



FlashBlade User Guide (2.1.0)

Thursday, August 10, 2017 15:15

Contents

Part 1: Purity//FB Overview

Chapter 1: Overview	7
Purity//FB Conventions	7
IP Addresses	8
Troubleshooting	9

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	20
Recent Alerts	21
Performance	21
Chapter 4: Storage	25
File Systems	25
Object Store	36
Chapter 5: Health	39
Hardware	39
Alerts	41
Chapter 6: Analysis	45
Performance	46
Capacity	48
Chapter 7: Settings	49
System	49
Network	58

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

Chapter 8: Purity//FB CLI Overview	65
CLI Command Syntax	65
CLI Output	66

CLI Login	72
CLI Command Help	72

Part 4: Appendices

Appendix A: Quick Reference Guide	149
Exporting a File System	149
Adding an Alert Watcher	159
Appendix B: Documentation	161
Related Documentation	161
Contact Us	161

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: through the browser-based graphical user interface (Purity//FB GUI), and through the command-driven 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.

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 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>**.

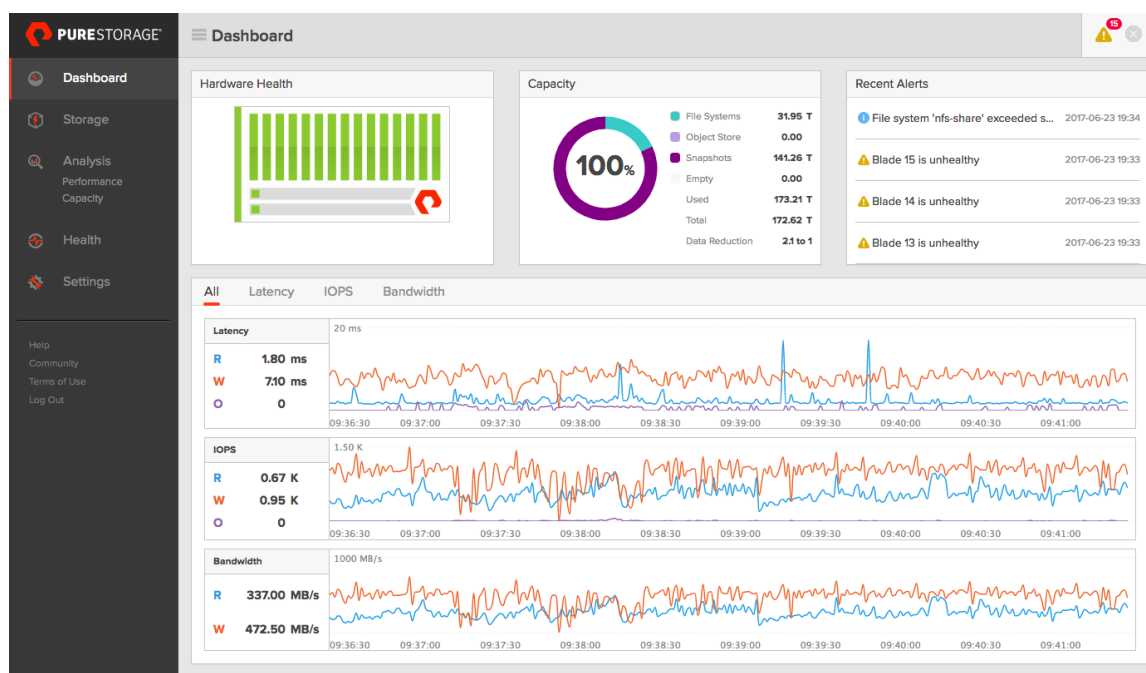
Part 2:

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.

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.

Health

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

Analysis

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

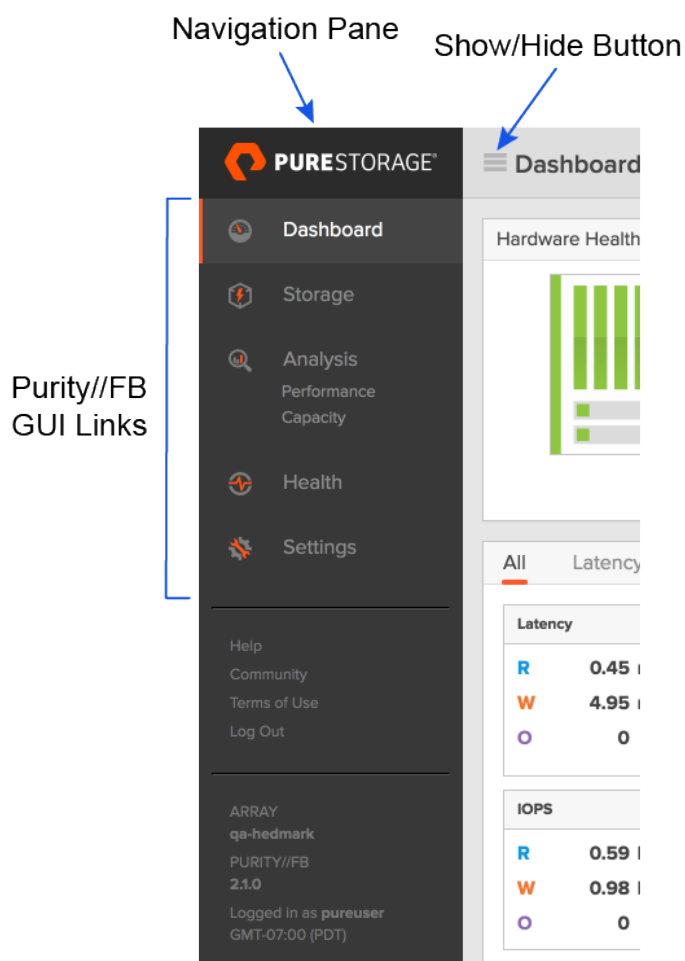
Settings

Displays array-wide system and network settings. Manage array-wide components, including network interfaces, system time, connectivity and connection configurations, 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.

Community

Launches the Pure1 community portal at <http://support.purestorage.com>.

Terms of Use

Launches the Pure Storage End User Agreement page.

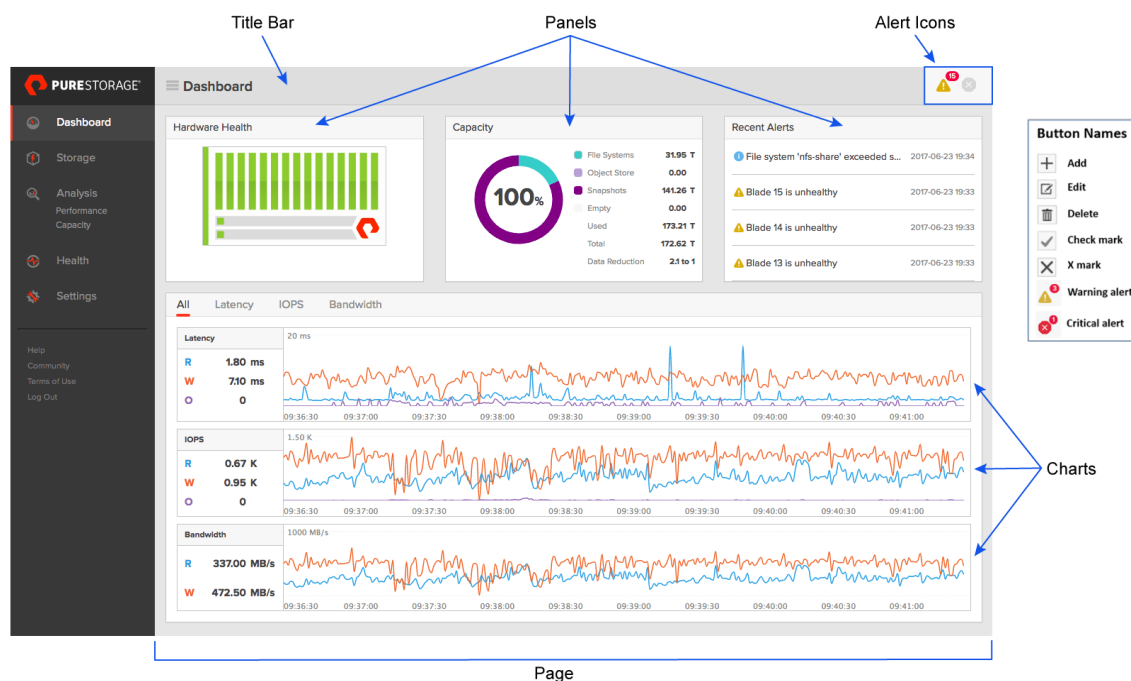
Log Out

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

The bottom of the navigation pane displays FlashBlade array 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 recent **Warning** and **Critical** alerts, respectively. A recent alert represents one that Purity//FB saw within the past 24 hours and still considers an open issue that requires attention. To analyze the recent 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.

Name	Size	Used	% Used	Data Reduction	Physical	Protocols	
1G	1.00 G	0.00	0%	-	0.00		
FS_01	15.00 T	0.00	0%	-	0.00	NFS	
FS_02	20.00 T	0.00	0%	-	0.00	NFS	
FS_03	20.00 T	0.00	0%	-	0.00	SMB	
FS_04	25.00 P	0.00	0%	-	0.00	HTTP	
FS_05	10.00 G	0.00	0%	-	0.00	NFS, SMB, HTTP	

Pagination divides a large list output into discrete pages. Pagination is enabled by default and is only in effect if the number of lines in the list output exceeds 25 rows. 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.

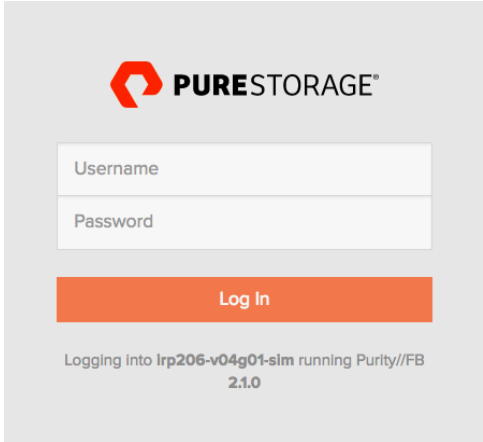
FlashBlade is installed with one administrative account with the username **pureuser**. The initial password for the account is **pureuser**. The password can be changed any time through the **pureadmin** CLI command.

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

The image shows the Purity//FB GUI login screen. At the top is the Pure Storage logo. Below it are two input fields: 'Username' and 'Password'. Underneath these fields is an orange 'Log In' button. At the bottom of the screen, there is a status message: 'Logging into lrp206-v04g01-slm running Purity//FB 2.1.0'.

Logging in to the Purity//FB GUI

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

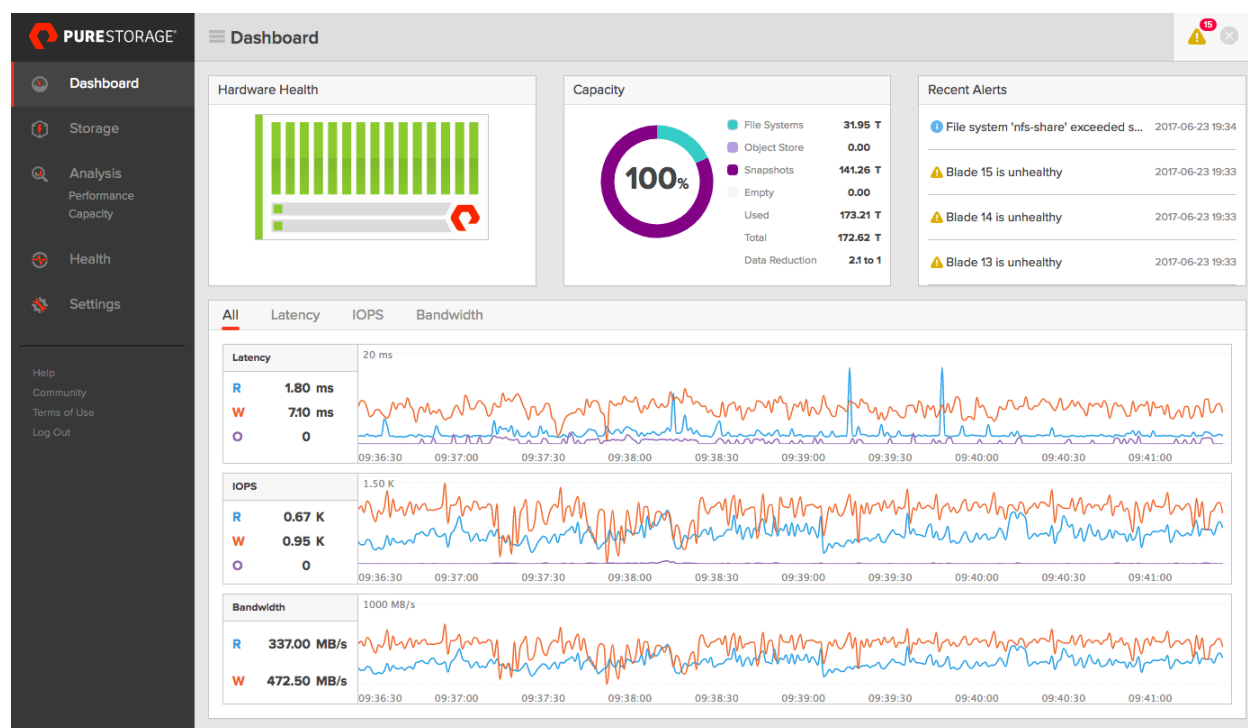
To log in to the Purity//FB GUI:

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.

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/Total**.

The capacity data is broken down into the following components:

File Systems

Amount of space that the written data occupies on the array 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.

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 within the past 24 hours and considers open issues that require attention. The list contains recent alerts of all severity levels.

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

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 3.2: Dashboard - Performance Charts



The performance metrics are displayed along a scrolling graph; incoming data appear along the right side of each graph every few minutes as older numbers drop off the left side. 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 all file systems. The data is counted in its expanded form rather than the reduced form stored in the array to truly reflect what is transferred over the storage network. Metadata bandwidth is not included in these numbers.

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

By default, the 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:

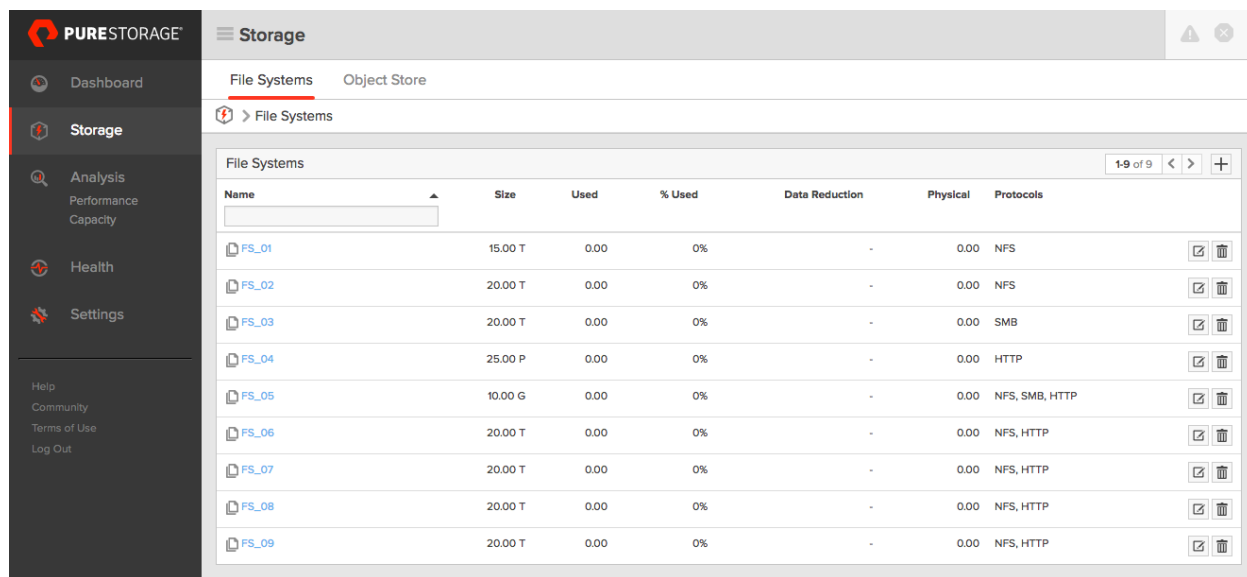
- At its shortest range (5m), the performance charts are updated once every second. As the range increases, the update frequency (and resolution) decreases.
- The performance charts display up to 1 year of historical data.
- The performance charts can be filtered to display metrics by protocol.

Chapter 4. Storage

The Storage page displays and manages the array's file systems and users' access keys for the array's buckets.

Exporting a file system 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



PURE STORAGE								Storage	
File Systems		Object Store							
> File Systems									
File Systems									
Name	Size	Used	% Used	Data Reduction	Physical	Protocols			
FS_01	15.00 T	0.00	0%	-	0.00	NFS			
FS_02	20.00 T	0.00	0%	-	0.00	NFS			
FS_03	20.00 T	0.00	0%	-	0.00	SMB			
FS_04	25.00 P	0.00	0%	-	0.00	HTTP			
FS_05	10.00 G	0.00	0%	-	0.00	NFS, SMB, HTTP			
FS_06	20.00 T	0.00	0%	-	0.00	NFS, HTTP			
FS_07	20.00 T	0.00	0%	-	0.00	NFS, HTTP			
FS_08	20.00 T	0.00	0%	-	0.00	NFS, HTTP			
FS_09	20.00 T	0.00	0%	-	0.00	NFS, HTTP			

File Systems

The File Systems list displays the file systems that have been created on the FlashBlade array.

Each file system in the list 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.
- **% Used:** Percentage of provisioned size occupied by file 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.

- **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.

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 **Settings > Network** page. For more information about data vips, refer to Chapter 7, *Settings* [49].

For more information about using directory services with file system protocols, refer to Chapter 7, *Settings* [49].

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 > System** 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-style permissions. 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. Extended Windows ACLs are not supported.
- 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* [49].

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* [49].
 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 filesystem 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.

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

File system snapshots are immutable, point-in-time images of the contents of one or more file systems. 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 a special directory named **.snapshot** 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.

The **.snapshot** directory can be removed from the root directory by disabling the feature. Once disabled, historical snapshots can be viewed but will not be accessible until the directory is enabled. Disabling the the directory will not delete any snapshots, they must be manually deleted.

The **.snapshot** special directory 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.

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 deleted, 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 > System** 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 **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.

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.

d. Click **Create**.

- **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.
- **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 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:

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.
3. Click **Save**.

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 deleted using the Purity//FB GUI (Storage > File Systems) or the CLI using the **purefs** command.

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 assigned if this field is left empty.
5. Click **Create**.

Deleting a file system snapshot

Once a file system snapshot has been deleted, it cannot be recovered.

To delete a snapshot:

1. Select **Storage > File Systems**.
2. In the Snapshots list, click the trash icon in the row of the snapshot you want to delete. The Delete Snapshot pop-up window appears.
3. Click **Delete**.

Enabling and disabling the snapshot directory

A snapshot directory is a special directory that can be enabled or disabled to store snapshots.

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.

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 deleted and unrecoverable.

Deleting a file system

Deleting a file system will also delete all associated snapshots. Once a file system has been deleted, it cannot be recovered, and all client connections to the file system are immediately terminated. All protocols of a file system must be removed before deletion is permitted.

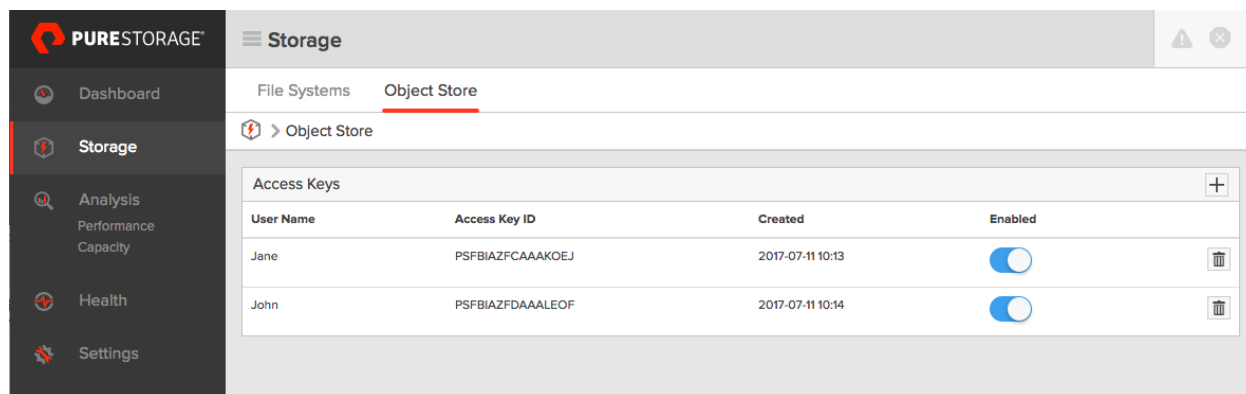
To delete a file system:

1. Select **Storage > File Systems**.
2. Click the **Delete** button in the row of the file system you want to delete. The Delete File System pop-up window appears.
3. Click **Delete**.

Object Store

The Object Store tab displays the access key IDs for buckets on the array. Each access key grants the specified user access to the array's buckets and their contents.

Figure 4.2: Object Store



Access Keys				
User Name	Access Key ID	Created	Enabled	
Jane	PSFBIAZFCAAKEJ	2017-07-11 10:13	☒	
John	PSFBIAZFDAALEOF	2017-07-11 10:14	☒	

The following information is listed for each access key:

- **User Name:** User name (owner of the access key).
- **Access Key ID:** Access key ID.
- **Created:** Access key creation date.
- **Enabled:** Enabled (green checkmark) or disabled (gray X) status.

Each user can have up to two sets of keys.

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.

Use the **purebucket** CLI command to list the array's buckets and the buckets' space usage. See `purebucket(1)` [93].

Process Steps

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 plus icon toward the top right of the page.
2. Enter the user name (owner) of this 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.

Note: There is no confirmation step or undo action. Be careful to delete only the intended key.

To delete an access key:

1. From the Storage > Object Store page, click the row of the access key that you want to delete. Ensure you identify the row for the correct access key.
2. Click the trash icon at the right end of the row.

Enabling and disabling an access key

Access keys are enabled when they are created.

Use the **purebucket** CLI command to disable and re-enable access keys. See `purebucket(1)` [93].

Chapter 5. Health

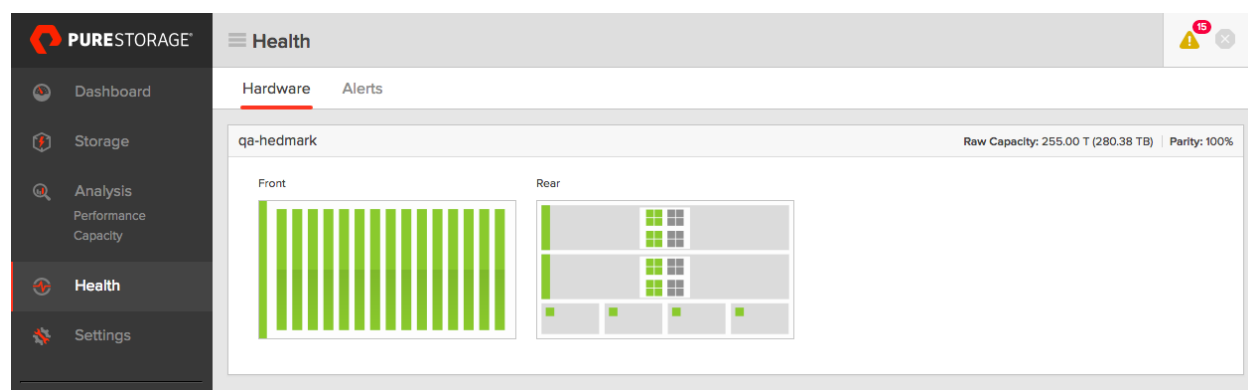
The Health page displays information to help gauge the health of the array.

Hardware

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

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

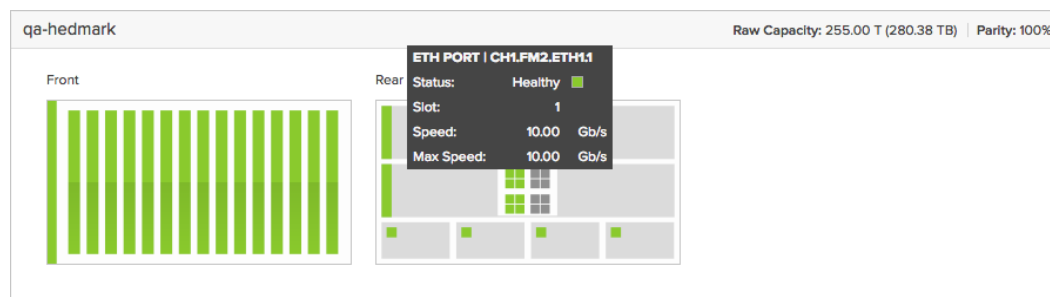


The view is a schematic representation of the array with colored indicators of each component's status. The colors within each hardware component represent the component status:

- **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 5.2: 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.

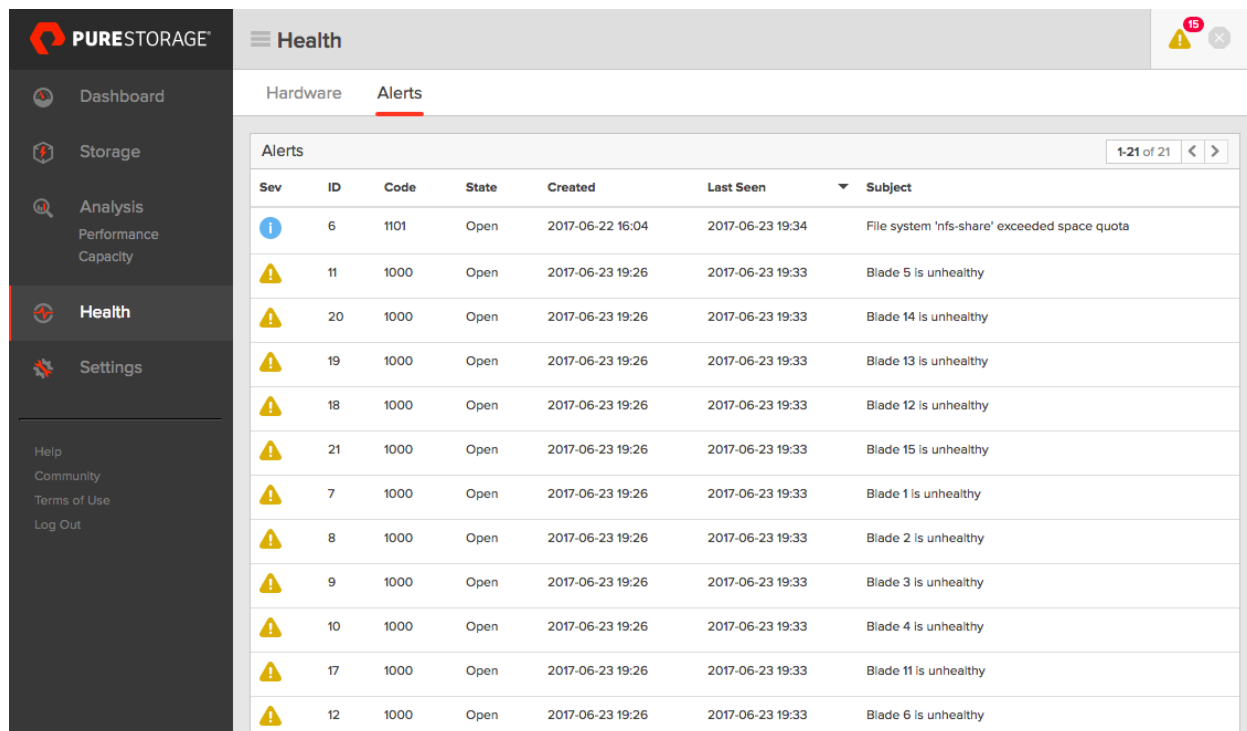
The raw capacity value represents the total usable capacity of the array, displayed in both tebibyte (T) and terabyte (TB) units.

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 5.3: Health - Alerts Page



Sev	ID	Code	State	Created	Last Seen	Subject
i	6	1101	Open	2017-06-22 16:04	2017-06-23 19:34	File system 'nfs-share' exceeded space quota
!	11	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 5 is unhealthy
!	20	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 14 is unhealthy
!	19	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 13 is unhealthy
!	18	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 12 is unhealthy
!	21	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 15 is unhealthy
!	7	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 1 is unhealthy
!	8	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 2 is unhealthy
!	9	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 3 is unhealthy
!	10	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 4 is unhealthy
!	17	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 11 is unhealthy
!	12	1000	Open	2017-06-23 19:26	2017-06-23 19:33	Blade 6 is unhealthy

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 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:

- **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.

- **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 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.

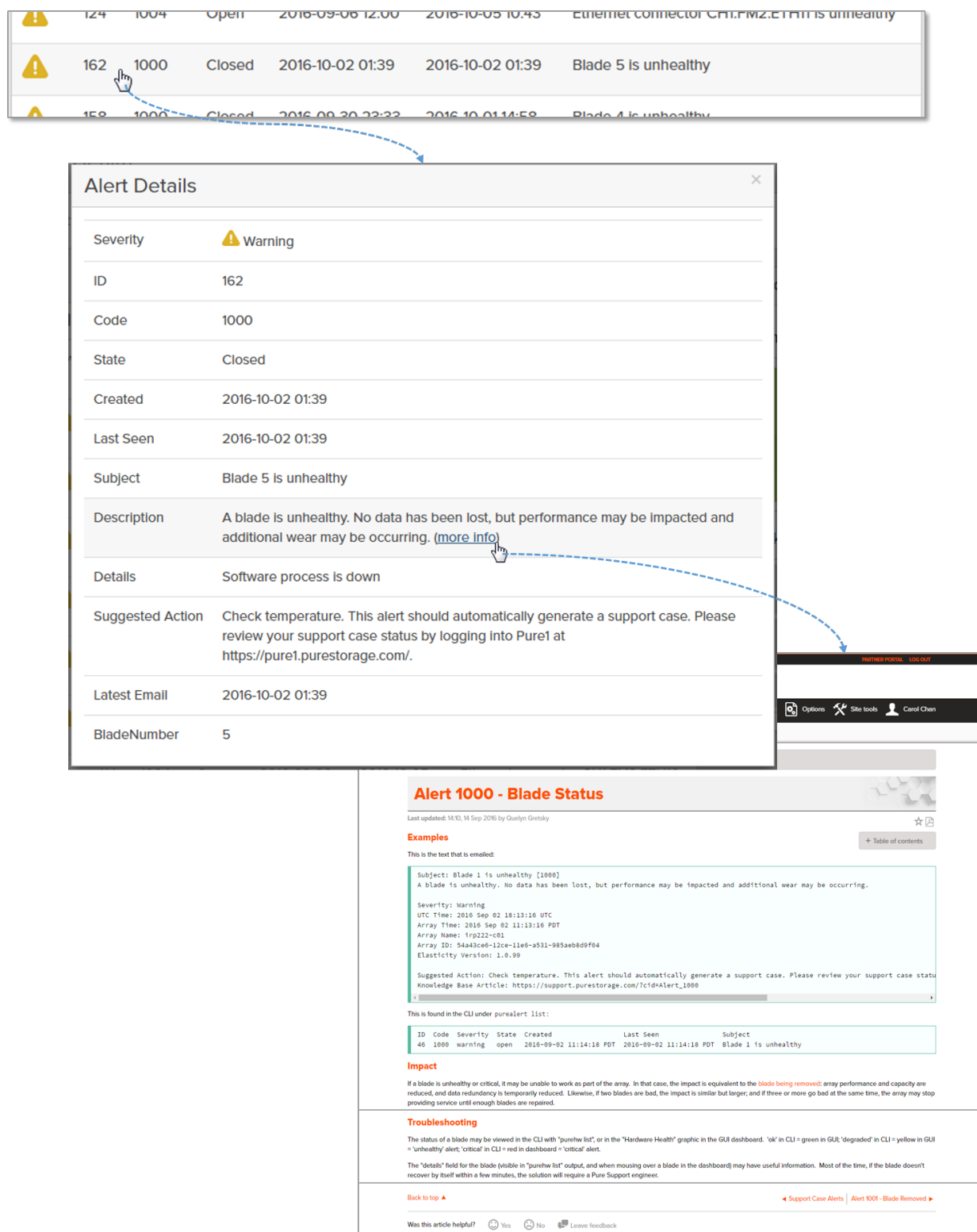
An alert goes from **open** state to **waiting to downgrade** state when the issue becomes less severe. After the alert remains in **waiting to downgrade** state for 24 hours without change, the alert severity is downgraded. For example, when array capacity reaches 100%, a critical alert is issued with 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.

Click anywhere in an alert row to display the alert details.

Figure 5.4: Health - Alerts - Alert Details

Click an alert row to display a detailed description of the alert and suggested actions. Some alert detail descriptions, such as the one in the following example, include a "more info" link providing additional information about the alert through the Pure Storage Knowledge Base.



The screenshot shows the Pure Storage Alerts interface. At the top, there is a table of alerts. The second row, with ID 162 and code 1000, is highlighted. A hand icon points to this row. Below the table, a modal window titled "Alert Details" is open, showing the details for alert 162. The modal includes fields for Severity (Warning), ID (162), Code (1000), State (Closed), Created (2016-10-02 01:39), Last Seen (2016-10-02 01:39), Subject (Blade 5 is unhealthy), Description (A blade is unhealthy. No data has been lost, but performance may be impacted and additional wear may be occurring. [more info](#)), Details (Software process is down), Suggested Action (Check temperature. This alert should automatically generate a support case. Please review your support case status by logging into Pure1 at <https://pure1.purestorage.com/>), Latest Email (2016-10-02 01:39), and BladeNumber (5). A hand icon points to the "more info" link in the Description field. To the right, a partial view of the "Alert 1000 - Blade Status" article is visible, showing examples of the alert message, CLI output, and troubleshooting steps.

ID	Code	Severity	State	Created	Last Seen	Subject
124	1004	Open	2016-09-06 12:00	2016-10-05 10:43	Etnermet connector C.HI.FM2.ET.HMIS unhealthy	
162	1000	Closed	2016-10-02 01:39	2016-10-02 01:39	Blade 5 is unhealthy	
158	1000	Closed	2016-09-30 22:22	2016-10-01 14:58	Blade 4 is unhealthy	

Alert Details

Severity: Warning

ID: 162

Code: 1000

State: Closed

Created: 2016-10-02 01:39

Last Seen: 2016-10-02 01:39

Subject: Blade 5 is unhealthy

Description: A blade is unhealthy. No data has been lost, but performance may be impacted and additional wear may be occurring. [more info](#)

Details: Software process is down

Suggested Action: Check temperature. This alert should automatically generate a support case. Please review your support case status by logging into Pure1 at <https://pure1.purestorage.com/>.

Latest Email: 2016-10-02 01:39

BladeNumber: 5

Alert 1000 - Blade Status

Last updated: 14/01/14 Sep 2016 by Gwyneth Gresham

Examples

This is the text that is emailed:

```
Subject: Blade 1 is unhealthy [1000]
A blade is unhealthy. No data has been lost, but performance may be impacted and additional wear may be occurring.

Severity: Warning
UTC Time: 2016 Sep 02 18:13:16 UTC
Array Time: 2016 Sep 02 11:13:16 PDT
Array Name: f9d222-c01
Array ID: 5443c0e-12ca-11e6-a531-985aebd9f04
Elasticity Version: 1.0.99

Suggested Action: Check temperature. This alert should automatically generate a support case. Please review your support case status
Knowledge Base Article: https://support.purestorage.com/7c1d-Alert\_1000
```

This is found in the CLI under purealert list:

ID	Code	Severity	State	Created	Last Seen	Subject
48	1000	warning	open	2016-09-02 11:14:18 PDT	2016-09-02 11:14:18 PDT	Blade 1 is unhealthy

Impact

If a blade is unhealthy or critical, it may be unable to work as part of the array. In that case, the impact is equivalent to the blade being removed: array performance and capacity are reduced, and data redundancy is temporarily reduced. Likewise, if two blades are bad, the impact is similar but larger, and if three or more go bad at the same time, the array may stop providing service until enough blades are repaired.

Troubleshooting

The status of a blade may be viewed in the CLI with "purehw list", or in the "Hardware Health" graphic in the GUI dashboard. 'ok' in CLI = green in GUI; 'degraded' in CLI = yellow in GUI; 'unhealthy' alert; 'critical' in CLI = red in dashboard = 'critical' alert.

The "details" field for the blade (visible in "purehw list" output, and when mousing over a blade in the dashboard) may have useful information. Most of the time, if the blade doesn't recover by itself within a few minutes, the solution will require a Pure Support engineer.

[Back to top](#) [Support Case Alerts](#) [Alert 1000 - Blade Removed](#)

Was this article helpful? Yes No [Leave feedback](#)

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 Performance panel displays a series of rolling charts consisting of real-time performance metrics; 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.

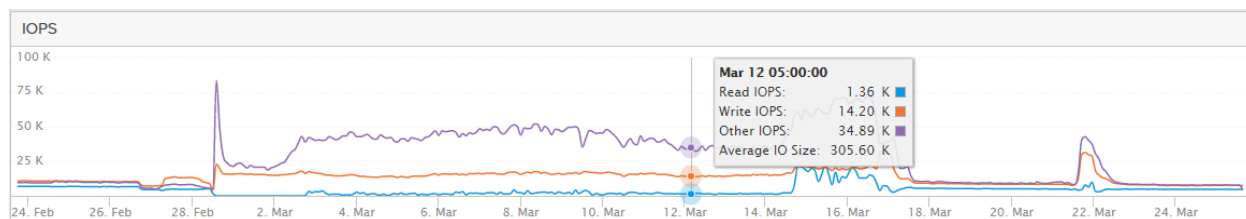
The file transfer can be performed in many chunks for one large file. The Raw view shows one operation per chunk transfer, while the User view shows only one operation per file. When transferring small files, the number of operations is the same in both the Raw and User views.

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

In the Analysis page:

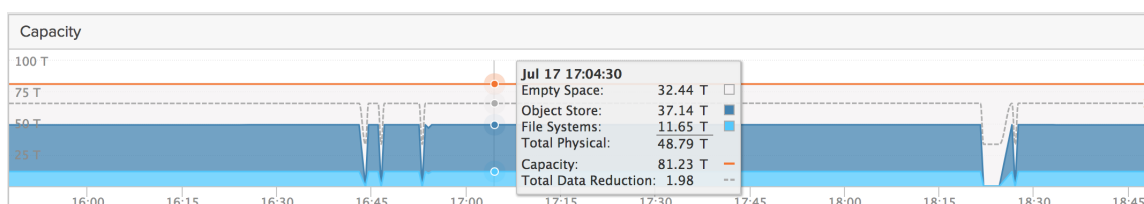
- At its shortest range (5m), the performance charts are updated once every second. As the range increases, the update frequency (and resolution) decreases.
- The performance charts display up to 1 year of historical data.
- The performance charts can be filtered to display metrics by protocol.

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.
- **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

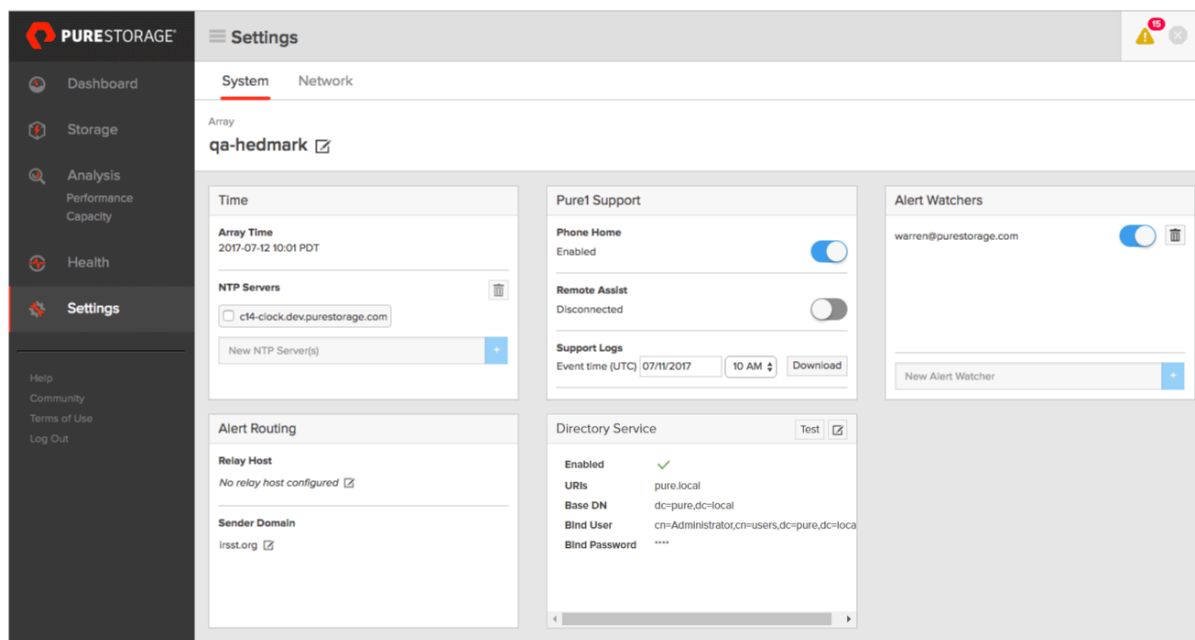
The Settings page displays and manages the general attributes and network settings of an array.

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, and alert settings.

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 UTC 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 start 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**.

Directory Service

The Directory Service panel manages the directory service settings. 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 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.

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.

To configure the directory service for 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.

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.

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 open (blue) and close (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 UTC time, representing the approximate array time the activity of interest occurred.
3. Click **Download** to save the password encrypted file to your administrative workstation. The file can only be opened by Pure Storage Support.

Configuring the proxy host name

1. Select **Settings > System**.
2. Hover over the Pure1 Support panel until the scroll bar appears, then scroll down. 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. Hover over the Pure1 Support panel until the scroll bar appears, then scroll down. 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 starts 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) and 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 icon 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 icon.

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.

Configuring the directory service

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 > Network**
 - b. In the DNS Settings panel, verify that the DNS settings can be used to resolve the directory server domain and server.
3. Configure the directory service.
 - a. Select **Settings > System**
 - 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 (**--base-dn**) of the directory service is used in place of the URI (**--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 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.

- 4. Click **Save**.

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 > System**.

2. In the Directory Service panel, 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.

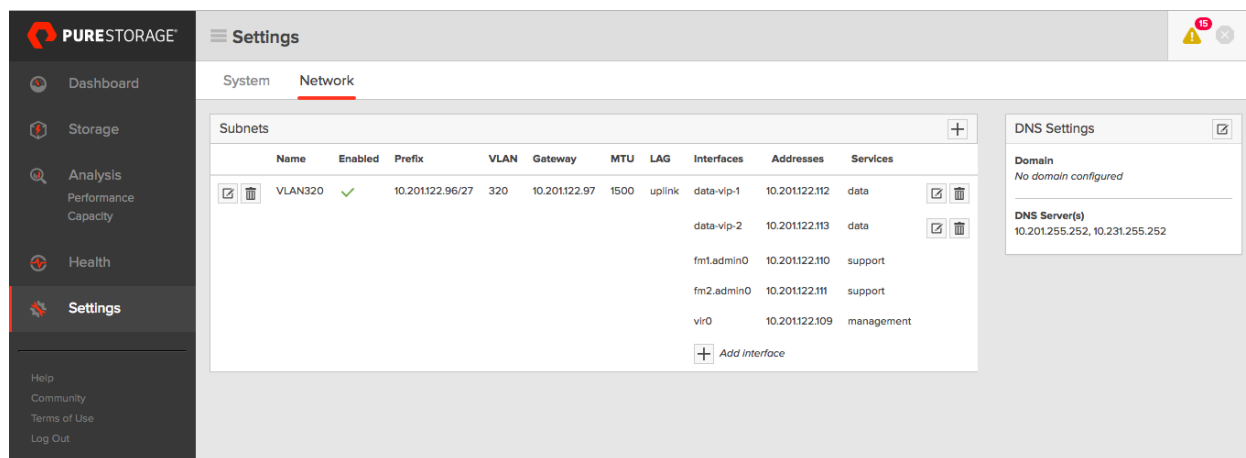
To enable the directory service:

1. Select **Settings > System**.
2. In the Directory Service panel, click the Edit icon. The Edit Directory Service Configuration pop-up window appears.
3. Click the toggle button to enable (blue) the directory service.
4. Click **Save**.

Network

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

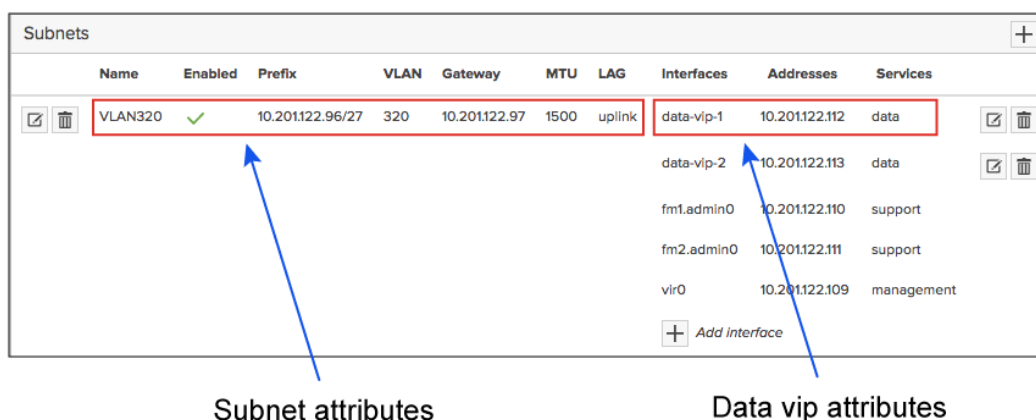
Figure 7.2: Settings - Network Page



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 vips **10.62.9.277** and **10.62.9.199** are attached to subnet **net1**.



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.
- **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 (under the Interfaces column) 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 button. 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 button. 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 Interface button 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 Subnet button next to the subnet you want to delete.

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.

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.

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 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] [--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
  --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.

pureadmin

pureadmin, pureadmin-create, pureadmin-delete, pureadmin-list, pureadmin-setattr — 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...*]

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.

--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.

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.

See Also

`pureads(1)` [100]

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

purealert

purealert, purealert-create, purealert-delete, purealert-disable, purealert-enable, purealert-list, 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 list [ --csv | --nvp ] [--notitle] [--watcher] [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.

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.

--watcher

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

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 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.
- **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.

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**.

See Also

`purearray(1)` [85], `puresupport(1)` [142]

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 [ --ntpserver | --relayhost | --senderdomain | --space ] [ --cli | --csv | --nvp ] [--notitle]
```

```
purearray monitor [ --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.

--historical **TIME**

Displays historical performance data over the specified range of time. Valid time range options include: 5 minutes, 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.

--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.

--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*]

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.

--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*]

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**

--size

Displays the average I/O sizes per read, write, and all operations. The information appears in the **B/op** columns of the output.

--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.

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 --relayhost option to display the hostname or IP address of the email relay server that is currently being used as a forwarding point for alert email notifications generated by the array.

Include the --senderdomain option to display the sender domain name.

Include the --space option to display the following array size and 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, 24 hours, 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.

Configuring the SMTP Relay Host

The **purearray 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 **purearray 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**.

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.

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
```

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

Example 4

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

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

Example 5

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

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

Example 6

```
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 7

```
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 8

```
purearray 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 9

```
purearray 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.

See Also

[purealert\(1\) \[80\]](#), [purearray\(1\) \[85\]](#), [puredns\(1\) \[98\]](#), [purefs\(1\) \[104\]](#), [purehw\(1\) \[119\]](#), [purenetwork\(1\) \[135\]](#), [puresubnet\(1\) \[138\]](#), [puresupport\(1\) \[142\]](#)

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

pureblade

pureblade, pureblade-list — displays the attributes of the blades

Synopsis

pureblade list [--csv | --nvp] [--notitle] [--total] [**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.

Description

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)` [85], `puresupport(1)` [142]

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

purebucket

purebucket, purebucket-create, purebucket-delete, purebucket-disable, purebucket-enable, purebucket-list — displays buckets and their contents; creates and manages access keys for buckets

Synopsis

purebucket create access-key --user-name *USER*

purebucket delete access-key *ID*...

purebucket disable access-key *ID*...

purebucket enable access-key *ID*...

purebucket list [--cli | --csv | --nvp] [--access-key] [--space] [--total] [--total-only] [--protocol *PROTOCOL*] [*BUCKET*...]

Arguments

BUCKET

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

ID

Access key ID or IDs (space-separated list).

USER

User name or names (comma-separated list). Filters output by the specified user names.

Options

-h | --help

Can be used with any command or subcommand to display a brief syntax description.

--access-key

Displays users' access key IDs, enabled or disabled status, user name, and key creation date.

--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.

--user-name

Filters output by the specified user names.

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.

Description

Access Keys for Buckets

Run the **purebucket create access-key** command to create an access key for the specified user. Access keys are enabled when they are created.

The access keys are used to authenticate access to the array's buckets. For example, the access keys are entered in the **.s3cfg** file for use with **s3cmd**, and in the **.s3curl** file for use with **s3curl**.

Each user can have up to two sets of keys.

Important note! At the time you create your access keys, the keys are displayed and you must save or download the key strings. Keys cannot be displayed in either the GUI or the CLI after this step.

Disabling and Enabling an Access Key

Access keys are enabled when they are created.

Run the **purebucket disable access-key** command to deny a user access to the array's buckets and their contents.

The user's next request will fail. Operations in progress when the key is disabled also fail.

Run the **purebucket enable access-key** command to restore the user's access to the array's buckets and their contents.

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.
- **Used:** Total amount of data written to the bucket.

Include the **--space** option to add the following size and 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 create access-key --user-name testuser
```

Creates, enables, and displays an access key for the specified user.

Example 2

```
purebucket list --access-key --user-name tst,tst2
```

Displays the access key IDs for two users.

Example output:

ID	Enabled	Secret Access Key	User Name	Created
PSFBIAZFMAAAJKBA	True	****	tst	2017-06-06 10:15:45 PDT
PSFBIAZFNAACEKB	True	****	tst2	2017-06-06 13:10:43 PDT

Example 3

```
purebucket disable access-key PSFBIAZFMAAAJKBA PSFBIAZFNAACEKB
```

Disables the specified access keys.

Example 4

```
purebucket list
```

Displays each bucket on the array and its Used space.

Example 5

```
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	Used	Data Reduction	Physical
(total)	110.74T	1.7 to 1	63.37T

Example 6

```
purebucket list --access-key --cli
```

Displays the CLI commands to create access keys for the users who currently have access keys configured.

Note that the exact keys cannot be recreated. Each key creation request generates an entirely new access key and secret key.

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 of 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.

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
purealert list --watcher --cli
purearray list --cli
purearray list --relayhost --cli
purearray list --senderdomain --cli
pureds list --cli
puredns 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) [80], purearray(1) [85], puredns(1) [98], purefs(1) [104], purenetwork(1) [135], puresupport(1) [142]

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) [85], puresupport(1) [142]

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

Synopsis

pureds disable

pureds enable

pureds list [--cli]

pureds setattr [--base-dn **BASE-DN**] [--bind-password] [--bind-user **BIND-USER**] [--uri **URI-LIST**]

pureds test

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.

--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**.

--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)` [98] 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.

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.

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.

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.

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**).

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.

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
pureds enable
```

Examples

Example 1

```
pureds setattr --uri ldaps://[2001:0db8:85a3::ae26:8a2e:0370:7334]
                --base-dn DC=mycompany,DC=com --bind-username ldapreader
```

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
```

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-password
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
Feature Status: Disabled

Testing from fm2:
Resolving ad1.mycompany.com... PASSED
Searching ldaps://ad1.mycompany.com... PASSED

Resolving ad2.mycompany.com... PASSED
Searching ldaps://ad2.mycompany.com... PASSED

pureds enable
```

Shows successful output of testing the current configuration and enabling the service following the successful test.

See Also

pureadmin(1) [77], puredns(1) [98]

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

purefs

purefs, purefs-add, purefs-create, purefs-delete, purefs-disable, purefs-enable, purefs-list, purefs-nfs, purefs-remove, purefs-setattr, purefs-snap — displays and manages the file systems

Synopsis

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

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

purefs delete **FILE-SYSTEM**...

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

purefs enable [--fast-remove-dir] [--snapshot-dir] **FILE-SYSTEM**...

purefs list [--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 remove --protocol **PROTOCOL** **FILE-SYSTEM**...

purefs setattr [--size **SIZE**] **FILE-SYSTEM**...

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

Arguments

FILE-SYSTEM

File system name or file system snapshot name. Represents the name of the directory to be exported.

Options

-h | --help

Can be used with any command or subcommand to display a brief syntax description.

--fast-remove-dir

Enables or disables the **.fast-remove** directory for one or more file system(s).

--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 existing set of rules.

Format is '**ddd.ddd.ddd.ddd(options)**' for IPv4 address, '**ddd.ddd.ddd.ddd/dd(options)**' for netmask, or '***(options)**' to represent 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)**'.

--size SIZE

Sets the provisioned size of the file system. If not set, the provisioned size is unlimited.

--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

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.

For more information on how to use the CLI options to control display formatting and manage display results, refer to pureman(1) [125]

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 **--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)` [125]

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)` [135].

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)` [100].
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-style permissions. 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. Extended Windows ACLs are not supported.
- 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 `purefs(1)` [100].

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 **purefs** command. For more information about the directory service, refer to `purefs(1)` [100].
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 filesystem when HTTP is enabled.
 - The use of HTTP connections, as opposed to HTTPS connections, potentially gives unauthenticated users access to the file system. Anyone with network access to the data vip has access to the file system.

Follow these steps to set up file sharing over HTTP access:

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

Important note! The use of HTTP connections, as opposed to HTTPS connections, potentially gives unauthenticated users access to the file system. Anyone with network access to the data vip has access to the file system.

NFS and File Locking

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

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

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

Creating File Systems

The **purefs create** command creates file systems as distributed file shares for export.

The **--size** option represents the provisioned size allocated to 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 and is entered 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 **'*(options)'** to represent all clients

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

Here is an example of the **purefs list** output with various rules set for each file system created:

```
purefs list
```

Name	Size	Used	Created	Protocols
File1	10T	3.56T	2016-08-16 14:10:52 PDT	http
File2	10T	8.99T	2016-08-16 14:16:16 PDT	nfs

File3	10T	5.65T	2016-08-16 14:16:47 PDT	nfs
File4	10T	9.50T	2016-08-16 14:18:36 PDT	http
File5	10T	9.50T	2016-08-16 14:18:36 PDT	http
File6	-	0.00	2016-08-16 14:56:02 PDT	-
File7	-	0.00	2016-08-16 14:56:02 PDT	-

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\(1\) \[100\]](#).
- 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, a **.snapshot** directory will appear in the root 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 deleted, 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 delete any snapshots, they must be actively deleted using the **purefs delete** command.

Deleting File Systems

The **purefs delete** command deletes a file system and all associated snapshots, permanently cutting off access to its contents. Once a file system has been deleted, it cannot be recovered, and all client connections to the file system are immediately terminated. As a precaution, protocols must be removed from a file system before deletion is permitted. Use the **purefs remove** command to remove all protocols from a file system.

Deleting File System Snapshots

The **purefs delete** command deletes a file system snapshot. Once a file system snapshot has been deleted, it cannot be recovered.

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 '*(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 5

```
purefs nfs setattr --rules "" MyFiles
```

Deletes all export rules from file system **MyFiles**, rendering the file system inaccessible to any client.

Example 6

```
purefs add --protocol nfs,smb MyFiles
```

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

Example 7

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

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

Example 8

```
purefs delete MyFiles
```

Deletes file system **MyFiles**.

Example 9

```
purefs delete MyFiles.snap01
```

Deletes file system snapshot **MyFiles.snap01**.

See Also

[purealert\(1\) \[80\]](#), [purearray\(1\) \[85\]](#), [purenetwork\(1\) \[135\]](#), [puresubnet\(1\) \[138\]](#),
[puresupport\(1\) \[142\]](#)

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

purehw

purehw, purehw-list — displays the hardware components of the array

Synopsis

```
purehw list [--spec] [--type COMPONENT-TYPE] [ --csv | --nvp ] [--notitle]  
[COMPONENT...]
```

Arguments

COMPONENT

Hardware component whose information is to be displayed or whose attribute is to be set to the specified value.

Options

-h | --help

Can be used with any command or subcommand to display a brief syntax description.

--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. 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.

--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

In the FlashBlade array, the chassis contains the blades and the bays that host the fabric modules. The FlashBlade chassis name has the form **CH1**. 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 **CH1.XXm** or **CH1.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. The fabric module contains one component named **ETH** (Ethernet port).

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
CH1	Chassis	Chassis
CH1 . FBm	Flash blade	Flash blade server.
CH1 . FMm	Fabric module	I/O modules (IOMs) within the chassis.
CH1 . PWRm	Power module	Power supplies within the chassis.
CH1 . FMm . ETHn	Ethernet port	Ethernet ports to the blades.

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.

Include the **--type** option to display information for all components of the specified type.

Examples

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 --type FB
```

Displays hardware component information for all blades.

Example 4

```
purehw list CH1.FB4
```

Displays hardware component information for the blade in slot 4 of the chassis.

See Also

[purealert\(1\)](#) [80], [purearray\(1\)](#) [85], [puresupport\(1\)](#) [142]

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] [**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.

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) [119], puresubnet(1) [138], purenetwork(1) [135]

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

pureman

pureman — displays man pages for Purity//FB commands

Synopsis

```
pureman { pureadmin | purealert | purearray | pureblade | purebucket | pureconfig  
| puredns | pureds | purefs | purehw | purelag | purenetwork | 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 2¹⁰, "**Mi**" denotes 2²⁰, 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 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] [--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
  --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] [--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.

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)` [138].
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)` [104].

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) [85], purefs(1) [104], puresubnet(1) [138], puresupport(1) [142]

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] [**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.

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) [135].

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) [122].

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) [85], purefs(1) [104], purenetwork(1) [135], puresupport(1) [142], purelag(1) [122]

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] [--cli | --csv | --nvp] [--notitle]

puresupport setattr { --proxy **HTTPS-PROXY** }

puresupport test { --connect | --phonehome }

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.

--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.

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 disabled.

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>](mailto:support@purestorage.com).

Examples

Example 1

```
puresupport list
```

Displays the remote assistance (RA), proxy, and phone home settings for the array.

Example 2

```
puresupport list --connect
```

Displays the connection details and status of a remote assistance (RA) session.

Example 3

```
puresupport connect
```

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 4

```
puresupport enable phonehome
```

Enables the automatic transmission of array logs and diagnostic information to Pure Storage Support.

Example 5

```
puresupport setattr --proxy http://proxy.example.com:8080
```

Sets the proxy host for HTTPS communication between the array and Pure Storage Support.

See Also

`purealert(1)` [80], `purearray(1)` [85],

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]`

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.

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

5. Enable the directory service.

pureds enable

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 > System**
 - 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. 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.

- a. From the Directory Service panel, click the Edit icon. The Edit Directory Service Configuration pop-up window appears.
- b. Click the toggle button to enable (blue) the directory service.
- c. Click **Save**.

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 at [<http://support.purestorage.com>](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>](mailto:documentfeedback@purestorage.com).

Product Support

If you are a registered Pure Storage user, log in to the **Pure Storage Support website**. [<http://www.purestorage.com/support.html>] 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>](mailto: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>](mailto:info@purestorage.com).



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