Pure Storage

FlashBlade CLI Quick Reference Guide

1.0.0

Table of Contents

Introduction

Elasticity is the operating environment that manages the FlashBlade array. Elasticity, which comes bundled with the FlashBlade array, can be administered through a graphical user interface (Elasticity GUI) or command line interface (Elasticity CLI).

This document provides process steps and examples to help you administer the FlashBlade through the Elasticity CLI and is a supplement to the FlashBlade Software Configuration Guide.

CLI Command Syntax and Conventions

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

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, `purenetwork list` with no interface names specified displays information about all interfaces in an array.

In the command synopses, OBJECT specifications enclosed in square brackets (for example, "[NAME]") represent ones that are optional.

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 the **-h** or **--help** switch.

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 subcommand 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
```

Process Steps

Logging in to Elasticity

The Elasticity release comes with a single pureuser administrative account, which is installed with the array. The pureuser account is password-protected (default password is "pureuser"). The pureuser account cannot be renamed or deleted.

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 username/password combination *pureuser*/*pureuser*.
- **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 username/password combination *pureuser*/*pureuser*.

Type **`exit`** or **`logout`** to log out of Elasticity and exit the shell or terminal emulator.

Creating a File System

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

puresubnet list

Note the subnet name.

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.

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, but with read-only access and root_squash privileges. Client 1.2.0.0/16 should also be able to access the exported files, but 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)'
              --size 1G 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.

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
```

Examples

purealert

Display Alerts

Run **purealert list** to display a list of all alerts on the FlashBlade array that have been generated by a hardware, software, or array event. For example,

```
$ purealert list
ID Code Severity State     Created              Last Seen            Subject
1  1000 warning  closed    2016-06-21 14:53:42  2016-06-22 13:13:21  Blade 1 is unhealthy
8  1006 info     closed    2016-06-21 18:58:10  2016-07-05 13:58:13  NFS service is unhealthy
30 1001 critical critical  2016-07-03 09:15:37  2016-07-03 09:15:37  Blade:7 was removed
```

List Alert Watchers

Run **purealert list --watcher** to display a list of alert watchers that have been created on the array. For example,

```
$ purealert list --watcher
Name                       Enabled
alexalert@example.com      True
walterwatcher@example.com  False
wendywatcher@example.com   True
```

Send Test Messages to Alert Watchers

Before adding an alert watcher, run **purealert test watcher** to send a test message to the alert watcher's email address to ensure alert notifications reach the intended destination. For example,

```
$ purealert test watcher aartiealert@example.com
Name                     Accepted  Error
aartiealert@example.com  True      -
```

At any time, run **purealert test watcher** to send a test message to the email addresses of all enabled alert watchers to ensure alert notifications reach the intended destinations. For example,

```
$ purealert test watcher
Name                       Accepted  Error
alexalert@example.com      True      -
wendywatcher@example.com   True      -
```

purearray

Display Array Attributes

Run **purearray list** to display array attributes, including array name, array ID, array version, and array revision number. For example,

```
$ purearray list
Name         ID                                    Version    Revision
MyArray-001  2a293b93-f8e3-4d06-821d-c78095214b89  0.9.2      20160722_3fd516d9
```

Display Array Capacity and Consumption Details

Run **purearray list --space** to display array size and storage consumption details. For example,

```
$ purearray list --space
Name           Capacity  Data Reduction  File Systems
MyArray-001    74.11T    1.3 to 1        23.34T
```

List NTP Server Addresses

Run **purearray list --ntpserver** to list the hostnames or IP addresses of the NTP servers that are currently configured on the array. For example,

```
$ purearray list --ntpserver
NTP Server
MyNTPServer1.com
MyNTPServer2.com
MyNTPServer3.com
```

Configure the Relay Host

Run **purearray setattr --relayhost** to set 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. For example,

```
$ purearray setattr --relayhost fake.relay.purestorage.com
Relay Host
fake.relay.purestorage.com
```

Run **purearray list --relayhost** to display the hostname or IP address of the relay server that is currently configured on the array.

To clear the configured relay hostname or IP address, set --**relayhost** to an empty value (**""**).

Configure the Sender Domain

Run **purearray setattr --senderdomain** to configure how logs are parsed and treated by Pure Storage Support and Escalations. For example,

```
$ purearray setattr --senderdomain MyCompany.com
Sender Domain
MyCompany.com
```

Important! It is crucial that you set the sender domain to the correct domain name. The default domain name is `purestorage.com`. If the array is not a Pure Storage test array, you must set the sender domain to your company domain name. If the sender domain is not set to your company, the array will be treated as an internal Pure Storage test array, potentially blocking alerts and other important warnings.

Run **purearray list --senderdomain** to display the sender domain name that is currently configured on the array.

puredns

Configure the Domain Suffix

Run **puredns setattr --domain** to set the domain suffix to be appended to DNS queries. For example,

```
$ puredns setattr --domain MyDomain.com
Domain        Nameservers
MyDomain.com  -
```

Configure the Name Servers

Run **puredns setattr --nameservers** to configure the list of DNS name server IP addresses. For example,

```
$ puredns setattr --nameservers 192.168.0.125,192.168.1.125,192.168.2.125
Domain        Nameservers
-             192.168.0.125
              192.168.1.125
              192.168.2.125
```

Notes:

- The `puredns setattr --nameservers` command overwrites any name server IP addresses that are already configured on the array.
- Elasticity queries the DNS servers in the order in which the IP addresses are listed.

Display DNS Attributes

Run **puredns list** to display the domain suffix and name servers configured on your array. For example,

```
$ puredns list
Domain        Nameservers
MyDomain.com  192.168.0.125
              192.168.1.125
              192.168.2.125
```

purefs

List File Systems

Run **purefs list** to display a list of file systems created on your array for export, and the rules that accompany each file system.

For example,

```
$ purefs list
Name  Size Used    % Used  Created                  Rules
FS1   5T   4.03T      81%  2016-06-20 23:26:47 UTC  *(rw,no_root_squash)
FS2   1T   976.90M    0%   2016-06-27 01:14:04 UTC  *(rw,no_root_squash)
FS3   15T  11.94T     80%  2016-06-20 22:07:02 UTC  10.20.0.0/16(rw,no_root_squash)
```

purehw

List Hardware Components

Run **purehw list** to display a list of FlashBlade array hardware components and their attributes. Include the **--type** option to display information for all components of the specified type. For example,

```
$ purehw list --type FM
Name     Status         Identify  Slot  Index  Speed  Temperature  Details
CH1.FM1  ok             off       -     1      -      -
CH1.FM2  ok             off       -     2      -      -
```

purenetwork

List Data Vips

Run **purenetwork list --service nfs** to display a list of data vips created for file system export. Data vips are configured with the **nfs** service.

For example,

```
$ purenetwork list --service nfs
Name         Enabled Subnet Address       VLAN  Mask           Gateway      Services
DataVIP01    True    net2   10.20.120.20  2120  255.255.248.0  10.20.120.1  nfs
DataVIP02    True    net2   10.20.120.28  2120  255.255.248.0  10.20.120.1  nfs
DataVIP03    True    net2   10.20.120.29  2120  255.255.248.0  10.20.120.1  nfs
10.20.79.10 True    net1   10.20.79.10   2072  255.255.248.0  10.20.72.1   nfs
```

puresubnet

List Subnets Created for File System Export

Run **puresubnet list --service nfs** to display a list of subnets created for file system export. Subnets created for file system export are configured with the **nfs** service.

For example,

```
$ puresubnet list --service nfs
Name  Enabled  Prefix          VLAN  Gateway      Services    Interfaces
SUB1  True     10.20.72.0/21   2072  10.20.72.1   nfs         fm1.admin0
                                                  support     fm2.admin0
                                                  management  vir0
                                                              10.20.79.10
SUB2  True     10.20.120.0/21  2120  10.20.120.1  nfs         DataVIP01
                                                              DataVIP02
                                                              DataVIP03
```

puresupport

Open a Remote Assist (RA) Session

Run **puresupport connect** to open a remote assist session. Opening a remote assist session gives Pure Storage Support the ability to log into the array and establish an administrative session. For example,

```
$ puresupport connect
Name         Opened                   Expires                  Proxied  Status
MyArray-001  2016-02-10 23:03:38 UTC  2016-02-12 23:03:38 UTC  False    Connected
```

Open remote assist sessions automatically expire (close) after two days have lapsed. Run **puresupport disconnect** to terminate an open RA session.

Enable the Phone Home Facility

Run **puresupport enable phonehome** to enable the transmission of diagnostic logs to Pure Storage support. For example,

```
$ puresupport enable phonehome
Phonehome
Enabled
```

Run **puresupport disable phonehome** to disable the phone home facility and effectively stopping log transmission.

Configure a Proxy Host

Run **puresupport setattr --proxy** to configure the proxy host for HTTPS communication. For example,

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

Display Connection Settings

Run **puresupport list** to display remote assist (RA), proxy, and phone home settings. For example,

```
$ puresupport list
RA Active  Proxy                          Phonehome
True       http://proxy.example.com:8080  enabled
```

Display Remote Assist (RA) Connection Details

Run **puresupport list --connect** to display the connection details and the status of a remote assist (RA) session. For example,

```
$ puresupport list --connect
Name         Opened                   Expires                  Proxied  Status
MyArray-001  2016-02-10 23:03:38 UTC  2016-02-12 23:03:38 UTC  False    Connected
```

Troubleshooting

Who do I contact about problems with the FlashBlade Array?

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

Pure Storage Support may ask you to open an RA session so that they can diagnose the issue.

Refer to the cheat sheet for steps on how to open an RA session and display FlashBlade configuration information.