



FlashBlade CLI Software Configuration Guide (1.0.0)

Contents

Part 1: Using the Elasticity CLI to Administer a FlashBlade Array

Chapter 1: Overview	7
Elasticity Conventions	8
CLI Command Syntax	9
CLI Login	10
CLI Command Help	10
Troubleshooting	12
Chapter 2: Elasticity CLI Process Steps	49
Creating a File System	49
Adding an Alert Watcher	50

Part 1:

Using the Elasticity CLI to
Administer a FlashBlade Array

Chapter 1. Overview

Elasticity is the operating environment that queries and manages the FlashBlade hardware, networking, and storage components. The Elasticity software is distributed with the FlashBlade array.

Elasticity provides two ways to administer the FlashBlade: through the browser-based graphical user interface (Elasticity GUI), and through the command-driven interface (Elasticity CLI).

The Elasticity command line interface (CLI) is a non-graphical, command-driven interface used to query and administer the FlashBlade.

This chapter covers general Elasticity CLI concepts and conventions.

Elasticity Conventions

Elasticity 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 Elasticity 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 Elasticity based on the array name, do not exceed the maximum allowed by DNS.

Array and File System Sizes

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

Time Zones

Dates and times appear throughout the Elasticity GUI and CLI to mark when an event occurred or when a change was made to the array. Elasticity GUI dates and times are displayed in the user's local time zone. Elasticity CLI dates and times are displayed in the time zone of the array, which is set during FlashBlade installation.

CLI Command Syntax

The Elasticity CLI is comprised of built-in commands specific to the Elasticity operating environment. Elasticity 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 Elasticity file systems. Run the **purehelp** command to see a list of Elasticity 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 virtual capacity 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 Login

Log in to the Elasticity CLI to query and administer the FlashBlade. Logging in to the Elasticity CLI requires a virtual IP address or fully-qualified domain name (FQDN) and a login username and password; this information is determined during the FlashBlade installation.

The Elasticity release comes with a single *pureuser* administrative account, which is installed with the array. The *pureuser* account is password protected and cannot be renamed or deleted. If you do not know the password to your *pureuser* account, contact a member of the Pure Storage account team or email Pure Storage Support at **<support@purestorage.com>**.

Logging in to the Elasticity CLI

To log in to the Elasticity 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 Elasticity 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 Elasticity with the *pureuser* username/password combination.

Type **exit** or **logout** to log out of Elasticity 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 Elasticity 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 Elasticity command, type:

COMMAND -h

For example,

```
$ purefs -h
usage: purefs [-h] {create,delete,list,setattr} ...
positional arguments:
  {create,delete,list,setattr}
    create              create a file system
    setattr             modify a file system
    delete              delete one or more file system(s)
    list                list file systems
```

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 Elasticity subcommand, type:

COMMAND SUBCOMMAND -h

For example,

```
$ purefs list -h
usage: purefs list [-h] [--cli | --csv | --nvp] [--notitle] [FILE-SYSTEM ...]
positional arguments:
  FILE-SYSTEM  file system name(s)
optional arguments:
  -h, --help  show this help message and exit
  --cli       display as CLI commands
  --csv       display as comma-separated values
  --nvp       display as name-value pairs
  --notitle   hide column titles
```

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>](mailto: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 Elasticity 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 Elasticity GUI or CLI operation again. If you still get the error message, contact Pure Storage Support at [<support@purestorage.com>](mailto:support@purestorage.com).

Unexpected Error

The Elasticity GUI or CLI operation that you performed generated an unexpected error. Contact Pure Storage Support at [<support@purestorage.com>](mailto:support@purestorage.com).

Elasticity CLI Commands

This section describes each of the Elasticity CLI commands.

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 Elasticity 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

Elasticity generates an alert when there is a change to the array or to one of the Elasticity hardware or software components. Alerts can also be sent as email notifications to designated alert watchers.

The **purealert** command displays and manages the alerts that are generated by Elasticity.

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 component experiencing the 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, Elasticity 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, Elasticity triggers a warning alert if it detects an unhealthy FlashBlade.

Informational (Info) alerts inform users of a general behavior change and require no action. For example, Elasticity triggers an informational alert if the NFS service is unhealthy.

- **State**: Current state of the alert. Possible states include: **open**, **closed**, **waiting to close**, and **waiting to downgrade**.
An alert goes from **open** state to **waiting to close** state when it is no longer an issue. After the alert remains in **waiting to close** 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 initially generated. Alert email notifications are sent to alert watchers.
- **Last Seen**: Most recent date and time Elasticity 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 Elasticity email alert notifications is the array name.

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 immediately starts receiving email alert 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 email alert 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 email alert 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 Elasticity 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 Elasticity from sending alert notifications to alert watcher **aartiealert@example.com**.

See Also

[purearray\(1\)](#) [20], [puresupport\(1\)](#) [44]

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 ]
```

```
purearray monitor [ --historical { 5m | 3h | 1d | 7d | 30d } | [--interval SECONDS] [--repeat REPEAT-COUNT] [--size] ] [--csv] [--notitle]
```

```
purearray rename NEW-NAME
```

```
purearray setattr { --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 day, 7 days, and 30 days.

--interval **SECONDS**

Sets the number of seconds between displays of real-time performance data. At each interval, the system displays a point-in-time snapshot of the performance data. If omitted, the interval defaults to every 5 seconds.

--ntpserver

Displays the hostnames or IP addresses of the NTP servers used by the array to maintain reference time.

--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 notifications generated by the array. Set to an empty value ("") to clear the configured hostname or IP address.

--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 provisioned (virtual) size 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 Elasticity version and revision numbers.

Include the **--ntpserver** option to display the Network Time Protocol (NTP) servers 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 (by default, time.purestorage.com). Up to four (4) NTP server IP addresses or hostnames can be assigned to the array. To change the NTP servers on the array, contact a member of the Pure Storage account team or email Pure Storage Support at [<support@purestorage.com>](mailto:support@purestorage.com).

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 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 provisioned (virtual) size of all file systems on 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 mapped sectors within a file system versus the amount of physical space the data occupies after data compression.
- **File Systems:** Physical storage space occupied by file system data.

Monitoring Array Performance

The **purearray monitor** command displays real-time and historical I/O performance information for the whole array. The output includes the following data about bandwidth, IOPS, and latency:

- **Time:** Current time.
- **B/s:** Number of bytes read/written.
- **op/s:** Number of read/write requests processed per second.
- **us/op:** Average arrival-to-completion time, measured in microseconds, for a host read/write operation.
- **B/op:** Average I/O size per read, write, and all operations. Include the **--size** option to see the **B/op** columns.

Include the **--historical** option to display historical performance data over any of the following ranges of time: 5 minutes, 3 hours, 1 day, 7 days, or 30 days.

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.

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 notifications generated by the array. To clear the configured hostname or IP address, set **--relayhost** to an empty value (""). Once cleared, Elasticity sends all email alert 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 Elasticity 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 Elasticity 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 --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 6

```
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 notifications generated by the array.

Example 7

```
purearray setattr --senderdomain mydomain.com
```

Sets the sender domain **mydomain.com** to be used as the forwarding point for alert notifications generated by the array.

See Also

purealert(1) [15], purearray(1) [20], puredns(1) [25], purefs(1) [27], purehw(1) [33],
purenetwork(1) [36], puresubnet(1) [41], puresupport(1) [44]

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 Elasticity DNS queries, set to an empty string ("").

--nameservers **NAMESERVERS**

Comma-separated list of up to three DNS server IP addresses. To unassign the DNS server IP addresses, set to an empty string (""). Once the DNS server IP addresses have been unassigned, Elasticity 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. Elasticity 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 Elasticity to use to resolve hostnames to IP addresses.

See Also

purearray(1) [20], puresupport(1) [44]

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

purefs

purefs, purefs-create, purefs-delete, purefs-list, purefs-setattr — displays and manages the file systems

Synopsis

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

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

purefs list [--cli | --csv | --nvp] [--notitle] [**FILE-SYSTEM**...]

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

Arguments

FILE-SYSTEM

File system 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.

--rules **RULES**

Defines the NFS export rules for the given file system. Format is

'**xxx.xxx.xxx.xxx(options)**' for IPv4 address, '**xxx.xxx.xxx.xxx/xx(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**, and **no_root_squash**. For example, '**10.20.225.2(ro,root_squash)**'. If export options are not specified, the client will have **rw** access and **no_root_squash** privileges to the file system.

--size **SIZE**

Sets the provisioned (virtual) size of the file system.

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 **purefs** command displays and manages the file systems for NFS export.

Exporting a file system requires at least one valid data virtual IP address (data vip), which are configured through the **purenetwork** command.

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:

- **Size:** Provisioned (virtual) size of the file system.
- **Used:** Physical storage space occupied by file system data.
- **% Used:** Percentage of provisioned size occupied by file system data.
- **Created:** File system creation date.
- **Rules:** File system NFS export rules.

Creating File Systems

The **purefs create** command creates file systems as NFS shares for export. Once a file system has been created, it is visible through any of the data vips configured on the subnet, ready to be mounted on the clients.

Each FlashBlade file system can include a set of NFS export rules that define the access rights and privileges a client has to the file system. The **--rules** option configures the NFS export rules. 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:

- IPv4 address in the form **'xxx.xxx.xxx.xxx(options)'**
- Netmask in the form **'xxx.xxx.xxx.xxx/xx(options)'**
- 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**, **no_root_squash**, and **root_squash**.

Enclose the entire rule set with single quotes. For example,

```
$ purefs create --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 create --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. To create a file system without NFS export rules and, therefore, be inaccessible to any client, set **--rules** to an empty value ("").

The following guidelines apply when creating NFS export rules:

- If rules are not defined during file system creation, then all clients will have **rw** access and **no_root_squash** privileges to the file system. For example,

```
purefs create MyFiles
```

This is equivalent to specifying export rule **'*(rw,no_root_squash)'**.

- If the entire rule is represented by the asterisk (*) wildcard character, then all clients will have **ro** access and **root_squash** privileges to the file system. For example,

```
purefs create --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 an IP address matches multiple rules, then the most specific IP-based rule wins. 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.

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

```
$ purefs list
```

Name	Size	Used	% Used	Created	Rules
MyFiles1	10T	3.56T	0%	2016-08-16 14:10:52 PDT	*
MyFiles2	10T	8.99T	0%	2016-08-16 14:16:16 PDT	* ()
MyFiles3	10T	5.65T	0%	2016-08-16 14:16:47 PDT	-
MyFiles4	10T	9.50T	0%	2016-08-16 14:18:36 PDT	*(rw,no_root_squash)
MyFiles5	-	0.00	-	2016-08-16 14:56:02 PDT	10.20.225.2(rw)
MyFiles6	-	0.00	-	2016-08-16 14:56:02 PDT	1.2.0.0(no_root_squash)

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

- * - All clients have default **ro** access and **root_squash** privileges to the file system.
- * () - All clients have default **ro** access and **root_squash** privileges to the file system.
- - - Export rules have been delete, rendering the file system inaccessible.
- *(**rw,no_root_squash**) - All clients have **rw** access and **no_root_squash** privileges.
- **10.20.225.2(rw)** - All clients have **rw** access and **root_squash** (default) privileges.
- **1.2.0.0(no_root_squash)** - All clients have the **ro** (default) access and **no_root_squash** privileges.

The **--size** option represents the provisioned (virtual) 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 physical storage space of the file system reaches a certain percentage of its quota, alerts are generated notifying administrators of the threshold reached. For example, Elasticity generates an **INFO** alert when the physical storage space occupied by data for a given file system exceeds 100% of its provisioned size.

Elasticity 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.

Configuring File Systems

The **purefs setattr** command changes the attributes of a file system. Include the **--rules** option to set the NFS export rules for the file system, replacing any export rules that currently exist. Include the **--size** option to change the virtual size of the file system.

Deleting File Systems

The **purefs delete** command deletes a file system, permanently cutting off access to its contents. Once a file system has been deleted, all client connections to the file system are immediately terminated.

Examples

Example 1

```
purefs list
```

Displays a list of file systems that have been created on the FlashBlade array for export, and the rules that accompany each file system.

Example 2

```
purefs create --rules '*(rw,no_root_squash) 10.20.225.2(ro,root_squash)' MyFiles
```

Creates file system **MyFiles**, granting **rw** access and **no_root_squash** privileges for all clients except client **10.20.225.2**, which has **ro** access and **root_squash** privileges.

Example 3

```
purefs create --size 10T '*(rw,no_root_squash)' MyFiles
```

or

```
purefs create --size 10T MyFiles
```

Creates file system **MyFiles** with a provisioned size of 10 terabytes. All hosts have **ro** access and **no_root_squash** privileges to the file system.

Example 4

```
purefs create --rules '10.20.225.2() 1.2.0.0/16(rw,no_root_squash)' MyFiles
```

Creates file system **MyFiles** with two export 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 5

```
purefs setattr --rules '*(ro,root_squash)' MyFiles
```

or

```
purefs setattr --rules '*' MyFiles
```

or

```
purefs 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 6

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

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

Example 7

```
purefs delete MyFiles
```

Deletes file system **MyFiles**.

See Also

purealert(1) [15], purearray(1) [20], purenetwork(1) [36], puresubnet(1) [41],
puresupport(1) [44]

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

purehw

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

Synopsis

```
purehw list [--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.

--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:** Each component reports its status as one of the following:
 - **ok** - Functioning properly at full capacity.
 - **critical** - Not functioning or requiring immediate attention.
 - **degraded** - Functioning, but not at full capability due to a non-fatal failure.
 - **device_off** - Installed, but powered off.
 - **identifying** - Functioning, but is not yet initialized.
 - **not_installed** - Does not appear to be installed.
 - **unknown** - Insufficient information to determine a state for this device.
- **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 **--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 --type FB
```

Displays information for all blades.

Example 3

```
purehw list CH1.FB4
```

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

See Also

[purealert\(1\)](#) [15], [purearray\(1\)](#) [20], [puresupport\(1\)](#) [44]

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
purenetwork-ping, purenetwork-trace — pings remote destinations and traces routes on the network

Synopsis

purenetwork create {vip} --address **ADDRESS** --subnet **SUBNET** --service **SERVICE** **NAME**

purenetwork delete **NAME...**

purenetwork list [--cli | --csv | --nvp | --service **SERVICE** | --vlan **VLAN**] [--notitle]

purenetwork ping [--count **COUNT**] [--interface **NAME**] [--no-hostname] [--packet-size **PACKET-SIZE**] [--user-to-user-latency] **DEST**

purenetwork setattr [--address **ADDRESS**] [--subnet **SUBNET**] **NAME**

purenetwork trace [--dont-fragment] [--interface **INTERFACE**] [--method **METHOD**] [--mtu] [--no-hostname] [--port **PORT**] [--source **SOURCE**] **DEST**

Arguments

DEST

Internet Protocol (IP) address or full hostname of a ping target or of a remote computer to which the network route is to be determined.

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**

Management address to be associated with the specified data vip. Specify the IPv4 address in the format **ddd.ddd.ddd.ddd**.

--count **COUNT**

Number of Internet Control Message Protocol (ICMP) ping messages to send in sequence.

--dont-fragment

Do not fragment the probe packets. For IPv4, it also sets the Don't Fragment (DF) bit. By default, IP allows the packets to be fragmented when it goes through a segment with a smaller maximum transmission unit (MTU).

--interface *INTERFACE*

For **purenetwork ping**, IPv4 address or interface name of the current controller. If the interface is an address, the source address is set to the specified interface address.

For **purenetwork trace**, interface name of the current controller.

--method *METHOD*

Method for trace operations. By default, the trace operation uses the UDP method. Trace operations can also be specified as ICMP (icmp) or TCP (tcp).

--mtu

Maximum transmission unit (MTU) along the path being traced between two Ethernet hosts.

--no-hostname

Do not map IP addresses to hostnames when displaying them.

--packet-size *PACKET-SIZE*

The number of data bytes to be sent. The default packet size is 56. For ICMP tracing, the default packet size is combined with 8 bytes of ICMP header data to equal 64 ICMP data bytes.

--port *PORT*

Initial destination port. The default destination port is 33434.

For UDP tracing, specifies the destination port base used by **purenetwork trace**. The destination port number will be incremented by each probe. For ICMP tracing, specifies the initial ICMP sequence value incremented by each probe. For TCP tracing, specifies the destination port to connect.

--service

Lists only the interfaces configured with the specified service. The supported service for data vips is **nfs**.

--source *SOURCE*

Alternative source address. The source address must be the address of one of the interfaces. By default, **purenetwork trace** uses the address of the outgoing interface.

--subnet *SUBNET*

For **purenetwork setattr**, name of the subnet which the data vip is to be attached. To detach a data vip from a subnet, set to an empty string ("").

For **purenetwork list**, lists only the interfaces that belong to the specified subnet.

--user-to-user-latency

Prints full user-to-user latency. By default, **purenetwork ping** prints network round-trip times.

--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 nfs** 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, if any, has been created. To view subnet details, run **puresubnet list**. For more information about subnets, refer to `puresubnet(1)` [41].
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 IPv4 address of the client in the format **ddd.ddd.ddd.ddd**. Include the required --**servicelist nfs** option to assign the **nfs** service to the data vip. Include the required --**subnet** option to attach the data vip to the subnet. Once the data vip is created, it inherits the gateway (if any) and VLAN ID from the subnet. Likewise, the subnet inherits the **nfs** 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)` [27].

Setting 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 management IP address of the data vip. Include the --**subnet** option to change the subnet. If the address is changed to one that belongs to a different subnet, then the subnet must also be changed.

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.

Network Ping and Trace

The **purenetwork ping** command is used to determine whether a remote computer can be accessed by the array, provided that the remote computer is ICMP-enabled. The ping target can be specified by IP address. If a DNS service is available and has been configured for the array, the ping target can be specified by hostname.

The **purenetwork trace** command traces and displays the route to a remote computer identified by IP address or, if a DNS service is available and configured for the array, by hostname.

Examples

Example 1

```
purenetwork list --service nfs
```

Displays a list of data vips created for file system export.

Example 2

```
purenetwork create vip --address 10.10.200.10 --service nfs --subnet SUB01 DataVip01
```

Adds data vip **DataVip01** to subnet **SUB01** 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**. Both addresses belong to the same subnet.

Example 4

```
purenetwork setattr --address 10.255.8.100 --subnet SUB08 DataVip03
```

Changes the IP address of **DataVip03** from **10.255.9.100** to **10.255.8.100**, which belongs to a different subnet (**SUB08**).

Example 5

```
purenetwork delete DataVip01 DataVip02
```

Deletes data vips **DataVip01** and **DataVip02** from the FlashBlade array.

Example 6

```
purenetwork ping --count 100 myhost.mydomain.com
```

Sends 100 ICMP ping messages to host **myhost.mydomain.com** and displays the response received for each one.

Example 7

```
purenetwork trace 192.168.0.1
```

Displays the route taken by a ping request to network address **192.168.0.1**.

See Also

`purearray(1)` [20], `purefs(1)` [27], `puresubnet(1)` [41], `puresupport(1)` [44]

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

puresubnet

puresubnet, puresubnet-list — displays the subnets used to create data vips for file system export purposes

Synopsis

```
puresubnet list [--service SERVICE] [ --cli | --csv | --nvp ] [--notitle]  
[SUBNET...]
```

Arguments

SUBNET

Subnet name.

Options

-h | --help

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

--service **SERVICE**

Lists only the interfaces configured with the specified service. For FlashBlade, the single supported service is **nfs**.

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

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, if any, are grouped into the same subnet.

The **puresubnet** command displays the subnets on the FlashBlade array.

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    Services Interfaces
SUB01   True    10.10.200.0/24  100  10.10.200.1 -          -
```

Before creating a data vip, verify the subnet with the correct network prefix, VLAN ID, and gateway, if any, appears in the list output. For more information about creating data vips, refer to purenetwork(1) [36].

After the data vip has been created and attached to the subnet, the subnet inherits the **nfs** service type from the data vip.

The **puresubnet list --service nfs** command displays a list of subnets on the array to which data vips are attached for file system export. For example,

```
puresubnet list --service nfs
Name    Enabled Prefix          VLAN Gateway    Services Interfaces
SUB01   True    10.10.200.0/24  100  10.10.200.1 nfs        DataVip01
```

Creating, Modifying, and Deleting Subnets

Subnets are created, configured, and deleted by Pure Storage representatives. To create, modify, or delete a subnet, contact a member of the Pure Storage account team or email Pure Storage Support at [<support@purestorage.com>](mailto:support@purestorage.com).

Creating a subnet requires a prefix and VLAN interface, so have this information ready in advance.

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 nfs
```

Displays a list of subnets to which data vips are attached for file system export.

See Also

purearray(1) [20], purefs(1) [27], purenetwork(1) [36], puresupport(1) [44]

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 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)` [15], `purearray(1)` [20],

Author

Pure Storage Inc. <documentfeedback@purestorage.com>

Chapter 2. Elasticity CLI Process Steps

Creating a File System

To create a file system, add a data virtual IP address (data vip) to an existing subnet, and then create and export the file system.

1. Find the subnet to which the data vips will be added.

```
puresubnet list
```

2. Add data virtual IP addresses (data vips) to the subnet. These addresses will be used to export the file system.

```
purenetwork create vip --address ADDRESS --service nfs --subnet SUBNET NAME
```

Optionally run **purenetwork list** to verify that the new data vip has been properly created and configured.

3. Create and export the file system. Define the NFS export rules so the file system is accessible to the appropriate clients.

```
purefs create [--rules RULES] [--size SIZE] FILE-SYSTEM
```

Optionally run **purefs list** to verify that the new file system and its export rules have been properly configured.

The file system is now visible through any of the data vips configured on the subnet, and ready to be mounted on the clients.

For example, run the following commands to:

- Export the **MyFiles** file system. Client **192.168.1.0/24** should be able to access the exported files with **read-only access** and **root_squash** privileges. Client **1.2.0.0/16** should also be able to access the exported files with **read-write** access and **root** privileges.
- Mount the exported **MyFiles** file system via data vip **10.10.200.10** to the **/mnt** directory of the local host.

On the server...

```
puresubnet list  
//Finds the subnet, named SUB01, to which the data vip will be added.  
purenetwork create vip --address 10.10.200.10 --service nfs --subnet SUB01 DataVip01  
purefs create --rules '192.168.1.0/24(ro) 1.2.0.0/16(rw,no_root_squash)' MyFiles
```

On the client...

```
mkdir -p ~/mnt  
mount 10.10.200.10:/MyFiles ~/mnt
```

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 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 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
```




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