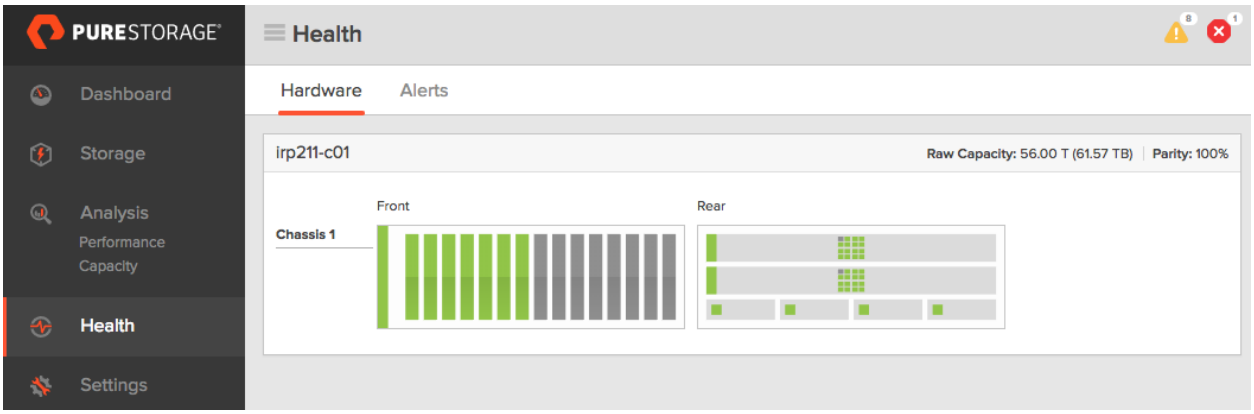
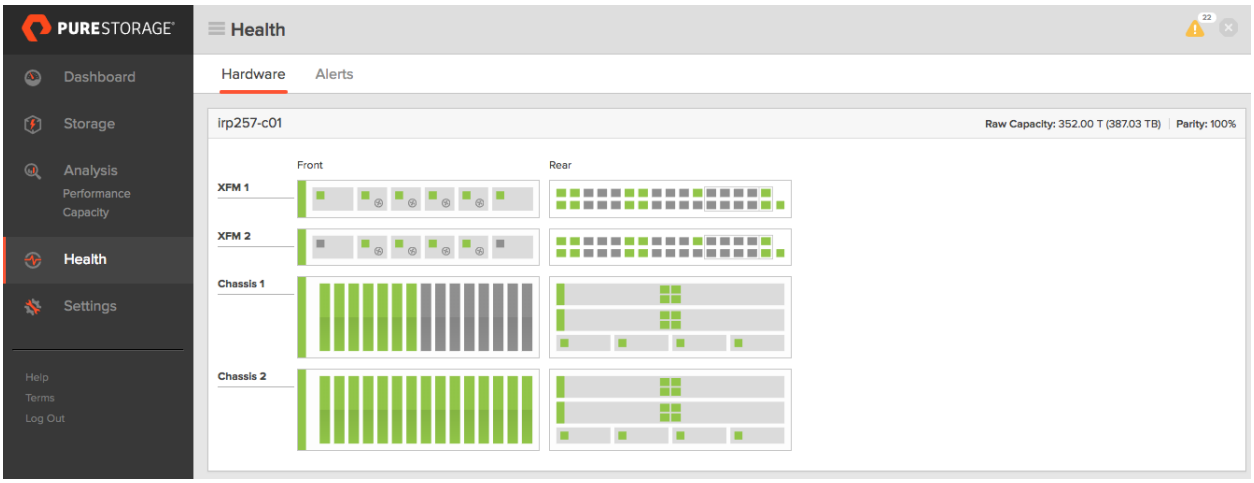


Figure 5.2: Health - Hardware Page (multi-chassis)



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

Figure 5.3: Health - Management Ports

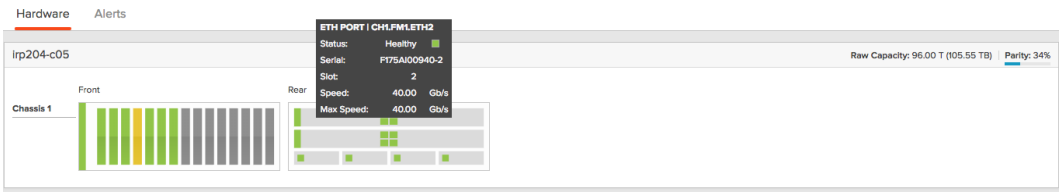
The view is a schematic representation of the array with colored indicators of each component's status.

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 5.4: 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.

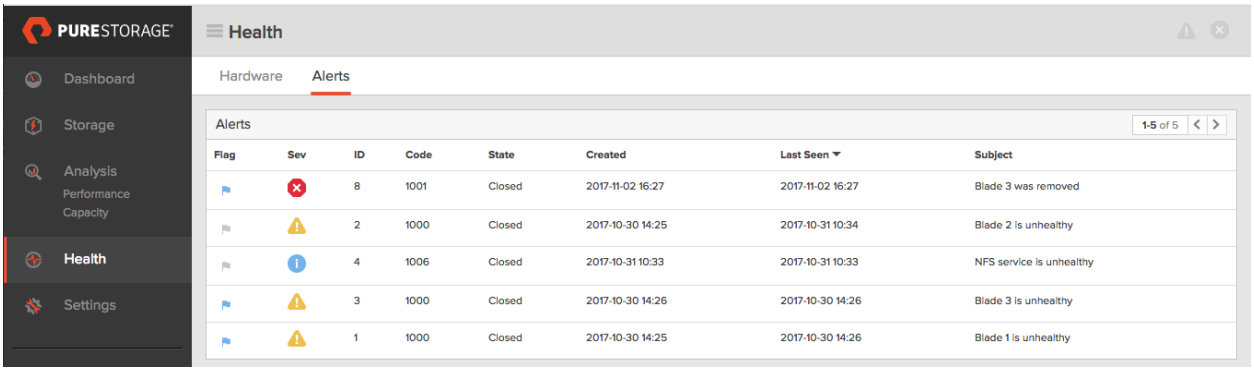
The raw capacity value represents the total usable capacity of the array, displayed in both tebibyte (T) and terabyte (TB) units.

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 5.5: 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:

- **Sev** (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.

- **ID**: Unique number assigned by the array to the alert. ID numbers are assigned to alerts in chronological ascending order.

- **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.
- **Subject**: Alert details.

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.

Flag 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.

Unflag Alert Messages

To unflag alert messages:

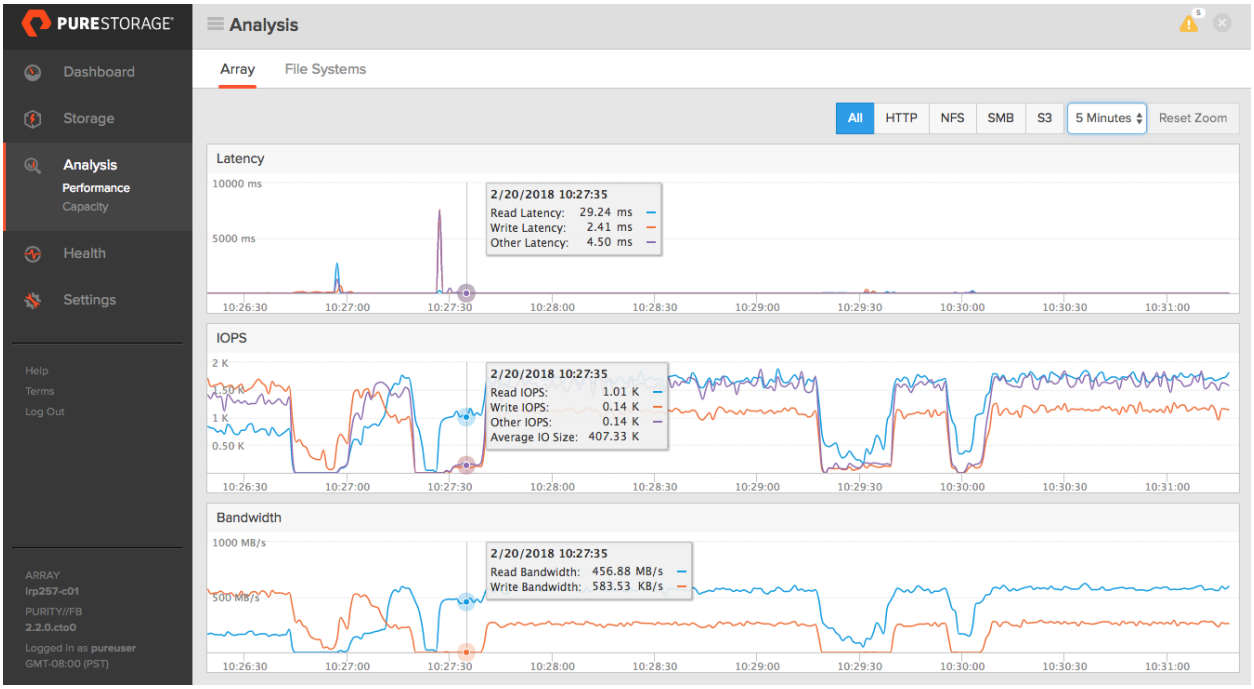
1. Select **Health > Alerts**.
2. Locate the alert you want to unflag.
3. Click the flag icon. When the icon becomes gray, the alert is unflagged.

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 6. Analysis

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

Figure 6.1: Analysis



Performance

The **Analysis > Performance** panel displays a series of rolling charts consisting of real-time performance metrics for the array and its file systems. By default, performance metrics are displayed for the array. To display performance metrics for the array's file systems, click **File Systems** and select one or more file systems from the left pane. The incoming data appear along the right side of each graph as older numbers drop off the left side.

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.

Display 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 Performance panel. 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:

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

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

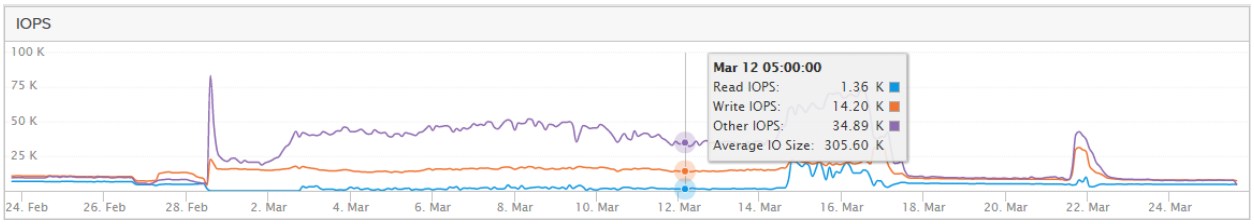
The file transfer can be performed in many chunks for one large file. The Raw view shows one operation per chunk transfer, while the User view shows only one operation per file. When transferring small files, the number of operations is the same in both the Raw and User views.

Display 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.

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.

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.

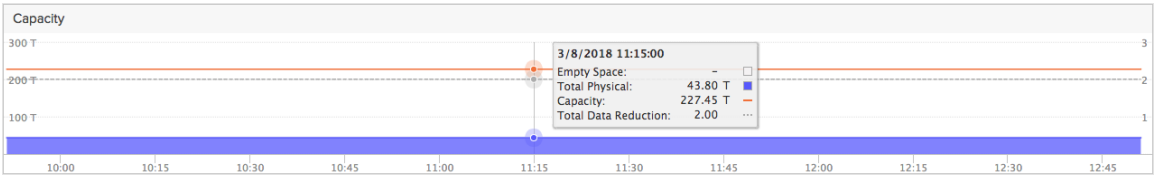
Capacity

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.

Chapter 7. Settings

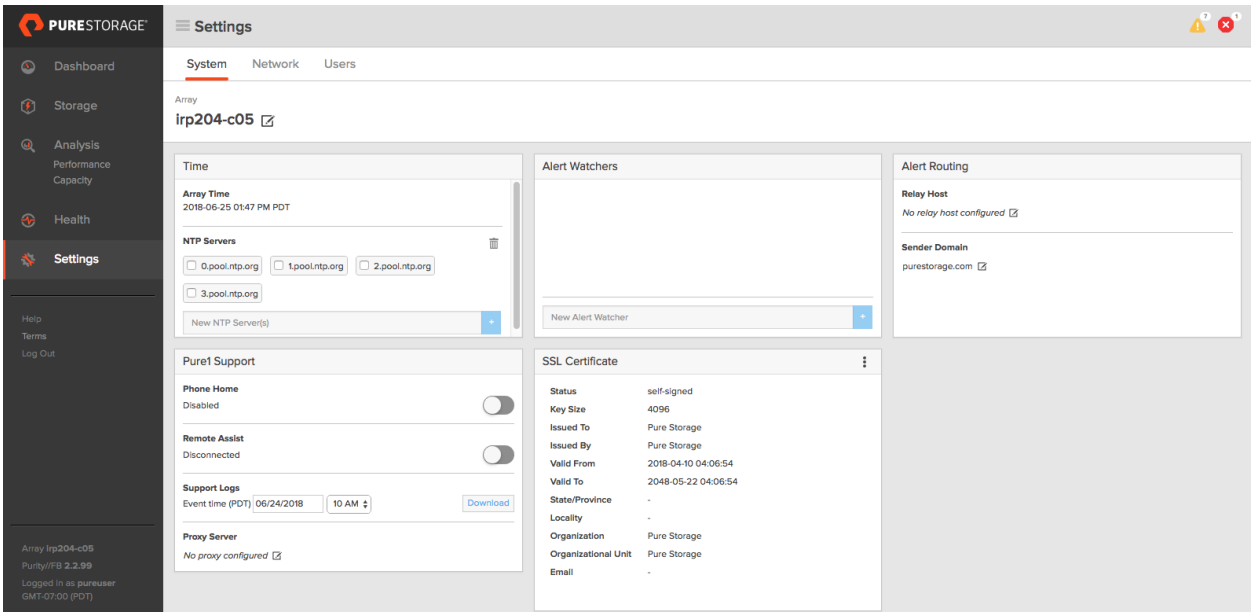
The **Settings** page displays and manages the general attributes, network settings, and directory service settings of an array.

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, Pure Storage support settings, alert settings, and the SSL certificate.

Figure 7.1: Settings - System Page



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.

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.

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.

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 Assistance label.

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

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.

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.

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 and the alert 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 starts receiving alert email notifications.

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. Alert watchers can be enabled and disabled at any time. The current alert watcher status is determined by the color of the toggle button that appears next to the alert watcher email address, where blue represents an enabled alert watcher and gray represents a disabled alert watcher.

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.

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.

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`**.

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, and cannot exceed 3000 characters in total length.

Process Steps

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.

Enabling and disabling phone home

1. Select **Settings > System**.
2. In the **Phone Home** section of the **Pure1 Support** panel, click the toggle button to switch between enabled (blue) and disabled (gray) status. Enabling phone home allows the array to transmit log and diagnostic information to Pure Storage Support.

Opening and closing a remote assistance (RA) session

1. Select **Settings > System**.
2. In the **Remote Assistance** section of the **Pure1 Support** panel, click the 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.

Downloading Support Logs

1. Select **Settings > System**.
2. In the **Support Logs** section of the **Pure1 Support** panel, specify the event date and time, representing the approximate array time the activity of interest occurred. Note that the time is based on your local browser's time zone.
3. Click **Download** to save the password encrypted file to your administrative workstation. The file can only be opened by Pure Storage Support.

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, and cannot exceed 3000 characters in length.
- **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**.

Configuring the proxy host name

1. Select **Settings > System**.
2. In the **Proxy Server** section of the **Pure1 Support** panel, click the edit icon. The field becomes an editable box.
3. In the editable box, 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 the check mark icon to confirm the change.

Deleting the proxy host name

1. Select **Settings > System**.
2. In the **Proxy Server** section of the **Pure1 Support** panel, click the edit icon next to the current proxy host name. The proxy host name becomes an editable box.
3. Delete the proxy host name.
4. Click the check mark icon to confirm the deletion.

Adding an alert watcher

1. Select **Settings > System**.
2. In the **Alert Watchers** panel, type the email address of the alert watcher.

3. Click the Add (**+**) button to add the email address to the list of alert watchers. Once added, the alert watcher immediately begins receiving alert email messages.

Enabling and disabling an alert watcher

1. Select **Settings > System**.
2. In the **Alert Watchers** panel, click the toggle button to enable (blue) or disable (gray) an alert watcher. Once enabled, an alert watcher starts receiving alert email notifications.

Deleting an alert watcher

1. Select **Settings > System**.
2. In the **Alert Watchers** panel, click the **Delete** button next to the alert watcher you want to delete.

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.

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.

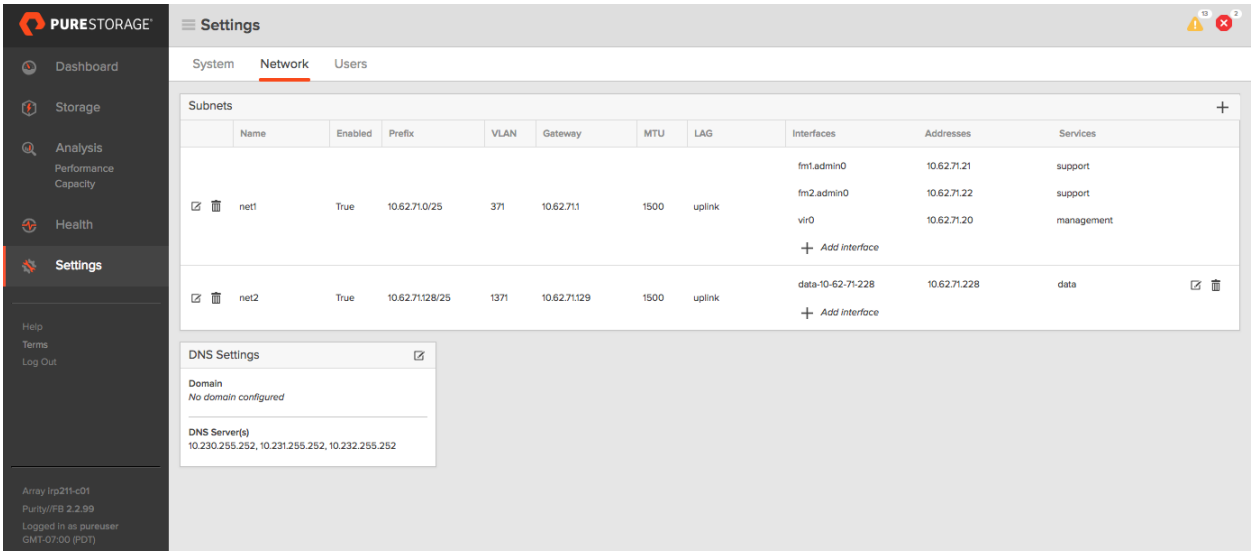
Configuring the sender domain

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.

Network

The **Settings > Network** page displays and manages the subnets and data vips required for file system export.

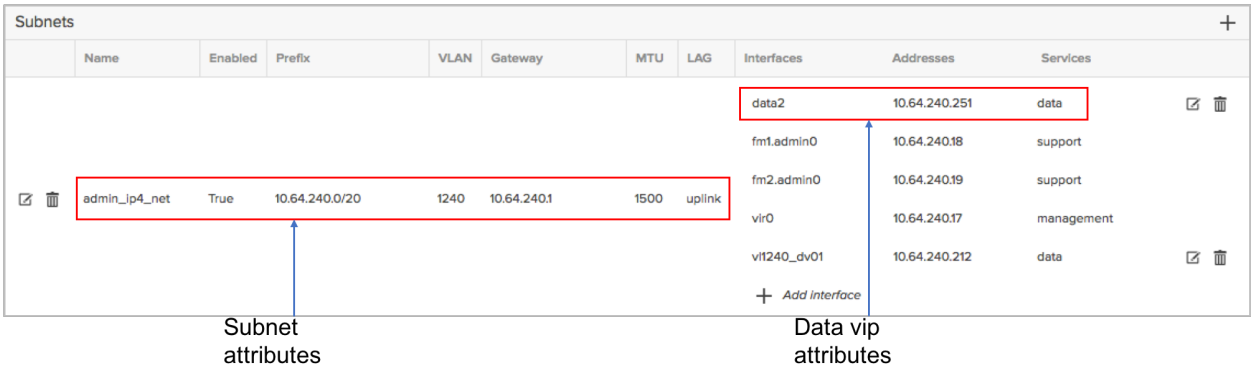
Figure 7.2: Settings - Network Page



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 **10.201.122.112** is attached to subnet **VLAN320**.



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. Data vips within the subnet that are in

disabled status remain disabled and cannot be reached. Newly created subnets are automatically enabled.

The Subnets list 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 list.

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 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.

DNS Settings

The **DNS Settings** panel displays and manages the DNS attributes for an array's administrative network.

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.

Process Steps

Creating a data vip

1. Select **Settings > Network**.
2. 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.
3. Click the Add interface button (**+**) belonging to the subnet to which the data vip will be attached. The **Create Network Interface** pop-up window appears.
4. In the **Name** field, type the name of the data vip.
5. In the **Address** field, type the IP address to be associated with the data vip.
6. In the **Services** field, leave the service type as **`data`**.
7. In the **Subnet** field, leave the subnet name as the default.
8. Click **Create**.

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.

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**.

Configuring the DNS domain

1. Select **Settings > Network**.
2. In the **DNS Settings** panel, click the **Edit** icon. All of the fields in the DNS Settings panel become editable boxes.
3. In the **Domain** field, type the domain suffix to be appended by the array when doing DNS lookups. For example, **`mydomain.com`**.
4. Click **Save**.

Configuring the DNS servers

The DNS servers manage the DNS domains that are configured for the array.

1. Select **Settings > Network**.
2. In the **DNS Settings** panel, click the **Edit** icon. All of the fields in the **DNS Settings** panel become editable boxes.
3. 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.
4. Click **Save**.

Deleting a data vip

1. Select **Settings > Network**.
2. In the Subnets list, click the **Delete** icon next to the data vip 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.

Users

The **Settings > Users** page provides configuration and management of directory services.

Directory Service

The **Directory Service** panel manages the integration of FlashBlades with an existing directory service. 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 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 support, 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 in place of the URI to represent the LDAP URL.

Important note! Anonymous bind for NFS is allowed. If configuring an anonymous bind for NFS, both the bind user and bind password are not 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 and write permissions to add computer objects in Active Directory. 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 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

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
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.

Role	Description
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, phonehome, 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.

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 or OpenLDAP. To export a file system over SMB, the FlashBlade array must be integrated with an Active Directory service.

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.

Process Steps

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.
   Important note! Anonymous bind for NFS is allowed. If configuring an anonymous bind for NFS, both the **Bind User** and **Bind Password** fields must be left blank.
   a. Select **Settings** > **Users**.
   b. In the **Directory Service** panel, select **Array Management**, **NFS**, or **SMB**.
   c. Click the **Edit** icon. 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, 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, 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 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`**.

g. In the **Bind Password** field, type the password for the bind user account.

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, "pureausers,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 **Save**.

Note: Directory service testing is applicable to the selected directory service (**Array Mangement**, **NFS**, or **SMB**).

After you configure the directory service settings, test and enable the directory service.

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**.

Part 3:
Using the Purity//FB CLI to Administer a FlashBlade Array

Chapter 8. Purity//FB CLI Overview

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 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; 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:

--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. The **--raw** output is used to sort and filter list results.

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
  ...
```

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 --api-token [ --cli | --csv | --nvp ] [--expose] [--notitle] [USER...]
pureadmin setattr [--password] [USER...]
pureadmin refresh {--clear} {USER...}
```

Arguments

USER
User login name. 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`
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. If not set, the API token will not expire.

`--password`
Changes the password for the pureuser administrative account.

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.

`--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 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.

Change 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.

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. An expiration can be set for an API token. 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      API Token                                Created                   Expires
pureuser  T-c3f9e02d-7ff7-44bd-b6a1-f7egg13419  2017-07-23 09:05:21 PDT  2017-07-25 09:05:21
```

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.

- **Name**: Lists the User names.

- **API Token**: Active tokens for API access.
- **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.

Examples

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.

See Also

pureds(1) [115]

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

purealert

purealert, purealert-create, purealert-delete, purealert-disable, purealert-enable, purealert-list, purealert-flag, purealert-unflag, purealert-test — manages alert history and the list of designated alert watchers to which Purity//FB sends email notifications when significant events occur in an array

Synopsis

```
purealert create { watcher } ADDRESS...
purealert delete { watcher } ADDRESS...
purealert disable { watcher } ADDRESS...
purealert enable { watcher } ADDRESS...
purealert flag [ID...]
purealert unflag [ID...]
purealert list [ --cli | --csv | --nvp ] [ --closed | --closed-only ] [--notitle] --watcher [ --unflagged | --unflagged-only ] [--filter FILTER] [--sort SORT] [--limit LIMIT] [--raw] [--page] [ID...]
purealert test { watcher } [ADDRESS...]
```

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.

`--unflagged`
Display alerts including unflagged alerts.

`--unflagged-only`
Display only unflagged alerts.

`--watcher`
Lists the email addresses of designated alert watchers and the alert status (enabled or disabled) of each watcher.

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`
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.

`--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.

`--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.

Add the **`--watcher`** option to display a list of email recipients that have been designated as alert watchers.

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 create watcher`** 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 test watcher`** 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 list --watcher`** command.

Sending Test Messages

The **`purealert test watcher`** 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 test watcher`** 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 test watcher`**.

Running the **`purealert test watcher`** command returns the following information for each 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 enable watcher`** and **`purealert disable watcher`** commands.

To view a list of alert watchers and their enabled/disabled statuses, run the **`purealert list --watcher`** command.

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 delete watcher`** 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.

Examples

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 list --watcher
```

Displays a list of alert watchers that have been created on the array and the enabled/disabled status of each watcher.

Example 3

```
purealert test watcher walterwatcher@example.com
purealert create watcher walterwatcher@example.com
purealert test watcher
```

Sends a test message to **`walterwatcher@example.com`**. After verifying receipt of the test message by **`walterwatcher@example.com`**, designates **`walterwatcher@example.com`** as a watcher to receive Purity//FB alert notifications. Sends a test message to all enabled watchers, including **`walterwatcher@example.com`**.

Example 4

```
purealert delete watcher wendywatcher@example.com alexalert@example.com
```

Removes **`wendywatcher@example.com`** and **`alexalert@example.com`** as alert watchers. Alert notifications will no longer be sent to **`wendywatcher@example.com`** and **`alexalert@example.com`**.

Example 5

```
purealert disable watcher aartiealert@example.com
```

Prevents Purity//FB from sending alert notifications to alert watcher **`aartiealert@example.com`**.

Example 6

```
purealert unflag 2
```

Unflag alert ID **2** by setting the parameter to **`False`**.

See Also

purearray(1) [97], puresupport(1) [184]

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

purearray

purearray, purearray list, purearray rename, purearray setattr — displays and manages the array attributes
purearray-monitor — monitors array I/O performance

Synopsis

```
purearray list [ --historical { 3h | 1d | 7d | 30d | 1y } ] [ --ntpserver | --relayhost | --senderdomain | --space ] [ --cli | --csv | --nvp ] [--notitle] [--raw]

purearray monitor --client [CLIENT]... --filter FILTER --limit LIMIT --sort SORT [ --historical { 5m | 3h | 1d | 7d | 30d | 1y } | [--interval SECONDS] [--repeat REPEAT-COUNT] [--size] ] [[ --protocol { http | nfs | smb | s3 } ] | [ --protocol-specific { http | s3 } ]] [--csv] [--notitle]

purearray rename NEW-NAME

purearray setattr { --ntpserver NTP-SERVER | --relayhost RELAY-HOST | --senderdomain SENDER-DOMAIN }
```

Arguments

NEW-NAME
Name by which the array is to be known after the command executes.

Options

`-h | --help`
Can be used with any command or subcommand to display a brief syntax description.

`--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.

`--filter` ***FILTER***
Specifies a custom filter to display client-specific performance information. This option is allowed for use with only the **`--client`** option.

`--historical` ***TIME***
Displays historical data for performance or storage consumption over the specified range of time.

Valid time range options for the **`monitor`** subcommand include: 5 minutes, 3 hours, 1 days, 7 days, 30 days, and 1 year.

Valid time range options for the **`list`** subcommand include: 3 hours, 1 days, 7 days, 30 days, and 1 year.

`--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.

`--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 (**`::`**).

`--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 only for **`http`** and **`s3`**.

`--relayhost [`***RELAY-HOST***`]`
Note: The **`purearray setattr --relayhost`** command has been deprecated and should not be used. To set SMTP attributes, use the **`puresmtp`** command.

`--repeat` ***REPEAT-COUNT***
Sets the number of times to display real-time performance data. If omitted, the repeat count defaults to 1.

`--senderdomain [`***SENDER-DOMAIN***`]`
Note: The **`purearray setattr --senderdomain`** command has been deprecated and should not be used. To set SMTP attributes, use the **`puresmtp`** command.

`--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.

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.

`--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. The **`--raw`** output is used to sort and filter list results.

Description

The **`purearray`** command displays and manages the FlashBlade array attributes.

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.

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 over any of the following ranges of time: 5 minutes, 3 hours, 1 day, 7 days, 30 days, or 1 year.

By default, the **`purearray monitor`** command displays real-time 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 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 information for the specified protocol.

With the `--protocol-specific` option, **`purearray monitor`** displays 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 for which HTTP access is enabled:

- **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.

With the **`--protocol-specific s3`** option, the performance information reflects the array's buckets and objects:

- **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.

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.

Examples

Example 1

```
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 2

```
purearray list --space
```

Displays array size and storage consumption details.

Example 3

```
purearray list --ntpserver
```

Displays the hostnames or IP addresses of the NTP servers that are currently configured on the array.

Example 4

```
purearray list --space --historical 3h
```

Displays the historical space usage of the array over the past three (3) hours.

Example 5

```
purearray monitor --historical 5m --csv
```

Displays a CSV output of historical performance data for the local array over the past five (5) minutes.

Example 6

```
purearray monitor --protocol smb
```

Displays real-time performance data for the SMB protocol only.

Example 7

```
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 8

```
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 9

```
purearray monitor --protocol nfs --client
```

Displays client-specific I/O performance statistics.

See Also

purealert(1) [91], purearray(1) [97], puredns(1) [113], purefs(1) [122], purehw(1) [141], purenetwork(1) [166], puresmtp(1) [177], puresubnet(1) [180], puresupport(1) [184]

Author

Pure Storage Inc. **`<documentfeedback@purestorage.com>`**

pureblade

pureblade, pureblade-evac, pureblade-list — evacuates data from a blade or displays blade attributes

Synopsis

```
pureblade list [ --csv | --nvp ] [--notitle] [--target] [--total] [--sort SORT]
[--limit LIMIT] [--page ] [--raw ] [BLADE...]
```

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.

Examples

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(1) [97], puresupport(1) [184]

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

purebucket

purebucket, purebucket-list — displays buckets and their contents.

Synopsis

```
purebucket list [ --cli | --csv | --nvp ] [--space] [--total] [--total-only] [--user-name USER_NAME] [--filter FILTER] [--sort SORT] [--raw] [--limit LIMIT] [--page] [BUCKET...]
```

Arguments

BUCKET
Bucket name or names (space-separated list). Filters output by the specified bucket names.

Options

`-h | --help`
Can be used with any command or subcommand to display a brief syntax description.

`--space`
Displays size and storage consumption details for each bucket.

`--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.

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.

`--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.

`--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 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.

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.

Examples

Example 1

```
purebucket list
```

Displays each bucket on the array and its Used space.

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
```

```
(total)   -       110.74T  1.7 to 1        63.37T
```

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

purecert

purecert, purecert-list, purecert-setattr — displays FlashBlade SSL certificate

Synopsis

```
purecert list [--certificate] [--intermediate-certificate] [ --cli | --csv | --
nvp ] [--notitle] CERTIFICATE...
```

```
purecert setattr --certificate [--intermediate-certificate] [--key] [--passphrase]
CERTIFICATE...
```

Arguments

CERTIFICATE
The SSL certificate name. Only the *global* certificate name is supported.

Options

`-h | --help`
Can be used with any command or subcommand to display a brief syntax description.

`--certificate`
Displays or imports the certificate.

`--intermediate-certificate`
Displays or imports the intermediate certificate chains.

`--key`
Private key of the certificate to import.

`--passphrase`
Passphrase used to decrypt the private key.

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.

`--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 **purecert** command allows you to perform tasks to manage your SSL 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 certificate or import a SSL certificate signed by a certificate authority.

Import 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.

Examples

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.

See Also

pureadmin(1) [87], purearray(1) [97]

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 ]
```

Options

None

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(1) [91], purearray(1) [97], puredns(1) [113], purefs(1) [122], purenetwork(1) [166], puresupport(1) [184]

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]
```

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. 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.

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.

Examples

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(1) [97], puresupport(1) [184]

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

pureds

pureds, pureds-disable, pureds-enable, pureds-list, 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] 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] [--uri URI-LIST] SERVICE-NAME
pureds test [--filter FILTER] [--sort SORT] [--raw] SERVICE-NAME
```

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 20 characters in length.

For OpenLDAP, enter the full DN of the user. For example, **CN=John,OU=Users,DC=example,DC=com**.

`--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.

`--uri` ***URI-LIST***
Comma-separated list of up to 30 Uniform Resource Identifiers (URIs) of the directory servers.

For file sharing over SMB, the base DN (**`--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. See puredns(1) [113] for more information.

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.

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.

`--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`
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.

The **`pureds setattr`** command configures the directory service settings, including the URIs, base DN, bind user, and bind password.

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.

Note: The **`pureds test`** command does not check SMB status. Additionally, if the OpenLDAP server is 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 SMB or NFS protocol support, 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 in place of the 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. 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 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.

Configuring the directory service for protocol support

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
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.

Role	Description
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.

Examples

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 --bind-user CN=John,OU=Users,DC=mycompany,DC=com smb
--bind-password smb
Enter bind password:
Retype bind password:
```

Shows 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 4

```
pureds test management
Name          Enabled        Component Name  Component Address  Destination                Te
management    True           fm1             10.64.240.191      ldap://ir-ldap-05.fifa.uk  co
                                                                                            bi
                                                                                            ro
                                                                                            ar
                                                                                            op
                                                                                            re
                                                                                            st
                             fm2             10.64.240.192      ldap://ir-ldap-05.fifa.uk  co
                                                                                            bi
                                                                                            ro
                                                                                            ar
                                                                                            op
                                                                                            re
                                                                                            st
```

Shows the output of testing the current management configuration.

Example 5

```
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 6

```
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**.

See Also

pureadmin(1) [87], puredns(1) [113]

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

purefs

purefs, purefs-add, purefs-create, purefs-destroy, purefs-disable, purefs-enable, purefs-eradicate, purefs-list, purefs-nfs, purefs-recover, purefs-remove, purefs-setattr, purefs-smb, purefs-snap — displays and manages the file systems

Synopsis

```
purefs add --protocol PROTOCOL FILE-SYSTEM...

purefs create [--size SIZE] FILE-SYSTEM...

purefs destroy FILE-SYSTEM...

purefs disable [--fast-remove-dir] [--snapshot-dir] [--hard-limit] FILE-SYSTEM...

purefs enable [--fast-remove-dir] [--snapshot-dir] [--hard-limit] [--ignore-usage]
FILE-SYSTEM...

purefs eradicate FILE-SYSTEM...

purefs list [ --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...]

purefs nfs {setattr} [--rules RULES] FILE-SYSTEM...

purefs recover FILE-SYSTEM...

purefs remove --protocol PROTOCOL FILE-SYSTEM...

purefs setattr [--ignore-usage] [--size SIZE] FILE-SYSTEM...

purefs smb {setattr} [ --acl-mode { native,shared } ] FILE-SYSTEM...

purefs snap [--suffix SUFFIX] FILE-SYSTEM...
```

Arguments

FILE-SYSTEM
File system name.

Options

`-h | --help`
Can be used with any command or subcommand to display a brief syntax description.

`--acl-mode` ***MODE***
Changes the SMB ACL mode of a protocol for one or more file system(s) to **`shared`** or **`native`** mode. The **`shared`** mode is set by default and shares UNIX-like ACL permissions between SMB and NFS protocols, while the **`native`** supports Windows and UNIX-like ACLs and restricted sharing of ACL permissions.

`--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.

`--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.

`--protocol` ***PROTOCOL***
Adds or removes file system protocol settings. FlashBlade file systems support the (case-sensitive) **`http`**, **`nfs`**, and **`smb`** protocols. Enter multiple protocols as comma-separated values. Adding **`http`** enables both **`http`** and **`https`**.

`--protocol-specific` ***PROTOCOL***
Displays only the file systems with the specified protocol enabled. FlashBlade file systems support the (case-sensitive) **`http`**, **`nfs`**, and **`smb`** protocols.

`--rules` ***RULES***
Defines the Network File System (NFS) export rules for the given file system, completely replacing the default set of rules. Options can be applied to an individual client or a range of clients. Valid export options are **`rw`**, **`ro`**, **`root_squash`**, **`no_root_squash`**, and **`fileid_32bit`**.

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

`--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.

`--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.

`--suffix` ***SUFFIX***
Assigns a suffix name to a file system snapshot. If not set, a snapshot suffix name will be assigned.

`--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.

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.

`--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.

`--sort` ***`SORT`***
Sorts the list output in ascending or descending order by the column specified.

For more information on how to use the CLI options to control display formatting and manage display results, refer to pureman(1) [156]

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**: Amount of space that the written data occupies on the array after reduction via data compression.

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(1) [156]

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 about data vips: 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(1) [166].

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 `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(1) [115].
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(1) [115].

  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(1) [115].
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.

SMB ACL Mode

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.

The SMB protocol supports two ACL modes, **`native`** and **`shared`**. Use the **`purefs smb setattr --acl-mode`** command to change ACL modes.

The **`shared`** ACL mode is set as the default and shares UNIX-like ACL permissions with the NFS protocol. In **`shared`** mode, both protocol ACL permissions are required to match. When one protocol creates files or modifies permissions, they must comply with the permission settings of the other protocol. For example, when an administrator attempts to change the SMB ACL permissions of a file to read and write, the attempt will prompt an action to perform a match on the file's NFS ACL permissions. If the NFS ACL permissions for that file is only allowed for read, then the attempt will fail.

The **`native`** ACL mode supports UNIX-like ACLs and Windows ACLs. In native mode, because SMB natively supports both ACLs while NFS only supports UNIX ACLs, ACLs sharing will be restricted between SMB and NFS. See the list of ACL sharing restrictions in **`native`** mode:

1. Permissions created or changed in SMB will not be applied to NFS permissions. For example, if User X access permissions are removed for File Y using SMB, then User X will still have access to File Y through NFS.
2. Permissions changed in NFS will not affect SMB permissions.
3. NFS permissions do not apply to directories and files created by SMB or permissions set by SMB.
4. If an SMB directory or file does not have an ACL applied to it, one will be applied to it from the NFS permissions.
5. If an NFS directory or file does not have an ACL applied to it, SMB ACLs will not be applied to that NFS directory/file.
6. UNIX and Windows ACLs grant full permissions by default.

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 the 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 an **`INFO`** alert when the amount of data written to the file system exceeds 100% of its provisioned size.

The file system size can be blank, as indicated by the dash (-) symbol in the **`purefs list`** output, or set to any value between **`0`** and **`8191`** petabytes. If the size is not set (blank) or if it is 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, administrators must decrease the percentage of storage space used by either deleting files or increasing the size of the file system.

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

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 host...'**
- Represent all clients with an asterisk: *****

host represents one of the following client specifications:

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

Valid export options are **ro**, **rw**, **root_squash**, **no_root_squash**, and **fileid_32bit**.

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 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)**

- 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** output with various rules set for each file system created:

```
purefs list
Name     Protocols   Rules
File1    nfs         *
File2    nfs         *()
File3    nfs         -
File4    nfs         *(rw,no_root_squash)
File5    nfs         *(rw,root_squash,fileid_32bit)
File6    nfs         10.20.225.2(rw)
File7    nfs         2620::/16(no_root_squash)
```

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

- ***** - 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.
- ***(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.
- **10.20.225.2(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.

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 Hypertext Transfer Protocol (HTTP), the Network File System (NFS), and Server Message Block (SMB) protocols.

The **`purefs add --protocol`** command enables protocols on a file system.

A file system must have at least one protocol (**`nfs`**, **`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(1) [115].
- 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.

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 systems 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 issuing 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.

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.

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.

Examples

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 nfs,smb MyFiles
```

Enables the NFS and SMB protocols on file system **MyFiles**.

Example 9

```
purefs smb setattr --acl-mode native MyFiles
```

Changes the SMB ACL mode to **native** on file system **MyFiles**.

Example 10

```
purefs disable --fast-remove-dir MyFiles
```

Disables the fast remove directory for file system **MyFiles**.

Example 11

```
purefs destroy MyFiles
```

Destroys file system **MyFiles** and places it in the eradication pending state until the time expires or is recovered.

Example 12

```
purefs recover MyFiles
```

Recovers file system **MyFiles** if performed before the eradication pending period expires.

Example 13

```
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 14

```
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  nfs
```

Enables a hard limit on file system fs1's provisioned size of 1M.

Example 15

```
purefs disable --hard-limit fs1
Name  Size  Used    Hard Limit  Created                  Protocols
fs1   1M    15.51M  False       2018-04-20 21:16:18 PDT  nfs
```

Disables the hard limit on file system fs1.

Example 16

```
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.

See Also

purealert(1) [91], purearray(1) [97], purenetwork(1) [166], puresubnet(1) [180], puresupport(1) [184]

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]
```

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.

`--identity {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), or **pwr** (power supply). 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. The **`--raw`** output is used to sort and filter list results.

`--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.
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 **FMx**, where **x** can be **1** or **2**.

The names of the other components within the chassis have the form **FMx.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).

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
FMx	External Fabric Module	External fabric module, modules can be numbered FM1 and FM2
FMx.ETHn	External Fabric Module	Ethernet port.
CHx.FANm	External Fabric Module	External fabric module cooling fan.
CHx.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**: Speed (in Gb/s) at which the component is operating.
- **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 FM1.ETH1
```

The QSFP connector FM1.ETH1 is set to 4 ports with 10Gb/s lane speed.

See Also

purealert(1) [91], purearray(1) [97], puresupport(1) [184]

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
```

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. 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. 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

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.

Examples

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(1) [141], puresubnet(1) [180], purenetwork(1) [166]

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

purelicense

purelicense — End User Agreement (EUA) between Pure Storage Inc. and end users of the company's products

END USER AGREEMENT (v11Apr17)

Available online at: http://www.purestorage.com/legal/productenduserinfo.html

IMPORTANT: PLEASE READ THIS END USER AGREEMENT ("**AGREEMENT**") BEFORE INSTALLING OR USING THE PURE STORAGE PRODUCT, INCLUDING THE HARDWARE AND SOFTWARE COMPONENTS THEREOF ("**PRODUCT**"). THIS AGREEMENT APPLIES TO THE PRODUCT THAT YOU OR THE ENTITY THAT YOU REPRESENT ("**END USER**") OBTAINED FROM PURE STORAGE, INC. ("**PURE**") OR FROM ANY THIRD PARTY AUTHORIZED BY PURE TO RESELL THE PRODUCTS AND SERVICES. BY INSTALLING OR USING THE PRODUCT OR SERVICES, YOU REPRESENT AND WARRANT THAT YOU HAVE THE AUTHORITY TO BIND END USER AND AGREE THAT END USER IS BOUND BY THIS AGREEMENT WITH PURE, UNLESS A SEPARATE WRITTEN AGREEMENT IS IN EFFECT THAT SPECIFICALLY GOVERNS THE SUBJECT MATTER HEREOF. IF END USER DOES NOT HAVE A SEPARATE WRITTEN AGREEMENT WITH PURE AND DOES NOT AGREE TO THIS AGREEMENT, THE END USER MAY, WITHIN 30 DAYS OF INITIAL RECEIPT OF THE PRODUCT, RETURN THE UNUSED PRODUCT FOR A REFUND.

1. SOFTWARE LICENSE.

1.1 Software Licnese.

Subject to End User’s compliance with the terms and conditions of this Agreement, Pure grants to End User, and any third party that End User authorizes to perform services involving the Product solely for End User’s benefit, a nontransferable, nonexclusive, perpetual license to use and execute the Pure software provided with, or incorporated in, the Product (the “**Software**”), in executable object code format only, and solely to the extent necessary to operate the Product in accordance with the applicable Product documentation and Product SKU description. Pure may make Software updates and new releases available for installation by the End User and such updates will be subject to the terms of this Agreement.

1.2 Termination of Software License.

The license in Section 1.1 and End User’s rights to use the Software will terminate immediately in the event that End User returns the Product to Pure as provided herein or in the event that End User materially breaches any provision of this Agreement. Upon termination, End User shall promptly discontinue all use of the Software and Sections 2, 3.2, 5.4, 6, 7, and 9 will survive.

2. PRODUCT RESTRICTIONS AND TITLE.

2.1 Restrictions.

Restrictions. End User will not directly or indirectly (i) reproduce, modify, distribute, assign, disclose or make available any portion of the Software (or any related documentation) to any third party (except as otherwise authorized herein); (ii) rent, lease or sublicense the Product, unless otherwise authorized by Pure in writing; (iii) reverse engineer, decompile, or disassemble any portion of the Product, or

otherwise attempt to decrypt, extract or derive source code for, or any algorithms or data structures embodied within, any portion of the Software (except to the extent the foregoing restriction is expressly prohibited by applicable law); (iv) use the Product to develop a similar product or service; (v) transfer, copy or use the Software to or on any other product or device; or (vi) publish or disclose to any third party any technical features, performance or benchmark tests, feedback or analyses relating to the Product, except for internal use by the End User or as may be authorized by Pure in writing. End User will remain fully and primarily responsible to Pure for compliance with this Agreement if End User permits any third party to access the Product. Any future release, update, or other addition to functionality of the Product made available by Pure to End User shall be subject to the terms and conditions of this Agreement, unless Pure expressly states otherwise. End User shall preserve and shall not remove, obscure or alter any copyright labels required by law or other proprietary notices in the Products or related documentation.

2.2 Title.

Pure and its suppliers shall exclusively retain all right, title and interest, in all intellectual property rights, including patent, trademark, trade name and copyright, whether registered or not registered, in and to the Software and related documentation. Pure and its suppliers reserve all rights not expressly granted herein, and no other license or other implied rights of any kind are granted or conveyed. Any non-public information relating to, or derived from, the Products and related services, including technical features, benchmark results, or performance results, is the confidential information of Pure, and End User shall not use or disclose such information except as expressly authorized in this Agreement. In the event that items of software code provided with the Product are subject to “open source” or “free software” licenses, nothing herein limits End User’s rights under, or grants rights that supersede, the applicable license therefore.

3. PRE-RELEASE PRODUCTS AND FEEDBACK.

3.1 Pre-Release Products.

Pure may make available to End User a beta or pre-release version of the hardware and/or Software (“**Pre-Release Products**”). End User acknowledges that the Pre-Release Products (i) are not at the level of performance or compatibility of final, generally available products; (ii) may not operate correctly; (iii) may be modified prior to being made generally available; (iv) may not be made available for general release; and (v) may not be used in a production environment. End User agrees to notify Pure of any bugs or problems in the Pre-Release Products.

3.2 Feedback.

End User may provide feedback to Pure regarding the use, operation, performance, and functionality of the Products and Pre-Release Products, including identifying potential errors and improvements (collectively, “**Feedback**”). End User grants to Pure a perpetual, irrevocable, worldwide, sublicenseable, fully paid-up and royalty-free right to modify and use the Feedback in any manner, provided that Feedback is anonymized and does not identify End User.

4. EVERGREEN SUBSCRIPTION.

At its option, End User may purchase an innovation and support subscription for a purchased Product (“**Evergreen Subscription**”), which provides End User with additional software and hardware benefits.

Under an applicable Evergreen Subscription, Pure will provide the generally available Product maintenance and technical support in accordance with the Pure Storage Customer Support Guide, during the term for which End User has purchased such Evergreen Subscription. Depending on the Product or Evergreen Subscription purchased, certain benefits of the Evergreen Storage Program Description may also apply. Pure may designate support partners and authorized resellers to deliver the Evergreen Subscription in accordance with the terms of this Agreement.

5. WARRANTY AND DISCLAIMER.

5.1. Limited Warranty.

Limited Warranty. Subject to Section 5.2, Pure warrants that the hardware in the Product will perform in substantial accordance with the corresponding Product documentation for three years from the date of shipment by Pure and that the Software will perform in substantial accordance with the corresponding Product documentation for 90 days from the date of shipment by Pure. The Evergreen Subscription benefits described in Section 4 extend beyond the limited warranty for the term purchased by End User.

5.2. Limited Warranty Process.

Limited Warranty Process. End User may contact Pure via email at **<support@purestorage.com>** or phone at +1 (866) 244-7121 for warranty service. If a return is required, End User must obtain a return material authorization number from Pure and return the Product in secure packaging, freight prepaid, as instructed by Pure. Under the hardware warranty, Pure, at its option, either (i) will repair or replace any defective Product with a new or refurbished Product or components of equal or greater functionality as the returned Product, or (ii) will refund the purchase price paid to Pure for such Product, reduced on a straight-line basis over a three-year life. Replacement Products or components will continue to be warranted for the remainder of the applicable warranty term. Repair, replacement or refund is the sole and exclusive remedy for breach of this warranty. Under the Software warranty, Pure will provide End User access to bug fixes and emergency patches. This warranty is provided to the original End User only and is not transferable. This warranty does not cover defects or damages resulting from: (a) use of Products other than in a normal and customary manner in accordance with Pure's documentation; (b) physical or electronic abuse or misuse, accident, or neglect; or (c) alterations or repairs made to Products that are not authorized by Pure in writing. Pure will use reasonable efforts to destroy (but have no liability for any loss or inadvertent disclosure of) data stored or remaining on a Product returned to Pure. Under this Agreement, all returned Products and components become the property of Pure.

5.3. No Warranty or Support for Pre-Release Products.

No Warranty or Support for Pre-Release Products. Pure provides Pre-Release Products for evaluation only on an "AS IS" basis (without warranty or liability of any kind), for use at End User's own risk. Evergreen Subscriptions are not available for Pre-Release Products. Pure will, at its discretion, use reasonable efforts to resolve problems identified in Pre-Release Products.

5.4. Disclaimer.

EXCEPT FOR THE EXPRESS WARRANTY IN SECTION 5.1, PURE DISCLAIMS ALL OTHER WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, TO THE EXTENT WARRANTIES MAY BE DISCLAIMED UNDER APPLICABLE LAW, INCLUDING ANY IMPLIED WARRANTIES OF

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. PURE DOES NOT WARRANT AGAINST LOSS OR INACCURACY OF DATA OR THAT THE OPERATION OF THE PRODUCT WILL BE UNINTERRUPTED OR ERROR FREE. EXCEPT AS EXPRESSLY STATED IN SECTION 5.1, PURE PROVIDES THE PRODUCTS (INCLUDING ANY SOFTWARE) ON AN “AS IS” BASIS. THE PRODUCT IS NOT DESIGNED OR INTENDED FOR USE WHERE FAILURE OF THE PRODUCT COULD REASONABLY BE EXPECTED TO RESULT IN PERSONAL INJURY, LOSS OF LIFE OR PROPERTY DAMAGE.

6. INDEMNIFICATION.

Pure will indemnify and defend End User, at Pure’s expense, against any action brought by a third party against End User to the extent that the action is based upon a claim that the Product directly infringes any copyrights or U.S. patents or misappropriates any trade secrets, and Pure will pay those costs and damages finally awarded by a court of competent jurisdiction against End User in any such action that are specifically attributable to such claim or those costs and damages agreed to by Pure in a monetary settlement of such action. If End User’s use of the Product is, or in Pure’s opinion is likely to become, enjoined as a result of an infringement claim, Pure will, at its option and expense, either (i) procure the right to continue using the Product; (ii) replace or modify the Product so that it becomes non-infringing and remains functionally equivalent; or (iii) if, despite its commercially reasonable efforts, Pure is unable to do either (i) or (ii), Pure will accept return of the Product, terminate the rights herein, and pay to End User a prorated refund of the money paid to Pure for the purchase of such Product, reduced on a straight-line basis over a three-year life. Notwithstanding the foregoing, Pure will have no obligation with respect to any infringement claim based upon (a) any use of the Product that is not in accordance with this Agreement or the corresponding Product documentation; (b) any use of the Product in combination with other products, equipment, software, or data not supplied by Pure if such infringement would not have arisen but for such combination; (c) the use of any release of the Software other than the current and immediately preceding version; or (d) any modification of the Product by any person other than Pure if such infringement would not have occurred but for such modification. This Section 6 states Pure’s entire liability, and End User’s sole and exclusive remedy, for infringement claims and actions. The foregoing obligations are subject to End User notifying Pure promptly in writing of such action, giving Pure sole control of the defense thereof and any related settlement negotiations, and cooperating and, at Pure’s reasonable request and expense (including reasonable attorneys’ fees), assisting in such defense.

7. LIMITATION OF LIABILITY.

OTHER THAN PURE’S INDEMNIFICATION OBLIGATIONS UNDER SECTION 6, TO THE MAXIMUM EXTENT PERMITTED BY LAW, PURE’S AGGREGATE LIABILITY UNDER OR RELATING TO THIS AGREEMENT SHALL NOT EXCEED THE AMOUNT PAID BY END USER FOR THE PRODUCTS OR SERVICES THAT GAVE RISE TO SUCH CLAIM. IN NO EVENT WILL PURE BE LIABLE FOR ANY SPECIAL, CONSEQUENTIAL, EXEMPLARY, INCIDENTAL, OR INDIRECT DAMAGES, (INCLUDING LOST PROFITS, ANTICIPATED SAVINGS OR REVENUES), LOST OR CORRUPTED DATA, OR INTERRUPTION OF BUSINESS ARISING IN CONNECTION WITH THE USE OF THE PRODUCT OR SERVICES OR IN CONNECTION WITH ANY OTHER CLAIM ARISING FROM THIS AGREEMENT, EVEN IF PURE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

8. PRODUCT DIAGNOSTIC REPORTING.

End User acknowledges that the Product will store certain diagnostic information about the routine operations of the Product (including performance, capacity usage, data reduction ratios, configuration data, and hardware faults) and will periodically transmit this diagnostic information to Pure and authorized End User partners. No actual user data of End User is accessed, transmitted or provided to Pure or any third party as part of this process, and no interruption of service is required to gather such detailed diagnostics. End User will control Pure or any third party's physical access to the Product. End User agrees that the collection and transmission of such Product information is necessary to facilitate the Evergreen Subscription.

9. GENERAL PROVISIONS.

9.1 Governing Law and Venue.

This Agreement will be governed and interpreted by and under the laws of the State of California, without giving effect to any conflicts of laws principles. The parties expressly consent to the personal jurisdiction and venue in the state and federal courts in Santa Clara County, California for any lawsuit filed there arising from or related to this Agreement. The U.N. Convention on Contracts for the International Sale of Goods shall not apply to this Agreement.

9.2 Notices.

Except as specifically stated, all notices or other communications required or permitted under this Agreement shall be in writing and shall be delivered by personal delivery, certified overnight delivery such as Federal Express, or registered mail (return receipt requested) and shall be deemed given upon personal delivery or upon confirmation of receipt.

9.3 Compliance with Laws.

The parties agree to comply with all laws applicable to the distribution and use of the Product and performance of its obligations under this Agreement.

9.4 Severability; Waiver.

If any provision of this Agreement is, for any reason, held to be invalid or unenforceable, the other provisions of this Agreement will remain enforceable and the invalid or unenforceable provision will be deemed modified so that it is valid and enforceable to the maximum extent permitted by law. Any waiver or failure to enforce any provision of this Agreement on one occasion will not be deemed a waiver of any other provision or of such provision on any other occasion.

9.5 Export.

The Product, Software, and related technology are subject to U.S. export control laws and may be subject to export or import regulations in other countries. End User agrees not to export, re-export, or transfer, directly or indirectly, any technical data acquired from Pure, or any products incorporating such data, in violation of applicable export laws or regulations.

9.6 No Assignment.

This Agreement, and End User's rights and obligations herein, may not be assigned by End User without Pure's prior written consent, which consent will not be unreasonably withheld, and any attempted assignment in violation of the foregoing will be null and void.

9.7 U.S. Government End Users.

The Product, Software, and related documentation are "commercial off the shelf items" as defined in FAR 2.101 and their use is subject to the policies set forth in FAR 12.211, FAR 12.212 and FAR 227.7202, as applicable.

9.8 Force Majeure.

Neither party shall be liable hereunder by reason of any failure or delay in the performance of its obligations under this Agreement on account of strikes, shortages, riots, insurrection, fires, flood, storm, explosions, acts of God, war, terrorism, governmental action, labor conditions, earthquakes, volcanic eruption, material shortages or any other cause that is beyond the reasonable control of the party.

9.9 Entire Agreement; Modification.

This Agreement, including any terms referenced herein, is the entire agreement between the End User and Pure with respect to the subject matter hereof. Any varying or additional terms relating to the subject matter hereof in any purchase order, discussion, or other written document will be of no effect. This Agreement, including any rights hereunder, may be extended or amended by the parties in writing.

pureman

pureman — displays man pages for Purity//FB commands

Synopsis

```
pureman { pureadmin | purealert | purearray | pureblade | purebucket | purecert
| pureconfig | puredns | pureds | purefs | purehw | purelag | purelicense |
purenetwork | pureobjaccount | pureobjuser | puresmtp | puresubnet | puresupport }
```

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 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. The **--raw** output is used to sort and filter list results.

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
  ...
```

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 create {vip} --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
```

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`**.

`--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. 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. 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

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(1) [180].
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(1) [122].

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.

Examples

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(1) [97], purefs(1) [122], puresubnet(1) [180], puresupport(1) [184]

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

pureobjaccount

pureobjaccount, pureobjaccount-create, pureobjaccount-delete, pureobjaccount-list — displays and manages the object account attributes

Synopsis

```
pureobjaccount create ACCOUNT...
pureobjaccount delete ACCOUNT...
pureobjaccount list [--space] [ --cli | --csv | --nvp ] [--notitle] [--total]
[--total-only] [--filter FILTER] [--sort SORT] [--raw] [--limit LIMIT] [--page]
[ACCOUNT...]
```

Arguments

ACCOUNT
The name of an object store account.

Options

`-h | --help`
Can be used with any command or subcommand to display a brief syntax description.

`--space`
Displays storage consumption details for each object store account.

`--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.

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.

`--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`
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.

`--sort`
Sorts the list output in ascending or descending order by the column specified.

Description

The **`pureobjaccount`** command manages the object store accounts. You can create multiple accounts on the array and delete them as needed. 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.

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.

- **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.

Creating an Account

The **`pureobjaccount create`** command creates a new object store account. Accounts must be created before an object store user can be created.

Deleting an Account

The **`pureobjaccount delete`** command deletes an object store account.

Examples

Example 1

```
pureobjaccount create pureaccount1 pureaccount2
```

Creates two new object store accounts named **pureaccount1** and **pureaccount2**.

Example 2

```
pureobjaccount delete pureaccount1
```

Deletes the object store account named **pureaccount1**.

See Also

purearray(1) [97], purebucket(1) [106] purefs(1) [122]

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

pureobjuser

pureobjuser, pureobjuser-access-key, pureobjuser-access-key-create, pureobjuser-access-key-setattr, pureobjuser-access-key-list, pureobjuser-create, pureobjuser-delete, pureobjuser-list — displays and manages the object user attributes

Synopsis

pureobjuser create ***USER...***

pureobjuser delete ***USER...***

pureobjuser list [--cli | --csv | --nvp] [--notitle] [--filter ***FILTER***] [--sort ***SORT***] [--raw] [--limit ***LIMIT***] [--page] [***USER...***]

pureobjuser access-key create --user ***USER***

pureobjuser access-key {delete} {enable} {disable} ***ACCESS-KEY-ID...***

pureobjuser access-key list [--cli | --csv | --nvp] [--notitle] [--filter ***FILTER***] [--sort ***SORT***] [--raw] [--limit ***LIMIT***] [--page] [***ACCESS-KEY-ID...***]

Arguments

USER
: The user name of an object store user, specified in the format of **accountname/user name**. For example, **account1/jsmith**.

ACCESS-KEY-ID
: The access key ID of the user.

Options

`-h | --help`
: Can be used with any command or subcommand to display a brief syntax description.

`--user` ***USER...***
: Specify the user name for the Object Store access key.

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.

`--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.

`--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.

`--sort`
Sorts the list output in ascending or descending order by the column specified.

Description

The **`pureobjuser`** command is used to create and delete users for object store accounts. Additionally, this command is used to manage user access keys by creating, deleting, enabling, or disabling keys. Each user can have up to two sets of access keys. Each key consists of an Access Key ID and a Secret Access Key.

Viewing Users and Access Keys

The **`pureobjuser 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 **`pureobjuser 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 key is encrypted and only the user can reveal the key.
- **User**: The name of the user.
- **Created**: The time the access key was created for the user.

Creating a User

The **pureobjuser 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 **pureuser** for an object store account named **pureaccount**, add **pureaccount** as a prefix followed by a forward slash. For example, **pureaccount/pureuser**.

```
pureobjuser create pureaccount/pureuser
```

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 **pureobjuser access-key create –user** command followed by the user name, including the account prefix. For example:

```
pureobjuser access-key create –user pureaccount/pureuser
```

An access key ID and secret access key is created for the user named **pureaccount/pureuser**.

Deleting, Disabling, Enabling an Access Key

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:

- **pureobjuser access-key disable** followed by the access key ID to disable a user's access key.
- **pureobjuser access-key enable** followed by the access key ID to re-enable a user's access key.
- **pureobjuser access-key delete** followed by the access key ID to delete a user's access key.

Examples

Example 1

```
pureobjuser create pureaccount/pureuser
```

Creates a new user named **pureaccount/pureuser** that is assigned to account **pureaccount**.

Example 2

```
pureobjuser access-key create --user pureaccount/pureuser
```

Creates an access-key for the User **pureaccount/pureuser**.

Example 3

```
pureobjuser access-key disable PSFBIAZFABAAMFP
```

Disables access key **PSFBIAZFABAAMFP**.

Example 4

```
pureobjuser delete pureaccount/pureuser
```

Deletes user **pureaccount/pureuser**.

See Also

purearray(1) [97], purebucket(1) [106] purefs(1) [122]

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

puresmtp

puresmtp, puresmtp-list, puresmtp-setattr — displays and sets SMTP attributes.

Synopsis

```
puresmtp list [ --cli | --csv | --nvp ] [--notitle]
puresmtp setattr [--relayhost RELAYHOST] [--senderdomain SENDERDOMAIN]
```

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. 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.

`--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`**.

Examples

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 the sender domain **mydomain.com** is set as the forwarding point for alert email notifications generated by the array.

See Also

pureadmin(1) [87], purearray(1) [97]

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***

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. 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. 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

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(1) [166].

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(1) [146].

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.

Examples

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(1) [97], purefs(1) [122], purenetwork(1) [166], puresupport(1) [184], purelag(1) [146]

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]
```

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. The **`--raw`** output is used to sort and filter list results.

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. A disconnected status of true confirms that the sessions has successfully closed.

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>`**.

Examples

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(1) [91], purearray(1) [97],

Author

Pure Storage Inc. **<documentfeedback@purestorage.com>**

Part 4:
Appendices

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, 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**.

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. Documentation

Related Documentation

The FlashBlade REpresentational State Transfer (REST) APIs use HTTP requests to interact with select resources on the FlashBlade. These are data path APIs.

The following guides provide overviews of these APIs and list all available resources.

- **FlashBlade HTTP Access to File Systems REST API Reference Guide.**
  The FlashBlade HTTP Access to File Systems REST API uses HTTP requests to interact with the NFS file systems resource on the FlashBlade.
- **FlashBlade Object Store REST API Reference Guide.**
  The FlashBlade Object Store REST API uses HTTP requests to interact with object stores on the FlashBlade.

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

Pure Storage is always eager to hear from you.

Documentation Feedback

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 Storage Support at **<support@purestorage.com>**.

General Feedback

For all other questions and comments about Pure Storage, including products, sales, service, and just about anything that interests you about data storage, email **<info@purestorage.com>**.

© 2018 Pure Storage, Inc. All rights reserved. Pure Storage, Pure1, and the Pure Storage logo are trademarks or registered trademarks of Pure Storage, Inc. in the U.S. and other countries. Other company, product, or service names may be trademarks or service marks of their respective owners.

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

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

Pure Storage, Inc.
Twitter: @purestorage

650 Castro Street, Suite 260
Mountain View, CA 94041
800-379-7873

Sales: sales@purestorage.com
Support: support@purestorage.com
Media: pr@purestorage.com
General: info@purestorage.com