Dell EMC ViPR Controller

Version 3.6.2.1

Command Line Reference

302-005-084

REV 02

DELL EMC

Copyright © 2013- 2018 Dell Inc. or its subsidiaries. All rights reserved.

Published August 2018

Dell believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS-IS." DELL MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. USE, COPYING, AND DISTRIBUTION OF ANY DELL SOFTWARE DESCRIBED IN THIS PUBLICATION REQUIRES AN APPLICABLE SOFTWARE LICENSE.

Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA.

Dell EMC
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.DellEMC.com

Revision history

Table 1 Revision history

Revision Date	Description of change
May 2018	Commands that have been updated: • volume create command • exportgroup add_vol command
August 2018	New commands: • snapshot expand command

CONTENTS

Revision history **3**

Tables **17**

Chapter 1 **ViPR Controller Command Line Interface Overview** **19**
Welcome to the ViPR Controller Command Line Interface Reference..... 20
Deploy the ViPR Controller CLI..... 20
Using viprcli.profile to simplify issuing viprcli commands..... 20
viprcli command format.....21
CLI help..... 23
Common Parameters.....24
Authenticating with viprcli.....24

Chapter 2 **viprcli approval commands** **27**
viprcli approval approve.....28
viprcli approval list.....28
viprcli approval reject..... 29
viprcli approval show..... 30

Chapter 3 **viprcli assetoptions commands** **33**
viprcli assetoptions list..... 34

Chapter 4 **viprcli authenticate commands** **35**
viprcli authenticate.....36

Chapter 5 **viprcli authentication commands** **37**
Authentication Provider.....38
viprcli authentication add-provider..... 44
viprcli authentication list-providers..... 45
viprcli authentication show-provider..... 46
viprcli authentication update.....47
viprcli authentication delete-provider..... 48
Active Directory integration for ViPR Controller.....48
viprcli authentication add-vdc-role..... 49
viprcli authentication list-vdc-role..... 50
viprcli authentication delete-role..... 50
Assigning a role to an active directory user..... 51
viprcli authentication add-user-group.....52
viprcli authentication user-group-add-attribute.....53
viprcli authentication user-group-add-values..... 53
viprcli authentication user-group-remove-attribute.....54
viprcli authentication user-group-remove-values..... 55
viprcli authentication show-user-group..... 56
viprcli authentication delete-user-group.....56
viprcli authentication list-user-groups..... 57

Chapter 6 **viprcli bucket commands** **59**
viprcli bucket acl 60
viprcli bucket create 61
viprcli bucket delete 62
viprcli bucket delete-acl 63
viprcli bucket list-acl 64
viprcli bucket show 64
viprcli bucket update 65

Chapter 7 **viprcli catalog commands** **67**
viprcli catalog get-category 68
viprcli catalog get-descriptor 72
viprcli catalog get-service 75
viprcli catalog execute 76

Chapter 8 **viprcli cluster commands** **79**
viprcli cluster create 80
viprcli cluster delete 81
viprcli cluster detach 82
viprcli cluster get-hosts 83
viprcli cluster list 84
viprcli cluster list-umexportmasks 84
viprcli cluster list-umvolumes 85
viprcli cluster show 86
viprcli cluster tasks 87
viprcli cluster update 88

Chapter 9 **viprcli computeimage commands** **91**
viprcli computeimage create 92
viprcli computeimage delete 92
viprcli computeimage list 93
viprcli computeimage show 94
viprcli computeimage update 94

Chapter 10 **viprcli computeimageserver commands** **97**
viprcli computeimageserver create 98
viprcli computeimageserver delete 99
viprcli computeimageserver list 99
viprcli computeimageserver show 99
viprcli computeimageserver update 100

Chapter 11 **viprcli computelement commands** **103**
viprcli computelement deregister 104
viprcli computelement list 104
viprcli computelement register 105
viprcli computelement show 105

Chapter 12 **viprcli computesystem commands** **107**
viprcli computesystem create 108
viprcli computesystem delete 109
viprcli computesystem deregister 109

viprcli computesystem discover....110
viprcli computesystem list....110
viprcli computesystem list-compute-elements....111
viprcli computesystem register....112
viprcli computesystem show....112
viprcli computesystem update....113

Chapter 13 viprcli computevpool commands 115
viprcli computevpool assign_computele....116
viprcli computevpool create....116
viprcli computevpool delete....118
viprcli computevpool list....119
viprcli computevpool show....119
viprcli computevpool update....120

Chapter 14 viprcli consistencygroup commands 123
viprcli consistencygroup accessmode....124
viprcli consistencygroup create....125
viprcli consistencygroup delete....126
viprcli consistencygroup failover....127
viprcli consistencygroup failover_cancel....128
viprcli consistencygroup list....130
viprcli consistencygroup show....131
viprcli consistencygroup swap....132
viprcli consistencygroup update....133

Chapter 15 viprcli executionwindow commands 135
viprcli executionwindow create....136
viprcli executionwindow delete....137
viprcli executionwindow list....138
viprcli executionwindow show....139
viprcli executionwindow update....140

Chapter 16 viprcli event commands 143
viprcli event approve....144
viprcli event delete....145
viprcli event list....146
viprcli event details....147
viprcli event decline....148
viprcli event show....149

Chapter 17 viprcli exportgroup commands 153
viprcli exportgroup add_cluster....154
viprcli exportgroup add_host....155
viprcli exportgroup add_initiator....156
viprcli exportgroup add_vol....158
viprcli exportgroup create....160
viprcli exportgroup delete....163
viprcli exportgroup list....164
viprcli exportgroup path_adjustment_preview....167
viprcli exportgroup path_adjustment....168
viprcli exportgroup remove_cluster....171

viprcli exportgroup remove_host....172
viprcli exportgroup remove_initiator....173
viprcli exportgroup remove_vol....175
viprcli exportgroup show....176
viprcli exportgroup tag....179
viprcli exportgroup tasks....180
viprcli exportgroup changeportgroup....180

Chapter 18 viprcli filepolicy commands 183
viprcli filepolicy assign....184
viprcli filepolicy create....185
viprcli filepolicy delete....187
viprcli filepolicy list....188
viprcli filepolicy show....189
viprcli filepolicy unassign....190
viprcli filepolicy update....191

Chapter 19 viprcli filesystem commands 195
viprcli filesystem assign-policy....196
viprcli filesystem change-vpool....197
viprcli filesystem create....198
viprcli filesystem create-replication-copy....200
viprcli filesystem delete....201
viprcli filesystem delete-acl....202
viprcli filesystem expand....203
viprcli filesystem export....204
viprcli filesystem export-rule....206
viprcli filesystem failback-replication....207
viprcli filesystem failover-replication....208
viprcli filesystem list....209
viprcli filesystem list-acl....210
viprcli filesystem list-policy....211
viprcli filesystem mount....212
viprcli filesystem mountlist....213
viprcli filesystem nfs-acl....214
viprcli filesystem nfs-delete-acl....216
viprcli filesystem nfs-list-acl....216
viprcli filesystem pause-replication....217
viprcli filesystem refresh-replication-copy....218
viprcli filesystem remove-replication-copy....219
viprcli filesystem resume-replication....220
viprcli filesystem schedule-snapshots-list....221
viprcli filesystem share-acl....222
viprcli filesystem show....223
viprcli filesystem show-exports....226
viprcli filesystem show-shares....227
viprcli filesystem start-replication....227
viprcli filesystem stop-replication....228
viprcli filesystem tag....229
viprcli filesystem tasks....230
viprcli filesystem unexport....231
viprcli filesystem unmanaged ingest....233
viprcli filesystem unmanaged show....233
viprcli filesystem unmount....234

viprcli filesystem update....235
viprcli filesystem unassign-policy....236

Chapter 20 viprcli host commands 239
viprcli host compute-host-os-install....240
viprcli host create....241
viprcli host delete....243
viprcli host detach....244
viprcli host discover....245
viprcli host discover-array-affinity....246
viprcli host list....246
viprcli host list-initiators....247
viprcli host list-ipinterfaces....248
viprcli host list-umexportmasks....249
viprcli host list-umvolumes....250
viprcli host provision-bare-metal-host....250
viprcli host show....251
viprcli host tasks....253
viprcli host update....254

Chapter 21 viprcli initiator commands 257
viprcli initiator aliasget....258
viprcli initiator aliasset....258
viprcli initiator create....260
viprcli initiator delete....261
viprcli initiator list....262
viprcli initiator show....263
viprcli initiator tasks....264
viprcli initiator update....265

Chapter 22 viprcli ipinterface commands 267
viprcli ipinterface create....268
viprcli ipinterface delete....269
viprcli ipinterface list....270
viprcli ipinterface show....271
viprcli ipinterface tasks....272
viprcli ipinterface update....273

Chapter 23 viprcli ipsec commands 275
viprcli ipsec change-status....276
viprcli ipsec rotate-key....276
viprcli ipsec status....277

Chapter 24 viprcli logout command 279
viprcli logout....280

Chapter 25 viprcli meter commands 281
viprcli meter....282

Chapter 26 viprcli monitor commands 285
viprcli monitor....286

Chapter 27 viprcli network commands 291
viprcli network create....292
viprcli network list....293
viprcli network show....294
viprcli network update....296
viprcli network assign....297
viprcli network delete....298
viprcli network endpoint add....298
viprcli network endpoint remove....300
viprcli network register....301
viprcli network deregister....302

Chapter 28 viprcli networksystem commands 303
viprcli networksystem aliases add....304
viprcli networksystem aliases remove....304
viprcli networksystem aliases show....305
viprcli networksystem aliases update....306
viprcli networksystem create....307
viprcli networksystem discover....308
viprcli networksystem list....310
viprcli networksystem list-connections....311
viprcli networksystem show....312
viprcli networksystem delete....313
viprcli networksystem register....314
viprcli networksystem deregister....315

Chapter 29 viprcli objectuser commands 317
viprcli objectuser create_secretkey....318

Chapter 30 viprcli order commands 319
viprcli order list....320
viprcli order show....320
viprcli order show-execution....323

Chapter 31 viprcli project commands 327
viprcli project create....328
viprcli project list....329
viprcli project show....330
viprcli project update....331
viprcli project delete....332
viprcli project get-acl....333
viprcli project update-acl....334
viprcli project tag....335

Chapter 32 viprcli protectionsystem commands 337
viprcli protectionsystem create....338
viprcli protectionsystem discover....339
viprcli protectionsystem list....340
viprcli protectionsystem show....341
viprcli protectionsystem update....343
viprcli protectionsystem connectivity....345
viprcli protectionsystem delete....346

Chapter 33 viprcli quotadirectory commands 349
viprcli quotadirectory create 350
viprcli quotadirectory delete 351
viprcli quotadirectory list 352
viprcli quotadirectory show 353
viprcli quotadirectory update 354

Chapter 34 viprcli sanfabrics commands 357
viprcli sanfabrics activate 358
viprcli sanfabrics get-sanzone 358
viprcli sanfabrics list 359
viprcli sanfabrics list-sanzones 360
viprcli sanfabrics show 360
viprcli sanfabrics update 361

Chapter 35 viprcli schedule-policy commands 363
viprcli schedule-policy create 364
viprcli schedule-policy delete 365
viprcli schedule-policy list 366
viprcli schedule-policy show 367
viprcli schedule-policy update 367

Chapter 36 viprcli scheduled_event commands 371
viprcli scheduled_event create 372
viprcli scheduled_event get 374
viprcli scheduled_event cancel 375
viprcli scheduled_event delete 375
viprcli scheduled_event update 375

Chapter 37 viprcli snapshot commands 379
viprcli snapshot activate 380
viprcli snapshot create 381
viprcli snapshot delete 384
viprcli snapshot delete-acl 385
viprcli snapshot export-file 386
viprcli snapshot export-rule 388
viprcli snapshot import-to-vplex 390
viprcli snapshot list 391
viprcli snapshot list-acl 393
viprcli snapshot restore 394
viprcli snapshot resync 396
viprcli snapshot share-acl 397
viprcli snapshot show 398
viprcli snapshot show-exports 400
viprcli snapshot show-shares 401
viprcli snapshot tasks 402
viprcli snapshot tag 404
viprcli snapshot unexport-file 405
viprcli snapshot expand 406

Chapter 38 viprcli snapshotsession commands 409
viprcli snapshotsession create 410

viprcli snapshotsession deactivate....411
viprcli snapshotsession linktarget....412
viprcli snapshotsession list....413
viprcli snapshotsession relinktargets....414
viprcli snapshotsession restore....415
viprcli snapshotsession show....416
viprcli snapshotsession unlinktargets....418

Chapter 39 viprcli storagepool commands 421
viprcli storagepool delete....422
viprcli storagepool deregister....423
viprcli storagepool list....424
viprcli storagepool register....425
viprcli storagepool show....426
viprcli storagepool update....428

Chapter 40 viprcli storageport commands 431
viprcli storageport create....432
viprcli storageport delete....433
viprcli storageport deregister....434
viprcli storageport list....435
viprcli storageport register....436
viprcli storageport show....438
viprcli storageport update....440

Chapter 41 viprcli storageportgroup commands 443
viprcli storageportgroup create....444
viprcli storageportgroup delete....445
viprcli storageportgroup deregister....446
viprcli storageportgroup list....447
viprcli storageportgroup register....448
viprcli storageportgroup show....449

Chapter 42 viprcli storageprovider commands 451
viprcli storageprovider list....452
viprcli storageprovider create....452
viprcli storageprovider delete....454
viprcli storageprovider scan....455
viprcli storageprovider show....455
viprcli storageprovider update....456

Chapter 43 viprcli storagesystem commands 459
viprcli storagesystem connectivity....460
viprcli storagesystem create....461
viprcli storagesystem delete....463
viprcli storagesystem deregister....464
viprcli storagesystem discover....465
viprcli storagesystem discover_arrayaffinity....466
viprcli storagesystem discover_unmanagedfilesystems....467
viprcli storagesystem discover_unmanagedvolumes....468
viprcli storagesystem get_unmanagedfilesystems....469
viprcli storagesystem get_unmanagedvolumes....470

viprcli storagesystem list..... 472
viprcli storagesystem register.....475
viprcli storagesystem show.....476
viprcli storagesystem show-unmanagedexportmask.....478
viprcli storagesystem update.....479

Chapter 44 viprcli system commands 481
viprcli system add-license.....483
viprcli system add-site.....483
viprcli system cluster-ipinfo.....484
viprcli system cluster-ipreconfig.....485
viprcli system cluster-poweroff..... 487
viprcli system cluster-recovery.....488
viprcli system cluster-recovery-status..... 488
viprcli system connectemc-ftps..... 489
viprcli system connectemc-smtp.....490
viprcli system create-backup..... 491
viprcli system db-consistency-check.....492
viprcli system db-consistency-check-cancel..... 492
viprcli system db-consistency-check-status.....493
viprcli system dbrepair-status..... 493
viprcli system delete-backup..... 494
viprcli system delete-site.....495
viprcli system delete-sites.....495
viprcli system delete-task.....496
viprcli system disable-update-check.....496
viprcli system download-backup.....497
viprcli system failover-site.....498
viprcli system get-alerts..... 498
viprcli system get-cluster-state..... 500
viprcli system get-diagnostics.....501
viprcli system get-esrsconfig..... 505
viprcli system get-health.....505
viprcli system get-license.....506
viprcli system get-log-level.....507
viprcli system get-logs..... 512
viprcli system get-properties..... 514
viprcli system get-properties-metadata..... 525
viprcli system get-stats..... 527
viprcli system get-storage.....530
viprcli system get-target-version..... 531
viprcli system install-image.....532
viprcli system ipreconfig-status.....533
viprcli system list-backup..... 534
viprcli system list-external-backup..... 534
viprcli system list-sites..... 536
viprcli system pause-site..... 537
viprcli system pause-sites.....538
viprcli system pull-backup..... 538
viprcli system pull-backup-cancel.....539
viprcli system query-backup.....540
viprcli system query-backup-info..... 540
viprcli system reboot-node.....541
viprcli system remove-image..... 542
viprcli system reset-properties..... 543

viprcli system restart-service 544
viprcli system restore-backup 545
viprcli system restore-backup-status 546
viprcli system resume-site 547
viprcli system retry-site 547
viprcli system send-alert 548
viprcli system send-heartbeat 550
viprcli system send-registration 550
viprcli system set-log-level 551
viprcli system set-properties 552
viprcli system show-site 555
viprcli system site-error 556
viprcli system site-time 557
viprcli system skip-setup 557
viprcli system switchover-site 558
viprcli system update-cluster 559
viprcli system update-site 560
viprcli system upload 561
viprcli system upload-backup 561
viprcli system upload-backup-status 562

Chapter 45 viprcli task commands 565
viprcli task 566

Chapter 46 viprcli tenant commands 569
viprcli tenant add-attribute 570
viprcli tenant add-group 570
viprcli tenant add-namespace 571
viprcli tenant add-role 572
viprcli tenant create 573
viprcli tenant delete-role 575
viprcli tenant delete 576
viprcli tenant get-clusters 577
viprcli tenant get-hosts 578
viprcli tenant get-role 578
viprcli tenant get-vcenters 579
viprcli tenant list 580
viprcli tenant list-object-namespaces 581
viprcli tenant remove-attribute 582
viprcli tenant show 582
viprcli tenant show-object-namespaces 583
viprcli tenant update-quota 584

Chapter 47 viprcli varray commands 587
viprcli varray create 588
viprcli varray get-acl 588
viprcli varray list 589
viprcli varray show 591
viprcli varray allow 592
viprcli varray disallow 593
viprcli varray update 594
viprcli varray list-storage-ports 594
viprcli varray list-storage-portsgroup 595
viprcli varray delete 596

Chapter 48 viprcli vcenter commands 599
viprcli vcenter create 600
viprcli vcenter delete 601
viprcli vcenter discover 602
viprcli vcenter get-clusters 603
viprcli vcenter get-datacenters 603
viprcli vcenter get-hosts 604
viprcli vcenter list 605
viprcli vcenter show 605
viprcli vcenter tasks 607
viprcli vcenter update 608

Chapter 49 viprcli vcenterdatacenter commands 611
viprcli vcenterdatacenter create 612
viprcli vcenterdatacenter create-cluster 612
viprcli vcenterdatacenter delete 613
viprcli vcenterdatacenter get-clusters 614
viprcli vcenterdatacenter get-hosts 614
viprcli vcenterdatacenter list 615
viprcli vcenterdatacenter show 616
viprcli vcenterdatacenter update-cluster 616

Chapter 50 viprcli vdc commands 619
viprcli vdc add 620
viprcli vdc delete 620
viprcli vdc disconnect 621
viprcli vdc list 621
viprcli vdc reconnect 622
viprcli vdc show 622
viprcli vdc update 623

Chapter 51 viprcli vnasserver commands 625
viprcli vnasserver assign 626
viprcli vnasserver list 626
viprcli vnasserver show 627
viprcli vnasserver unassign 628

Chapter 52 viprcli volume commands 631
viprcli volume clone 632
viprcli volume clone-activate 633
viprcli volume clone-deactivate 634
viprcli volume clone-checkprogress 635
viprcli volume clone-detach 637
viprcli volume clone-list 638
viprcli volume clone-restore 639
viprcli volume clone-resync 640
viprcli volume clone-show 641
viprcli volume continuous_copies copy 644
viprcli volume continuous_copies delete 645
viprcli volume continuous_copies establish 646
viprcli volume continuous_copies failover 647
viprcli volume continuous_copies failover-test 648
viprcli volume continuous_copies failover-test-cancel 649

viprcli volume continuous_copies list....650
viprcli volume continuous_copies pause....651
viprcli volume continuous_copies restore....653
viprcli volume continuous_copies resume....654
viprcli volume continuous_copies show....656
viprcli volume continuous_copies start....658
viprcli volume continuous_copies stop....660
viprcli volume continuous_copies swap....662
viprcli volume continuous_copies update-access-mode....663
viprcli volume create....664
viprcli volume delete....667
viprcli volume expand....668
viprcli volume list....669
viprcli volume migration-cancel....671
viprcli volume migration-deactivate....672
viprcli volume migration-list....674
viprcli volume migration-pause....675
viprcli volume migration-resume....676
viprcli volume migration-show....678
viprcli volume protectionset show....679
viprcli volume show....679
viprcli volume tag....681
viprcli volume tasks....682
viprcli volume unmanaged ingest....684
viprcli volume unmanaged show....685
viprcli volume update....685

Chapter 53 viprcli vpool commands 687
viprcli vpool create....688
viprcli vpool list....694
viprcli vpool show....696
viprcli vpool update....698
viprcli vpool delete....702
viprcli vpool allow....703
viprcli vpool disallow....704
viprcli vpool add_pools....705
viprcli vpool refresh_pools....706
viprcli vpool get_pools....707
viprcli vpool remove_pools....708

TABLES

1 Revision history ... 3
2 Descriptions of command tokens ... 22
3 Help examples ... 23
4 List of Common Parameters ... 24
5 Authentication provider configuration file fields ... 39
6 Description of command output ... 345

CHAPTER 1

ViPR Controller Command Line Interface Overview

This chapter contains the following topics:

- Welcome to the ViPR Controller Command Line Interface Reference 20
- Deploy the ViPR Controller CLI 20
- viprcli command format 21
- Common Parameters 24
- Authenticating with viprcli 24

Welcome to the ViPR Controller Command Line Interface Reference

The EMC® ViPR® Command Line Interface (CLI) allows data center personnel to use ViPR Controller to manage storage resources.

Note

Throughout this document, Virtual Storage Pools is known as virtual pools and Virtual Storage Arrays is known as virtual arrays.

Deploy the ViPR Controller CLI

For instructions on installing the ViPR Controller CLI on Linux and Windows, and also uninstalling or upgrading the CLI, refer to the EMC ViPR Controller Installation, Upgrade, and Maintenance Guide, which is available in the ViPR Controller Product Documentation Index.

Using viprcli.profile to simplify issuing viprcli commands

Use the `viprcli.profile` (Linux) or `viprcli.profile.bat` (Windows) commands to avoid appending `-hostname xx.xx.xx.xx` every time you issue a viprcli command.

Specify a ViPR Controller instance in the viprcli.profile (Linux) or viprcli.profile.bat (Windows) file so that any viprcli command is issued against that ViPR Controller instance.

Procedure

1. Change directory to `/opt/storageos/cli` or to the directory where the CLI is installed.
2. Edit the `viprcli.profile` or `viprcli.profile. bat` files.

   a. For Linux, edit the `viprcli.profile` file using the `vi` command and set the *VIPR_HOSTNAME* to the ViPR Controller public virtual IP address and *VIPR_PORT*=`4443` environment variable. Save the file. For example,

```
cd /opt/storageos/cli
vi viprcli.profile
#!/usr/bin/sh


# Installation directory of ViPR Controller CLI
ViPR Controller_CLI_INSTALL_DIR=/opt/storageos/cli

# Add the ViPR Controller install directory to the PATH and
PYTHONPATH env variables
if [ -n $ViPR Controller_CLI_INSTALL_DIR ]
then
        export PATH=$ViPR Controller_CLI_INSTALL_DIR/bin:
$PATH
        export PYTHONPATH=$ViPR Controller_CLI_INSTALL_DIR/
bin:$PYTHONPATH
fi
```

```
# USER CONFIGURABLE ViPR Controller VARIABLES

# ViPR Controller Host fully qualified domain name
ViPR Controller_HOSTNAME=example.mydomain.com

# ViPR Controller Port Number
ViPR Controller_PORT=4443

:wq
```

b. For Windows, edit `viprcli.profile.bat` and set the following variables:

Variable	Value
SET *VIPR_HOSTNAME*	Set to fully qualified. It is the ViPR Controller hostname.
Domain name	Set to the fully qualified domain name (FQDN) or virtual IP address of the ViPR Controller host or configuration.
SET *VIPR PORT*	The ViPR Controller port. Default value is 4443.

3. Set the path environment variable for the ViPR Controller executable.

   a. For Linux, run the source command to set the path environment variable:

   `source ./viprcli.profile`

   b. For Windows, change directories to the location where viprcli was installed and run `viprcli.profile.bat`.

   The default path is: `C:\EMC\ViPR\cli`.

After you finish

For either the Linux or Windows environment, authenticate (log in to) the ViPR Controller instance with the viprcli to confirm that the installation was successful. See Authenticating with viprcli.

Note

On ViPR Controller nodes, the `viprcli.profile` and `viprcli.profile.bat` files are not persisted across node restart. If the `viprcli.profile` or `viprcli.profile.bat` files are not used or, if left to default, the parameter, `-hostname`, must be specified for any viprcli command as the last parameter.

---

viprcli command format

viprcli commands follow a consistent format viprcli <resource> <operation> <-parameter> <value> [<-parameter>...] [<value>...].

Every command must start with the keyword viprcli. For example, `viprcli vpool create -h`.

Table 2 Descriptions of command tokens

Token	Description
viprcli	The name of the CLI executable.
resource	The storage resource on which the operation is performed. For example, host, filesystem, network, networksystem and so on are all objects. For example, in `viprcli vpool create -h` command, 'vpool' is the resource name.
operation	Some objects support all of these operations; others support only a subset of them. Valid operations for each object are listed in the help. For example, create, list, show, delete, update and so on are all operations. For example, in `viprcli vpool create -h` command, 'create' is the operation.
parameters or arguments	Following the above three terms every CLI command has a list of parameters. Most of the command set include both mandatory and optional parameters, with the exception of few. For example, name, hostname, type, port, ssl, cookiefile, and so on are parameters. • Parameters are either mandatory or optional. If a mandatory parameter is not specified, the command returns a contextual error message. In some commands, one or more optional parameters are required following the default mandatory parameters. For example, in `viprcli project create -n <projectname> -tn <tenantname>` command -n is mandatory and -tn is optional. • Parameters can be specified in any order. For example, in `viprcli project create -n <projectname> -tn <tenantname>` is same as `viprcli project create -tn <tenantname> -n <projectname>`. • Parameters have both long and short forms and can be specified either in a long or short forms. For example, in `viprcli project create -n <projectname> -tn <tenantname>` is same as `viprcli project create -name <projectname> -tenant <tenantname>`. • Parameters require a value to be specified. Some parameters may have a

Table 2 Descriptions of command tokens (continued)

Token	Description
	predefined set of values. For example, projectname and tenantname are values. **Note** The CLI help (`viprcli -h`) available from the command prompt refers to the parameters as arguments. They both mean the same.
value	Following every parameter you must specify a value. In some cases the values are from a predefined set. For example type file or type block; provisiontype Thin or provisiontype Thick. Occasionally, there could be parameters which don't require any values to be specified. **Note** Values are case-sensitive and must be used as specified in the command help and in this document. Include values containing spaces within double quotes. For example, -description "My ViPR Pool".

CLI help

Help is available on all commands by appending the -h switch to the command name, from the command prompt of both `Windows` and `Linux` hosts.

Example 1

Table 3 Help examples

Help examples	Description
viprcli -h	Entire viprcli command set is displayed.
viprcli authentication -h	A complete list of authentication command set is displayed.
viprcli vpool create -h	Only the help on vpool create command is displayed.

Common Parameters

All viprcli commands take these optional parameters in the following table.

Table 4 List of Common Parameters

Parameters	Description
hostname|hn	The Fully Qualified Domain Name (FQDN), IPv4 or IPv6 address of a ViPR Controller host. This setting overrides the *ViPR_HOSTNAME* environment variable. This is an optional parameter. For IPv6 appliances, the IPv6 address must be entered using square brackets. For example, [2520:0:140:280f::106]. **Note** 'hn' is not the short form for 'hostname' in the `viprcli host` commands.
port|po	The port of the ViPR Controller specified in the hostname argument. The default ViPR Controller port is 443. It is an optional parameter.
portui|pu	The https port of the ViPR Controller Portal UI. The default ViPR Controller port is 443. It is an optional parameter.
cookiefile|cf	A valid cookie file as returned by the viprcli authenticate operation. It is an optional parameter.
version|v	Version number of the ViPR Controller CLI program.
help|h	Show a help page for the command operation.

Authenticating with viprcli

You should authenticate a user before any viprcli commands can be successfully executed.

Before you can authenticate, you must have configured the environment variable with the path to the Python installation directory. If you did not do set the environment variable before installing the CLI, you must do it now to use the ViPR Controller CLI.

Logging in to the ViPR Controller command line interface is different on `Windows` and `Linux` hosts.

When logging in to the ViPR Controller, if you do not type a host when authenticating, you will automatically log in to the ViPR Controller you provided during installation. If you want to log into a different ViPR instance, you can type the hostname as shown.

Authenticate on Windows

To log in to the default ViPR Controller instance use:

```
C:/>
          viprcli authenticate -u root -d c:\tmp
```

To specify the ViPR Controller instance use:

```
C:/>
          viprcli -hostname <fqdn, or host ip> authenticate  -u
root -d c:\tmp
```

Do not end the directory path with a '\'. For example, `c:\tmp\`

Type the password when prompted.

Authenticate on Linux

To log into the default ViPR Controller instance use:

```
#
          viprcli authenticate -u root -d /tmp
```

To specify the ViPR Controller instance use:

```
#
          viprcli -hostname <fqdn, or host ip> authenticate -u root
-d /tmp
```

Type the password when prompted.

Note

The non-root users must have read, write, and run permissions to use the CLI installed by root. However, they do not need all these permissions for installing and running the CLI in their home directory.

CHAPTER 2

viprcli approval commands

This chapter contains the following topics:

- viprcli approval approve 28
- viprcli approval list 28
- viprcli approval reject 29
- viprcli approval show 30

viprcli approval approve

Approve request for a service catalog order.

Syntax

```
viprcli approval approve
-uri|id <uri>
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli approval approve` command approves the request for a service catalog order. This command is silent on success. Only a TENANT_APPROVER role can approve a request.

Note

There is an option called "Approval Required" in the Service catalog. This option can be set only through the ViPR GUI.

Options

-uri|id

Approval request URI or Id. This is a mandatory parameter. The Id can be known using the `viprcli approval list` command.

common arguments

This operation also takes the Common Parameters on page 24.

Example 2

```
# viprcli approval approve -id  urn:storageos:ApprovalRequest:
511e53ba-4e20-4dd3-9357-e8d539d1e7f9:
```

viprcli approval list

List the pending approval requests.

Syntax

```
viprcli approval list
[-pending <pending>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
```

```
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-h]
```

Description

`viprcli approval list` command lists all the approved, pending, and rejected approval requests with the URI and request ID. Only a TENANT_APPROVER role can approve a request.

Note

There is an option called "Approval Required" in the Service catalog. This option can be set only through the ViPR GUI.

Options

-pending

This option is used to list all pending approval requests. This parameter doesn't require you to supply any input. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 3

```
# viprcli approval list -pending
HREF
              ID
/api/approvals/urn:storageos:ApprovalRequest:
513e74ba-7e20-4dd3-9287-e8d539d1e7f9: urn:storageos:ApprovalRequest:
513e74ba-7e20-4dd3-9287-e8d539d1e7f9:
```

viprcli approval reject

Reject the approval request.

Syntax

```
            viprcli approval reject
            -uri|id <uri>
            [-hostname|hn <hostname>]
            [-cookiefile|cf <cookie_file>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-h]
```

Description

`viprcli approval reject` command rejects the approval request. This command is silent on success. Only a TENANT_APPROVER role can approve a request.

Options

-uri|id

Approval request URI or Id. This is a mandatory parameter. The Id can be known using the `viprcli approval list` command.

common arguments

This operation also takes the Common Parameters on page 24.

Example 4

```
# viprcli approval reject -id urn:storageos:ApprovalRequest:
96766f3a-9753-4c04-a874-547d098445a2:
```

viprcli approval show

Return the approval request details for the specified Service Catalog order.

Syntax

```
viprcli approval show
[-uri|id <uri>]
[-xml <xml>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli approval show` command shows the approval request details for the specified Service Catalog order.

Options

-uri|id

Approval request URI or Id. This is a mandatory parameter. The Id can be known using the `viprcli approval list` command.

-xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 5

Example 5 (continued)

```
# viprcli approval show -id  urn:storageos:ApprovalRequest:
513e74ba-7e20-4dd3-9287-e8d539d1e7f9:
```

CHAPTER 3

viprcli assetoptions commands

This chapter contains the following topics:

- viprcli assetoptions list 34

viprcli assetoptions list

List the URN information of all ViPR Controller entities.

Syntax

```
viprcli assetoptions list
-type|t <type>
[-params <namevalue>[<namevalue>...]]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli assetoptions list` command lists the URN information of all ViPR Controller entities.

Options

-type|t

Virtual pool type to list. Supply the input enclosed within quote marks. This is a mandatory parameter.

-param

List of name-value pairs to retrieve the asset options, each separated by commas. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 6

```
# viprcli assetoptions list -t "vipr.blockVirtualPool"
  KEY
VALUE
  urn:storageos:VirtualPool:3696644a-4431-4757-9ab4-dec694bd5023:
basepool
  urn:storageos:VirtualPool:4d47f8da-abd4-44a9-802e-9d7655ea1eaf:
srdfpool2
  urn:storageos:VirtualPool:712d587d-f1f8-4473-bfc5-19df40904a42:
srdfpool8
  urn:storageos:VirtualPool:6d26c025-2763-452a-4071-f3f0011309db:
suvpool
  urn:storageos:VirtualPool:a4a87fdb-e865-4d31-a566-e9f6851e746e:
srdfpool
  urn:storageos:VirtualPool:d1ace54b-a6f2-47b8-be6e-710b17a2b06d:
srdfpool1
  urn:storageos:VirtualPool:d2c25d36-d611-43cc-a701-6306a15fef6c:
source1
```

CHAPTER 4

viprcli authenticate commands

This chapter contains the following topics:

- viprcli authenticate 36

viprcli authenticate

Authenticate with the ViPR CLI.

Syntax

```
viprcli authenticate
[-username|u <username>]
[-cookiedir|d <cookiedir>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli authenticate` command authenticates you with the ViPR CLI.

Options

username|u

The username to login with. This is a mandatory parameter.

cookiedir|d

The cookie directory used to store cookie files. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 5

viprcli authentication commands

This chapter contains the following topics:

- Authentication Provider 38
- viprcli authentication add-provider 44
- viprcli authentication list-providers 45
- viprcli authentication show-provider 46
- viprcli authentication update 47
- viprcli authentication delete-provider 48
- Active Directory integration for ViPR Controller 48
- viprcli authentication add-vdc-role 49
- viprcli authentication list-vdc-role 50
- viprcli authentication delete-role 50
- Assigning a role to an active directory user 51
- viprcli authentication add-user-group 52
- viprcli authentication user-group-add-attribute 53
- viprcli authentication user-group-add-values 53
- viprcli authentication user-group-remove-attribute 54
- viprcli authentication user-group-remove-values 55
- viprcli authentication show-user-group 56
- viprcli authentication delete-user-group 56
- viprcli authentication list-user-groups 57

Authentication Provider

Authentication Providers are required to integrate ViPR Controller with either active directory or LDAP.

User authentication in ViPR Controller is done through an authentication provider. Except for the special built-in administrative users (root, sysmonitor, svcuser, and proxyuser) there are no local users in ViPR Controller. Users who can log in, and who are assigned roles or ACLs, must be found on an authentication provider added to ViPR Controller.

To implement an authentication provider in ViPR Controller, you must build a configuration file that describes an authentication provider, then follow the steps in the article Active Directory integration for ViPR Controller on page 48.

You can build a description of an authentication provider with a standard text editor. The following shows an example configuration file.

Note

The configuration file must include all of the attributes in the following example. However, if you are using LDAP, provide correct values for `groupofobjectclasses` and `groupmemberattributes`. If you are using Active Directory, the value can be blank for these two attributes.

```
[section one]
#The below skeleton of the file should not be altered.
#however if you want to set the value of a
#particular field to empty string, you can do so.
mode:ldap
url:ldap://192.168.0.0
add-urls:
remove-urls:
managerdn:CN=Manager,DC=maxabc,DC=com
searchbase:OU=ViPR,DC=maxabc,DC=com
searchfilter:userPrincipalName=%u
groupattr:CN
name:maxabc.com
domains:maxabc.com
remove-domains:
add-domains:
whitelist:
disable:false
add-whitelist:
remove-whitelist:
searchscope:ONELEVEL
maxpagesize:
validatecertificate:true
description:"maxabc.com"
groupobjectclasses:groupOfNames,groupOfUniqueNames,posixGroup,organi
zationalRole
groupmemberattributes:member,uniqueMember,memberUid,roleOccupant
add-groupobjectclassnames:groupOfNames, posixGroup,
groupOfUniqueNames, organizationalRole
remove-groupobjectclassnames:
add-groupmemberattributetypenames:memberUid, member, roleOccupant,
uniqueMember
remove-groupmemberattributetypenames:

force-groupattributeupdate:false
```

```
##########################################

[section two]
mode:ad
description:multi-domain
searchscope:SUBTREE
validatecertificate:false
maxpagesize:4096000
disable:
url:ldap://<servername>.mycompany.com:3268
certificate:system_test_cert_3
passwd_user:Dangerous1
managerdn:CN=manager_bind,OU=Test1,OU=Test,DC=secqe,DC=com
searchbase:DC=secqe,DC=com
searchfilter:userPrincipalName=%u
groupattr:CN
name:multi-domain
domains:secqe.com,domain2.secqe.com,domain3.secqe.com
whitelist:*
groupobjectclasses:
groupmemberattributes:
```

Table 5 Authentication provider configuration file fields

Field	Description
name	The name of the authentication provider. You can have multiple providers for different domains.
mode	Active Directory or LDAP. In Provider.cfg (CLI), use ad or ldap.
url	The Active Directory or LDAP server IP address. ldap or ldaps (secure LDAP) with the domain controller IP address. Default port for ldap is 389. Usage: ldap://<Domain controller IP >:<port> (if not default port). If the authentication provider supports a multidomain forest, use the global catalog server IP and always specify the port number. Default is 3268 for ldap. Usage: ldap://<Global catalog server IP>:<port>
passwd_user	The password of the manager user. The example shows administrator (managerdn:CN=Administrator).Usually it will be an active directory user who has fewer privileges than Administrator, but has sufficient privileges to query Active Directory for users attributes and group Information. **Note** You must update this value in ViPR Controller if the managerdn credentials change in Active Directory.

Table 5 Authentication provider configuration file fields (continued)

Field	Description
managerdn	Indicates the Active Directory Bind user account that ViPR Controller uses to connect to Active Directory or LDAP server. This account is used to search Active Directory. For example, when a ViPR Controller administrator specifies a user for role assignment. This user must have **Read all inetOrgPerson information** in Active Directory. The `inetOrgPerson` object class is used in several non-Microsoft, Lightweight Directory Access Protocol (LDAP) and X.500 directory services to represent people in an organization. To set this privilege in Active Directory: 1. Open **Active Directory Users and Computers**. 2. Right-click on the domain, and select **Delegate Control...**. 3. Click **Next**. 4. Select the user that you are using for managerdn 5. Click **Next**. The required permission is on the next screen **Read all inetOrgPerson information**. Example: CN=Administrator,CN=Users,DC=sanity,DC=local In this example, the Active Directory Bind user is Administrator, in the Users tree of the sanity.local domain. Usually managerdn is a user who has fewer privileges than Administrator, but has sufficient privileges to query Active Directory for users attributes and group information. **Note** You must update this value in ViPR Controller if the managerdn credentials change in Active Directory.
searchbase	Indicates the Base Distinguished Name that ViPR Controller uses to search for users at login time and when assigning roles or setting ACLs. Example: CN=Users,DC=sanity,DC=com

Table 5 Authentication provider configuration file fields (continued)

Field	Description
	In this example, all users in the Users container will be searched for. Example: CN=Users,OU=myGroup,DC=sanity,DC=com In this example, all users in the Users container in the myGroup organization unit will be searched for. **Note** The structure of the searchbase value begins with the "leaf" level and goes up to the domain controller level--the reverse of the structure seen in the Active Directory Users and Computers UI.
searchfilter	Indicates the string used to select subsets of users. Example: userPrincipalName=%u **Note** ViPR Controller does not validate this value when you add the authentication provider.
The searchbase and searchfilter parameters are used to pinpoint the Active Directory / LDAP search starting point. The values of these fields are site-specific. The example above shows some typical values.	
groupattr	Indicates the Active Directory attribute that is used to identify a group name. Used for searching the directory by groups. Example: CN Active Directory only. Does not apply to other authentication providers. **Note** Once set this value cannot be changed for a provider, because the tenants that are using this provider may already have role assignments and permissions configured using group names in this current format.
name	The name of the provider. You can have multiple providers for difference domains.
domains	Active Directory and LDAP allow administrators to organize objects of a network (such as users, computers, and devices) into a hierarchical collection of containers known as the logical structure. The

Table 5 Authentication provider configuration file fields (continued)

Field	Description

	top-level logical container in this hierarchy is the forest. Within a forest are domain containers, and within domains are organizational units.
	
	Domains are container objects. Domains are a collection of administratively defined objects that share a common directory database, security policies, and trust relationships with other domains. In this way, each domain is an administrative boundary for objects. A single domain can span multiple physical locations or sites and can contain millions of objects. It is also possible to have multi-domain configuration.
	
	A single domain entry in this field of the authentication provider would look like this:
	
	`mydomain.com`
	
	A multi-domain entry in this field of the authentication provider would look like this:
	
	`mydomain.com,cli.vipr,sanity.local`

whitelist	ViPR Controller supports an optional whitelist capability. One or more group names as defined by the authentication provider. This setting will filter the group membership information that ViPR Controller retrieves about a user. The example shows how to allow folks listed in the Admins* and *Test* groups to authenticate. • When a group or groups are included in the whitelist, it means that ViPR Controller will be aware of a user's membership in the specified group[s] only. Multiple values (one per line in ViPR Controller UI, comma-separated in CLI and API) and wildcards (for example MyGroup*,TopAdminUsers*) are allowed. • Blank value (default) means that ViPR Controller will be aware of any and all groups that a user belongs to. Asterisk (*) is the same as blank. Example:UserA belongs to Group1 and Group2.

Table 5 Authentication provider configuration file fields (continued)

Field	Description
	If the whitelist is blank, ViPR Controller knows that UserA is a member of Group1 and Group2. If the whitelist is "Group1", ViPR Controller knows that UserA is a member of Group1, but does not know that UserA is a member of Group2 (or of any other group). Use care when adding a whitelist value. For example, if mapping a user to a tenant is based on group membership, then ViPR Controller must be aware of the user's membership in the group. To restrict access to a tenant to users of certain group(s) only, one must: • add these group(s) to the tenant user mapping (using the CLI command viprcli tenant add-group), so the tenant is configured to accept only users of these group(s). • add these group(s) to the white list, so that ViPR Controller is authorized to receive information about them. Active Directory only. Does not apply to other authentication providers. **Note** By default, if no groups are added to the tenant user mapping, users from any groups are accepted, regardless of the white list configuration.
disable	Set it to `false`, if you want to add the server to ViPR Controller but not immediately use it for authentication.
searchscope	The scope of search within the Active Directory/LDAP. It could be ONELEVEL, meaning the scope is limited to one level below or SUBTREE, meaning the scope is extended to the sub-tree levels.
maxpagesize	Value that controls the maximum number of objects returned in a single search result. This is independent of size of the each returned object.
validatecertificate	Validates the certificate on an LDAP environment.
description	Free text description of the authentication provider.

Table 5 Authentication provider configuration file fields (continued)

Field	Description
groupobjectclasses	Required for LDAP. All of the object classes that represent the group in the LDAP schema. You can remove the classes you know are not being used, or add custom classes that are being used in your LDAP schema. **Note** For Active Directory, the configuration file must have this attribute but left blank without any member attributes. If you do not include this line, then the addition of the authentication provider will fail.
groupmemberattributes	Required for LDAP. All the attributes of a group object that represents members in the LDAP schema. You can remove the attributes you know are not being used, or add custom attributes that are being used in the LDAP schema. **Note** For Active Directory, the configuration file must have this attribute but left blank without any member attributes. If you do not include this line, then the addition of the authentication provider will fail.

viprcli authentication add-provider

Add an authentication provider to ViPR Controller.

Syntax

```
viprcli authentication add-provider
-configfile <configfile>
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli authentication add-provider` command adds an authentication provider to ViPR Controller. This is done as part of Active Directory or LDAP integration. This operation requires a configuration file formatted with specific fields.

The article Authentication Provider on page 38 shows an example of an authentication provider configuration file. This command is silent on success.

Options

-configfile

Specifies the file containing the authentication provider. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 7

```
# ls ad*
ad_provider

# viprcli authentication add-provider -configfile ad_provider

# viprcli authentication list-providers
  NAME                  SERVER_URLS                  MODE
  ad configuration 2    ldap://192.168.1.2           ad
```

viprcli authentication list-providers

Display a list of authentication providers.

Syntax

```
viprcli authentication list-providers
[{-verbose|v | -long|l}]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli authentication list-providers` command displays a list of all authentication providers registered under the current tenant. You can have multiple authentication providers enabled for different domains.

Options

{verbose|v |long|l}

These options provide additional information. Use either verbose or long and not both. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 8

```
# viprcli authentication list-providers
  NAME                SERVER_URLS                    MODE
  ad configuration 2 ldap://192.168.1.2              ad
```

viprcli authentication show-provider

Display the extended information for an authentication provider.

Syntax

```
viprcli authentication show-provider
-name|n <name>
[-xml]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli authentication show-provider` command provides extended information about an authentication provider added in ViPR Controller. The information provided by this operation should be identical to the information in the authentication provider file used to call `viprcli authentication add-provider`.

Options

name|n

The name of an authentication provider. This is a mandatory parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 9

```
# viprcli authentication show-provider
-name "ad configuration8"
{
   "creation_time": 1378056235219,
   "description": "test",
   "disable": false,
   "domains": [
```

Example 9 (continued)

```
      "cli.vipr"
   ],
   "group_attribute": "CN",
   "group_whitelist_values": [
      "*Admins*",
      "*Test*"
   ],
   "id": "urn:storageos:AuthnProvider:
9025069e-d8fd-431c-83b7-d4a2887c7e3a:",
   "inactive": false,
   "link": {
      "href": "/vdc/admin/authnproviders/urn:
storageos:AuthnProvider:9025069e-d8fd-431c-83b7-d4a2887c7e3a:",
      "rel": "self"
   },
   "manager_dn": "CN=Administrator,CN=Users,DC=cli,DC=vipr",
   "max_page_size": 2,
   "mode": "ad",
   "name": "ad configuration8",
   "search_attribute_key": "userPrincipalName",
   "search_base": "CN=Users,DC=cli,DC=vipr",
   "search_filter": "userPrincipalName=%u",
   "search_scope": "ONELEVEL",
   "server_urls": [
      "ldap://192.168.66.66"
   ],
   "tags": [],
   "validate_certificates": false
}
```

viprcli authentication update

Update an authentication provider.

Syntax

```
viprcli authentication update
-configfile <configfile>
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli authentication update` command updates the specified authentication provider. This command is silent on success.

Options

-configfile

Specifies the file containing the authentication provider. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli authentication delete-provider

Delete the authentication provider.

Syntax

```
viprcli authentication delete-provider
-name|n <name>
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli authentication delete-provider` command deletes the specified authentication provider. This command is silent on success.

Options

name|n

The name of an authentication provider. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Active Directory integration for ViPR Controller

Before you begin

- The Active Directory (or LDAP) server must be running.
- The bind account that is used by ViPR Controller must be created in Active Directory and it must meet the requirements that are described under "managerdn" in Authentication Provider on page 38.

Procedure

1. Use the CLI to log in to one of the ViPR Controller nodes as a user with system administrator privileges.
2. Create the file `Provider.cfg` in the local directory on the host. The structure of this configuration file is shown in Authentication Provider on page 38.

```
[section one]
mode:ad
url:ldap://192.168.66.66
certificate:test_cert
managerdn:CN=Administrator,CN=Users,DC=mydomaincontroller,DC=c
om
searchbase:CN=Users,DC=mydomaincontroller,DC=com
searchfilter:userPrincipalName=%u
searchkey:userPrincipalName
```

```
groupattr:CN
name:ad configuration 2
domains:mydomaincontroller.com
whitelist:*Admins*,*Test*
```

3. Log in to ViPR Controller CLI using default root credentials.

   viprcli authenticate -u root -d /temp

4. Add the authentication provider - Provider.cfg - using the following command.

   viprcli authentication add-provider -configfile /tmp/Provider.cfg

5. Verify that the authentication provider has been successfully added.

```
# viprcli authentication list-providers

  NAME                      SERVER_URLS              MODE
  ad configuration 2        ldap://192.168.66.66      ad
```

6. Login to ViPR Controller CLI using Active Directory user credentials.

```
# viprcli authenticate -u  -d /test/temp
```

viprcli authentication add-vdc-role

Add a user role to the vdc group.

Syntax

```
            viprcli authentication add-vdc-role
            [-hostname|hn <hostname>]
            [-port|po <port_number>
            [-coookiefile|cf <cookiefile>
            -role <role> {SYSTEM_ADMIN | SECURITY_ADMIN |
SYSTEM_MONITOR | SYSTEM_AUDITOR}
            [{-subject-id|sb <subjectid> |group|g <group>}]
            [-h]
```

Description

`viprcli authentication add-vdc-role` command adds a user role to access the VDC. The command is silent on success.

Options

role

The user role to be added to the VDC. This must be one of SYSTEM_ADMIN, SECURITY_ADMIN, SYSTEM_MONITOR, or SYSTEM_AUDITOR. This is a mandatory parameter.

subject-id|sb|group|g

The subject ID or user group to be authenticated. The subject ID or user group must be in your AD/LDAP list.

common arguments

This operation also takes the Common Parameters on page 24.

Example 10

```
# viprcli authentication add-vdc-role -r SYSTEM_MONITOR -sb aduser
```

viprcli authentication list-vdc-role

Display the list of roles authenticated to the VDC.

Syntax

```
viprcli authentication list-vdc-role
[-hostname|hn <hostname>]
[-port|po <port_number>
[-portui|pu <ui_port_number>]
[-cooookiefile|cf <cookiefile>
[-h]
```

Description

`viprcli authentication list-vdc-role` command displays the list of roles added to access the VDC.

Options

common arguments

This operation also takes the Common Parameters on page 24.

Example 11

```
# viprcli authentication list-vdc-role
{
   "link": {
      "href": "/vdc/role-assignments",
      "rel": "self"
   },
   "role_assignments": []
}
```

viprcli authentication delete-role

Delete a user role from the VDC group.

Syntax

```
viprcli authentication delete-role
[-hostname|hn <hostname>]
```

```
            [-port|po <port_number>
            [-portui|pu <ui_port_number>]
            [-coookiefile|cf <cookiefile>
            -role <role> {SYSTEM_ADMIN | SECURITY_ADMIN |
SYSTEM_MONITOR | SYSTEM_AUDITOR}
            [{-subject-id|sb <subjectid> |group|g <group>}]
            [-h]
```

Description

`viprcli authentication delete-role` command deletes a user role from accessing the VDC. The command is silent on success.

Options

role

The user role to be deleted from the VDC. This must be one of SYSTEM_ADMIN, SECURITY_ADMIN, SYSTEM_MONITOR, or SYSTEM_AUDITOR. This is a mandatory parameter.

subject-id|sb|group|g

The subject ID or user group to be authenticated. The subject ID or user group must be in your AD/LDAP list.

common arguments

This operation also takes the Common Parameters on page 24.

Example 12

```
# viprcli authentication delete-vdc-role -r SYSTEM_MONITOR
```

Assigning a role to an active directory user

If you have active directory integrated into your ViPR Controller environment, you can assign either a Project Admin or Tenant Admin role to active directory users.

Before you begin

You must integrate Active Directory into your ViPR Controller environment before performing this procedure.

Procedure

1. Login to ViPR Controller CLI using the default root credentials.

   viprcli authenticate -u root -d /temp

2. Run viprcli tenant add-role. You can assign either a TENANT_ADMIN or PROJECT_ADMIN role to your user. This example assigns the role PROJECT_ADMIN to an active directory user in the cli.vipr domain.

   viprcli tenant add-role -name "Provider Tenant" -role PROJECT_ADMIN -subject-id cliuser@mydomain.com

3. Verify that the role assignment was successful by running viprcli tenant get-role command.

viprcli tenant get-role -name "Provider Tenant"

```
# viprcli tenant get-role -name "Provider Tenant"
{
   "link": {
      "href": "/tenants/urn:storageos:TenantOrg:
4e76d46b-7706-4979-826b-
de1afee7
                        687b:/role-assignments",
      "rel": "self"
   },
   "role_assignments": [
      {
         "role": [
            "TENANT_ADMIN"
         ],
         "subject_id": "root"
      },
      {
         "role": [
            "PROJECT_ADMIN"
         ],
         "subject_id": "cliuser@mydomain.com"
      }
   ]
}
```

viprcli authentication add-user-group

Add a user group.

Syntax

```
        viprcli authentication add-user-group
        -name <name>
        -domain <domain>
        -key <key>
        -values <values>
        [-hostname <hostname>]
        [-port|po <port_number>
        [-portui|pu <ui_port_number
        [-coookiefile|cf
   <cookiefile>
        [-h]
```

Description

`viprcli authentication add-user-group` command adds a user group to access the VDC.

Options

name

The name of the user group to be created. This is a mandatory parameter.

domain

The domain to which this user group is to be mapped.

key

The attribute key. This is a mandatory parameter.

values

Attribute values. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli authentication user-group-add-attribute

Add an attribute to a user group.

Syntax

```
        viprcli authentication user-group-add-attribute
        -name <name>
        -key <key>
        -values <values>
        [-hostname <hostname>]
        [-port|po <port_number>
        [-portui|pu <ui_port_number
        [-coookiefile|cf
    <cookiefile>
        [-h]
```

Description

`viprcli authentication user-group-add-attribute` command adds an attribute to a user group.

Options

name

The name of the user group to which the attribute is being added. This is a mandatory parameter.

key

The attribute key. This is a mandatory parameter.

values

Attribute values. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli authentication user-group-add-values

Add values to an attribute of a user group.

Syntax

```
        viprcli
  authentication user-group-add-values
```

```
            -name <name>
            -key <key>
            -values <values>
            [-hostname <hostname>]
            [-port|po <port_number>
            [-portui|pu <ui_port_number
            [-coookiefile|cf
        <cookiefile>
            [-h]
```

Description

`viprcli authentication user-group-add-values` command adds values to an attribute of a user group.

Options

name

The name of the user group to which the attribute will be modified. This is a mandatory parameter.

key

Name of the attribute key to which the values will be added. This is a mandatory parameter.

values

Attribute values to add. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli authentication user-group-remove-attribute

Remove an attribute from a user group.

Syntax

```
            viprcli authentication
    user-group-remove-attribute
            -name <name>
            -keys <keys>
            [-hostname <hostname>]
            [-port|po <port_number>
            [-portui|pu <ui_port_number
            [-coookiefile|cf
        <cookiefile>
            [-h]
```

Description

`viprcli authentication user-group-remove-attribute` command removes an attribute from a user group.

Options

name

The name of the user group from which the attribute will be removed. This is a mandatory parameter.

keys

Names of the attribute keys which are being removed. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli authentication user-group-remove-values

Remove values from an attribute of a user group.

Syntax

```
viprcli authentication user-group-remove-values
-name <name>
-key <key>
-values <values>
[-hostname <hostname>]
[-port|po <port_number>
[-portui|pu <ui_port_number
[-coookiefile|cf
<cookiefile>
[-h]
```

Description

`viprcli authentication user-group-remove-values` command removes values from an attribute of a user group.

Options

name

The name of the user group in which the attribute is being modified. This is a mandatory parameter.

key

Name of the attribute key from which the values will be removed. This is a mandatory parameter.

values

The attribute values to remove. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli authentication show-user-group

Show the details of a user group.

Syntax

```
        viprcli authentication show-user-group
        -name <name>
        [-xml]
        [-hostname <hostname>]
        [-port|po <port_number>
        [-portui|pu <ui_port_number
        [-coookiefile|cf
    <cookiefile>
        [-h]
```

Description

`viprcli authentication show-user-group` command shows the details of a user group.

Options

name

The name of the user group in which the attribute is being modified. This is a mandatory parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli authentication delete-user-group

Deletes a user group.

Syntax

```
        viprcli authentication delete-user-group
        -name <name>
        [-hostname <hostname>]
        [-port|po <port_number>
        [-portui|pu <ui_port_number
        [-coookiefile|cf
    <cookiefile>
        [-h]
```

Description

`viprcli authentication delete-user-group` command deletes a user group.

Options

name

The name of the user group being deleted. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli authentication list-user-groups

List all user groups.

Syntax

```
viprcli authentication list-user-groups
[-verbose]
[-hostname <hostname>]
[-port|po <port_number>
[-portui|pu <ui_port_number
[-coookiefile|cf
<cookiefile>
[-h]
```

Description

`viprcli authentication list-user-groups` command lists all user groups.

Options

verbose

List the user groups with details. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 6

viprcli bucket commands

This chapter contains the following topics:

- viprcli bucket acl 60
- viprcli bucket create 61
- viprcli bucket delete 62
- viprcli bucket delete-acl 63
- viprcli bucket list-acl 64
- viprcli bucket show 64
- viprcli bucket update 65

viprcli bucket acl

Adds, updates, or deletes the ACL rules for a bucket.

Syntax

```
        viprcli bucket acl
        [-h]
        [-hostname|hn <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        -name|-n <objectname>
        -operation|-op
    <acloperation>
        [-permissions|-perm
    <permissions>]
        [-tenant|-tn
    <tenantname>]
        -project|-pr
    <projectname>
        [-user|-u <user>]
        [-domain|-dom <domain>]
        [-group|-grp <group>]
        [-customgroup|-custgrp
    <customgroup>]
```

Description

The `viprcli bucket acl` command adds, updates, or deletes the ACL rules for a bucket.

Options

name|n

Name of bucket. This is a mandatory parameter.

operation|op

Bucket ACL operation. This is a mandatory parameter.

permissions|perm

Permission for ACL. This is an optional parameter. Use pipe delimited when providing permissions for ACL, for example, `execute|delete|write|read`. Allowable values are:

- read
- write
- execute
- delete
- privileged_write
- full_control
- read_acl
- write_acl

- none

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

user|u

User. This is an optional parameter.

domain|dom

Domain. This is an optional parameter.

group|grp

Group. This is an optional parameter.

customgroup|custgrp

Custom group. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli bucket create

Creates a bucket.

Syntax

```
    viprcli bucket create
    [-h]
    [-hostname|hn <hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|-n <name>
    -project|-pr
<projectname>
    -varray|-va <varray>
    -vpool|-vpool <vpool>
    -softquota|-squota
<softquota>
    -hardquota|-hquota
<hardquota>
    -retention|-ret
<retention_period>
    -owner|-own <owner>
```

Description

The `viprcli bucket create` command creates a bucket.

Options

name|n

Name for the bucket. This is a mandatory parameter.

project|pr

Name of project. This is a mandatory parameter.

varray|va

Name of the virtual array. This is a mandatory parameter.

vpool|vpool

Name of a virtual pool. This is a mandatory parameter.

softquota|squota

Soft quota size. This is a mandatory parameter.

hardquota|hquota

Hard quota size. This is a mandatory parameter.

retention|ret

Retention period. This is a mandatory parameter.

owner|own

Owner. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli bucket delete

Deletes a bucket.

Syntax

```
    viprcli bucket delete
    [-h]
    [-hostname|hn <hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|-n <objectname>
    -tenant|-tn <tenantname>
    -project|-pr
<projectname>
    [-forceDelete|-fd
<forcedelete>]
    [-deleteType|-dt
<delete_type>]
```

Description

The `viprcli bucket delete` command deletes a bucket.

Options

name|n

Name of bucket. This is a mandatory parameter.

tenant|tn

Name of tenant. This is a mandatory parameter.

project|pr

Name of project. This is a mandatory parameter.

forceDelete|fd

Force delete option. This is an optional parameter.

deleteType|dt

Specifies whether to delete the bucket from inventory only or perform a full delete. The default is a full delete. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli bucket delete-acl

Deletes an ACL from a bucket.

Syntax

```
    viprcli bucket delete-acl
    [-h]
    [-hostname|hn <hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|-n <objectname>
    [-tenant|-tn
<tenantname>]
    -project|-pr
<projectname>
```

Description

The `viprcli bucket delete-acl` command deletes an ACL from a bucket.

Options

name|n

Name of bucket. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli bucket list-acl

Lists the ACL for a bucket.

Syntax

```
        viprcli bucket list-acl
        [-h]
        [-hostname|hn <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        -name|-n <objectname>
        [-tenant|-tn
    <tenantname>]
        -project|-pr
    <projectname>
```

Description

The `viprcli bucket list-acl` command lists the ACL for a bucket.

Options

name|n

Name of bucket. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Names of project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli bucket show

Shows a bucket.

Syntax

```
        viprcli bucket show
        [-h]
        [-hostname|hn <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        -name|-n <objectname>
        -project|-pr
```

```
        <projectname>
            -tenant|-tn <tenantname>
            [-xml]
```

Description

The `viprcli bucket show` command shows a bucket.

Options

name|n

Name of bucket. This is a mandatory parameter.

project|pr

Name of project. This is a mandatory parameter.

tenant|tn

Name of tenant. This is a mandatory parameter.

xml

XML response. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli bucket update

Updates a bucket.

Syntax

```
            viprcli bucket update
            [-h]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|-n <name>
            -project|-pr
        <projectname>
            [-varray|-va <varray>]
            [-softquota|-squota
        <softquota>]
            [-hardquota|-hquota
        <hardquota>]
            [-retention|-ret
        <retention_period>]
            -tenant|-tn <tenant>
```

Description

The `viprcli bucket update` command updates a bucket.

Options

name|n

Name for the bucket. This is a mandatory parameter.

project|pr

Name of project. This is a mandatory parameter.

varray|va

Name of the virtual array. This is an optional parameter.

softquota|squota

Soft quota size. This is an optional parameter.

hardquota|hquota

Hard quota size. This is an optional parameter.

retention|ret

Retention period. This is an optional parameter.

tenant|tn

Name of tenant. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

CHAPTER 7

viprcli catalog commands

This chapter contains the following topics:

- viprcli catalog get-category 68
- viprcli catalog get-descriptor 72
- viprcli catalog get-service 75
- viprcli catalog execute 76

viprcli catalog get-category

Show service catalog category details.

Syntax

```
viprcli catalog get-category
[-urn|id <urn>]
[-path <path>]
[-xml <xml>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli catalog get-category` command shows the service catalog category details.

Options

-urn|id

Catalog category or sub-category URN or Id. This is an optional parameter.

-path

Path to the catalog category. This is an optional parameter.

-xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 13

```
 # viprcli catalog get-category -id
urn:storageos:CatalogCategory:d2fb4d15-e163-4355-aa55-370d70c75b8b:
{
   "description": "Block storage services for fibre channel and
iSCSI",
   "id": "urn:storageos:CatalogCategory:d2fb4d15-e163-4355-
aa55-370d70c75b8b:",
   "image": "icon_host.png",
   "inactive": false,
   "link": {
      "href": "/api/categories/
urn:storageos:CatalogCategory:d2fb4d15-e163-4355-
aa55-370d70c75b8b:",
      "rel": "self"
   },
   "name": "BlockStorageServices",
   "services": [
      {
```

Example 13 (continued)

```
         "approvalRequired": false,
         "baseService": "CreateBlockStorageForHost",
         "description": "Create block volume and export it for a
host",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:057bd4f2-0d0e-4f53-
bc54-c90566988342:",
         "image": "icon_array_host_add.png",
         "inactive": false,
         "link": {
            "href": "/api/services/urn:storageos:CatalogService:
057bd4f2-0d0e-4f53-bc54-c90566988342:",
            "rel": "self"
         },
         "name": "CreateBlockVolumeforaHost",
         "title": "Create Block Volume for a Host"
      },
      {
         "approvalRequired": false,
         "baseService": "RemoveBlockStorageForHost",
         "description": "Removes an unmounted block volume assigned
to a host and all of its exports. The volume will no longer be
available from any host.",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:ff21dba2-e547-43fe-
aa72-d6b1d30e25d2:",
         "image": "icon_array_host_remove.png",
         "inactive": false,
         "link": {
            "href": "/api/services/
urn:storageos:CatalogService:ff21dba2-e547-43fe-aa72-d6b1d30e25d2:",
            "rel": "self"
         },
         "name": "RemoveVolumebyHost",
         "title": "Remove Volume by Host"
      },
      {
         "approvalRequired": true,
         "baseService": "CreateVolume",
         "defaultExecutionWindowId":
"urn:storageos:ExecutionWindow:NEXT:",
         "description": "Create a block volume",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:4265cbf7-a261-4736-
ae96-71aa0bdcdb68:",
         "image": "icon_db_create.png",
         "inactive": false,
         "link": {
            "href": "/api/services/urn:storageos:CatalogService:
4265cbf7-a261-4736-ae96-71aa0bdcdb68:",
            "rel": "self"
         },
         "maxSize": 0,
         "name": "CreateBlockVolume",
         "title": "Create Block Volume"
      },
      {
         "approvalRequired": false,
         "baseService": "RemoveBlockStorage",
         "description": "Removes block volumes and related
exports.",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:
67ce5ff4-865a-4945-9b66-a90437777b60:",
         "image": "icon_db_remove.png",
```

Example 13 (continued)


```
         "inactive": false,
         "link": {
            "href": "/api/services/urn:storageos:CatalogService:
67ce5ff4-865a-4945-9b66-a90437777b60:",
            "rel": "self"
         },
         "name": "RemoveBlockVolumes",
         "title": "Remove Block Volumes"
      },
      {
         "approvalRequired": false,
         "baseService": "ExpandBlockStorage",
         "description": "Expand storage on a block volume",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:c3450225-
d254-472b-836a-5968caba4e0e:",
         "image": "icon_db_expand.png",
         "inactive": false,
         "link": {
            "href": "/api/services/
urn:storageos:CatalogService:c3450225-d254-472b-836a-5968caba4e0e:",
            "rel": "self"
         },
         "name": "ExpandBlockVolume",
         "title": "Expand Block Volume"
      },
      {
         "approvalRequired": false,
         "baseService": "ExportVolume",
         "description": "Export a block volume to a host",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:
6a517cf3-0754-469d-93e0-7ba26e3a8739:",
         "image": "icon_host_add.png",
         "inactive": false,
         "link": {
            "href": "/api/services/urn:storageos:CatalogService:
6a517cf3-0754-469d-93e0-7ba26e3a8739:",
            "rel": "self"
         },
         "name": "ExportVolumetoaHost",
         "title": "Export Volume to a Host"
      },
      {
         "approvalRequired": false,
         "baseService": "UnexportVolume",
         "description": "Remove a volume from an export. The volume
will still be accessible using another export.",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:0b335406-1740-4dfa-
bc36-e231b5e286b9:",
         "image": "icon_host_remove.png",
         "inactive": false,
         "link": {
            "href": "/api/services/urn:storageos:CatalogService:
0b335406-1740-4dfa-bc36-e231b5e286b9:",
            "rel": "self"
         },
         "name": "UnexportVolume",
         "title": "Unexport Volume"
      },
      {
         "approvalRequired": false,
         "baseService": "ChangeVolumeVirtualPool",
         "description": "Move a volume to a different virtual pool.
```

Example 13 (continued)

```
Imports will occur when moving from a non-VPLEX storage system pool
to a VPLEX storage pool.",
         "executionWindowRequired": false,
         "id":
"urn:storageos:CatalogService:be600eb1-7a13-4a35-7118-
d679f09929fa:",
         "image": "Service_Change_Virtual_Pool.png",
         "inactive": false,
         "link": {
            "href": "/api/services/
urn:storageos:CatalogService:be600eb1-7a13-4a35-7118-d679f09929fa:",
            "rel": "self"
         },
         "name": "ChangeVirtualPool",
         "title": "Change Virtual Pool"
      },
      {
         "approvalRequired": false,
         "baseService": "ChangeVolumeVirtualArray",
         "description": "Move a VPLEX local volume to a different
virtual array (virtual arrays must have connectivity via VPLEX)",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:c6ed10c5-2cb1-4fe3-
b782-dad430e131bb:",
         "image": "Service_Change_Virtual_Array.png",
         "inactive": false,
         "link": {
            "href": "/api/services/
urn:storageos:CatalogService:c6ed10c5-2cb1-4fe3-b782-dad430e131bb:",
            "rel": "self"
         },
         "name": "ChangeVirtualArray",
         "title": "Change Virtual Array"
      },
      {
         "approvalRequired": false,
         "baseService": "DiscoverUnmanagedVolumes",
         "description": "Discovers volumes on arrays that were not
created by ViPR",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:056262cd-a41c-468a-
b634-9e3c4dcdf8e4:",
         "image": "Service_Discover_Unmanaged_Volumes.png",
         "inactive": false,
         "link": {
            "href": "/api/services/urn:storageos:CatalogService:
056262cd-a41c-468a-b634-9e3c4dcdf8e4:",
            "rel": "self"
         },
         "name": "DiscoverUnmanagedVolumes",
         "title": "Discover Unmanaged Volumes"
      },
      {
         "approvalRequired": false,
         "baseService": "IngestUnmanagedVolumes",
         "description": "Import unmanaged volumes to bring them
under management of ViPR",
         "executionWindowRequired": false,
         "id": "urn:storageos:CatalogService:1fbcbb4d-95f5-4d79-
aef3-b42cdbb6e840:",
         "image": "Service_Ingest_Unmanaged_Volumes.png",
         "inactive": false,
         "link": {
            "href": "/api/services/urn:storageos:CatalogService:
1fbcbb4d-95f5-4d79-aef3-b42cdbb6e840:",
```

Example 13 (continued)

```
            "rel": "self"
         },
         "name": "IngestUnmanagedVolumes",
         "title": "Ingest Unmanaged Volumes"
      }
   ],
   "title": "Block Storage Services"
}
```

viprcli catalog get-descriptor

Show service catalog category options.

Syntax

```
viprcli catalog get-descriptor
[-urn|id <urn>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli catalog get-descriptor` command shows the service catalog category options.

Options

-urn|id

Catalog category or sub-category URN or Id. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 14

```
# viprcli catalog get-descriptor -id urn:storageos:CatalogService:
087bd4f2-0d0e-4f83-bc54-c90566988342:
{
   "category": "Block Services",
   "description": "Create Block Volume and export it for a Host",
   "destructive": false,
   "fields": {
      "consistencyGroup": {
         "label": "Consistency Group",
         "lockable": false,
         "name": "consistencyGroup",
         "options": {},
         "required": false,
         "select": "one",
```

Example 14 (continued)

```
         "type": "assetType.vipr.consistencyGroup",
         "validation": {
            "failureMessage": "",
            "max": 2147483647,
            "min": 0,
            "regEx": ""
         }
      },
      "hlu": {
         "description": "Host logical unit number for this volume.
A value of '-1' means automatically assign.",
         "initialValue": "-1",
         "label": "HLU",
         "lockable": true,
         "name": "hlu",
         "options": {},
         "required": false,
         "select": "one",
         "type": "number",
         "validation": {
            "failureMessage": "",
            "max": 2147483647,
            "min": -1,
            "regEx": ""
         }
      },
      "host": {
         "label": "Host",
         "lockable": false,
         "name": "host",
         "options": {},
         "required": true,
         "select": "one",
         "type": "assetType.vipr.host",
         "validation": {
            "failureMessage": "",
            "max": 2147483647,
            "min": 0,
            "regEx": ""
         }
      },
      "name": {
         "description": "User assigned description of the volume",
         "label": "Name",
         "lockable": false,
         "name": "name",
         "options": {},
         "required": true,
         "select": "one",
         "type": "text",
         "validation": {
            "failureMessage": "",
            "max": 128,
            "min": 2,
            "regEx": ""
         }
      },
      "numberOfVolumes": {
         "initialValue": "1",
         "label": "Number Of Volumes",
         "lockable": false,
         "name": "numberOfVolumes",
         "options": {},
         "required": true,
         "select": "one",
```

Example 14 (continued)

```
            "type": "number",
            "validation": {
               "failureMessage": "",
               "max": 2147483647,
               "min": 1,
               "regEx": ""
            }
         },
         "project": {
            "label": "Project",
            "lockable": true,
            "name": "project",
            "options": {},
            "required": true,
            "select": "one",
            "type": "assetType.vipr.project",
            "validation": {
               "failureMessage": "",
               "max": 2147483647,
               "min": 0,
               "regEx": ""
            }
         },
         "size": {
            "label": "Size (GB)",
            "lockable": false,
            "name": "size",
            "options": {},
            "required": true,
            "select": "one",
            "type": "storageSize",
            "validation": {
               "failureMessage": "",
               "max": 2147483647,
               "min": 1,
               "regEx": ""
            }
         },
         "storageType": {
            "initialValue": "exclusive",
            "label": "Storage Type",
            "lockable": true,
            "name": "storageType",
            "options": {},
            "required": true,
            "select": "one",
            "type": "assetType.vipr.blockStorageType",
            "validation": {
               "failureMessage": "",
               "max": 2147483647,
               "min": 0,
               "regEx": ""
            }
         },
         "virtualArray": {
            "label": "Virtual Array",
            "lockable": true,
            "name": "virtualArray",
            "options": {},
            "required": true,
            "select": "one",
            "type": "assetType.vipr.virtualArray",
            "validation": {
               "failureMessage": "",
               "max": 2147483647,
```

Example 14 (continued)

```
            "min": 0,
            "regEx": ""
         }
      },
      "virtualPool": {
         "label": "Virtual Pool",
         "lockable": true,
         "name": "virtualPool",
         "options": {},
         "required": true,
         "select": "one",
         "type": "assetType.vipr.blockVirtualPool",
         "validation": {
            "failureMessage": "",
            "max": 2147483647,
            "min": 0,
            "regEx": ""
         }
      }
   },
   "roles": [],
   "serviceId": "CreateBlockStorageForHost",
   "title": "Create Block Volume for a Host"
}
```

viprcli catalog get-service

Show service catalog service details.

Syntax

```
viprcli catalog get-service
-urn|id <urn>
[-path <path>]
[-xml <xml>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli catalog get-service` command shows the service catalog service details.

Options

-urn|id

Catalog category or sub-category URN or Id. This is a mandatory parameter.

-path

Path to the catalog service. This is an optional parameter.

-xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 15

```
 # viprcli catalog get-service -id urn:storageos:CatalogService:
087bd4f2-0d0e-4f83-bc54-c90566988342:
{
   "approvalRequired": false,
   "baseService": "CreateBlockStorageForHost",
   "description": "Create block volume and export it for a host",
   "executionWindowRequired": false,
   "id": "urn:storageos:CatalogService:087bd4f2-0d0e-4f83-bc54-
c90566988342:",
   "image": "icon_array_host_add.png",
   "inactive": false,
   "link": {
      "href": "/api/services/urn:storageos:CatalogService:
087bd4f2-0d0e-4f83-bc54-c90566988342:",
      "rel": "self"
   },
   "name": "CreateBlockVolumeforaHost",
   "title": "Create Block Volume for a Host"
}
```

viprcli catalog execute

Execute the catalog service for the specified request to place the order.

Syntax

```
viprcli catalog execute
-urn|id <urn> | -path        <path>
[-params <name value> [name        value>...]]
[-xml <xml>]
[-hostname|hn        <hostname>]
[-cookiefile|cf        <cookie_file>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-h]
```

Description

`viprcli catalog execute` command executes the catalog service for the specified request to place the order.

Options

-urn|id | -path

You can provide either the catalog category or sub-category URN or ID, or the path to the catalog service, but not both. This is a mandatory parameter.

-params

List of name-value pairs to run the catalog service. The following example shows the format when creating a single volume. If creating multiple volumes repeat for each volume.

```
volumes[0].name test01
volumes[0].size 1
volumes[0].numberOfVolumes 1
```

This is a mandatory parameter.

-xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Note

Provide the ID of a virtual array, virtual pool, or project when passing them as a parameter to a catalog service.

```
viprcli catalog execute -id ${catalogServiceUID} -params
storageType Exclusive host ${hostUID} virtualArray $
{virtualArrayUID} virtualPool ${virtualPoolUID} project $
{projectUID} volumes[0].name volTest volumes[0].size 1
volumes[0].numberOfVolumes 1
```

CHAPTER 8

viprcli cluster commands

This chapter contains the following topics:

- viprcli cluster create 80
- viprcli cluster delete 81
- viprcli cluster detach 82
- viprcli cluster get-hosts 83
- viprcli cluster list 84
- viprcli cluster list-umexportmasks 84
- viprcli cluster list-umvolumes 85
- viprcli cluster show 86
- viprcli cluster tasks 87
- viprcli cluster update 88

viprcli cluster create

Create a cluster.

Syntax

```
        viprcli cluster create
        -name|n <viprhostname>
        [-datacenter|dc
    <datacenter>]
        [-vcenter|vc <vcenter>]
        [-tenant|tn <tenant>]
        [-hostname|hn
    <hostname>]
        [-port|po
    <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        [-h]
```

Description

`viprcli cluster create` command creates a cluster. This command is silent on success.

Options

name|n

The name of cluster. This is a mandatory parameter.

datacenter|dc

The name of the datacenter for the host. This is an optional parameter.

vcenter|vc

The name of the vcenter. This is an optional parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 16

```
# viprcli cluster create -n viprclusterhost
```

Use the list command to view the result.

Example 16 (continued)

```
# viprcli cluster list
  NAME
  viprclusterhost
```

viprcli cluster delete

Delete the cluster and provide the datacenter and vCenter options if the cluster is a part of Datacenter-vCenter.

Syntax

```
            viprcli cluster delete
-name|n <viprhostname>
[-datacenter|dc <datacentername>]
[-vcenter|vc <vcentername>]
[-detachstorage|ds]
[-tenant|tn <tenant>]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-h]
```

Options

name|n

The cluster name . This is mandatory parameter.

datacentername|dc

The datacenter name . This is mandatory parameter.

vcentername|vc

The vCenter name . This is mandatory parameter.

hostname|hn

The ViPR Controller host name (fully qualified domain name or IPv4 address,, or IPv6 address inside quotes and brackets, for example: `"[2001:db8::1]"` . This is an optional parameter.

detachstorage|ds

Detach storage before deactivation. This is an optional parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 17

Example 17 (continued)

```
# viprcli cluster delete -n viprclusterhost
```

viprcli cluster detach

Detach a cluster and provide the datacenter and vCenter options if the cluster is a part of Datacenter-vCenter.

Syntax

```
            viprcli cluster detach
[-name|n <cluster_name>]
[-datacenter|dc <datacentername>]
[-vcenter|vc <vcentername>]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Options

name|n

The cluster name . This is mandatory parameter.

datacentername|dc

The datacenter name . This is mandatory parameter.

vcentername|vc

The vCenter name . This is mandatory parameter.

hostname|hn

The ViPR Controller host name (fully qualified domain name or IPv4 address,, or IPv6 address inside quotes and brackets, for example: `"[2001:db8::1]"` . This is an optional parameter.

tenant|tn

The name of the tenant.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli cluster get-hosts

Display the list of hosts in the cluster and provide the datacenter and vCenter options if the cluster is a part of Datacenter-vCenter.

Syntax

```
            viprcli cluster get-hosts
[-name|n <clustername>]
[-datacenter|dc <datacentername>]
[-vcenter|vc <vcentername>]
[-verbose|v <verbose> | -long|l        <long>]
[-tenant|tn <tenant>]
[-hostname|hn        <hostname>]
[-port|po        <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-h]
```

Options

name|n

The cluster name . This is mandatory parameter.

datacentername|dc

The datacenter name . This is mandatory parameter.

vcentername|vc

The vCenter name . This is mandatory parameter.

hostname|hn

The ViPR Controller host name (fully qualified domain name or IPv4 address,, or IPv6 address inside quotes and brackets, for example: `"[2001:db8::1]"` . This is an optional parameter.

verbose|v | long|l

Verbose lists the details of the hosts. Long lists the details of the hosts in tabular form. Use either verbose or long but not both. This is an optional parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 18

```
# viprcli cluster get-hosts
```

viprcli cluster list

Display the list of clusters.

Syntax

```
viprcli cluster list
[-verbose|v <verbose>]
[-long|l <long>]
[-tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli cluster list` command displays the list of clusters.

Options

verbose|v | long|l

These options show additional information. Use either verbose or long and not both.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 19

```
# viprcli cluster list
  NAME
  viprclusterhost
```

viprcli cluster list-umexportmasks

Display the list of unmanaged export masks on the cluster and provides the datacenter and vCenter options if the cluster is a part of Datacenter-vCenter..

Syntax

```
viprcli cluster list-umexportmasks
[-name|n <name>]
[-datacentername|dc <datacentername>]
[-vcentername|vc <vcentername>]
[-hostname|hn        <hostname>]
```

```
[-verbose|v <verbose>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-h]
```

Options

name|n

The cluster name . This is mandatory parameter.

datacentername|dc

The datacenter name . This is mandatory parameter.

vcentername|vc

The vCenter name . This is mandatory parameter.

hostname|hn

The ViPR Controller host name (fully qualified domain name or IPv4 address,, or IPv6 address inside quotes and brackets, for example: `"[2001:db8::1]"` . This is an optional parameter.

verbose|v

This option shows additional information. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli cluster list-umvolumes

Display the list of unmanaged volumes on the cluster and provides the datacenter and vCenter options if the cluster is a part of Datacenter-vCenter.

Syntax

```
            viprcli cluster list-umvolumes
[-name|n <name>]
[-datacentername|dc <datacentername>]
[-vcentername|vc <vcentername>]
[-hostname|hn        <hostname>]
[-verbose|v <verbose>]
[-port|po        <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-h]
```

Options

name|n

The cluster name . This is mandatory parameter.

datacentername|dc

The datacenter name . This is mandatory parameter.

vcentername|vc

The vCenter name . This is mandatory parameter.

hostname|hn

The ViPR Controller host name (fully qualified domain name or IPv4 address,, or IPv6 address inside quotes and brackets, for example: "`[2001:db8::1]`" . This is an optional parameter.

verbose|v

This option shows additional information. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli cluster show

Return the cluster details and provides the datacenter and vCenter options if the cluster is a part of Datacenter-vCenter.

Syntax

```
            viprcli cluster show
[name|n <clustername>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-tenant|tn <tenant>]
[-datacentername|dc <datacentername>]
[-vcentername|vc <vcentername>]
[-xml <xmlresponse>]
[-h]
```

Options

name|n

The cluster name . This is a mandatory parameter.

datacentername|dc

The datacenter name . This is a mandatory parameter.

vcentername|vc

The vCenter name . This is a mandatory parameter.

hostname|hn

The ViPR Controller host name (fully qualified domain name or IPv4 address,, or IPv6 address inside quotes and brackets, for example: "`[2001:db8::1]`" . This is an optional parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

xml

Information is returned in JSON format by default. This parameter returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

```
 # viprcli cluster show -n viprclusterhost
{
   "creation_time": 1377066647244,
   "id": "urn:storageos:Cluster:
3265815e-752d-419d-8169-3ec4ec175183:",
   "inactive": false,
   "link": {
      "href": "/compute/clusters/urn:storageos:Cluster:
3265815e-752d-419d-8169-3ec4ec175183:",
      "rel": "self"
   },
   "name": "viprclusterhost",
   "tags": [],
   "tenant": {
      "id": "urn:storageos:TenantOrg:f8c0562f-33ef-4994-
a325-98ef7ffc1c03:",
      "link": {
         "href": "/tenants/
urn:storageos:TenantOrg:f8c0562f-33ef-4994-a325-98ef7ffc1c03:",
         "rel": "self"
      }
   }
}
```

viprcli cluster tasks

Check the tasks of a cluster and provide the datacenter and vCenter options if the cluster is a part of Datacenter-vCenter.

Syntax

```
            viprcli cluster tasks
[-name|n <cluster_name>]
[-datacenter|dc <datacentername>]
[-vcenter|vc <vcentername>]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-id <opid>]
[-verbose|v]
[-h]
```

Options

name|n

The cluster name . This is mandatory parameter.

datacentername|dc

The datacenter name . This is mandatory parameter.

vcentername|vc

The vCenter name . This is mandatory parameter.

hostname|hn

The ViPR Controller host name (fully qualified domain name or IPv4 address,, or IPv6 address inside quotes and brackets, for example: `"[2001:db8::1]"` . This is an optional parameter.

tenant|tn

The name of the tenant.

id

The operation ID of a cluster.

verbose|v

List all the tasks of a cluster.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli cluster update

Update the cluster with specified details and provide the datacenter and vCenter options if the cluster is a part of Datacenter-vCenter.

Syntax

```
            viprcli cluster update
-name|n <viprhostname>
[-label|l <labelname>
[-datacentername|dc        <datacentername>]
[-newdatacentername|ndc        <newdatacentername>]
[-vcentername|vc <vcentername>]
[-newvcentername|nvc <newvcentername>]
[-tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-autoExportsEnabled|autoEx        <true|false>]
[-h]
```

Options

name|n

The ViPR Controller cluster name. This is a mandatory parameter.

datacentername|dc

The name of the datacenter for the host. This is a mandatory parameter.

vcentername|vc

The name of the vcenter. This is a mandatory parameter.

newdatacentername|ndc

The new name of the datacenter for the host. This is an optional parameter.

newvcentername|nvc

The new name of the vcenter. This is an optional parameter.

label|l

The new label cluster name. This is an optional parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 20

```
# viprcli cluster update -n viprclusterhost -l viprcluster
```

Use the list command to view the result.

```
# viprcli cluster list
  NAME
  viprclustercluster
```

CHAPTER 9

viprcli computeimage commands

This chapter contains the following topics:

- viprcli computeimage create 92
- viprcli computeimage delete 92
- viprcli computeimage list 93
- viprcli computeimage show 94
- viprcli computeimage update 94

viprcli computeimage create

Adds a compute image to the ViPR Controller physical assets.

Syntax

```
    viprcli computeimage create
    -name|n
<computeimagename>
    -imageurl|url
<imageurl>
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

The viprcli computeimage create command creates a compute image.

Options

name|n

The name of the compute image.

imageurl|url

The url path of the compute image.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computeimage delete

Deletes a compute image.

Syntax

```
viprcli computeimage delete
-name|n <computeimagename>
-forcedelete|fd
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computeimage delete command deletes a compute image.

Options

name|n

The name of the compute image.

forcedelete|fd

Force the delete of the compute image.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computeimage list

Lists all the compute images that have been added to ViPR Controller.

Syntax

```
    viprcli computeimage list
    -imagetype|t
<imagetype>
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
    [-verbose|v]
    [-long|l]
```

Description

The viprcli computeimage list command lists a compute image.

Options

imagetype|t

The type of the compute image.

verbose|v

List compute systems with details.

long|l

List compute systems with more details in tabular form.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computeimage show

Shows the details of a compute image.

Syntax

```
        viprcli computeimage show
        -name|n
    <computeimagename>
        [-hostname|hn
    <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        [-h]
        [-xml]
```

Description

`viprcli computeimageshow` provides command shows a compute image and the status of whether the compute image successfully uploaded to the compute image server or not.

Options

name|n

The name of the compute image.

tbd

The compute image was successfully uploaded to the compute image servers listed here.

tbd

The compute image failed to be uploaded to the compute image servers listed here.

xml

The xml response.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computeimage update

Updates the properties of a compute image.

Syntax

```
        viprcli computeimage update
        -name|n
    <computeimagename>
```

```
            -imageurl|url
        <imageurl>
            -label|l <label>
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

The viprcli computeimage update command updates a compute image.

Options

name|n

The name of the compute image.

imageurl|url

The url path of the compute image.

label|l

The new name for the compute image.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 10

viprcli computeimageserver commands

This chapter contains the following topics:

- viprcli computeimageserver create 98
- viprcli computeimageserver delete 99
- viprcli computeimageserver list 99
- viprcli computeimageserver show 99
- viprcli computeimageserver update 100

viprcli computeimageserver create

Adds the compute image server to the ViPR Controller physical assets

Syntax

```
            viprcli computeimageserver create
            -name <imageserver_name>
            -imageserver
        <imageserver_ip_address>
            -imageserversecondip
        <os_install_network_ip>
            -u <username>
            -tftd <tftpBootDir>
            -itm <os install timeout in
        ms>
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli computeimageserver create` adds the compute image server to the ViPR Controller physical assets.

Options

name

Unique name to identify the compute image server.

imageserver

FQDN or IP address of the compute image server.

imageserversecondip

The IP address of the OS installation network.

u

Log in name of the user to access the compute image server from the ViPR Controller.

tftd

Path to TFTPBOOT directory on the compute image server. Default is `/opt/tftpboot/`.

itm

Timeout value for OS installation (in seconds). Default value is 3600.

Common Arguments

This operation also takes the Common Parameters on page 24.

```
viprcli computeimageserver create -name lgly5185 -imageserverip
10.247.85.185 -imageserversecondip 12.0.51.7 -imageimporttimeout
1500 -tftpbootdir /opt/tftpboot/ -user root -itm 3600 -sshtimeout 20
```

viprcli computeimageserver delete

Deletes a compute image server from ViPR Controller.

Syntax

```
viprcli computeimageserver delete
-imageserver
<imageserver_ip_address>
[-cookiefile|cf
<cookie_file>]
[-h]
```

Description

`viprcli computeimageserver delete` Deletes a compute image server from ViPR Controller.

Options

imageserver

FQDN or IP address of the compute image server.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computeimageserver list

Provides a list of the compute image servers that have been added to the ViPR Controller.

Syntax

```
viprcli computeimageserver list
```

Description

viprcli computeimageserver show

Provides the details of a compute image server.

Syntax

```
viprcli computeimageserver show
-is
```

```
        <imageserver_ip_address>
```

Description

`viprcli computeimageserver show` provides the details of a compute image server, and the status of which compute images successfully uploaded to the compute image server. "AVAILABLE" being the image was successfully uploaded, or "NOT_AVAILABLE," if the image did not successfully upload to the compute image server.

Options

is

FQDN or IP address of the compute image server.

viprcli computeimageserver update

Edits the compute image server properties.

Syntax

```
            viprcli computeimageserver update
            -name <imageserver_name>
            -imageserver
        <imageserver_ip_address>
            -label
        <new_imageserver_ip_address>
            -imageserversecondip
        <os_install_network_ip>
            -u <username>
            -tftd <tftpBootDir>
            -itm <os install timeout in
        ms>
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli computeimageserver update` edits the compute image server properties.

Options

name

Unique name to identify the compute image server.

imageserver

FQDN or IP address of the compute image server.

label

To change the FQDN or IP address of the image server.

imageserversecondip

The IP address of the OS installation network.

u

Log in name of the user to access the compute image server from the ViPR Controller.

tftd

Path to TFTPBOOT directory on the compute image server. Default is `/opt/tftpboot/`.

itm

Timeout value for OS installation (in seconds). Default value is 3600.

Common Arguments

This operation also takes the Common Parameters on page 24.

```
viprcli computeimageserver update -name lgly5185 -imageserverip
10.247.85.185 -imageserversecondip 12.0.51.7 -imageimporttimeout
1500 -tftpbootdir /opt/tftpboot/ -user root -itm 3600 -sshtimeout 25
```

CHAPTER 11

viprcli computelement commands

This chapter contains the following topics:

- viprcli computelement deregister 104
- viprcli computelement list 104
- viprcli computelement register 105
- viprcli computelement show 105

viprcli computelement deregister

De-register a compute element.

Syntax

```
viprcli computelement deregister
-name|n <computelementename>
-computesystem|cs <computesystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computelement deregister command de-registers a compute element.

Options

name|n

The name of the compute element.

computesystem|cs

The name of the compute system.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computelement list

Lists a compute element.

Syntax

```
viprcli computelement list
-name|n <computelementename>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-xml]
[-h]
[-verbose|v]
[-long|l]
```

Description

The viprcli computelement list command lists a compute element.

Options

name|n

The name of the compute element.

verbose|v

List the compute elements with details.

long|l

List the compute elements with more details in tabular form.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computelement register

Register a compute element.

Syntax

```
viprcli computelement register
-name|n <computelementename>
-computesystem|cs <computesystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computelement register command registers a compute element.

Options

name|n

The name of the compute element.

computesystem|cs

The name of the compute system.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computelement show

Show a compute element.

Syntax

```
viprcli computelement show
-name|n <computelementename>
-computesystem|cs <computesystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
```

```
            [-cookiefile|cf <cookie_file>]
            [-xml]
            [-h]
```

Description

The viprcli computelement show command shows a compute element.

Options

name|n

The name of the compute element.

computesystem|cs

The name of the compute system.

xml

The XML response.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 12

viprcli computesystem commands

This chapter contains the following topics:

- viprcli computesystem create 108
- viprcli computesystem delete 109
- viprcli computesystem deregister 109
- viprcli computesystem discover 110
- viprcli computesystem list 110
- viprcli computesystem list-compute-elements 111
- viprcli computesystem register 112
- viprcli computesystem show 112
- viprcli computesystem update 113

viprcli computesystem create

Adds a compute system (Cisco UCS) to the ViPR Controller physical assets.

Syntax

```
    viprcli computesystem create
    -name|n
<computesystemname>
    -computeip|cip
<computesystemipaddress>
    -computeport|cpn
<computesystemportnumber>
    -user|u <username>
    -type|t {ucs}
    [-usessl|ssl]
    [-osinstallnetwork]
<osinstallnetworkID>
    [-computeimageserver]
<compute_image_server_ip>
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

The viprcli computesystem create command Adds a compute system (Cisco UCS) to the ViPR Controller physical assets.

Options

name|n

The name of the compute system. This is a mandatory parameter.

computeip|cip

The IP address of the compute system. This is a mandatory parameter.

computeport|cpn

The port number of the compute system. This is a mandatory parameter.

user|u

The username of the compute system. This is a mandatory parameter.

type|t

The type of the compute system. This is a mandatory parameter.

usessl|ssl

Indicates whether to use ssl. This is an optional parameter.

osinstallnetwork

The os installation network ID. This is the network required if performing os installation during a Vblock system service provisioning operation.

computeimageserver

The IP address of the compute image server to associate with this compute system. This is required if performing os installation during a Vblock system service provisioning operation.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computesystem delete

Delete a compute image.

Syntax

```
viprcli computesystem delete
-name|n <computesystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computesystem delete command deletes a compute system.

Options

name|n

The name of the compute system.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computesystem deregister

De-registers a compute image.

Syntax

```
viprcli computesystem deregister
-name|n <computesystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computesystem deregister command de-registers a compute system.

Options

name|n

The name of the compute system.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computesystem discover

Discovers a compute image.

Syntax

```
viprcli computesystem discover
-name|n <computesystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computesystem discover command discovers a compute system.

Options

name|n

The name of the compute system.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computesystem list

Lists a compute image.

Syntax

```
viprcli computesystem list
-long|l
-verbose|v
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computesystem list command lists a compute system.

Options

long|l

List compute systems with details.

verbose|v

List compute systems with more details in tabular form.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computesystem list-compute-elements

Lists the compute element for a compute image.

Syntax

```
viprcli computesystem list-compute-elements
-name|n <computesystemname>
-long|l
-verbose|v
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computesystem list-compute-elements command lists the compute elements for a compute system.

Options

name|n

The name of the compute system.

long|l

List compute systems with details.

verbose|v

List compute systems with more details in tabular form.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computesystem register

Registers a compute image.

Syntax

```
viprcli computesystem register
-name|n <computesystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computesystem register command registers a compute system.

Options

name|n

The name of the compute system.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computesystem show

Shows a compute system, and the compute system attribtutes.

Syntax

```
    viprcli computesystem show
    -name|n
<computesystemname>
    [-xml]
<computesystemname>
    [-hostname|hn <hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

The viprcli computesystem show command shows a compute system, and the compute system attributes.

Options

name|n

The name of the compute system. This is a mandatory parameter.

xml

The name of the compute system. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computesystem update

Edit the properties of a Vblock compute system (UCS).

Syntax

```
    viprcli computesystem update
    -name|n
<computesystemname>
    -osinstallnetwork|oin
<osinstallnetwork>
    -computeport|cpn
<computesystemportnumber>
    -user|u <username>
    -label|l <label>
    -ssl|usessl

    [-computeimageserver]
<compute_image_server_ip>
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

The viprcli computesystem update command updates a compute system.

Options

name|n

The name of the compute system.

osinstallnetwork|oin

The os installation network ID. This is the network required if performing os installation during a Vblock system service provisioning operation.

computeport|cpn

The port number of the compute system.

computeimageserver

The IP address of the compute image server to associate with this compute system. This is required if performing os installation during a Vblock system service provisioning operation.

user|u

The username of the compute system.

label|l

The new label of the compute system.

ssl|usessl

Set to use SSL or not.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 13

viprcli computevpool commands

This chapter contains the following topics:

- viprcli computevpool assign_computele 116
- viprcli computevpool create 116
- viprcli computevpool delete 118
- viprcli computevpool list 119
- viprcli computevpool show 119
- viprcli computevpool update 120

viprcli computevpool assign_computele

Assigns a compute element to a virtual pool.

Syntax

```
            viprcli computevpool assign_computele
            -name|n <computevpoolname>
            -computesystem|cs <computesystemname>
            -computele_remove|ele_remove <computele_remove>
[<computele_remove> ...]
            -computele_add|ele_add <computele_add>
[<computele_add> ...]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

The viprcli computevpool assign_computele command assigns a compute element to a virtual pool.

Options

name|n

The name of the compute virtual pool.

computesystem|cs

The url path of the compute system.

computele_remove|ele_remove

The compute element to be removed from the compute virtual pool.

computele_add|ele_add

The compute element to be added to the compute virtual pool.

common arguments

This operation also takes the Common Arguments.

viprcli computevpool create

Creates a compute virtual pool.

Syntax

```
            viprcli computevpool create
            -name|n <computevpoolname>
            -servicetemplates|stp <ServiceProfileTemplates>
[<ServiceProfileTemplates> ...]
            -varrays|va <varrays> [<varrays> ...]
            -maxhbas|maxh <MaxHBAs>
            -minhbas|minh <MinHBAs>
```

```
            -maxnics|maxn <MaxNICs>
            -minnics|minn <MinNICs>
            -maxmemory|maxm <MaxMemory>
            -minmemory|minm <MinMemory>
            -maxcpuspeed|maxs <MaxCpuSpeed>
            -mincpuspeed|mins <MinCpuSpeed>
            -maxthreads|maxt <MaxTotalThreads>
            -minthreads|mint <MinTotalThreads>
            -maxcores|maxc <MaxTotalCores>
            -mincores|minc <MinTotalCores>
            -maxprocessors|maxp <MaxProcessors>
            -minprocessors|minp <MinProcessors>
            -type|t {Cisco_UCSM,Cisco_CSeries,Generic}
            -usematchedpools|ump <useMatchedPools>
            -description|desc <description>
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

The viprcli computevpool create command creates a compute virtual pool.

Options

name|n

The name of the compute virtual pool.

servicetemplates|stp

The service profile template.

varrays|va

The virtual arrays for the compute virtual pool.

maxhbas|maxh

The max number of HBAs.

minhbas|minh

The min number of HBAs.

maxnics|maxn

The max number of NICs.

minnics|minn

The min number of NICs.

maxmemory|maxm

The max memory.

minmemory|minm

The min memory.

maxcpuspeed|maxs

The max CPU speed.

mincpuspeed|mins

The min CPU speed.

maxthreads|maxt
The max number of total threads.

minthreads|mint
The min number of total threads.

maxcores|maxc
The max number of total cores.

mincores|minc
The min number of total cores.

maxprocessors|maxp
The max number of processors.

minprocessors|minp
The min number of processors.

type|t
The type of the compute system.

usematchedpools|ump
The matched pools for the virtual pool.

description|desc
The description of the virtual pool.

common arguments
This operation also takes the Common Parameters on page 24.

viprcli computevpool delete

Deletes a compute virtual pool.

Syntax

```
viprcli computevpool delete
-name|n <computevpoolname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computevpool delete command deletes a compute virtual pool.

Options

name|n
The name of the compute virtual pool.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computevpool list

Lists a compute virtual pool.

Syntax

```
viprcli computevpool list
-long|l
-verbose|v
[-hostname|hn
<hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookie_file>]
[-h]
```

Description

The viprcli computevpool list command lists a compute virtual pool.

Options

long|l

List the compute virtual pool with details.

verbose|v

List the compute virtual pool with details in table format.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli computevpool show

Shows a compute virtual pool.

Syntax

```
viprcli computevpool show
-name|n <computevpoolname>
-xml
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli computevpool show command shows a compute virtual pool.

Options

name|n

The name of the compute virtual pool.

xml

The XML response.

common arguments

This operation also takes the Common Arguments.

viprcli computevpool update

Updates a compute virtual pool.

Syntax

```
            viprcli computevpool update
            -name|n <computevpoolname>
            -templates_remove|tp_rm
<service_profile_templates_remove>
[<service_profile_templates_remove> ...]
            -templates_add|tp_add <service_profile_templates_add>
[<service_profile_templates_add> ...]
            -varray_remove|va_rm <varray_remove>
[<varray_remove> ...]
            -varray_add|va_add <varray_add> [<varray_add> ...]
            -label|l <label>
            -maxhbas|maxh <MaxHBAs>
            -minhbas|minh <MinHBAs>
            -maxnics|maxn <MaxNICs>
            -minnics|minn <MinNICs>
            -maxmemory|maxm <MaxMemory>
            -minmemory|minm <MinMemory>
            -maxcpuspeed|maxs <MaxCpuSpeed>
            -mincpuspeed|mins <MinCpuSpeed>
            -maxthreads|maxt <MaxTotalThreads>
            -minthreads|mint <MinTotalThreads>
            -maxcores|maxc <MaxTotalCores>
            -mincores|minc <MinTotalCores>
            -maxprocessors|maxp <MaxProcessors>
            -minprocessors|minp <MinProcessors>
            -type|t {Cisco_UCSM,Cisco_CSeries,Generic}
            -usematchedpools|ump <useMatchedPools>
            -description|desc <description>
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

The viprcli computevpool update command updates a compute virtual pool.

Options

name|n

The name of the compute virtual pool.

templates_remove|tp_rm

The service profile template to be removed from the compute virtual pool.

templates_add|tp_add
The service profile template to be added to the compute virtual pool.

varray_remove|va_rm
The virtual array to be removed from the compute virtual pool.

varray_add|va_add
The virtual array to be added to the compute virtual pool.

label|l
The label of the compute virtual pool.

maxhbas|maxh
The max number of HBAs.

minhbas|minh
The min number of HBAs.

maxnics|maxn
The max number of NICs.

minnics|minn
The min number of NICs.

maxmemory|maxm
The max memory.

minmemory|minm
The min memory.

maxcpuspeed|maxs
The max CPU speed.

mincpuspeed|mins
The min CPU speed.

maxthreads|maxt
The max number of total threads.

minthreads|mint
The min number of total threads.

maxcores|maxc
The max number of total cores.

mincores|minc
The min number of total cores.

maxprocessors|maxp
The max number of processors.

minprocessors|minp
The min number of processors.

type|t

The type of the compute system.

usematchedpools|ump

The matched pools for the virtual pool.

description|desc

The description of the virtual pool.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 14

viprcli consistencygroup commands

This chapter contains the following topics:

- viprcli consistencygroup accessmode 124
- viprcli consistencygroup create 125
- viprcli consistencygroup delete 126
- viprcli consistencygroup failover 127
- viprcli consistencygroup failover_cancel 128
- viprcli consistencygroup list 130
- viprcli consistencygroup show 131
- viprcli consistencygroup swap 132
- viprcli consistencygroup update 133

viprcli consistencygroup accessmode

Updates the access mode on a target copy for a RecoverPoint consistency group. This command is silent on success. In this mode, the journal is not kept and a full sweep is done after direct access mode is complete. This is an option for long term tests (for RecoverPoint test copy) where the journal volume may not have enough space for long term image access mode.

Syntax

```
            viprcli consistencygroup accessmode
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n
        <consistencygroupname>
            -project|pr
        <projectname>
            [-tenant|tn <tenant>]
            -copyvarray|cv
        <copyvarray>
            [-pit|p <pit>]
            [-type|t <type> {native | rp |
        srdf}]
    [-accessmode|am <accessmode> {DIRECT_ACCESS}]
```

Options

name|n

The name of the consistency group in ViPR Controller. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

Note

The project to be use here must exist in ViPR Controller.

tenant|tn

The name of the tenant. This is an optional parameter.

copyvarray|cv

The name of the copy virtual array. This is a mandatory parameter.

pit|p

Any UTC point in time formatted as yyyy-MM-dd_HH:mm:ss or datetime in milliseconds. This is an optional parameter.

type|t

The type of protection. Values are `native`, `rp`, and `srdf`.

accessmode|am

Updates the access mode for RecoverPoint consistency groups only. Currently, the only supported value is `DIRECT_ACCESS`.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 21

```
viprcli consistencygroup accessmode -name sg17924b -project sanity -
tenant standalone -copyvarray varray1 -type rp -accessmode
DIRECT_ACCESS)
```

viprcli consistencygroup create

Create a consistency group.

Syntax

```
            viprcli consistencygroup create
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n
        <consistencygroupname>
            -project|pr
        <projectname>
            [-tenant|tn <tenant>]
            [-h]
```

Description

`viprcli consistencygroup create` command creates a consistency group with the given name. This command is silent on success.

Options

name|n

The name of the consistency group in ViPR Controller. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

Note

The project to be use here must exist in ViPR Controller.

tenant|tn

The name of the tenant. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 22

```
# viprcli consistencygroup create -n consgroup01 -pr finance
```

Use the list command to view the result.

```
# viprcli consistencygroup list -pr finance
  NAME        VOLUMES STORAGE_CONTROLLER
  consgroup01
```

viprcli consistencygroup delete

Delete the consistency group.

Syntax

```
            viprcli consistencygroup delete
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n
        <consistencygroupname>
            -project|pr
        <projectname>
            [-tenant|tn
        <tenantname>]
            [-vipronly|vo]
```

Description

`viprcli consistencygroup delete` command deletes the consistency group from the specified project. The command is silent on execution.

Options

name|n

The name of the consistency group in ViPR Controller. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

Note

The project must exist in ViPR Controller.

tenant|tn

The name of the tenant. This is an optional parameter.

vipronly|vo

Delete only from ViPR. This option only applies to VPLEX consistency groups. This option prevents ViPR Controller from removing the consistency group from VPLEX . This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 23

```
# viprcli consistencygroup delete -n consgroup01 -pr finance
```

Use the list command to verify the consistency group has been deleted.

```
# viprcli consistencygroup list -pr finance
  NAME        VOLUMES STORAGE_CONTROLLER
  consgroup02
```

viprcli consistencygroup failover

Provide access to the latest image at the remote site.

Syntax

```
            viprcli consistencygroup failover
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n
        <consistencygroupname>
            -project|pr
        <projectname>
            [-tenant|tn <tenant>]
            -copyvarray|cv
        <copyvarray>
            [-pit|p <pit>]
            [-type|t <type> {native | rp |
```

```
        srdf}]
```

Description

`viprcli consistencygroup failover` command provides access to the latest image at the remote site. This command is silent on success.

Options

name|n

The name of the consistency group in ViPR Controller. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

Note

The project to be use here must exist in ViPR Controller.

tenant|tn

The name of the tenant. This is an optional parameter.

copyvarray|cv

The name of the copy virtual array. This is a mandatory parameter.

pit|p

Any UTC point in time formatted as yyyy-MM-dd_HH:mm:ss or datetime in milliseconds. This is an optional parameter.

type|t

The type of protection. Values are **`native`**, **`rp`**, and **`srdf`**.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 24

```
 viprcli consistencygroup failover -name cgName -project proj -
tenant standalone -copyvarray varray1 -type rp -pit
2016-01-19_12:03:00
```

viprcli consistencygroup failover_cancel

Cancel the operation started by viprcli consistencygroup failover.

Syntax

```
viprcli consistencygroup failover_cancel
```

```
            -name|n
        <consistencygroupname>
            -project|pr
        <projectname>
            -copyvarray|cv
        <copyvarray>
            [-tenant|tn <tenant>]
            [-type|t <type of protection {native
        | rp | srdf}]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli consistencygroup failover_cancel` command cancels the operation started by viprcli consistencygroup failover. This command is silent on success.

Options

name|n

The name of the consistency group in ViPR Controller. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

Note

The project to be use here must exist in ViPR Controller.

copyvarray|cv

The name of the copy virtual array. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

type|t

The type of protection. Values are **`native`**, **`rp`**, and **`srdf`**.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 25

```
 # viprcli consistencygroup failover_cancel -name
mpSanity-10247103-196-cg -project sanity -tenant standalone -
copyvarray varray3 -type rp
```

viprcli consistencygroup list

Display the list of consistency groups within a project.

Syntax

```
            viprcli consistencygroup list
            [-verbose|v]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -project|pr <projectname>
            [-tenant|tn <tenant>]
            [-h]
```

Description

`viprcli consistencygroup list` command displays the list of consistency groups available within a given project in ViPR Controller.

Options

project|pr

The name of the project. This is a mandatory parameter.

Note

The project to be used must exist in ViPR Controller and should contain a minimum one consistency group.

---

tenant|tn

The name of the tenant. This is an optional parameter.

verbose|v

List consistency groups with details. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 26

```
# viprcli consistencygroup list -pr finance
 NAME        VOLUMES STORAGE_CONTROLLER
 consgroup01
 consgroup02
```

viprcli consistencygroup show

Return the details of a consistency group.

Syntax

```
viprcli consistencygroup show
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
-name|n <consistencygroupname>
-project|pr <projectname>
[-xml <XML response>]
[-tenant|tn <tenant>]
[-h]
```

Description

`viprcli consistencygroup show` command returns details of a consistency group within the specified project.

Options

name|n

The name of the consistency group in ViPR Controller. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

Note

The project to be used must exist in ViPR Controller and should contain a minimum one consistency group to show.

---

tenant|tn

The name of the tenant.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

common arguments

This operation also takes the Common Parameters on page 24.

Example 27

```
# viprcli consistencygroup show -n consgroup02 -pr finance
{
   "id": "urn:storageos:BlockConsistencyGroup:27742e1f-1f53-40b1-
a12a-dda71ccc5aaf:",
   "inactive": false,
```

Example 27 (continued)

```
   "link": {
      "href": "/block/consistency-groups/
urn:storageos:BlockConsistencyGroup:27742e1f-1f53-40b1-a12a-
dda71ccc5aaf:",
      "rel": "self"
   },
   "name": "consgroup02",
   "project": {
      "id": "urn:storageos:Project:66b692e3-
fcc1-4be8-9d7a-3fe8f519387e:",
      "link": {
         "href": "/projects/urn:storageos:Project:66b692e3-
fcc1-4be8-9d7a-3fe8f519387e:",
         "rel": "self"
      }
   },
   "tags": []
}
```

viprcli consistencygroup swap

Swap the personalities of the source and target so that the source becomes the target and the target becomes the source.

Syntax

```
            viprcli consistencygroup swap
            -name|n
        <consistencygroupname>
            -project|pr
        <projectname>
            -copyvarray|cv
        <copyvarray>
            [-tenant|tn <tenant>]
            [-type|t <type of protection {native
        | rp | srdf}]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli consistencygroup swap` command swaps the personality of the source and target. The source becomes the target and the target becomes the source. This command is silent on success.

Options

name|n

The name of the consistency group in ViPR Controller. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

Note

The project to be use here must exist in ViPR Controller.

copyvarray|cv

The name of the copy virtual array. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

type|t

The type of protection. Values are **`native`**, **`rp`**, and **`srdf`**.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 28

```
 # viprcli consistencygroup swap -name mpSanity-10247103-196-cg -
project sanity -tenant standalone -copyvarray varray3 -type rp
```

viprcli consistencygroup update

Update the consistency group.

Syntax

```
            viprcli consistencygroup update
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n
        <consistencygroupname>
            -project|pr
        <projectname>
            [-tenant|tn <tenant>]
            [-add_volumes|av <add_volumes>
        [<add_volumes>...]]
            [-remove_volumes|rv
        <remove_volumes> [<remove_volumes>...]]
            [-synchronous|sync]
            [-synctimeout]
```

Description

`viprcli consistencygroup update` command updates the specified consistency group and project. The update command works only when there you specify a volume to be added or removed. This command is silent on success.

Note

The add_volume and remove_volume parameters works only when the volume is created on a VPLEX array and not on any other supported arrays.

Options

name|n

The name of the consistency group in ViPR Controller. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

Note

The project must exist in ViPR Controller.

add_volume|av

The name of the volume in ViPR Controller to be added to the consistency group.

Note

The add_volume parameter is applicable only for volumes created on a VPLEX array.

remove_volume|rv

The name of the volume in ViPR Controller to be removed from the consistency group.

Note

The remove_volume parameter is applicable only for volumes created on a VPLEX array.

tenant|tn

The name of the tenant. This is an optional parameter.

synchronous|sync

Execute in synchronous mode. This is an optional parameter.

synctimeout

Synchronous timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 15

viprcli executionwindow commands

This chapter contains the following topics:

- viprcli executionwindow create 136
- viprcli executionwindow delete 137
- viprcli executionwindow list 138
- viprcli executionwindow show 139
- viprcli executionwindow update 140

viprcli executionwindow create

Create a schedule for the orders to be executed at a scheduled time or date or recurring intervals.

Syntax

```
    viprcli executionwindow create
    -name <name>
    -type <type>
    -hourOfDay <hourOfDay>
    -length <length>
    -lengthType <lengthType> {WEEKLY
| MONTHLY}
    -tenantname|tn <tenantname>
    [-minuteOfHour <minuteOfHour>]
    [-dayOfWeek <dayOfWeek>]
    [-dayOfMonth
<dayOfMonth>]
    [-lastDayOfMonth
<lastDayOfMonth>]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-h]
```

Description

`viprcli executionwindow create` command creates an execution window. This command is silent on success.

An execution window is the pre-defined time or date when the orders are scheduled to be executed. It is preferred to schedule an execution window to run when the system is least loaded or when it has been taken offline for maintenance.

Options

name

Name of the execution window. This is a mandatory parameter.

type

Type of the execution window. This is a mandatory parameter.

hourOfDay

Hour of the day the execution window should run. Specify it in UTC format between 00 to 23. This is a mandatory parameter.

length

Time length of the execution window. This is a mandatory parameter.

lengthType

Time length unit for the execution window. The allowed options are WEEKLY and MONTHLY. This is a mandatory parameter.

tenantname|tn

Name of the tenant. This is a mandatory parameter.

minuteOfHour

Minute of the hour that the execution window should run. This is an optional parameter.

dayOfWeek

Day of the week, when the `lengthType` is WEEKLY. This is an optional parameter.

dayOfMonth

Last day of the month, when the `lengthType` is MONTHLY. This is an optional parameter.

lastDayOfMonth

A number from 1 to 31 indicating the day of the month. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 29

```
# viprcli executionwindow create -n Test_Execution-Window1 -type
DAILY -hourOfDay 22 -length 2 -lengthType HOURS -tenantname
TEST_TENANT1
```

viprcli executionwindow delete

Delete the execution window.

Syntax

```
viprcli executionwindow delete
-uri|id <uri>
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli executionwindow delete` command deletes the execution window. This command is silent on success.

Options

-uri|id

Execution window URI or Id. This is a mandatory parameter. You must list the execution windows using `viprcli executionwindow list` command to know the Id.

common arguments

This operation also takes the Common Parameters on page 24.

Example 30

```
# viprcli executionwindow delete -id
urn:storageos:ExecutionWindow:de764792-d1d6-42b2-8efc-c53e0a9afd61:
```

viprcli executionwindow list

List the execution window.

Syntax

```
viprcli executionwindow list
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli executionwindow list` command lists the execution windows.

Options

common arguments

This operation also takes the Common Parameters on page 24.

Example 31

```
# viprcli executionwindow list

ID
LABEL EXECUTIONWINDOWTYPE EXECUTIONWINDOWLENGTHTYPE DAYOFMONTH
DAYOFWEEK
  urn:storageos:ExecutionWindow:de764792-d1d6-42b2-8efc-
c53e0a9afd61: ew2   DAILY               MINUTES
  urn:storageos:ExecutionWindow:f131ec2c-61d4-4d62-b818-
ea4de9067fec: ew1   DAILY               MINUTES
```

viprcli executionwindow show

Show the execution window details.

Syntax

```
viprcli executionwindow show
-uri|id <uri>
[-xml <xml>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli executionwindow show` command shows the execution window details.

Options

-uri|id

Execution window URI or Id. This is a mandatory parameter.

-xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 32

```
# viprcli executionwindow show -id
urn:storageos:ExecutionWindow:f131ec2c-61d4-4d62-b818-ea4de9067fec:
{
   "executionWindowLength": 31,
   "executionWindowLengthType": "MINUTES",
   "executionWindowType": "DAILY",
   "hourOfDayInUTC": 5,
   "id": "urn:storageos:ExecutionWindow:f131ec2c-61d4-4d62-b818-
ea4de9067fec:",
   "label": "ew1",
   "lastDayOfMonth": false,
   "tenant": "urn:storageos:TenantOrg:f7e8a505-4ed2-4944-92b4-
a12755994628:"
}
```

viprcli executionwindow update

Update the execution window.

Syntax

```
viprcli executionwindow update
-uri|id <uri>
-newname <newname>
-type <type>
-hourOfDay <hourOfDay>
-length <length>
-lengthType <lengthType> {WEEKLY | MONTHLY}
[-dayOfWeek <dayOfWeek>]
[-dayOfMonth <dayOfMonth>]
[-lastDayOfMonth <lastDayOfMonth>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli executionwindow update` command updates the execution window. This command is silent on success.

Options

-uri|id

Execution window URI or Id. This is a mandatory parameter.

newname

New name of the execution window. This is a mandatory parameter.

type

Type of the execution window. This is a mandatory parameter.

hourOfDay

Hour of the day the execution window should run. Specify it in UTC format between 00 to 23. This is a mandatory parameter.

length

Time length of the execution window. This is a mandatory parameter.

lengthType

Time length unit for the execution window. The allowed options are MINUTES, HOURS, DAYS, WEEKLY, and MONTHLY. This is a mandatory parameter.

dayOfWeek

Day of the week, when the `lengthType` is WEEKLY. This is an optional parameter.

dayOfMonth

Last day of the month, when the `lengthType` is MONTHLY. This is an optional parameter.

lastDayOfMonth

A number from 1 to 31 indicating the day of the month. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 33

```
# viprcli executionwindow update -id
urn:storageos:ExecutionWindow:f131ec2c-61d4-4d62-b818-ea4de9067fec:
           -t MONTHLY -dayOfMonth 5 -length 31 -lengthType MONTHLY -
newname ew1
```

CHAPTER 16

viprcli event commands

This chapter contains the following topics:

- viprcli event approve 144
- viprcli event delete 145
- viprcli event list 146
- viprcli event details 147
- viprcli event decline 148
- viprcli event show 149

viprcli event approve

Approve the event.

Syntax

```
            viprcli event approve
            [-h]
[-hostname|hn <hostname>]
[-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-uri <uri>]
```

Description

`viprcli event approve` command approves the event with the specified details. This command is silent on success.

Options

hostname

The ViPR Controller hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0 or a ViPR Controller IPv6 address inside quotes and brackets. For example, "[2001:db8::1]"

port|po

The ViPR Controller port number.

portui|pu

The https port number of ViPR Controller Portal UI.

cookiefile|cf

The ViPR Controller cookie file full name.

uri

Universal resource identifier that provides name and location of the resource.

common arguments

This operation also takes the Common Parameters on page 24.

Example 34

```
# ./viprcli event approve -uri
urn:storageos:ActionableEvent:bc76aa07-ebb5-4afc-8a60-
d00e8f6657b9:vdc1
```

viprcli event delete

Delete the event.

Syntax

```
            viprcli event delete
            [-h]
[-hostname <hostname>]
[-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-uri <uri>]
```

Description

`viprcli event delete` command deletes the event with the specified details. This command is silent on success.

Options

hostname

The ViPR Controller hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0 or a ViPR Controller IPv6 address inside quotes and brackets. For example, "[2001:db8::1]"

port|po

The ViPR Controller port number.

portui|pu

The https port number of ViPR Controller Portal UI.

cookiefile|cf

The ViPR Controller cookie file full name.

uri

Universal resource identifier that provides name and location of the resource.

common arguments

This operation also takes the Common Parameters on page 24.

Example 35

```
./viprcli event delete -uri urn:storageos:ActionableEvent:bc76aa07-
ebb5-4afc-8a60-d00e8f6657b9:vdc1
```

viprcli event list

List the events.

Syntax

```
            viprcli event list
            [-h]
[-hostname <hostname>]
[-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-tenant <tenant>]
```

Description

`viprcli event list` command lists the event with the specified details. This command is silent on success.

Options

hostname

The ViPR Controller hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0 or a ViPR Controller IPv6 address inside quotes and brackets. For example, "[2001:db8::1]"

port|po

The ViPR Controller port number.

portui|pu

The https port number of ViPR Controller Portal UI.

cookiefile|cf

The ViPR Controller cookie file full name.

tenant|tn

The ViPR Controller tenant name.

common arguments

This operation also takes the Common Parameters on page 24.

Example 36

```
linux:/opt/storageos/cli/bin # ./viprcli event list

ID
    EVENT_STATUS NAME                 WARNING
  urn:storageos:ActionableEvent:bc76aa07-ebb5-4afc-8a60-
d00e8f6657b9:vdc1 approved     lglw7149.lss.emc.com Host storage
will be modified
  urn:storageos:ActionableEvent:7abfc925-f325-4aac-a1d7-
bc55dea9f85a:vdc1 approved     lglw7144.lss.emc.com Host storage
```

Example 36 (continued)

```
will be modified
  urn:storageos:ActionableEvent:853ed040-cf56-4759-
bb2f-2ee6522dfb76:vdc1 approved     lglw7144.lss.emc.com Host
storage will be modified
  urn:storageos:ActionableEvent:e712d356-c507-4f30-
aa66-15b43e0298fa:vdc1 approved     lglw7144.lss.emc.com Host
storage will be modified
  urn:storageos:ActionableEvent:4ad45a39-e390-470c-98f9-
d180cde03007:vdc1 declined     lglw7144.lss.emc.com Host storage
will be modified
```

viprcli event details

Provides details about the event.

Syntax

```
            viprcli event details
            [-h]
[-hostname <hostname>]
[-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-uri <uri>]
```

Description

`viprcli event details` command output shows whether an event failed, is approved, or is declined. This command is silent on success.

Options

hostname

The ViPR Controller hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0 or a ViPR Controller IPv6 address inside quotes and brackets. For example, "[2001:db8::1]"

port|po

The ViPR Controller port number.

portui|pu

The https port number of ViPR Controller Portal UI.

cookiefile|cf

The ViPR Controller cookie file full name.

uri

Universal resource identifier that provides name and location of the resource.

common arguments

This operation also takes the Common Parameters on page 24.

Example 37

```
./viprcli event details -uri urn:storageos:ActionableEvent:bc76aa07-
ebb5-4afc-8a60-d00e8f6657b9:vdc1
{
   "approve_details": [ “Host will gain access to volume: Project
CL1 c11 ID: urn:storageos:Volume:525bedee-b446-4785-bc2f-
f2077baa2e07:vdc1”,
“Host will lose access to volume: Project CL2 c22 ID:
urn:storageos:Volume:bf741dbe-cbd7-49dc-8469-c74ebf037cdf:vdc1”],
   "decline_details": [
      "Host lglw7149.lss.emc.com will remain associated with its
current vCenter DataCenter. If the host is re-discovered in a
different vCenter DataCenter during the next vCenter discovery
cycle, a new event will be created."
   ]
}
```

viprcli event decline

Declines action on the event.

Syntax

```
            viprcli event decline
            [-h]
[-hostname <hostname>]
[-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-uri <uri>]
```

Description

`viprcli event decline` command declines action on the event. This command is silent on success.

Options

hostname

The ViPR Controller hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0 or a ViPR Controller IPv6 address inside quotes and brackets. For example, "[2001:db8::1]"

port|po

The ViPR Controller port number.

portui|pu

The https port number of ViPR Controller Portal UI.

cookiefile|cf

The ViPR Controller cookie file full name.

uri

Universal resource identifier that provides name and location of the resource.

common arguments

This operation also takes the Common Parameters on page 24.

Example 38

```
# ./viprcli event decline -uri
urn:storageos:ActionableEvent:bc76aa07-ebb5-4afc-8a60-
d00e8f6657b9:vdc1
```

viprcli event show

Shows the event.

Syntax

```
            viprcli event show
            [-h]
[-hostname <hostname>]
[-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-uri <uri>]
```

Description

`viprcli event show` command shows the event details. This command is silent on success.

Options

hostname

The ViPR Controller hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0 or a ViPR Controller IPv6 address inside quotes and brackets. For example, "[2001:db8::1]"

port|po

The ViPR Controller port number.

portui|pu

The https port number of ViPR Controller Portal UI.

cookiefile|cf

The ViPR Controller cookie file full name.

uri

Universal resource identifier that provides name and location of the resource.

common arguments

This operation also takes the Common Parameters on page 24.

Example 39

```
./viprcli event show -uri urn:storageos:ActionableEvent:bc76aa07-
ebb5-4afc-8a60-d00e8f6657b9:vdc1
{
   "creation_time": 1473299459501,
   "decline_details": [
      "Host lglw7149.lss.emc.com will remain associated with its
current vCenter DataCenter. If the host is re-discovered in a
different vCenter DataCenter during the next vCenter discovery
cycle, a new event will be created."
   ],
   "description": "Host lglw7149.lss.emc.com will be removed from
shared exports for cluster TrevorDC2Cluster and added to shared
exports for cluster TestCluster2",
   "event_code": "104",
   "event_execution_time": 1473299592377,
   "event_status": "approved",
   "global": false,
   "id": "urn:storageos:ActionableEvent:bc76aa07-ebb5-4afc-8a60-
d00e8f6657b9:vdc1",
   "inactive": false,
   "internal": false,
   "link": {
      "href": "/",
      "rel": "self"
   },
   "name": "Moved datacenter from TrevorDC to Bourne-4",
   "remote": false,
   "resource": {
      "id": "urn:storageos:Host:
391e98c1-1c1e-480b-87cd-02e9ebec78a7:vdc1",
      "link": {
         "href": "/compute/hosts/urn:storageos:Host:
391e98c1-1c1e-480b-87cd-02e9ebec78a7:vdc1",
         "rel": "self"
      },
      "name": "lglw7149.lss.emc.com"
   },
   "tags": [],
   "task_ids": [
      {
         "id": "urn:storageos:Task:9902dcf8-2193-421a-b5ca-
b17626c07005:vdc1",
         "link": {
            "href": "/vdc/tasks/urn:storageos:Task:
9902dcf8-2193-421a-b5ca-b17626c07005:vdc1",
            "rel": "self"
         }
      }
   ],
   "tenant": {
      "id": "urn:storageos:TenantOrg:ee91950d-ebbb-4653-
af60-193fe0aa9316:global",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:ee91950d-
ebbb-4653-af60-193fe0aa9316:global",
         "rel": "self"
      }
   },
```

Example 39 (continued)

```
   "vdc": {
      "id": "urn:storageos:VirtualDataCenter:4d27511c-55e4-4e29-
addb-df3fa0fd7ea4:vdc1",
      "link": {
         "href": "/vdc/urn:storageos:VirtualDataCenter:
4d27511c-55e4-4e29-addb-df3fa0fd7ea4:vdc1",
         "rel": "self"
      }
   },
   "warning": "Host storage will be modified"
}
```

CHAPTER 17

viprcli exportgroup commands

This chapter contains the following topics:

- viprcli exportgroup add_cluster 154
- viprcli exportgroup add_host 155
- viprcli exportgroup add_initiator 156
- viprcli exportgroup add_vol 158
- viprcli exportgroup create 160
- viprcli exportgroup delete 163
- viprcli exportgroup list 164
- viprcli exportgroup path_adjustment_preview 167
- viprcli exportgroup path_adjustment 168
- viprcli exportgroup remove_cluster 171
- viprcli exportgroup remove_host 172
- viprcli exportgroup remove_initiator 173
- viprcli exportgroup remove_vol 175
- viprcli exportgroup show 176
- viprcli exportgroup tag 179
- viprcli exportgroup tasks 180
- viprcli exportgroup changeportgroup 180

viprcli exportgroup add_cluster

Add a cluster to the export group.

Syntax

```
            viprcli exportgroup add_cluster
[-h]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
-name|n        <exportgroupname>
-cluster|cl        <clustername>
-project|pr        <projectname>
[-tenant|tn        <tenantname>]
[-varray|va <varray>]
[-synchronous|sync]
[-synctimeout|syncto]
```

Description

`viprcli exportgroup add_cluster` command adds a cluster to the export group. This command is silent on success.

Note

You must have created an export group with type 'Cluster' to add cluster to it.

Options

name|n

The name of a valid export group. This is a mandatory parameter.

cluster|cl

The cluster name to add. This must be a valid cluster configured to work in ViPR Controller. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

varray|va

The virtual array name. This is an optional parameter.

synchronous|sync

Perform the operation in synchronous or asynchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 40

```
# viprcli exportgroup add_cluster -n EXPORT_GROUP3 -cl clustername -
pr BLOCK_PROJECT
```

viprcli exportgroup add_host

Add a host to the export group.

Syntax

```
            viprcli exportgroup add_host
[-h]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
-name|n        <exportgroupname>
-hostlabel|hl        <hostlabel>
-project|pr        <projectname>
[-tenant|tn        <tenantname>]
[-varray|va <varray>]
[-synchronous|sync]
[-synctimeout|syncto]
```

Description

`viprcli exportgroup add_host` command adds a host to the export group. This command is silent on success.

Note

You must have created an export group with type 'Host' to add host to it.

Options

name|n

The name of a valid export group. This is a mandatory parameter.

hostlabel|hl

The host label name for which the initiators must be searched. This must be a valid host within ViPR Controller. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

varray|va

The virtual array name. This is an optional parameter.

synchronous|sync

Perform the operation in synchronous or asynchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 41

```
# viprcli exportgroup add_host -n EXPORT_GROUP2 -hl host_100 -pr
BLOCK_PROJECT
```

viprcli exportgroup add_initiator

Add an initiator to the export group. An initiator is a host port.

Syntax

```
            viprcli exportgroup add_initiator
[-h]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
-name|n        <exportgroupname>
-initiator|in <initiator>        [<initiator> ...]
-hostlabel|hl        <hostlabel>
-project|pr        <projectname>
[-tenant|tn <tenantname>]
[-varray|va <varray>]
[-synchronous|sync]
[-synctimeout|syncto]
```

Description

`viprcli exportgroup add_initiator` command adds an initiator to the export group. An initiator is a host port.

Note

You must have created an export group without specifying any type (or specify type Exclusive) to add initiator to it.

Options

name|n

The name of a valid export group. This is a mandatory parameter.

initiator|in

The initiator host name. Its format depends upon the protocol. For Fibre Channel, specify the Worldwide Node Name (WWNN). For iSCSI specify the initiator node in IQN format. The CLI will not validate the initiator node string. This is a mandatory parameter.

hostlabel|hl

A host label for the initiator host. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

varray|va

The virtual array name. This is an optional parameter.

synchronous|sync

Perform the operation in synchronous or asynchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 42

(iSCSI)

```
viprcli exportgroup add_initiator -n EXPORT_GROU21
-pr BLOCK_PROJECT -hl host_098 -initiator 'iqn.2013-08.com
.mycompany:localhost-4c92c642'
```

(Fiber Channel)

```
 # viprcli exportgroup add_initiator -n exportgroup1
-in 21:00:00:E0:8B:92:F2:85 -hl viprhost.example.mycompany.com -pr
HD
```

viprcli exportgroup add_vol

Add a volume or a snapshot to the export group.

Syntax

```
viprcli exportgroup add_vol
[-h]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-name|n <exportgroupname>]
[-volume|v <volumename>        [<Volume> ...]
[-project|pr    <projectname>]
[-tenant|tn <tenantname>]
[-snapshot|sh <snapshot for    volume> [<Snapshotname for
volume> ...]]
[-blockmirror|bmr <Block Mirror for        volume>[<Block Mirror
for volume> ...]]
[-consistencygroup|cg    <consistencygroup>]
[-varray|va <varray>]
[-maxpaths|mxp        <max_paths>]
[-minpaths|mnp        <min_paths>]
[-pathsperinitiator|ppi        <paths_per_initiaton];
[-portgroup|port_group_name  <portgroupname>]
[-serialnumber|sn]
[-synchronous|sync]
[-synctimeout|syncto]
```

Description

`viprcli exportgroup add_vol` command adds a volume or a snapshot to the export group. You can add a snapshot of a volume or a snapshot from a consistency group to an export group even when the parent volume is not part of that export group. A volume and its snapshots are treated as separate and independent objects for export purposes. An export group can also export snapshots to the host initiators. This command is silent on success.

Options

name|n

The name of a valid export group. This is a mandatory parameter.

volume|v

A list of volume and Host LUN ID (HLU) pairs in the format: `<volume_name>:<HLU>`. For example, volk:-1. The syntax for specifying multiple volumes is, <volumename_1>:<HLU1> <volumename_2>:<HLU2>, and so on. The volume names must be valid volumes within ViPR Controller.

Note

If you set HLU equal to -1, then the array will assign the HLU number.

This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

snapshot|sh

The name of a snapshot associated with the volume specified in the volume argument. Include the LUN ID, for example, `<snapshot_name>:<lun_id>`. This is an optional parameter.

blockmirror|bmr

List of block mirrors (continuous copies). Include the LUN ID, for example, `<block_mirror_name>:<lun_id>`. This is an optional parameter.

consistencygroup|cg

Name of the consistency group. This is an optional parameter, but must be specified if the snapshot is from a consistency group.

varray|va

The virtual array name. This is an optional parameter.

maxpaths|mxp

The maximum number of paths that can be used between a host and a storage volume. This value will override the value set in the virtual pool. This is an optional parameter.

minpaths|mnp

The minimum number of paths that can be used between a host and a storage volume. This value will override the value set in the virtual pool. This is an optional parameter.

pathsperinitiator|ppi

The number of paths per initiator. This value will override the value set in the virtual pool. This is an optional parameter.

synchronous|sync

Perform the operation in synchronous or asynchronous mode. This is an optional parameter.

portgroupname|port group name

Name of the Port group. This is an optional parameter.

serial number|sn

Serial number of the port groups's storage system. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters.

Example 43

Example 43 (continued)

```
# viprcli exportgroup add_vol -n Example_ExpGroup_1G_7822726523
            -v Example_ExpGroup_1G_7822726523 -pr Marketing
```

In this example, a single volume and LUN ID pair is specified as one of the parameters.

```
viprcli exportgroup add_vol -name 192.168.0.0 -pr myproj -volume
volk:-1 -sync

Exportgroup : 192.168.0.0
Volume name : volk
HLU : -1
```

In this example ViPR Controller will use the maximum path, minimum path, and paths per initiator set here to override the maximum path, minimum path, and paths per initiator set in the virtual pool when exporting the volume to a host.

```
viprcli exportgroup add_vol -n host3_eg -volume testvolume-1 -pr
project1 -mxp 1 -mnp 1 -ppi 1
```

In this example ViPR Controller will use the maximum path, minimum path, and paths per initiator set here to override the maximum path, minimum path, and paths per initiator set in the virtual pool when exporting a snapshot to the same host.

```
viprcli exportgroup add_vol -n host3_eg -volume testvolume-1 -pr
project1 -mxp 1 -mnp 1 -ppi 1
```

In this example ViPR Controller will use the port group number with a serial number.

```
viprcli exportgroup add_vol -n hostone.lss.emc.com -pr HR -vol
DemoVolApril20 -varray Hn1 -sn 351 -pgname PG-351-distributed
```

viprcli exportgroup create

Create a new export group.

Syntax

```
viprcli exportgroup create
```

```
            -name|n <exportgroupname>
            -project|pr <project>
            -varray|va <varray>
            [-type|t <exportgrouptype>]
            [-exportdestination|ed <exportdestination>]
            [-tenant|tn <tenantname>]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli exportgroup create` command creates a new export group. You must specify a name for the group, a project, and a virtual storage array. Initially this will be empty, with no initiators and no volumes.

Options

name|n

The name of an export group. This is a mandatory parameter.

project|pr

The name of a project. This is a mandatory parameter.

varray|va

The name of a virtual storage array. This is a mandatory parameter.

type|t

The export group type. It could be Exclusive, Host, or Cluster. The default is Exclusive when no type is specified. This is an optional parameter.

Note

You must have created an export group with type 'Host' to add a host or 'Cluster' to add a cluster and without any type (Exclusive) to add an initiator to an export group. However, to add volumes to an export group the type declaration doesn't matter.

---

exportdestination|ed

The export destination of cluster or host port type. Specify a host, initiator, or cluster. This is an optional parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 44

Export group general example

Example 44 (continued)

```
# viprcli exportgroup create -n my_export_group -pr myvipr_project -
va new_varray
# viprcli exportgroup list -pr myvipr_project
  NAME
  my_export_group
```

Example 45

Export group used with add_initiator

```
# viprcli exportgroup create -n EXPORT_GROUP2 -pr BLOCK_PROJECT -
varray new_varray2
```

Example 46

Export group used with add_host

```
# viprcli exportgroup create -n EXPORT_GROUP2 -pr BLOCK_PROJECT -
varray new_varray2 -type Host
```

Example 47

Export group used with add_cluster

```
# viprcli exportgroup create -n EXPORT_GROUP3 -pr BLOCK_PROJECT -
varray new_varray2 -type Cluster
```

Example 48

Export group with export destination for Cluster type

```
# viprcli exportgroup create -n eg10 -pr HD -va "site 1" -t Cluster
-ed Prov_Cluster_1
```

Example 49

Export group with export destination for Host type

```
# viprcli exportgroup create -n eg2 -pr HD -va "site 1" -t Host -ed
myfqdn.company.com
```

viprcli exportgroup delete

Delete an export group.

Syntax

```
viprcli exportgroup delete
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n <exportgroupname>
    -project|pr <project>
    [-varray|va <varray>]
    [-tenant|tn
<tenantname>]
    [-synchronous|sync]
    [-synctimeout|syncto]
```

Description

`viprcli exportgroup delete` command deletes an export group.

Options

name|n

The name of an export group. This is a mandatory parameter.

project|pr

The name of a project. This is a mandatory parameter.

varray|va

The name of a virtual storage array. This is an optional parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

synchronous|sync

Perform the operation in synchronous or asynchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 50

Example 50 (continued)

```
# viprcli exportgroup list -pr myvipr_project
  NAME
  my_export_group

# viprcli exportgroup delete -n my_export_group -pr myvipr_project
```

Use list command to view the results.

```
#viprcli exportgroup list -pr myvipr_project
```

viprcli exportgroup list

Return the list of all export groups for a specified project.

Syntax

```
            viprcli exportgroup list
            -project|pr <project>
            [{-verbose|v|-long|l <verbose |
        long> }]
            [-tenant|tn
        <tenantname>]
            [-varray <varrayname>]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli exportgroup list` command displays the names of all export groups in a specified project. A verbose listing (-v) allows you to get a detailed listing of all the initiators in the export group, the volumes exported using that export group, and all snapshots in the export group. A long listing (-l) display the initiators, volume, and snapshot information along with export group names.

Options

project|pr

The name of a project. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

varray

Is an optional parameter which allows you to view the export groups for a specific virtual array.

{verbose|v| long|l}

These options display additional information about export groups. Use either verbose or long and not both.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 51

```
# viprcli exportgroup list -pr exampleproj
  NAME
  Example_ExpGroup_1G_7822726523
```

Example 52

```
# viprcli exportgroup list -pr exampleproj -v
[
   {
   "clusters": [],
   "creation_time": 1376989761627,
   "generated_name": "ProviderT_Example
   _f79ca67e-ed18-46cc-91b9-f7751d3568cf",
   "hosts": [],
   "id": "urn:storageos:ExportGroup:
    f79ca67e-ed18-46cc-91b9-f7751d3568cf:",
   "inactive": false,
   "initiators": [
      {
         "creation_time": 1376989750907,
         "host": {
            "id": "urn:storageos:Host:
             6d98ccaf-9b47-431d-99ae-d1296f4461c1:",
            "link": {
               "href": "/compute/hosts/urn:
                storageos:Host:6d98ccaf-9b47-431d-99ae-
d1296f4461c1:",
               "rel": "self"
            }
         },
         "hostname": "host.3007850695349452",
         "id": "urn:storageos:Initiator:
          6060ed9a-e982-420a-be43-e0f74b55ee1f:",
         "inactive": false,
         "initiator_node": "20:00:00:00:00:00:00:00",
         "initiator_port": "21:00:98:92:71:80:05:85",
         "link": {
            "href": "/compute/initiators/urn:
           storageos:Initiator:6060ed9a-e982-420a-be43-
e0f74b55ee1f:",
            "rel": "self"
         },
         "protocol": "FC",
         "tags": []
      },
      {
```

Example 52 (continued)


```
        "creation_time": 1376989751025,
        "host": {
           "id": "urn:storageos:Host:
         6d98ccaf-9b47-431d-99ae-d1296f4461c1:",
           "link": {
              "href": "/compute/hosts/urn:
            storageos:Host:6d98ccaf-9b47-431d-99ae-d1296f4461c1:",
              "rel": "self"
           }
        },
        "hostname": "host.3007850695349452",
        "id": "urn:storageos:
       Initiator:99641cfd-03ac-47e2-afa9-24e2f8257f4f:",
        "inactive": false,
        "initiator_node": "20:00:00:00:00:00:00:00",
        "initiator_port": "21:00:87:57:04:65:04:85",
        "link": {
           "href": "/compute/initiators/urn:
       storageos:Initiator:99641cfd-03ac-47e2-afa9-24e2f8257f4f:",
           "rel": "self"
        },
        "protocol": "FC",
        "tags": []
     }
  ]
     "name": "Example_ExpGroup_1G_7822726523",
     "project": {
        "id": "urn:storageos:Project:
        b6395993-28b0-44fa-ba9c-f2ede0814148:",
        "link": {
           "href": "/projects/urn:storageos:Project:
        b6395993-28b0-44fa-ba9c-f2ede0814148:",
           "rel": "self"
        }
     },
     "tags": [],
     "tenant": {
        "id": "urn:storageos:TenantOrg:
      f8c0562f-33ef-4994-a325-98ef7ffc1c03:",
        "link": {
           "href": "/tenants/urn:storageos:TenantOrg:
      f8c0562f-33ef-4994-a325-98ef7ffc1c03:",
           "rel": "self"
        }
     },
     "type": "Exclusive",
     "varray": {
        "id": "urn:storageos:VirtualArray:
     501ed39e-834c-4dc0-baad-d16a5174482d:",
        "link": {
           "href": "/vdc/varrays/urn:storageos:VirtualArray:
     501ed39e-834c-4dc0-baad-d16a5174482d:",
           "rel": "self"
        }
     },
     "volumes": [
        {
           "id": "urn:storageos:Volume:
     7c8f0bc0-4ad5-4964-9421-3068f1fc35c4:"
        }
     ],
     "volumes_snapshots": [
        "Example_ExpGroup_1G_7822726523"
     ]
```

Example 52 (continued)

```
    }
]
```

viprcli exportgroup path_adjustment_preview

Preview export path adjustments (such as paths that will be added or removed) for a storage system within an export group before issuing the `viprcli exportgroup path_adjustment` command.

Syntax

```
viprcli exportgroup path_adjustment
            [-port|po <port_number>]
            [-portui|pu        <ui_port_number>]
            [-cookiefile|cf        <cookiefile>]
            [-project|pr <projectname>]
            [-tenant|tn <tenantname>]
            [-storagesystem|ss <storagesystem>]
            [-name|n <exportgroupname>]
            [-varray|va <varray>]
            [-minpaths|minp        <minpaths>]
            [-maxpaths|maxp        <maxpaths>]
            [-pathsperinitiator|ppi        <pathsperinitiator>]
            [-storageports|sp <storageports>]
            [-useexistingpaths|useex]
            [-h]
            [-hostname]
```

Description

The `viprcli exportgroup path_adjustment_preview` command displays what will change when the `viprcli exportgroup path_adjustment` is run. You can view the resources that will be affected when you issue the path_adjustment command. You can preview new paths and removed paths plus initiators and storage ports that will be affected by the changes.

Options

project|pr

The name of a project. This is a mandatory parameter.

storagesystem|ss

A valid storage system. This is a mandatory parameter.

name|n

The export group name. This is a mandatory parameter.

varray|va

Allows you to view the export groups for a specific virtual array. This is a mandatory parameter.

minp

Minimum number of paths allowed from the initiator to the Storage Port on the storage array.

maxp

Maximum number of paths allowed from the initiator to the Storage Port on the storage array. This is a mandatory parameter.

ppi

The number of storage port paths that may be provisioned for each initiator.

useexistingpaths|useex

If this option is specified, then it is True. If not specified, it is False.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 53

```
./viprcli exportgroup path_adjustment_preview -n losathost.emc.com -
pr rrprj1 -va testva -ss 999334388821  -minp 1 -maxp 4 -ppi 4 -useex
```

In this example, **rrprj1** is the `project`. **testva** is the `virtual array`. A minimum of one initiator is attached to the host and it can have up to four paths (`ppi`) connected to `storage system`, **999334388821**. Existing paths will be used if possible because `useex` is included in the command.

viprcli exportgroup path_adjustment

Adjust the export paths for a storage system within an export group. Paths may be added, removed, or updated with a different selection of ports. This feature is supported for VPLEX and VMAX exports.

Syntax

```
viprcli exportgroup path_adjustment
            [-port|po <port_number>]
            [-portui|pu        <ui_port_number>]
            [-cookiefile|cf        <cookiefile>]
            [-project|pr <projectname>]
            [-tenant|tn <tenantname>]
            [-verbose|v]
            [-storagesystem|ss <storagesystem>]
            [-name|n <exportgroupname>]
            [-varray|va <varray>]
            [-minpaths|minp        <minpaths>]
            [-maxpaths|maxp        <maxpaths>]
            [-pathsperinitiator|ppi        <pathsperinitiator>]
            [-storageports|sp <storageports>]
            [-useexistingpaths|useex]
            [-wait]
            [-h]
            [-hostname]
```

Description

`viprcli exportgroup path_adjustment` is used to adjust the export paths for a storage system within an export group. Use the `-wait` option if you want to pause the operation, review the impact of the changes, and then resume the path adjustments. The `-wait` option returns the Task ID which is required when issuing the command to resume operations. A verbose listing (-v) allows you to get a detailed listing of all the port and path changes in the export group.

Note

Including `-v`, `-wait` or `-useex` in the command enables these options. Omit these options if not needed.

Note

ViPR Controller runs the `viprcli exportgroup path_adjustment_preview` command and uses the results to adjust the paths. Exceptions can occur if the minp, maxp, or ppi values differ between the preview and actual path adjustment commands. If the `wait` option is included in the `viprcli exportgroup path_adjustment` command and this results in tasks being suspended, you have two choices:

- Resume adding and/or removing paths using the `viprcli task resume` command.
- Stop the path adjustment task using the `viprcli task rollback` command.

Note

Use the `viprcli task rollback` command to stop the path adjustment operation. This does not roll back provisioning that has already taken place. Any paths that have already been added remain. Rollback terminates the order but does not remove existing paths.

Options

project|pr

The name of a project. This is a mandatory parameter.

storagesystem|ss

The serial number of a storage system in the export that is adjusted. This is a mandatory parameter.

name|n

The export group name. This is a mandatory parameter.

varray|va

Allows you to view the export groups for a specific virtual array. This is a mandatory parameter.

minp

Minimum number of paths allowed from the initiator to the Storage Port on the storage array.

maxp

Maximum number of paths allowed from the initiator to the Storage Port on the storage array. This is a mandatory parameter.

ppi

The number of storage port paths that may be provisioned for each initiator.

useexistingpaths|useex

If this option is specified, then it is True. If not specified, it is False.

wait

In the path_adjustment this is the "wait_before_remove_paths" parameter. If this parameter is specified, then it is True. If not specified, it is False.

verbose|v

Use the verbose option to display additional information about export groups.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example syntax for `viprcli exportgroup path-adjustment`:

```
./viprcli exportgroup path_adjustment -n
"testty11_20170315085001826" -pr proj1 -va va -ss 000196801612 -
minp 2 -maxp 4 -ppi 2 -useex -wait
```

In this example, **`proj1`** is the `project`. **`va`** is the `virtual array`. A minimum of two initiators are attached to the host and up to four paths (`ppi`) may be connected to `storage system`, **`000196801612`**. Existing paths will be used if possible because `useex` is specified in the command. The absence of `verbose` means this will return only the short description.

The presence of the `-wait` option means the provisioning process will be suspended until you have a chance to review the impact of the path adjustments. When you are ready to accept the results, you have to issue a command to resume processing of the path adjustment. You can resume activity with the `viprcli task` command and the Task ID.

Example of Task ID obtained when `-wait` is in the command:

```
ExportGroup Path Adjustment
There are tasks (URIs listed below) that are suspended as part of
this operation. Manually resume the tasks.
urn:storageos:Task:6de37504-b145-4ef2-8afe-b38ea6d14e57:vdc1
```

Example of resuming a task:

```
viprcli task resume urn:storageos:Task:6de37504-b145-4ef2-8afe-
b38ea6d14e57:vdc1
```

Example of rolling back a task

```
./viprcli task rollback -tid urn:storageos:Task:6de37504-
b145-4ef2-8afe-b38ea6d14e57:vdc1
```

viprcli exportgroup remove_cluster

Delete a cluster from the export group.

Syntax

```
            viprcli exportgroup remove_cluster
[-h]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu    <ui_port_number>]
[-cookiefile|cf <cookie_file>]
-name|n <exportgroupname>
-cluster|cl    <clustername>
-project|pr    <projectname>
[-varray|va <varray>]
[-tenant|tn    <tenantname>]
[-synchronous|sync]
-force|f
[-synctimeout|syncto]
```

Description

`viprcli exportgroup remove_cluster` command deletes a cluster from the export group.

Options

name|n

The name of a valid export group. This is a mandatory parameter.

cluster|cl

The cluster name to be removed. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

varray|va

The name of a virtual storage array. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

force|f

The force flag is mandatory. Use this command with care, as it removes cluster access to all underlying storage.

synchronous|sync

Perform the operation in synchronous or asynchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 54

```
# viprcli exportgroup remove_cluster -n EXPORT_GROUP3 -cl
clustername -pr BLOCK_PROJECT -f
```

viprcli exportgroup remove_host

Delete a host from the export group.

Syntax

```
            viprcli exportgroup remove_host
[-h]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
-name|n <exportgroupname>
-hostlabel|hl <hostlabel>
-project|pr <projectname>
[-varray|va <varray>]
[-tenant|tn <tenantname>]
[-synchronous|sync]
-force|f
[-synctimeout|syncto]
```

Description

`viprcli exportgroup remove_host` command deletes a host from the export group. This command is silent on success.

Options

name|n

The name of a valid export group. This is a mandatory parameter.

hostlabel|hl

The host label of the host initiator to be searched. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

varray|va

Varray name. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

force|f

The force flag is mandatory. Use this command with care, as it removes host access to all underlying storage.

synchronous|sync

Perform the operation in synchronous or asynchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 55

```
# viprcli exportgroup remove_host -n EXPORT_GROUP2 -hl host_100 -pr
BLOCK_PROJECT -f
```

viprcli exportgroup remove_initiator

Remove an initiator from an export group. An initiator is a host port.

Syntax

```
            viprcli exportgroup remove_initiator
[-h]
[-hostname|hn        <hostname>]
[-port|po        <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
-name|n        <exportgroupname>
-initiator|in <initiator>        [<initiator> ...]
-hostlabel|hl        <hostlabel>
-project|pr        <projectname>
[-tenant|tn        <tenantname>]
[-varray|va <varray>]
[-synchronous|sync]
[-synctimeout|syncto]
```

Description

`viprcli exportgroup remove_initiator` command removes an initiator from an export group. An initiator is a host port.

Options

name|n

The name of a valid export group. This is a mandatory parameter.

initiator|in

The initiator name to be removed. This is a mandatory parameter.

hostlabel|hl

The host label from which the initiator to be searched. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

varray|va

The name of a virtual storage array. This is an optional parameter.

synchronous|sync

Perform the operation in synchronous or asynchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 56

```
viprcli exportgroup remove_initiator -n EXPORT_GROUP1 -pr
BLOCK_PROJECT -hl host_098 -initiator 'iqn.
2013-08.com.mycompany:localhost-4c92c642'
```

Example 57

```
# viprcli exportgroup remove_initiator -n EXPORT_GROUP1 -pr
BLOCK_PROJECT -hl host_100 -initiator 'iqn.
2013-08.com.mycompany:host_100-7e9a6791'
```

viprcli exportgroup remove_vol

Remove a volume or volume snapshot from an export group.

Syntax

```
viprcli exportgroup remove_vol
[-h]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-name|n        <exportgroupname>]
[-volume|v <Volume>        [<Volume> ...]
[-project|pr        <projectname>]
[-snapshot|sh <Snapshot for        volume> [<Snapshot for
volume> ...]]
[-consistencygroup|cg        <consistencygroup>]
[-tenant|tn        <tenantname>]
[-blockmirror|bmr <Block Mirror for        volume> [<Block Mirror
for volume> ...]]
[-varray|va <varray>]
[-synchronous|sync]
[-synctimeout|syncto]
```

Description

`viprcli exportgroup remove_vol` command removes a volume or volume snapshot from an export group. This command is silent on success

Options

name|n

The name of a valid export group. This is a mandatory parameter.

volume|v

The name of the volume to remove. This must be a valid volume within ViPR Controller. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

snapshot|sh

The name of a snapshot associated with the volume specified in the volume argument. This is an optional parameter.

consistencygroup|cg

Name of the consistency group. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

blockmirror|bmr

List of block mirrors (continuous copies). Include LUN ID, for example, <block_mirror_name>:<lun_id>. This is an optional parameter.

varray|va

The name of a virtual storage array.. This is an optional parameter.

synchronous|sync

Perform the operation in synchronous or asynchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 58

```
# viprcli exportgroup remove_vol
-n Example_ExpGroup_1G_7822726523 -v Example_ExpGroup_1G_7822726523
-pr Marketing
```

viprcli exportgroup show

Return the details for a specified export group in a project.

Syntax

```
            viprcli exportgroup show
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n
        <exportgroupname>
            -project|pr <project>
            [-tenant|tn
        <tenantname>]
            [-varray|va <virtual
        array>]
            [-xml]
```

Description

`viprcli exportgroup show` command returns information about an export group such as the following:

- A list of volumes in the export group
- A list of initiators in the export group
- The virtual storage arrays to which the export group belongs
- The project to which an export group belongs
- Whether the export group is active or inactive

Options

name|n

The name of a valid export group. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

varray|va

The name of the virtual storage array. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 59

```
viprcli exportgroup show -n Example_ExpGroup_1G_7822726523 -pr
exampleproj
{
   "clusters": [],
   "creation_time": 1376989761627,
   "generated_name": "ProviderT_Example
     _f79ca67e-ed18-46cc-91b9-f7751d3568cf",
   "hosts": [],
   "id": "urn:storageos:ExportGroup:
    f79ca67e-ed18-46cc-91b9-f7751d3568cf:",
   "inactive": false,
   "initiators": [
      {
         "creation_time": 1376989750907,
         "host": {
            "id": "urn:storageos:Host:
              6d98ccaf-9b47-431d-99ae-d1296f4461c1:",
            "link": {
               "href": "/compute/hosts/urn:storageos:Host:
                6d98ccaf-9b47-431d-99ae-d1296f4461c1:",
               "rel": "self"
            }
         },
         "hostname": "host.3007850695349452",
         "id": "urn:storageos:Initiator:
           6060ed9a-e982-420a-be43-e0f74b55ee1f:",
         "inactive": false,
         "initiator_node": "20:00:00:00:00:00:00:00",
         "initiator_port": "21:00:98:92:71:80:05:85",
         "link": {
            "href": "/compute/initiators/urn:storageos:Initiator:
               6060ed9a-e982-420a-be43-e0f74b55ee1f:",
            "rel": "self"
         },
         "protocol": "FC",
         "tags": []
      },
```

Example 59 (continued)


```
      {
         "creation_time": 1376989751025,
         "host": {
            "id": "urn:storageos:Host:
             6d98ccaf-9b47-431d-99ae-d1296f4461c1:",
            "link": {
               "href": "/compute/hosts/urn:storageos:Host:
                6d98ccaf-9b47-431d-99ae-d1296f4461c1:",
               "rel": "self"
            }
         },
         "hostname": "host.3007850695349452",
         "id": "urn:storageos:Initiator:
            99641cfd-03ac-47e2-afa9-24e2f8257f4f:",
         "inactive": false,
         "initiator_node": "20:00:00:00:00:00:00:00",
         "initiator_port": "21:00:87:57:04:65:04:85",
         "link": {
            "href": "/compute/initiators/urn:storageos:Initiator:
            99641cfd-03ac-47e2-afa9-24e2f8257f4f:",
            "rel": "self"
         },
         "protocol": "FC",
         "tags": []
      }
   ],
   "link": {
      "href": "/block/exports/urn:storageos:ExportGroup:
       f79ca67e-ed18-46cc-91b9-f7751d3568cf:",
      "rel": "self"
   },
   "name": "Example_ExpGroup_1G_7822726523",
   "project": {
      "id": "urn:storageos:Project:
            b6395993-28b0-44fa-ba9c-f2ede0814148:",
      "link": {
         "href": "/projects/urn:storageos:Project:
            b6395993-28b0-44fa-ba9c-f2ede0814148:",
         "rel": "self"
      }
   },
   "tags": [],
   "tenant": {
      "id": "urn:storageos:TenantOrg:
          f8c0562f-33ef-4994-a325-98ef7ffc1c03:",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:
           f8c0562f-33ef-4994-a325-98ef7ffc1c03:",
         "rel": "self"
      }
   },
   "type": "Exclusive",
   "varray": {
      "id": "urn:storageos:VirtualArray:
          501ed39e-834c-4dc0-baad-d16a5174482d:",
      "link": {
         "href": "/vdc/varrays/urn:storageos:VirtualArray:
                501ed39e-834c-4dc0-baad-d16a5174482d:",
         "rel": "self"
      }
   },
   "volumes": [
      {
         "id": "urn:storageos:Volume:
           7c8f0bc0-4ad5-4964-9421-3068f1fc35c4:"
```

Example 59 (continued)

```
        }
    ]
}
```

viprcli exportgroup tag

Add or delete a tag name to the export group.

Syntax

```
            viprcli exportgroup tag
-name|n        <exportgroupname>
-project|pr        <projectname>
[-add <addtag>        [<addtag>...]]
[-remove <removetag>        [<removetag>...]]
[-tenant|tn        <tenantname>]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-h]
```

Description

`viprcli exportgroup tag` command adds or deletes a tag name to the export group.

Options

name|n

The name of a valid export group. This is a mandatory parameter.

project|pn

The name of a ViPR Controller project. This is a mandatory parameter.

-add

The tag name to be added to the export group. This is an optional parameter.

-remove

The tag name to be deleted from the export group. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli exportgroup tasks

List all tasks within a given project of the export group.

Syntax

```
            viprcli exportgroup tasks
-project|pr <projectname>
[-name|n        <exportgroupname>]
[-verbose|v <verbose>
[-id <opid>]
[-tenant|tn        <tenantname>]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-h]
```

Description

`viprcli exportgroup tasks` command lists tasks within a specific project of the export group.

Options

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

name|n

The name of a valid export group. This is an optional parameter.

verbose|v

Lists all tasks. This is an optional parameter.

id

The operation identity of the export group. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli exportgroup changeportgroup

Changes to a portgroup can be done using the `viprcli exportgroup change_port_group` command.

Syntax

```
viprcli exportgroup change_port_group
            [-port|po <port_number>]
```

```
            [-portui|pu        <ui_port_number>]
            [-cookiefile|cf        <cookiefile>]
            [-project|<exportgroupname>]
            [-portgroup|<projectname>
            [-tenant|tn <tenantname>]
            [-varray|va <new_portgroup_name>]
            [-sn|<serialnumber>
            [-tenant<tenantname>]
            [-currentportgroup|<current_portgroup_name>
            {-exportmask|<exportmask>]
            {-wait][-verbose]
            [-h]
            [-hostname]
```

Description

The `viprcli exportgroup changeportgroup` command allows to change port group name.

Options

port|po

Port numver of the ViPR.

portui

https port number of ViPR Portal UI.

cookiefile|cf

Name of the current port group.

tenant|tn

Name of the container tenant.

currentportgroup|cpg

Name of the current port group.

exportmask|em

Export mask URI.

wait|w

Wait before removal of paths.

verbose|v

Print verbose output.

project|pr

The name of the container project anme. This is a mandatory parameter.

portgroup|new portgroup name

The name of the new storage port group. This is a mandatory parameter.

exportgroupname|n

The export group name. This is a mandatory parameter.

varray|va

Allows you to view the export groups for a specific virtual array. This is a mandatory parameter.

serial number|sn

Serial number o the storage system.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 60

```
viprcli exportgroup change_port_group -n demo-host -pr  TEST  -pg
Demo_PG -va v197 -sn 197
```

In this example, **TEST** is the `project`. **V197** is the `virtual array`. In the export group `Demo_PG` is the port group name that is changed from `demo_host`. Where`Demo_PG` is the current portgroup.

CHAPTER 18

viprcli filepolicy commands

This chapter contains the following topics:

- viprcli filepolicy assign 184
- viprcli filepolicy create 185
- viprcli filepolicy delete 187
- viprcli filepolicy list 188
- viprcli filepolicy show 189
- viprcli filepolicy unassign 190
- viprcli filepolicy update 191

viprcli filepolicy assign

Assigns a file policy to a vPool, project, or file system.

Syntax

```
            viprcli filepolicy assign
            -name|n <policy_name>
                                                [-applyontargetsite|
aptrgtsite] <apply_on_target_site>]
            [-assigntovpools|asignvpls <assign_to_vpools>]
                                                [-assigntoprojects|
asignprjs <assign_to_projects>]
            [-assigntoprojectsvpool|asignprjvpool
<project_assign_vpool>]
                                                [-sourcevarray|
srcvarray <source_varray>]
            [-targetvarrays|trgvarrays <target_varrays>]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu    <ui_port_number>]
            [-cookiefile|cf    <cookiefile>]
            [--help|h]
```

Description

The `viprcli filepolicy assign` command assigns a file policy to a vPool, project, or file system.

Mandatory Arguments

name|n

The name of the policy.

Optional Arguments

applyontargetsite|aptrgtsite

Apply on target site; true/false.

assigntovpools|asignvpls

Required for assigning file policies to vPool.

assigntoprojects|asignprjs

Required for assigning file policies to projects.

assigntoprojectsvpool|asignprjvpool

vPool of to-be-assigned projects. Required for assigning file policies to project.

sourcevarray|srcvarray

Source vArray for file replication.

targetvarrays|trgvarrays

Target vArrays for file replication.

hostname

ViPR Hostname (fully qualified domain name) or IPv4 address (i.e. 192.0.2.0) or IPv6 address inside quotes and brackets (i.e. "[2001:db8::1]").

port|po

Port number of ViPR.

portui|pu

https port number of ViPR Portal UI.

cookiefile|cf

Full name of cookiefile.

help|-h

Show this help message and exit.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filepolicy create

Creates a new file policy.

Syntax

```
            viprcli filepolicy create
            -name|n <policy_name>
            -type|t <policy_type>
            tenants_access|ta <is_access_to_tenants>
                                                [-apply_at|aplat
<apply_at>]
            [-description|dc <policy_description>]
                                                [-priority|pr
<priority>]
            [-num_worker_threads|wt <num_worker_threads>]
                                                [-
policyscheduleweek|plscwk <policy_schedule_week>]
            [-policyschedulemonth|plscmn <policy_schedule_month>]
            [-replicationconfiguration|repconf
<replicate_configuration>]
            [-snapshotnamepattern|snpnmptrn <snapshot_name_pattern>]
            [-snapshotexpiretype|snpexptp <snapshot_expire_type>]
            [-snapshotexpirevalue|snpexpv <snapshot_expire_value>]
            [-policyschedulefrequency|plscfr
<policy_schedule_frequency>]
            [-policyscheduleerepeat|plscrp <policy_schedule_repeat>]
            [-policyscheduletime|plsctm <policy_schedule_time>]
            [-replicationtype <replication_type>]|reptype]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu    <ui_port_number>]
            [-cookiefile|cf    <cookiefile>]
            [--help|h]
```

Description

The `viprcli filepolicy create` command creates a new file policy.

Mandatory Arguments

name|n

The name of the policy.

type|t

The policy type. Valid values are : file_snapshot, file_replication, file_quota.

tenants_access|ta

Tenants access.

Optional Arguments

apply_at|aplat

Level at which policy will be applied. Valid values are: vpool, project, file_system. Default: vpool.

description|dc

Policy description.

priority|pr

The policy priority. Valid values are: HIGH, LOW.

num_worker_threads|wt

Number of worker threads. Range: 3-10. Default: 3.

policyscheduleweek|plscwk

Day of the week when the policy runs.

policyschedulemonth|plscmn

Day of the month when the policy runs.

replicationconfiguration|repconf

Whether to replicate file system configurations (i.e., CIFS shares, NFS Exports) at the time of failover/failback. Default: False.

snapshotnamepattern|snpnmptrn

Snapshot pattern.

snapshotexpiretype|snpexptp

Snapshot expire type (e.g., hours, days, weeks, months, or never). Default: days.

snapshotexpirevalue|snpexpvl

the snapshot expires after this value. Default: 2.

policyschedulefrequency|plscfr

Frequency of schedule policy (e.g., days, weeks, or months). Default: days.

policyschedulerepeat|plscrp

The policy run on every <value>. Default: 1.

policyscheduletime|plsctm

Time when the policy runs. Default: 00:00.

replicationtype|reptype

The file replication type. Valid values: LOCAL, REMOTE. Default: REMOTE.

hostname

ViPR Hostname (fully qualified domain name) or IPv4 address (i.e. 192.0.2.0) or IPv6 address inside quotes and brackets (i.e. "[2001:db8::1]").

port|po

Port number of ViPR.

portui|pu

https port number of ViPR Portal UI.

cookiefile|cf

Full name of cookiefile.

help|-h

Show this help message and exit.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filepolicy delete

Deletes a file policy.

Syntax

```
viprcli filepolicy delete
-name|n <policy_name>
                    [-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu    <ui_port_number>]
[-cookiefile|cf    <cookiefile>]
[--help|h]
```

Description

The `viprcli filepolicy delete` command deletes a file policy.

Mandatory Arguments

name|n

The name of the policy.

Optional Arguments

hostname

ViPR Hostname (fully qualified domain name) or IPv4 address (i.e. 192.0.2.0) or IPv6 address inside quotes and brackets (i.e. "[2001:db8::1]").

port|po

Port number of ViPR.

portui|pu

https port number of ViPR Portal UI.

cookiefile|cf

Full name of cookiefile.

help|-h

Show this help message and exit.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filepolicy list

Outputs a list of file policies.

Syntax

```
viprcli filepolicy list
[-verbose|v]
[-long|l]
                        [-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu    <ui_port_number>]
[-cookiefile|cf    <cookiefile>]
[--help|h]
```

Description

The `viprcli filepolicy list` command outputs a list of file policies.

Optional Arguments

verbose|v

Lists file policies with details.

long|l

Lists file policies with details in table format.

hostname

ViPR Hostname (fully qualified domain name) or IPv4 address (i.e. 192.0.2.0) or IPv6 address inside quotes and brackets (i.e. "[2001:db8::1]").

port|po

Port number of ViPR.

portui|pu

https port number of ViPR Portal UI.

cookiefile|cf

Full name of cookiefile.

help|-h

Show this help message and exit.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filepolicy show

Displays a file policy.

Syntax

```
viprcli filepolicy show
-name|n <policy_name>
[-xml]
                        [-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu    <ui_port_number>]
[-cookiefile|cf    <cookiefile>]
[--help|h]
```

Description

The `viprcli filepolicy show` command displays a file policy.

Mandatory Arguments

name|n

The name of the policy.

Optional Arguments

xml

XML response.

hostname

ViPR Hostname (fully qualified domain name) or IPv4 address (i.e. 192.0.2.0) or IPv6 address inside quotes and brackets (i.e. "[2001:db8::1]").

port|po

Port number of ViPR.

portui|pu

https port number of ViPR Portal UI.

cookiefile|cf

Full name of cookiefile.

help|-h

Show this help message and exit.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filepolicy unassign

Unassigns a file policy.

Syntax

```
            viprcli filepolicy unassign
            -name|n <policy_name>
            -unassignresourcetype|unasngrestp
                                                    [-unassignvpools|
unasignvpls <unassign_from_vpools>]
            [-unassignfromprojects|unasignprjs
<unassign_from_projects>]
                                                    [-unassignfromfs|
unasignfs <unassign_from_filesystem>]
            [-tenant|tn <tenantname>]
                                                    [-project|pr
<projectname>]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu    <ui_port_number>]
            [-cookiefile|cf    <cookiefile>]
            [--help|h]
```

Description

The `viprcli filepolicy unassign` command unassigns a file policy from a vPool, project, or file system.

Mandatory Arguments

name|n

The name of the policy.

unassignresourcetype|unasngrestp

Resource type to be unassigned from. Type values: vpools, projects, filesystem.

Optional Arguments

unassignvpools|unasignvpls

Unassign from vPools.

unassignfromfs|unasignfs

Unassign from files ystem.

tenant|tn

Name of tenant.

project|pr

Name of project.

hostname

ViPR Hostname (fully qualified domain name) or IPv4 address (i.e. 192.0.2.0) or IPv6 address inside quotes and brackets (i.e. "[2001:db8::1]").

port|po

Port number of ViPR.

portui|pu

https port number of ViPR Portal UI.

cookiefile|cf

Full name of cookiefile.

help|-h

Show this help message and exit.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filepolicy update

Updates a file policy.

Syntax

```
            viprcli filepolicy update
            -existingpolname|epn <existing_policy_name>
            [-name|n <policy_name>]
                                                [-apply_at|aplat
<apply_at>]
            [-description|dc <policy_description>]
                                                [-priority|pr
<priority>]
            [-num_worker_threads|wt <num_worker_threads>]
                                                [-
policyscheduleweek|plscwk <policy_schedule_week>]
            [-policyschedulemonth|plscmn <policy_schedule_month>]
            [-replicationconfiguration|repconf
<replicate_configuration>]
            [-snapshotnamepattern|snpnmptrn <snapshot_name_pattern>]
            [-snapshotexpiretype|snpexptp <snapshot_expire_type>]
            [-snapshotexpirevalue|snpexpv <snapshot_expire_value>]
            [-policyschedulefrequency|plscfr
<policy_schedule_frequency>]
            [-policyscheduleerepeat|plscrp <policy_schedule_repeat>]
            [-policyscheduletime|plsctm <policy_schedule_time>]
            [-replicationtype <replication_type>]|reptype]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu    <ui_port_number>]
            [-cookiefile|cf    <cookiefile>]
            [--help|h]
```

Description

The `viprcli filepolicy update` command updates a file policy.

Mandatory Arguments

existingpolname|epn

The name of the existing policy to be updated.

Optional Arguments

name|n

The name of the policy.

apply_at|aplat

Level at which policy will be applied. Valid values are: vpool, project, file_system. Default: vpool.

description|dc

Policy description.

priority|pr

The policy priority. Valid values are: HIGH, LOW.

num_worker_threads|wt

Number of worker threads. Range: 3-10. Default: 3.

policyscheduleweek|plscwk

Day of the week when the policy runs.

policyschedulemonth|plscmn

Day of the month when the policy runs.

replicationconfiguration|repconf

Whether to replicate file system configurations (i.e., CIFS shares, NFS Exports) at the time of failover/failback. Default: False.

snapshotnamepattern|snpnmptrn

Snapshot pattern.

snapshotexpiretype|snpexptp

Snapshot expire type (e.g., hours, days, weeks, months, or never). Default: days.

snapshotexpirevalue|snpexpvl

The snapshot expires after this value. Default: 2.

policyschedulefrequency|plscfr

Frequency of schedule policy (e.g., days, weeks, or months). Default: days.

policyscheduleerepeat|plscrp

The policy run on every <value>. Default: 1.

policyscheduletime|plsctm

Time when the policy runs. Default: 00:00.

replicationtype|reptype

The file replication type. Valid values: LOCAL, REMOTE. Default: REMOTE.

hostname

ViPR Hostname (fully qualified domain name) or IPv4 address (i.e. 192.0.2.0) or IPv6 address inside quotes and brackets (i.e. "[2001:db8::1]").

port|po

Port number of ViPR.

portui|pu

https port number of ViPR Portal UI.

cookiefile|cf

Full name of cookiefile.

help|-h

Show this help message and exit.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

CHAPTER 19

viprcli filesystem commands

This chapter contains the following topics:

- viprcli filesystem assign-policy 196
- viprcli filesystem change-vpool 197
- viprcli filesystem create 198
- viprcli filesystem create-replication-copy 200
- viprcli filesystem delete 201
- viprcli filesystem delete-acl 202
- viprcli filesystem expand 203
- viprcli filesystem export 204
- viprcli filesystem export-rule 206
- viprcli filesystem failback-replication 207
- viprcli filesystem failover-replication 208
- viprcli filesystem list 209
- viprcli filesystem list-acl 210
- viprcli filesystem list-policy 211
- viprcli filesystem mount 212
- viprcli filesystem mountlist 213
- viprcli filesystem nfs-acl 214
- viprcli filesystem nfs-delete-acl 216
- viprcli filesystem nfs-list-acl 216
- viprcli filesystem pause-replication 217
- viprcli filesystem refresh-replication-copy 218
- viprcli filesystem remove-replication-copy 219
- viprcli filesystem resume-replication 220
- viprcli filesystem schedule-snapshots-list 221
- viprcli filesystem share-acl 222
- viprcli filesystem show 223
- viprcli filesystem show-exports 226
- viprcli filesystem show-shares 227
- viprcli filesystem start-replication 227
- viprcli filesystem stop-replication 228
- viprcli filesystem tag 229
- viprcli filesystem tasks 230
- viprcli filesystem unexport 231
- viprcli filesystem unmanaged ingest 233
- viprcli filesystem unmanaged show 233
- viprcli filesystem unmount 234
- viprcli filesystem update 235
- viprcli filesystem unassign-policy 236

viprcli filesystem assign-policy

Assigns a snapshot schedule policy to a file system.

Syntax

```
            viprcli filesystem assign-policy
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <filesystemname>
            -policyname|polnm
        <policy_name>
            -tenant|tn <tenantname>
            -project|pr <projectname>
```

Description

The `viprcli filesystem assign-policy` command assigns a snapshot schedule policy to a file system.

Options

name|n

A valid file system name. This is a mandatory parameter.

policyname|polnm

Name of policy, for example, policy5. This is a mandatory parameter.

tenant|tn

Name of tenant. This is a mandatory parameter.

project|pr

Name of ViPR Controller project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 61

```
viprcli filesystem assign-policy -n fs1 -polnm policy5 -tn
"Provider Tenant" -pr Project1
```

viprcli filesystem change-vpool

Moves a file system from one virtual pool to another.

Syntax

```
viprcli filesystem change-vpool
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -name|n <filesystemname>
    [-tenant|tn
<tenantname>]
    -project|pr <projectname>
    -vpool|vp <vpoolname>
```

Description

The `viprcli filesystem change-vpool` command moves a file system from one virtual pool to another.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

vpool|vp

Name of the target virtual pool. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 62

```
viprcli filesystem change-vpool -n fs3 -pr Project1 -vpool
vpoolprotectremote
```

viprcli filesystem create

Create a file system with the given parameters.

Syntax

```
viprcli filesystem create

[-h]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-name|n <filesystemname>
-size|s <size>
-project|pr
<projectname>
[-tenant|tn
<tenantname>]
-vpool|vp <vpoolename>
-varray|va <varray>
[-id <filesystemid>]
[-advisorylimit|advlmt
<advisorylimit>]
[-softlimit|softlmt
<softlimit>]
[-graceperiod|grace
<graceperiod>]
[-syncronous|sync]
[-synctimeout|syncto]
```

Description

`viprcli filesystem create` command creates a file system with the given parameters.

Options

name|n

The name of the file system. This is a mandatory parameter.

size|s

Filesystem size. Size must be followed by a size specifier (kK, mM, gG or tT) unit. K implies multiples of 1024 bytes, M implies multiples of (1024 * 1024) bytes, and so on. The default unit is bytes. Minimum size required to create a file system is 20 M. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

vpool|vp

A valid virtual pool. This is a mandatory parameter.

varray|va

A valid virtual storage array name. This is a mandatory parameter.

id

A predetermined filesystem ID to be used instead of the filesystem ID that ViPR Controller would generate. Supported only for VNX for File. This is an optional parameter.

advisorylimit|advlmt

Advisory limit in percentage for the file system This is an optional parameter.

softlimit|softlmt

Soft limit in percentage for the file system. This is an optional parameter.

graceperiod|grace

Grace period in days for the soft limit. This is an optional parameter.

synchronous|sync

Perform the file system creation synchronously. If this parameter is not specified, the file system creation is performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 63

```
viprcli filesystem create -n IMG -s 1G -project HR -vp SILVER -va
myvarray
{
"id": " urn:storageos:FileShare:8cd928a7-8679-48a2-b64c-
c94c5bb6aeb1 :",
 "task": "777b6960-048c-4212-afc3-cabd511d2749"
}
```

Example 64

```
viprcli filesystem create -n fstest5 -size 4GB -pr Project1 -vpool
vpool4 -varray varray1 -advlmt 20 -softlmt 30 -grace 5
```

viprcli filesystem create-replication-copy

Creates a replication copy of a file system.

Syntax

```
    viprcli filesystem create-replication-copy
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -name|n <filesystemname>
    [-tenant|tn <tenantname>
    -project|pr <projectname>
    [-targetname|tgn
<targetname>
    [-synchronous|sync]
    [-synctimeout|syncto]
```

Description

The `viprcli filesystem create-replication-copy` command creates a continuous copy of a file system.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

targetname|tgn

Name of target. This is an optional parameter.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 65

Example 65 (continued)

```
viprcli filesystem create-replication-copy -n fsnew4 -pr Project1
```

viprcli filesystem delete

Delete a file system.

Syntax

```
        viprcli filesystem delete
        [-h]
        [-hostname|hn
    <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        -name|n <filesystemname>
        [-tenant|tn
    <tenantname>]
        -project|pr <projectname>
        [-synchronous|sync]
        [-synctimeout|syncto]
        [-forceDelete|fd
    <forceDelete>]
        [-deleteType|dt
    <delete_type>]
```

Description

`viprcli filesystem delete` command deletes a file system. This command is silent on success.

A file system cannot be deleted until the following conditions are met:

- The file system is in unexported state. If the file system is exported to a host, you must unexport it (use the -unexport command) first before deleting the file system.
- All snapshots associated with the file system are deleted.

Options

name|n

A valid file system name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

synchronous|sync

Perform the file system deletion synchronously. If this argument is not specified, the file system deletion will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

forceDelete|fd

Forcibly delete the file system. By default this is set to false. This is an optional parameter.

deleteType

The full value (default) removes the file system and all objects referencing it from both the ViPR Controller database and its backend storage system. The inventory only value deletes the file system and all objects referencing it from the ViPR Controller.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filesystem delete-acl

Delete an ACL of a SMB share.

Syntax

```
        viprcli filesystem delete-acl
        -project|pr <projectname>
        -share|sh <SHARE>
        -name|n
    <filesystemname>
        [-tenant|tn
    <tenantname>]
        [-hostname <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        [-h]
```

Description

`viprcli filesystem delete-acl` command deletes an ACL of a SMB share.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

share|sh

The name of a SMB share. This is a mandatory parameter.

name|n

A valid file system name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filesystem expand

Expand the size of a file system.

Syntax

```
viprcli filesystem expand
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -name|n
<filesystemname>
    [-tenant|tn
<tenantname>]
    -project|pr <projectname>
    -size|s
<filesystemsize[kKmMgGtT]>
    [-synchronous|sync]
    [-synctimeout|syncto]
```

Description

`viprcli filesystem expand` command expands the size of a file system.

Options

name|n

A valid file system name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

size|s

The new size of the file system. This value must be higher than the current size of the file system, or the operation will fail. By default, this value is in bytes. You can append k, m, g, or t to the numeric value to allocate kilobytes, megabytes, gigabytes or terabytes. This is a mandatory parameter.

synchronous|sync

Perform the file system deletion synchronously. If this argument is not specified, the file system deletion will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 66

```
viprcli filesystem expand -size 2G -n isilon_fileshare -pr
TEST_PROJECT

viprcli filesystem list -pr TEST_PROJECT -l
```

Name	Capacity	Maxsnapshots	Multivolumeconsistency	Protocols
isilon_fileshare	2147483648	10	false	NFS

viprcli filesystem export

Export the file system to specified host protocol endpoints.

Syntax

```
            viprcli filesystem export

            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n
        <filesystemname>
            [-tenant|tn <tenant]
            -project|pr <projectname>
            [-security|sec {sys, krb5, krb5i,
        krb5p}]
            [-permission|pe {r | w | rw}]
            [-rootuser|ru
        <root_user>]
            [-endpoint|e <endpoint>
        [<endpoint>...] ]
            -protocol|pl {NFS | CIFS}
            [-share|sh <share>]
            [-description|desc
        <description>]
            [-permission_type|pt {allow |
        deny}]
            [-subdir| <sub
        directory>]
```

```
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

`viprcli filesystem export` command exports the file system to specified host protocol endpoints.

Options

name|n

A valid file system name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

security|sec

Security Type. Choose one or more of sys, krb5, krb5i, or krb5p. This is an optional parameter.

permission|pe

This is an optional parameter. One of the following:

- r (read permissions)
- w (write permissions)
- rw (read and write permissions)

This is an optional parameter.

rootuser|ru

Root user mapping for anonymous accesses (either nobody or root). This is an optional parameter.

endpoint|e

Endpoints to export a file system. For CIFS, NFS, or NFSv4 protocols the endpoint is an IP address or a file system share name. This is an optional parameter.

protocol|pl

Specify NFS (also for NFSv4), or CIFS. This is a mandatory parameter.

share|sh

The name of a Windows (SMB) share. This parameter should be used for CIFS exports only. This is an optional parameter.

description|desc

A string description of a windows share. This is an optional parameter.

permission_type|pt

File system access permissions. Allowable values are allow or deny. This is an optional parameter.

subdir

Unexport from File system subdirectory. This is an optional parameter.

synchronous|sync

Perform the file system deletion synchronously. If this argument is not specified, the file system deletion will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 67

```
viprcli filesystem export -n HR_Filesys
-e 192.168.1.2 -pl NFS -pr my_proj -ru root -pe rw -t sys
"id": "urn:storageos:FileShare:8cd928a7-8679-48a2-b64c-
c94c5bb6aeb1 :",
"task": "3436c953-d388-4d37-b2cb-6738ce81a45d"
```

viprcli filesystem export-rule

The export rules for a file system.

Syntax

```
            viprcli filesystem export-rule
-name|n <filesystemname>
-security|-sec {sys | krb5 | krb5i | krb5p}
-project|pr <projectname>
-operation|op <exportruleoperation>
[-roothosts|rhosts <roothosts> [<roothosts> ...]]
[-readonlyhosts|rohosts <readonlyhosts> [<readonlyhosts> ...]]
[-readwritehosts|rwhosts <readwritehosts> [<readwritehosts> ...]]
[-user|u <user>]
[-tenant|tn <tenant>]
[-subdir <sub directory>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli filesystem export-rule` command updates the export rules for a particular filesystem.

Options

name|n

A valid file system name. This is a mandatory parameter.

security|sec

Specify one or more of sys, krb5, krb5i, krb5p. This is a mandatory parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

operation|op

The export rule operation. This is a mandatory parameter

roothosts|rhosts

The root host names. This is an optional parameter.

readonlyhosts|rohosts

The read only host names. This is an optional parameter.

readwritehosts|rwhosts

The read/write only host names. This is an optional parameter.

user|u

A valid user. This is an optional parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

subdir

Unexport from File system subdirectory. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filesystem failback-replication

After a failover, use to failback from the target file system back to the original source file system.

Syntax

```
            viprcli filesystem failback-replication
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <filesystemname>
            [-tenant|tn
        <tenantname>]
            -project|pr <projectname>
                                                -replicateconfig
<true|false>
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

After a failover, use the `viprcli filesystem failback-replication` command to failback from the target file system back to the original source file system.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

replicateconfig

Specify true or false. If the configuration is replicated, CIFS shares, ACLs, NFS exports, and export rules are also included.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 68

```
viprcli filesystem failback-replication -n fssrc -pr Project1
```

viprcli filesystem failover-replication

Use to failover from the source file system to the target file system.

Syntax

```
            viprcli filesystem failover-replication
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <filesystemname>
            [-tenant|tn
        <tenantname>]
            -project|pr <projectname>
```

```
                                                        -replicateconfig
<true|false>
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

Use the `viprcli filesystem failover-replication` command to failover from the source file system to the target file system.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

replicateconfig

Specify true or false. If the configuration is replicated, CIFS shares, ACLs, NFS exports, and export rules are also included.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 69

```
viprcli filesystem failover-replication -n fssrc -pr Project1
```

viprcli filesystem list

Return a list of all file systems in the given project.

Syntax

```
            viprcli filesystem list
            -project|pr <projectname>
            [-tenant|tn <tenantname>]

            [-verbose|v]
            [-long|l]
```

```
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli filesystem list` command returns a list of all file systems in the given project. If the '-verbose' option is specified, additional details are provided. If the '-long' option is specified, thinly provisioned information along with filesystem name, capacity, and protocols is displayed.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

{verbose|v| long|l}

These options provide additional information. Use either verbose or long and not both.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 70

```
viprcli filesystem list -pr TEST_PROJECT -l
```

Name	Capacity	Maxsnapshots	Multivolumeconsistency	Protocols
isilon_fileshare	2147483648	10	false	NFS

viprcli filesystem list-acl

List the ACLs of a SMB share.

Syntax

```
viprcli filesystem list-acl
-project|pr <projectname>
-share|sh <SHARE>
```

```
            -name|n
        <filesystemname>
            [-tenant|tn
        <tenantname>]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli filesystem list-acl` command lists the ACLs of a SMB share.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

share|sh

The name of a SMB share. This is a mandatory parameter.

name|n

A valid file system name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filesystem list-policy

Lists the schedule policies that are assigned to a file system.

Syntax

```
            viprcli filesystem list-policy
            [-h]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n
        <filesystemname>
            -tenant|tn <tenantname>
            -project|pr <projectname>
```

Description

The `viprcli filesystem list-policy` command lists the schedule policies that are assigned to a file system.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 71

```
viprcli filesystem list-policy -n fs1 -tn "Provider Tenant" -pr
Project1
```

viprcli filesystem mount

Mount a previously created NFS Export of a file system to a Linux Host.

Syntax

```
            viprcli filesystem mount
            -project|pr <projectname>
            -hstname|hst <hostname>
            -mountpath|mp <mountpath>
               -fstype|f <fstype>
                                                  -filesystem|fs
<filesystem>
            -security|sec <security>
            [-hostname <hostname>]
            [-subdirectory|dir <subdirectory>]
            [-tenant|tn <tenantname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-synchronous|sync <synchronous>]
            [-synctimeout|syncto <SYNCTIMEOUT>]
            [-cookiefile|cf <cookiefile>]
            [-h|help]
```

Description

`viprcli filesystem mount` command mounts the target file system on the specified host. This allows the client to access the file system on the array.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

hostname|hst

The name of the Host on which the export is to be mounted. This is a mandatory parameter.

mountpath|mp

The mount path. This is a mandatory parameter.

fstype|f

The mount file system type. This is a mandatory parameter.

filesystem|fs

The name of the file system. This is a mandatory parameter.

security|sec

The security type for the mount. This is a mandatory parameter.

hostname

The hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0, or an IPv6 address inside quotes and brackets (for example, [2001:db8::1]) of ViPR. This is an optional parameter.

subdirectory|dir

The mount subdirectory. This is an optional parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

port|po

The ViPR Controller port number. This is an optional parameter.

portui|pu

The https port number of the ViPR Portal UI. This is an optional parameter.

synchronous|sync

Do a synchronous mount create. This is an optional parameter.

synctimeout|syncto

The sync timeout, in seconds. This is an optional parameter.

cookiefile|cf

The full name of the cookie file. This is an optional parameter.

common arguments

This operation also takes the Common Arguments.

viprcli filesystem mountlist

Return a list of all NFS exports of a file system in the specified project.

Syntax

```
viprcli filesystem mountlist
-project|pr <projectname>
```

```
                -filesystem|fs <filesystem>
            [-hostname <hostname>]
            [-tenant|tn <tenantname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookiefile>]
            [-h|help]
```

Description

`viprcli filesystem mountlist` command returns a list of all NFS exports of a file system in the specified project.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

filesystem|fs

The name of the file system. This is a mandatory parameter.

hostname

The hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0, or an IPv6 address inside quotes and brackets (for example, [2001:db8::1]) of ViPR. This is an optional parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

port|po

The ViPR Controller port number. This is an optional parameter.

portui|pu

The https port number of the ViPR Portal UI. This is an optional parameter.

cookiefile|cf

The full name of the cookie file. This is an optional parameter.

common arguments

This operation also takes the Common Arguments.

viprcli filesystem nfs-acl

To add, update, or delete an NFS ACL from a filesystem or subdirectory.

Syntax

```
        viprcli filesystem nfs-acl
        [-name|n
    <filesystem_name>]
        [-subdirectory
    <subdirectory_name>]
        [-project|pr
    <project_name>]
        [-operation
```

```
        <operation_name>]
            [-permission
        <permission_name>]
            [-permissiontype
        <type_of_permission>]
            [-domain <acl_domain>]
            [-type <user|group>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli filesystem nfs-acl` Delete the NFS access control list from a filesystem or subdirectory..

The following are mandatory options. See the CLI help for a list and description of optional parameters.

Options

name|n

Name of filesystem.

subdirectory

Name of subdirectory, which is only mandatory when performing the operation on a subdirectory.

project|pr

Name of project.

operation

ACL operation being performed on the filesystem or subdirectory: add, update, or delete.

permission

The permissions being applied to the access control entry: Read, Write, Execute, Read,write, Read,Execute, Read,Write,Execute

permissiontype

The type of permission to assign to the user or group: allow or deny.

domain

The user or group domain.

type

The type of users: user or group.

The following example is used to setup an NFS ACL on a subirectory of a file system:

```
viprcli filesystem nfs-acl -name CLI-FS-NFSv4-1 -operation add -
permissions Write,Execute -pr cli-nfsv4 -user jai -domain
provisioning.bourne.local -type user -permissiontype allow -
subdirectory W-E
```

viprcli filesystem nfs-delete-acl

Delete the NFS access control list from a filesystem or subdirectory.

Syntax

```
            viprcli filesystem nfs-delete-acl
            [-name|n
        <filesystem_name>]
            [-subdirectory
        <subdirectory_name>]
            [-project|pr
        <project_name>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli filesystem nfs-delete-acl` Delete the NFS access control list from a filesystem or subdirectory..

The following are mandatory options. See the CLI help for a list and description of optional parameters.

Options

name|n

Name of the filesystem.

subdirectory

Name fo subdirectory, which is only mandatory when removing the ACL from a subdirectory.

project|pr

Name of the project.

The following example is used to delete an NFS ACL from a subdirectory.

```
viprcli filesystem nfs-delete-acl -name CLI-FS-NFSv4-1 -project cli-
nfsv4 -subdirectory W-E
```

viprcli filesystem nfs-list-acl

Lists the NFS ACLs on a filesystem or a subdirectory.

Syntax

```
            viprcli filesystem nfs-list-acl
            [-name|n
```

```
        <filesystem_name>]
            [-subdirectory
        <subdirectory_name>]
            [-project|pr
        <project_name>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli filesystem nfs-list-acl` Lists the NFS ACLs on a filesystem or a subdirectory.

The following are mandatory options. See the CLI help for a list and description of optional parameters.

Options

name|n

Name of the filesystem.

subdirectory

Name fo subdirectory, which is only mandatory when removing the ACL from a subdirectory.

project|pr

Name of the project.

To list all the NFS ACL configured on all subdirectories of a filesystem:

```
viprcli filesystem nfs-list-acl -name CLI-FS-NFSv4-1 -project cli-
nfsv4 -alldir
DOMAIN USER PERMISSIONS PERMISSION_TYPE TYPE
provisioning.bourne.local jai Write,Execute allow user
```

viprcli filesystem pause-replication

Pauses the replication process between the source and target file system.

Syntax

```
            viprcli filesystem pause-replication
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <filesystemname>
            [-tenant|tn
        <tenantname>]
```

```
            -project|pr <projectname>
            [-synchronus|sync]
            [-synchtimeout|syncto]
```

Description

The `viprcli filesystem pause-replication` command pauses the replication process between the source and target file system.

Options

name|n

A valid file system name. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a valid ViPR Controller project. This is a mandatory parameter.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 72

```
viprcli filesystem pause-replication -n fssrc -pr Project1
```

viprcli filesystem refresh-replication-copy

Refreshes a replication copy of a file system.

Syntax

```
            viprcli filesystem refresh-replication-copy
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|puo
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <filesystemname>
            [-tenant|tn
        <tenantname>]
```

```
            -project|pr <projectname>
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

The `viprcli filesystem refresh-replication-copy` command refreshes a replication copy of a file system.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 73

```
viprcli filesystem refresh-replication-copy -n fs3 -pr Project1
```

viprcli filesystem remove-replication-copy

Removes a replication copy of the file system.

Syntax

```
            viprcli filesystem remove-replication-copy
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <filesystemname>
            [-tenant|tn
        <tenantname>]
```

```
            -project|pr <projectname>
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

The `viprcli filesystem remove-replication-copy` command removes a replication copy of the file system.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 74

```
viprcli filesystem remove-replication-copy -n fsnew4 -pr Project1
```

viprcli filesystem resume-replication

Resumes the replication process between the source and target file system.

Syntax

```
            viprcli filesystem resume-replication
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <filesystemname>
            [-tenant|tn
        <tenantname>]
```

```
            -project|pr <projectname>
            -synchronous|sync
            -synctimeout|syncto
```

Description

The `viprcli filesystem resume-replication` command resumes the replication process between the source and target file system.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 75

```
viprcli filesystem resume-replication -n fssrc -pr Project1
```

viprcli filesystem schedule-snapshots-list

Lists schedule snapshots.

Syntax

```
            viprcli filesystem schedule-snapshots-list
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <filesystemname>
            -policyname|polnm
        <policyname>
```

```
            -tenant|tn <tenantname>
            -project|pr <projectname>
```

Description

The `viprcli filesystem schedule-snapshots-list` command lists schedule snapshots.

Options

name|n

Name of file system. This is a mandatory parameter.

policyname|polnm

Name of policy. This is a mandatory parameter.

tenant|tn

Name of tenant. This is a mandatory parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filesystem share-acl

Add, update, and delete ACL rules for a file share.

Syntax

```
        viprcli filesystem share-acl
        -project|pr <projectname>
        -share|sh <SHARE>
        -name|n
    <filesystemname>
        -operation|op
    <acloperation>
        [-permission|perm
    <permission>]
        [-tenant|tn
    <tenantname>]
        [-user|-u <user>]
        [-domain|dom <domain>]
        [-group|grp <group>]
        [-hostname <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        [-h]
```

Description

`viprcli filesystem share-acl` command adds, updates, and deletes ACL rules for a file share.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

share|sh

The name of a SMB share. This is a mandatory parameter.

name|n

A valid file system name. This is a mandatory parameter.

operation|op

The CIFS ACL operation. This is an optional parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

user|u

The user. This is an optional parameter.

domain|dom

The domain. This is an optional parameter.

group|grp

The group. This is an optional parameter

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filesystem show

Returns details of a file system.

Syntax

```
viprcli filesystem show

-name|n <filesystemname>
-project|pr <projectname>
[-tenant|tn <tenantname>]
[-xml]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

The viprcli filesystem show command returns details of a file system.

Options

name|n

The name of the file system. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 76

```
viprcli filesystem show -n isilon_fileshare -pr TEST_PROJECT
{
"capacity": 1024,
 "vpool": {
 "vpool_params": [
{
"name": "multiVolumeConsistency",
"value": "false"
 },
{
 "name": "maxSnapshots",
 "value": "10"
}
 ],
"id": "urn:storageos:VirtualPool:96e1fe0f-78e5-4ed3-ae1a-
c4023e71de90:",
 "link": {
"href": "/file/vpool/urn:storageos:VirtualPool:96e1fe0f-78e5-4ed3-
ae1a-c4023e71de90:",
 "rel": "self"
}
 },
 "fs_exports": {
 "NFS.sys.rw.root": {
"clients": [
 "192.168.181.40"
 "fileExportKey": "NFS.sys.rw.root",
"isilon_id": "336",
 "mount_point": "192.168.96.132:/ifs/vipr/urn:storageos:FileShare:
0cc73ee8-d2e2-4eff-93e8-cfa904b9fb54:",
 "permissions": "rw",
"protocol": "NFS",
 "root_user_mapping": "root",
 "security_type": "sys",
 "storage_port": "192.168.96.132",
 "storage_port_name": "smartconnect"
 }
 },
 "id": "urn:storageos:FileShare:0cc73ee8-d2e2-4eff-93e8-
cfa904b9fb54:",
"inactive": false,
```

Example 76 (continued)


```
 "link": {
 "href": "/file/filesystems/urn:storageos:FileShare:0cc73ee8-
d2e2-4eff-93e8-cfa904b9fb54:",
"rel": "self"
 },
"mount_path": "/ifs/vipr/urn:storageos:FileShare:0cc73ee8-
d2e2-4eff-93e8-cfa904b9fb54:",
"name": "isilon_fileshare",
"varray": {
 "id": "urn:storageos:VirtualArray:3616d16f-c8b6-425f-bec3-
a6b5f025ca8c:",
"link": {
 "href": "/vdc/varrays/urn:storageos:VirtualArray:3616d16f-
c8b6-425f-bec3-a6b5f025ca8c:",
 "rel": "self"
 }
 },
 "project": {
"id": "urn:storageos:Project:a9f799de-c054-4a64-b0f3-866ebfefd413:",
 "link": {
 "href": "/projects/urn:storageos:Project:a9f799de-c054-4a64-
b0f3-866ebfefd413:",
 "rel": "self"
 }
 },
 "protocols": [
 "NFS"
],
 "storage_device": {
"id": "urn:storageos:StorageSystem:8fa06707-0614-43da-
a9e2-8d83100a773a:",
"link": {
 "href": "/vdc/storage-systems/urn:storageos:StorageSystem:
8fa06707-0614-43da-a9e2-8d83100a773a:",
"rel": "self"
}
},
 "storage_pool": {
 "id": "urn:storageos:StoragePool:51278dbf-35ac-4b93-
a95c-7fb151d03d94:",
 "link": {
 "href": "/vdc/storage-pools/urn:storageos:StoragePool:
51278dbf-35ac-4b93-a95c-7fb151d03d94:",
 "rel": "self"
}
},
"storage_port": {
 "id": "urn:storageos:StoragePort:95444e90-9820-4689-939b-
cd7d739fe098:",
 "link": {
 "href": "/vdc/storage-ports/urn:storageos:StoragePort:
95444e90-9820-4689-939b-cd7d739fe098:",
 "rel": "self"
 }
 }
}
```

viprcli filesystem show-exports

Return the export details of a filesystem.

Syntax

```
        viprcli filesystem show-exports
        -name|n
    <filesystemname>
        -project|pr
    <projectname>
        [-tenant|tn
    <tenantname>]
        [-subdir <subdirectory>
        [-alldir]
        [-hostname|hn
    <hostname>]
        [-port|po
    <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        [-h]
```

Description

`viprcli filesystem show-exports` command returns export details of a filesystem.

Options

name|n

The name of the file system. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

subdir

Name of the sub directory. This is an optional parameter.

alldir

Show file system export information for all directories. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filesystem show-shares

Return the shares of a filesystem.

Syntax

```
viprcli filesystem show-shares
-name|n <filesystemname>
-project|pr <projectname>
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli filesystem show-shares` command returns shares of a filesystem.

Options

name|n

The name of the file system. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filesystem start-replication

Manually starts, or restarts the replication process for continuous copies of file systems.

Syntax

```
viprcli filesystem start-replication
[-h]
[-hostname <hostname>]
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-name|n <filesystemname>
[-tenant|tn
<tenantname>]
-project|pr <projectname>
```

```
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

The `viprcli filesystem start-replication` command manually starts, or restarts the replication process for continuous copies of file systems.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 77

```
viprcli filesystem start-replication -n fssrc -pr Project1
```

viprcli filesystem stop-replication

Stops the replication process between the source and target file system.

Syntax

```
            viprcli filesystem stop-replication
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <filesystemname>
            [-tenant|tn <tenantname>
            -project|pr <projectname>
            [-synchronous|sync]
```

```
            [-synctimeout|syncto]
```

Description

The `viprcli filesystem stop-replication` command stops the replication process between the source and target file system.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 78

```
viprcli filesystem stop-replication -n fssrc -pr Project1
```

viprcli filesystem tag

Add or remove tags to the file system.

Syntax

```
            viprcli filesystem tag
[-h]
[-hostname|hn <hostname>]
[-port|po        <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
-name|n <filesystemname>
[-tenant|tn        <tenantname>]
-project|pr        <projectname>
[-add <addtag>        [<addtag>...]]
[-remove <removetag>        [<removetag>...]]
```

Description

`viprcli filesystem tag` command adds or deletes a tag name to the file systems.

Options

name|n

The name of a valid file system. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

add

The tag name to be added to the file system. This is an optional parameter.

remove

The tag name to be deleted from the file system. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli filesystem tasks

Check the status of an asynchronous task.

Syntax

```
viprcli filesystem tasks
-project|pr <projectname>
[-tenant|tn <tenantname>]
[-name|n <filesystemname>]
[-id <operationID>]
[-verbose|v]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli filesystem tasks` command checks the status of an asynchronous task.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

name|n

A valid file system name. This is an optional parameter.

id

An operation ID. Each asynchronous operation within ViPR Controller issues an operation id. This id must be provided through this CLI operation to retrieve information about an ongoing asynchronous task. This is an optional parameter.

verbose|v

Show additional details about asynchronous tasks. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 79

This command creates a virtual storage pool with the NFS protocol and type file.

```
viprcli filesystem tasks -pr TEST_PROJECT -name isilon_fileshare

OP_ID                                    NAME                 STATE
98f3c809-e22c-40eb-87e1-43579f2ae5d1     isilon_fileshare     ready
```

viprcli filesystem unexport

Unexport a file system.

Syntax

```
            viprcli filesystem unexport
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            name|n <filesystemname>
            [-tenant|tn <tenant>]
            -project|pr <projectname>
            -protocol|pl {NFS | NFSv4 |
        CIFS}
            [-share|sh <sharename>]
            [-subdir <sub
        directory>]
            [alldir]
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

`viprcli filesystem unexport` command unexports the specified file system. This makes the file system unavailable to users.

Options

name|n

A valid file system name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

protocol|pl

Specify NFS, NFSv4, or CIFS. This is a mandatory parameter.

share|sh

The name of a Windows (SMB) share. This parameter should be used for CIFS exports only. This is an optional parameter.

subdir

Unexport from file system subdirectory. This is an optional parameter.

alldir

Unexport file system from all directories, including sub-directories. This is an optional parameter.

synchronous|sync

Perform the file system deletion synchronously. If this argument is not specified, the file system deletion will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 80

```
viprcli filesystem unexport -n HR_Filesys -pl NFS -pr my_proj -ru
root -pe rw -sec sys
{
   "id": "urn:storageos:FileShare:8cd928a7-8679-48a2-b64c-
c94c5bb6aeb1:",
   "task": "13d06473-8609-4b74-903a-911a87d93d9f"
}
```

viprcli filesystem unmanaged ingest

Ingest unmanaged filesystems into ViPR Controller.

Syntax

```
viprcli filesystem unmanaged ingest
-vpool|vp <vpoolname>
-project|pr <projectname>
-varray|va <varray>
-filesystems|fs <filesystems> [<filesystems>...]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli filesystem unmanaged ingest` command ingests the specified unmanaged file system into ViPR Controller.

Options

vpool|vp

A valid virtual pool. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

varray|va

A valid virtual array name. This is a mandatory parameter.

filesystems|fs

Names or Ids of the file systems to ingest. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli filesystem unmanaged show

Return the details of unmanaged file systems.

Syntax

```
viprcli filesystem unmanaged show
-filesystems|fs <filesystems> [<filesystems>...]
[-hostname|hn <hostname>]
[-port|po <port_number>]
```

```
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookiefile>]
            [-h]
```

Description

`viprcli filesystem unmanaged show` command returns the details of unmanaged file systems.

Options

filesystems|fs

Names or Ids of the file systems. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli filesystem unmount

Unmount a previously mounted NFS Export from a Linux Host.

Syntax

```
            viprcli filesystem unmount
            -project|pr <projectname>
            -hstname|hst <hostname>
            -mountpath|mp <mountpath>
                  -filesystem|fs <filesystem>
            [-hostname <hostname>]
            [-tenant|tn <tenantname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-synchronous|sync <synchronous>]
            [-synctimeout|syncto <SYNCTIMEOUT>]
            [-cookiefile|cf <cookiefile>]
            [-h|help]
```

Description

`viprcli filesystem unmount` command unmounts the target file system on the specified host.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

hostname|hst

The name of the Host on which the export is mounted. This is a mandatory parameter.

mountpath|mp

The mount path. This is a mandatory parameter.

filesystem|fs

The name of the file system. This is a mandatory parameter.

hostname

The hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0, or an IPv6 address inside quotes and brackets (for example, [2001:db8::1]) of ViPR. This is an optional parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

port|po

The ViPR Controller port number. This is an optional parameter.

portui|pu

The https port number of the ViPR Portal UI. This is an optional parameter.

synchronous|sync

Do a synchronous unmount. This is an optional parameter.

synctimeout|syncto

The sync timeout, in seconds. This is an optional parameter.

cookiefile|cf

The full name of the cookie file. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli filesystem update

Updates the file system.

Syntax

```
    viprcli filesystem update
    [-h]
    [-hostname <hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -name|n <filesystemname>
    [-tenant|tn <tenantname>]
    -project|pr
<projectname>
    [-advisorylimit|advlmt
<advisorylimit>]
    [-softlimit|softlmt
<softlimit>]
    [-graceperiod|grace
<graceperiod>]
```

Description

The `viprcli filesystem update` command updates the file system.

Options

name|n

Name of file system. This is a mandatory parameter.

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of a ViPR Controller project. This is a mandatory parameter.

advisorylimit|advlmt

Advisory limit in percentage for the file system This is an optional parameter.

softlimit|softlmt

Soft limit in percentage for the file system. This is an optional parameter.

graceperiod|grace

Grace period in days for the soft limit. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 81

```
viprcli filesystem update -n fs4 -pr Project1 -advlmt 40
```

viprcli filesystem unassign-policy

Unassigns a schedule policy that is assigned to a file system.

Syntax

```
        viprcli filesystem usassign-policy
        [-h]
        [-hostname <hostname>]
        [-port|po
    <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        -name|n <filesystemname>
        -policyname|polnm
    <policyname>
        -tenant|tn <tenantname>
        -project|pr <projectname>
```

Description

The `viprcli filesystem usassign-policy` unassigns a schedule policy that is assigned to a file system.

Options

name|n

Name of file system. This is a mandatory parameter.

policyname|polnm

Name of policy, for example, policy1. This is a mandatory parameter.

tenant|tn

Name of tenatn. This is a mandatory parameter.

project|pr

The name of a ViPR Controller project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 82

```
viprcli filesystem unassign-policy -n fs1 -polnm policy1 -tn
"Provider Tenant" -pr Project1
```

CHAPTER 20

viprcli host commands

This chapter contains the following topics:

- viprcli host compute-host-os-install 240
- viprcli host create 241
- viprcli host delete 243
- viprcli host detach 244
- viprcli host discover 245
- viprcli host discover-array-affinity 246
- viprcli host list 246
- viprcli host list-initiators 247
- viprcli host list-ipinterfaces 248
- viprcli host list-umexportmasks 249
- viprcli host list-umvolumes 250
- viprcli host provision-bare-metal-host 250
- viprcli host show 251
- viprcli host tasks 253
- viprcli host update 254

viprcli host compute-host-os-install

Installs an operating system on a host.

Syntax

```
    viprcli host compute-host-os-install
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -computeimage|ci
<computeimagenametname>
    -volume|vol <volume>

    -project|pr <project>
    [-tenant|t <tenant>]
    -name|n <hostname>
    -newhostname|nhn
<newhostname>
    [-hostip|ip <hostip>]
    [-netmask|nm <netmask>]
    [-gateway|gw <gateway>]
    [-ntpserver|ns
<ntpserver>]
    [-dnsservers|ds
<dnsservers>]
    [-managementnetwork|mn
<managementnetwork>]
    [-forceinstallation|ump
<forceinstallation>]
    [-synchronous|sync]
    [-synctimeout|syncto]
```

Description

`viprcli host compute-host-os-install` command installs an operating system on a host.

Options

computeimage|ci

The name of the compute image. This is a mandatory parameter.

volume|vol

The name of the volume. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

tenant|t

The name of the tenant. This is an optional parameter.

name|n

The name of a host. This is a mandatory parameter.

newhostname|nhn

The new name of a host. This is a mandatory parameter.

hostip|ip

Host IP of the new host. This is an optional parameter.

netmask|nm

IP address of net mask. This is an optional parameter.

gateway|gw

IP address of gateway. This is an optional parameter.

ntpserver|ns

IP address of NTP server. This is an optional parameter.

dnsservers|ds

IP address of DNS server. This is an optional parameter.

managementnetwork|mn

Management network. This is an optional parameter.

forceinstallation|ump

Specifies whether to force the installation. This is an optional parameter.

synchronous|sync

Run in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli host create

Create a host with the specified name.

Syntax

```
    viprcli host create
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -type <type>
{Windows,HPUX,Linux,Esx,Other,AIXVIO,AIX,No_OS,SUN-VCS}
    -hostlabel|hl
<hostlabel>
    -viprhostname|hn
<viprhostname>
    [-hostport|hp <hostport>]
    -hostusername|un
```

```
        <hostusername>
            [-tenant|tn <tenant>]
            [-osversion|ov
        <hostosversion>]
            [-cluster|c <cluster>]
            [-datacenter|dc
        <datacenter>]
            [-vcenter|vc
        <vcentername>]
            [-hostusessl|hostssl <hostusessl>
        [<hostusessl>...] {true,false}]
            [-autodiscovery|discover
        {true,false}]
            [-bootvolume|bvol
        <bootvolume>]
            [-project|pr <project>]
            [-testconnection|tc]
```

Description

`viprcli host create` command creates a host with the specified name. This command is silent on success.

Note

After creating a host, you can discover it from ViPR Controller Admin and Self-Service UI.

Options

type

Operating system type of the host. This is a mandatory parameter. Possible types are: Windows, HPUX, Linux, Esx, Other, AIXVIO, AIX, No_OS, or SUNVCS.

host label|hl

Label for the host. This is a mandatory parameter.

viprhostname|hn

The ViPR Controller host name. This is a mandatory parameter.

Note

`hn` is not the short form for `hostname` in all `viprcli host` commands.

---

hostport|hp

The management interface port number for the host. This is a mandatory parameter.

hostusername|un

The ViPR Controller host user name. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

osversion|ov

The operating system version number of host. This is an optional parameter.

cluster|c

The name of the cluster host. This is an optional parameter.

datacenter|dc

The name of the datacenter for the host. This is an optional parameter.

vcenter|vc

The name of the vCenter. This is an optional parameter.

hostusessl|hostssl

SSL flag for the host. Possible values are: true or false. This is an optional parameter.

autodiscovery|discover

Boolean value to enable or disable auto discovery of host. This is an optional parameter.

bootvolume|bvol

The name of the boot volume. This is an optional parameter.

project|pr

The name of the project. This is an optional parameter.

testconnection|tc

Used to validate the connection. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 83

```
viprcli host create -hn 192.168.1.2 -type Linux
-hl viprhost25.example.mycompany.com -hp 4443 -un root
Enter password of the host:
Retype password:
```

viprcli host delete

Delete a host.

Syntax

```
            viprcli host delete
            -name|n <viprhostname>
            -type|t <type> { Windows | Linux
        | HPUX | Esx | AIXVIO | AIX | No_OS |SUN_VCS |Other }
            [-detachstorage|ds]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
```

```
            [-tenant|tn <tenant>]
            [-h]
```

Description

`viprcli host delete` command deletes the host of specified name and type. This command is silent on success.

Options

name|n

The ViPR Controller host name to delete. This is a mandatory parameter.

type|t

Operating system type of the host. This is a mandatory parameter. Possible types are: Windows, Linux, HPUX, Esx, AIXVIO, AIX, No_OS, SUN_VCS or Other. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

detachstorage|ds

Detach storage before activation. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 84

```
 # viprcli host delete -n xyz825.lss.emc.com -t Linux
```

Use list command to view the results.

```
# viprcli host list
  NAME                 HOST_NAME            TYPE
  vipr_vh.example.mycompany.com vipr_vh.example.mycompany.com Esx
```

viprcli host detach

Detach a host.

Syntax

```
            viprcli host detach
            [-name|n <viprhostname>]
            [-type|t {Windows| HPUX | Linux | Esx |
        AIXVIO | AIX | No_OS | SUN_VCS | Other}]
            [-tenant|tn
```

```
<tenantname>]
    [-hostname|hn <hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

`viprcli host detach` command detaches a host.

Options

name|n

The name of the host. This is a mandatory parameter.

type|t

The type of the host. This is a mandatory parameter. Possible types are Windows, HPUX, Linus, Esx, AIXVIO, AIX, No_OS, SUN_VCS, or Other.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli host discover

Discover a host.

Syntax

```
    viprcli host discover
    [-name|n <viprhostname>]
    [-type|t {Windows | HPU | Linux | Esx |
AIXVIO | AIX | No_OS | SUN_VCS | Other}]
    [-tenant|tn
<tenantname>]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

`viprcli host discover` command discovers a host.

Options

name|n

The name of the host. This is a mandatory parameter.

type|t

The type of the host. This is a mandatory parameter. Possible types are Windows, HPU, Linux, Esx, AIXVIO, AIX, No_OS, SUN_VCS, or Other .

tenant|tn

The name of the tenant.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli host discover-array-affinity

Discovers host/array affinity for a given host, or hosts when the storage is provisioned ot the host from VMAX, VNX for Block, Unity, and XtremIO storage systems.

Syntax

```
viprcli host discover_array_affinity
[-hostnames|hn    <hostnames>]
```

Options

hostname|hn

The name of a host. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

```
./viprcli host discover-array-affinity -hn lglw7145.lss.emc.com
```

viprcli host list

Display the list of hosts.

Syntax

```
            viprcli host list
            [-hosttype|ht <hosttype> {
        Windows | Linux | HPUX | Esx | AIXVIO | AIX | No_OS |
SUN_VCS |
        Other}]
            [-tenant|tn <tenant>]
            [{-verbose|v <verbose> | -long|l
        <long>|]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
```

```
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli host list` command displays the list of hosts.

Note

If you created a host with the viprcli host create command using the -tenant option, the hosts do not appear after running the viprcli host list command unless you also include the -tenant option.

Options

hosttype|ht

Type or operating system of the host. This is an optional parameter. Possible host types are: Windows, Linux, HPUX, Esx, AIXVIO, AIX, No_OS, SUN_VCS, or Other.

tenant|tn

The name of the tenant. This is an optional parameter.

{verbose|v| long|l}

These options display additional information about host. Use only either verbose or long and not both.

common arguments

This operation also takes the Common Parameters on page 24.

Example 85

```
# viprcli host list
  NAME                  HOST_NAME            TYPE
  vipr_vh.example.mycompany.com vipr_vh.example.mycompany.com Esx
  vipr_host25.example.mycompany.com  192.168.1.2         Linux
```

viprcli host list-initiators

Display the list of initiators for the specified host label.

Syntax

```
            viprcli host list-initiators
            -hostlabel|hl
        <hostlabel>
            [{-verbose|v <verbose> | -long|l
        <long>}]
            [-hostname <hostname>]
            [-port|po
```

```
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli host list-initiators` command lists the initiators for the specified host label.

Options

hostlabel|hl

Label of the host for which initiators are listed. This is a mandatory parameter.

{verbose|v | long|l}

The verbose parameter lists initiators with details. The long parameter lists initiators in a large table. Use only either verbose or long and not both. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 86

```
# viprcli host list-initiators -hl vipr_host.example.mycompany.com
 NAME    PROTOCOL HOST_NAME
 vmhba2  FC
 vmhba3  FC
 vmhba33 iSCSI
```

viprcli host list-ipinterfaces

Display the list of IP interfaces for a specified host label.

Syntax

```
            viprcli host list-ipinterfaces
            -hostlabel|hl
        <labelname>
            [{-verbose|v <verbose> | -long|l
        <long> }]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
```

```
            [-h]
```

Description

`viprcli host list-ipinterfaces` command displays the list of IP interfaces for the specified host label.

Options

hostlabel|hl

Label of the host for which IP interfaces are listed. This is a mandatory parameter.

{verbose|v | long|l }

The verbose parameter lists initiators with details. The long parameter lists initiators in a large table. Use only either verbose or long and not both. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 87

```
# viprcli host list-ipinterfaces -hl vipr_host.example.mycompany.com
  NAME      IP_ADDRESS     PROTOCOL
  vmk0/ipv4 192.168.1.1  IPV4
  vmk1/ipv4 192.168.1.2 IPV4
```

viprcli host list-umexportmasks

Display the list of un-managed export masks on a host.

Syntax

```
            viprcli host list-umexportmasks
            -hostlabel|hl HOSTLABEL
            [{-verbose|v <verbose>]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli host list-umexportmasks` command displays the list of un-managed export masks on a host.

Options

hostlabel|hl

The label of the host. This is an mandatory parameter.

verbose|v

This option displays additional information about the host. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli host list-umvolumes

Display the list of un-managed volumes on a host.

Syntax

```
viprcli host list-umvolumes
-hostlabel|hl HOSTLABEL
[{-verbose|v <verbose>]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookie_file>]
[-h]
```

Description

`viprcli host list-umvolumes` command displays the list of un-managed volumes on a host.

Options

hostlabel|hl

The label of the host. This is an mandatory parameter.

verbose|v

Lists initiators with details. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli host provision-bare-metal-host

Creates hosts using compute elements from the virtual compute pool.

Syntax

```
viprcli host provision-bare-metal-host
[-h]
```

```
            [hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-tenant|t <tenant>]
            [-cluster|c <cluster>]
            -computevpool|cvp
        <computevpoolname>
            -hostnames|hn <hostnames>
        [<hostnames...]]

            -varray|va <varrayname>
```

Description

`viprcli host provision-bare-metal-host` command creates hosts using compute elements from the compute virtual pool.

Options

tenant|t

Name of the tenant. This is an optional parameter.

cluster|c

Name of the cluster for the host. This is an optional parameter.

computevpool|cvp

Name of a compute virtual pool. This is a mandatory parameter.

hostnames|hn

A list of host names separated by a space. This is a mandatory parameter.

varray|va

Name of the virtual storage array. This is a mandatory parameter.

tenant|t

Name of the tenant. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli host show

Return the details of a host.

Syntax

```
            viprcli host show
-type|t <type> { Windows | Linux        | HPUX | Esx | AIXVI | AIX
| No_OS | SUN_VCS | Other }
[-name|n <name>]
[-xml <xmlresponse>]
[-tenant|tn <tenant>]
```

```
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-perferred_pools        <preferred_pools>]
[-cookiefile|cf        <cookie_file>]
[-h]
```

Description

`viprcli host show` command returns the host details.

Options

type|t

Operating system type of the host. This is a mandatory parameter. Possible types are: Windows, Linux, HPUX, Esx, AIXVIO, AIX, No_OS, SUN_VCS or Other.

name|n

Name of the host. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 88

```
# viprcli host show -name vipr_host25.example.mycompany.com
-type Linux
{
   "compatibility_status": "UNKNOWN",
   "creation_time": 1379759104944,
   "host_name": "192.168.1.2",
   "id": "urn:storageos:Host:00c87f9a-d7fd-4035-a69e-29c1d12f19ca:",
   "inactive": false,
   "job_discovery_status": "CREATED",
   "job_metering_status": "CREATED",
   "last_discovery_run_time": 0,
   "last_metering_run_time": 0,
   "link": {
      "href": "/compute/hosts/urn:storageos:Host:
      00c87f9a-d7fd-4035-a69e-29c1d12f19ca:",
      "rel": "self"
   },
   "name": "vipr_host25.example.mycompany.com",
   "native_guid": "",
   "next_discovery_run_time": 0,
   "next_metering_run_time": 0,
   "port_number": 4443,
   "preferred_pools": [],
   "registration_status": "REGISTERED",
   "tags": [],
   "tenant": {
      "id": "urn:storageos:TenantOrg:
```

Example 88 (continued)

```
        dbe23c11-4d96-4fa5-a617-01c8906eef9d:",
        "link": {
           "href": "/tenants/urn:storageos:TenantOrg:
           dbe23c11-4d96-4fa5-a617-01c8906eef9d:",
           "rel": "self"
        }
    },
    "type": "Linux",
    "use_ssl": false,
    "user_name": "root"
}
```

viprcli host tasks

Check the tasks of a host.

Syntax

```
viprcli host tasks
[-name|n <viprhostname>]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-id <opid>]
[-verbose|v]
[-h]
```

Description

`viprcli host tasks` command checks the tasks of a host.

Options

name|n

The name of the host. This is a mandatory parameter.

tenant|tn

The name of the tenant.

id

The operation ID of a host.

verbose|v

List all the tasks of a host.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli host update

Update a host with specified details.

Syntax

```
            viprcli host update
            -hostlabel|hl
        <hostlabel>
            [-newviprhostame|nhn
        <newviprhostname>]
            [-newtype|nt <newhosttype> {
        Windows | Linux | HPUX | Esx | Other | AIXVIO | AIX | SUN-
VCS |
        No_OS}]
            [-newlabel|nl
        <newlabelname>]
            [-newhostport|nhp
        <newhostport>]
            [-newhostusername|nun
        <newhostusername>]
            [-tenant|tn <tenant>]
            [-hostusessl|hostssl <hostusessl>
        {true | false}]
            [-newosversion|nov
        <newhostosversion>]
            [-newcluster|nc
        <newcluster>]
            [-newdatacenter|ndc
        <newdatacenter>]
            [-vcenter|vc
        <vcentername>]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-autodiscovery|discover
        <autodiscover> {true | false}]
            [-bootvolume|bvol
        <bootvolume>]
            [-project|pr <project>]
            [-testconnection|tc
        <testconnection>]

            [-h|help]
```

Description

`viprcli host update` command updates the host with the specified details. This command is silent on success.

Note

After updating a host, you can discover it from ViPR Controller Admin and Self-Service UI.

Options

hostlabel|hl

Search label for the host. This is a mandatory parameter.

newviprhostname|nhn

The new ViPR Controller host name. This is an optional parameter.

newtype|nt

New operating system type of the host. This is an optional parameter. Possible types are: Windows, Linux, HPUX, Esx, Other, AIXVIO, AIX, SUN-VCS, and No_OS.

newlabel|nl

New label for the host. This is an optional parameter.

newhostport|nhp

The new port number for the host. This is an optional parameter.

newhostusername|nun

The new host user name. This is an optional parameter.

hostusessl|hostssl

SSL flag for the host. Possible values are: true or false. This is an optional parameter.

osversion|ov

The operating system version number of host. This is an optional parameter.

newcluster|nc

The name of the new cluster host. This is an optional parameter.

newdatacenter|ndc

The name of the new datacenter for the host. This is an optional parameter.

vcenter|vc

The name of the vCenter. This is an optional parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

hostname

The fully qualified domain name, IPv4 address, or the IPv6 address inside quotes and brackets (for example, "[2001:db8::1]") of ViPR Controller. This is an optional parameter.

port|po

The port number of ViPR Controller. This is an optional parameter.

portui|pu

The https port number of the ViPR Controller Portal UI. This is an optional parameter.

cookiefile|cf

The full name of the cookie file. This is an optional parameter.

autodiscovery|discover

Boolean value to enable and disable automatic discovery of the host. Possible values: true or false. This is an optional parameter.

bootvolume|bvol

The name of the boot volume. This is an optional parameter.

project|pr

The name of the project. This is an optional parameter.

testconnection|tc

Validate connection to the host. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 89

```
# viprcli host update -hl vipr_host25.example.mycompany.com
-nl xyz825.example.mycompany.com
```

CHAPTER 21

viprcli initiator commands

This chapter contains the following topics:

- viprcli initiator aliasget 258
- viprcli initiator aliasset 258
- viprcli initiator create 260
- viprcli initiator delete 261
- viprcli initiator list 262
- viprcli initiator show 263
- viprcli initiator tasks 264
- viprcli initiator update 265

viprcli initiator aliasget

Lists the aliases used for VMAX storage system initiator world wide port names (WWPN).

Syntax

```
viprcli initiator aliasget
-initiatorportwwn|pwwn
hostlabel|hl
name|n
type|t
```

Options

The following are the mandatory options that must be used to list the aliases.

initiatorportwwn|pwwn

Specify a Worldwide Port Name (WWPN) for Fibre Channel and IQN, or EUI for iSCSI.

hostlabel|hl

The label name for the host.

name|n

The storage system name.

-type|t

The type of storage system.

```
./viprcli initiator aliasget -pwwn  10:00:00:90:FA:66:22:A5 -hl
HarshaTest -n  SYMMETRIX+000196801612 -t vmax

TestNode/TestPort
```

Where Test/Node/TestPort is the output. The output format is Nodename/Portname.

viprcli initiator aliasset

Adds an alias to a VMAX storage system initiator world wide port name (WWPN).

Syntax

```
viprcli initiator aliasset
-initiatorportwwn|pwwn
hostlabel|hl
name|n
type|t
-initiatoralias|ia
```

Options

The following are the mandatory options that must be used to list the aliases.

initiatorportwwn|pwwn

Specify a Worldwide Port Name (WWPN) for Fibre Channel and IQN, or EUI for iSCSI.

hostlabel|hl

The label name for the host.

name|n

The storage system name.

-type|t

The type of storage system.

--initiatoralias|ia

The alias to assign to the initiator. For the initiator alias you can use either the alias name or Nodename/Portname.

Example using the alias name format

```
./viprcli initiator aliasset -pwwn  10:00:00:90:FA:66:22:A5 -hl
HarshaTest -n  SYMMETRIX+000196801612 -t vmax -ia 10000090fa6622a5
```

```
/viprcli initiator aliasget -pwwn  10:00:00:90:FA:66:22:A5 -hl
HarshaTest -n  SYMMETRIX+000196801612 -t vmax
10000090fa6622a5/10000090fa6622a5
```

Where 10000090fa6622a5/10000090fa6622a5 are the assigned aliases.

Example using the Nodename/Portname format:

```
./viprcli initiator aliasset -pwwn  10:00:00:90:FA:66:22:A5 -hl
HarshaTest -n  SYMMETRIX+000196801612 -t vmax -ia HTestnode/
HTestPort
```

```
./viprcli initiator aliasget -pwwn  10:00:00:90:FA:66:22:A5 -hl
HarshaTest -n  SYMMETRIX+000196801612 -t vmax
HTestnode/HTestPort
```

Where HTestnode/HTestPort are the assigned aliases.

viprcli initiator create

Create an initiator. An initiator is a host port.

Syntax

```
        viprcli initiator create
        [-h]
        [-hostname <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        -hostlabel|hl <hostlabel>
        -protocol|pl <protocol>
    {FC,iSCSI}
        [-initiatorwwn|wwn
    <initatorwwn>]
        [-synchronous|sync]

        [-synctimeout|syncto]

        [-initiatorname|initname
    <initiatorname>]

        [-tenantname|tn <tenantname>]

        -initiatorportwwn|pwwn
    <initiatorportwwn>
```

Description

`viprcli initiator create` command creates an initiator. An initiator is a host port.

Options

hostlabel|hl

The label name for the host. This is a mandatory parameter.

protocol|pl

Specify FC or iSCSI. This is a mandatory parameter.

initiatorwwn|wwn

The initiator specification is a string. Its format depends upon the protocol.

- For Fibre Channel, specify the Worldwide Node Name (WWNN).
- For iSCSI specify the initiator node in IQN format.

This is an optional parameter.

synchronous|sync

Execute in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

initiatorname|initname

Initiator alias name. This is an optional parameter.

tenantname|tn

Tenant name. This is an optional parameter.

initiatorportwwn|pwwn

Specify a Worldwide Port Name (WWPN) for Fibre Channel and IQN, or EUI for iSCSI. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 90

```
# viprcli initiator create -pwwn iqn.
1998-01.com.vmware:vipr_host-7e9a6791 -pl iSCSI -hl
vipr_host.example.mycompany.com
```

viprcli initiator delete

Delete the specified initiator.

Syntax

```
            viprcli initiator delete
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -initiatorportwwn|pwwn
        <initatorportwwn>
            -hostlabel|hl
        <hostlabel>
            [-tenantname|tn
        <tenantname>]
```

Description

`viprcli initiator delete` command deletes the specified initiator. This command is silent on success.

Options

initiatorportwwn|pwwn

Initiator Worldwide Port Name (WWPN) for Fibre Channel and IQN or EUI for iSCSI to be deleted. This is a mandatory parameter.

hostlabel|hl

The host label for which the initiator is to be deleted . This is a mandatory parameter.

tenantname|tn

Name of the tenant. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 91

```
# viprcli initiator delete
-pwwn iqn.1998-01.com.vmware:vipr_host-7e9a6791
-hl vipr_host.example.mycompany.com
```

viprcli initiator list

Display the list of initiator port details.

Syntax

```
            viprcli initiator list
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -hostlabel|hl
        <hostlabel>
            [-protocol|pl <protocol>
        {FC,iSCSI}]
            [-tenantname|tn
        <tenantname>]
            [-verbose|v <verbose>]
            [-long|l <long>]
```

Description

`viprcli initiator list` command displays the list of initiator ports of the specified host.

Options

hostlabel|hl

The host label name to display the initiator port details. This is a mandatory parameter.

protocol|pl

Specify FC or iSCSI. This is an optional parameter.

tenantname|tn

Name of the tenant. This is an optional parameter.

verbose|v

The verbose parameter lists initiators with details. Use only either verbose or long and not both. This is an optional parameter.

long|l

The long parameter lists initiators in a large table. Use only either verbose or long and not both. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 92

```
viprcli initiator list -hl vipr_host.example.mycompany.com
  INITIATOR_PORT
  21:00:00:E0:8B:92:F2:85
  21:01:00:E0:8B:B2:F2:85
  iqn.1998-01.com.vmware:vipr_host-7e9a6791
```

viprcli initiator show

Shows initiator details.

Syntax

```
            viprcli initiator show
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-xml]
            -initiatorportwwn|pwwn
        <initatorportwwn>
            [-tenantname|tn
        <tenantname>]
            -hostlabel|hl <hostlabel>
```

Description

`viprcli initiator show` command shows initiator details.

Options

xml

XML response. This is an optional parameter.

initiatorportwwn|pwwn

Specify a Worldwide Port Name (WWPN) for Fibre Channel and IQN or EUI for iSCSI. This is a mandatory parameter.

tenantname|tn

Name of the tenant. This is an optional parameter.

hostlabel|hl

The host for which initiator tasks are to be searched. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli initiator tasks

Check the tasks of an initiator.

Syntax

```
    viprcli initiator tasks
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -initiatorportwwn|pwwn
<initiatorportwwn>
    -hostlabel|hl
<hostlabel>
    [-tenantname|tn
<tenantname>
    [-id <id>]
    [-verbose|v]
```

Description

`viprcli initiator tasks` command checks the tasks of an initiator.

Options

initiatorportwwn|pwwn

The Worldwide Port Name (WWPN) for Fibre Channel and IQN or EUI for iSCSI of the initiator. This is a mandatory parameter.

hostlabel|hl

The host initiator to be searched. This is an mandatory parameter.

tenantname|tn

The name of the tenant. This is an optional parameter.

id

The operation ID of a initiator. This is an optional parameter.

verbose|v

List all the tasks of a initiator. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli initiator update

Update an initiator with specified details.

Syntax

```
    viprcli initiator update
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -initiatorportwwn|pwwn
<initatorportwwn>
    -hostlabel|hl
<hostlabel>
    [-newprotocol|npl <newprotocol>
{FC | iSCSI}]
    [-newinitiatorwwn|nwwn
<newinitiatorwwn>]
    [-newinitiatorname|newinitname
<newinitiatorname>]
    [-tenantname|tn
<tenantname>]
    -newinitiatorportwwn|npwwn
<newinitatorportwwn>
```

Description

`viprcli initiator update` command updates an initiator with the specified details. This command is silent on success.

Options

initiatorportwwn|pwwn

Specify a Worldwide Port Name (WWPN) for Fibre Channel and IQN or EUI for iSCSI. This is a mandatory parameter.

hostlabel|hl

The host label name for which the details to be updated . This is a mandatory parameter.

newprotocol|npl

Specify a new FC or iSCSI. This is an optional parameter.

newinitiatorwwn|nwwn

The new initiator specification string to be updated. Its format depends upon the protocol. For Fibre Channel, specify the Worldwide Node Name (WWNN). For iSCSI specify the initiator node in IQN format. This is an optional parameter.

newinitiatorname|newinitname

Initiator alias name. This is an optional parameter.

tenantname|tn

Name of the tenant. This is an optional parameter.

newinitiatorportwwn|npwwn

Specify a new Worldwide Port Name (WWPN) for Fibre Channel and IQN or EUI for iSCSI. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 93

```
# viprcli initiator update -pwwn iqn.
1998-01.com.vmware:vipr_host-7e9a6791
          -hl vipr_host.example.mycompany.com -npwwn iqn.
2013-01.com.vmware:viprhost01-7e9a6791
```

CHAPTER 22

viprcli ipinterface commands

This chapter contains the following topics:

- viprcli ipinterface create 268
- viprcli ipinterface delete 269
- viprcli ipinterface list 270
- viprcli ipinterface show 271
- viprcli ipinterface tasks 272
- viprcli ipinterface update 273

viprcli ipinterface create

Create an IP interface.

Syntax

```
viprcli ipinterface create
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -hostlabel|hl
<hostlabel>
    -protocol|pl <protocol> {IPV4 |
IPV6}
    -ipaddress|ipadr
<ipaddress>
    [-netmask|nm <netmask>]
    [-prefixlength|pxl
<prefixlength>]
    [-scopeid|sid <scopeid>]
    [-tenant|tn <tenantname>]
```

Description

`viprcli ipinterface create` command creates an IP interface. This command is silent on success.

Options

hostlabel|hl

The host label name for which to create the IP address. This is a mandatory parameter.

protocol|pl

The IP interface protocol. Specify IPV4 or IPV6. This is a mandatory parameter.

ipaddress|ipadr

The IP address of IP interface node. This is a mandatory parameter.

netmask|nm

Netmaks of the IP interface node. This is an optional parameter.

prefixlength|pxl

Prefix length of the IP interface node. This is an optional parameter.

scopeid|sid

Scope ID of the IP interface node. This is an optional parameter.

tenantname|tn

The name of the tenant. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 94

```
viprcli ipinterface create -hl viprhost20 -pt IPV4 -ipadr
192.168.98.20
```

viprcli ipinterface delete

Delete an IP interface.

Syntax

```
            viprcli ipinterface delete
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -ipaddress|ipadr
        <ipaddress>
            -hostlabel|hl
        <hostlabel>
            [-tenantname|tn
        <tenantname>
```

Description

`viprcli ipinterface delete` command deletes an IP interface. This command is silent on success.

Options

ipaddress|ipadr

The IP address of IP interface node. This is a mandatory parameter.

hostlabel|hl

The host label name of the IP interface node. This is a mandatory parameter.

tenantname|tn

The name of the tenant. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 95

Example 95 (continued)

```
viprcli ipinterface delete -hl viprhost2 -ipadr 192.168.1.2
```

viprcli ipinterface list

Display the list of IP interfaces.

Syntax

```
            viprcli ipinterface list
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -hostlabel|hl
        <hostlabel>
            [-protocol|pl <protocol>
        {IPV4,IPV6}]
            [-tenantname|tn
        <tenantname>]
            [-verbose|v <verbose>]
            [-long|l <long>]
```

Description

`viprcli ipinterface list` command displays the list of IP interface.

Options

hostlabel|hl

The host label name to list the details. This is a mandatory parameter.

protocol|pl

The IP interface protocol. Specify IPV4 or IPV6. This is an optional parameter.

tenantname|tn

The name of the tenant. This is an optional parameter.

verbose|v

Lists IP interfaces with details. Use either verbose or long. This is an optional parameter.

long|l

Lists IP interfaces in a table. Use either verbose or long. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 96

```
viprcli ipinterface list -hl viprhost20
  NAME IP_ADDRESS   PROTOCOL
       192.168.1.2 IPV4
```

viprcli ipinterface show

Return the IP interface details.

Syntax

```
            viprcli ipinterface show
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-xml <xmlresponse>]
            -ipaddress|ipadr
        <ipaddress>
            -hostlabel|hl <hostlabel>
            [-tenantname|tn
        <tenantname>
```

Description

`viprcli ipinterface show` command returns the IP interface details.

Options

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

ipaddress|ipadr

The IP address of the IP interface. This is a mandatory parameter.

hostlabel|hl

The host label name to return the details. This is a mandatory parameter.

tenantname|tn

The name of the tenant. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 97

```
 # viprcli ipinterface show -hl viprhost2 -ipadr 192.168.1.2
{
   "creation_time": 1379772227736,
   "host": {
      "id": "urn:storageos:Host:cdb2dc8c-4756-44a8-
b6a3-25f6aca81694:",
      "link": {
         "href": "/compute/hosts/
urn:storageos:Host:cdb2dc8c-4756-44a8-b6a3-25f6aca81694:",
         "rel": "self"
      }
   },
   "id": "urn:storageos:IpInterface:e3808ab2-63cd-46b9-bb71-
ce1783f15578:",
   "inactive": false,
   "ip_address": "192.168.1.2",
   "link": {
      "href": "/compute/ip-interfaces/
urn:storageos:IpInterface:e3808ab2-63cd-46b9-bb71-ce1783f15578:",
      "rel": "self"
   },
   "netmask": "",
   "protocol": "IPV4",
   "tags": []
}
```

viprcli ipinterface tasks

Check the tasks of an IP interface.

Syntax

```
            viprcli ipinterface tasks
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -ipaddress|ipadr
        <ip_address>
            -hostlabel|hl <hostlabel>
            [-id <opid>]
            [-tenantname|tn
        <tenantname>]
            [-verbose|v]
```

Description

`viprcli ipinterface tasks` command checks the tasks of an IP interface.

Options

ipaddress|ipadr

The IP address of the IP interface. This is a mandatory parameter.

hostlabel|hl

The host of the IP interface to be searched. This is a mandatory parameter.

id

The operation ID of a IP interface. This is an optional parameter.

tenantname|tn

The name of the tenant. This is an optional parameter.

verbose|v

List all the tasks of a IP interface. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli ipinterface update

Update an IP interface.

Syntax

```
    viprcli ipinterface update
    [-h]
    [-hostname|hn <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -ipaddress|ipadr
<ipaddress>
    -hostlabel|hl
<hostlabel>
    [-newprotocol|npl <newprotocol>
{IPv4,IPv6}]
    [-newipaddress|nipadr
<newipaddress>]
    [-newnetmask|nnm
<newnetmask>]
    [-newprefixlength|npxl
<newprefixlength>]
    [-tenantname|tn
<tenantname>]
    [-newscopeid|nsid
<newscopeid>]
```

Description

`viprcli ipinterface update` command updates an IP interface with the specified details. This command is silent on success.

Options

ipaddress|ipadr

The IP address of IP interface node. This is a mandatory parameter.

hostlabel|hl

The host label for which the IP interface to be searched. This is a mandatory parameter.

newprotocol|npl

Specify a new IPv4 or IPv6. This is an optional parameter.

newipaddress|nipadr

The new IP address of IP interface node. This is an optional parameter.

newnetmask|nnm

The new netmaks of the IP interface node. This is an optional parameter.

newprefixlength|npxl

The new prefix length of the IP interface node. This is an optional parameter.

tenantname|tn

The name of the tenant. This is an optional parameter.

newscopeid|nsid

The new scope ID of the IP interface node. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 98

```
# viprcli ipinterface update -hl viprhost2
-ipadr 192.168.1.1 -nipadr 192.168.1.2
```

CHAPTER 23

viprcli ipsec commands

This chapter contains the following topics:

- viprcli ipsec change-status 276
- viprcli ipsec rotate-key 276
- viprcli ipsec status 277

viprcli ipsec change-status

Changes the IPsec status.

Syntax

```
viprcli ipsec change-status
[-h]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-status|s <status>
```

Description

The `viprcli ipsec change-status` command changes the IPsec status.

Options

status|s

The IPsec status. Valid values are enabled or disabled. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli ipsec rotate-key

Use to start key rotation.

Syntax

```
viprcli ipsec rotate-key
[-h]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
```

Description

The `viprcli ipsec rotate-key` command starts key rotation.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli ipsec status

Returns the IPsec status.

Syntax

```
viprcli ipsec status
[-h]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
[-xml <xmlresponse>]
```

Description

The `viprcli ipsec status` command returns the IPsec status.

Options

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

CHAPTER 24

viprcli logout command

This chapter contains the following topic:

- viprcli logout 280

viprcli logout

Log out of ViPR Controller CLI.

Syntax

```
 viprcli logout
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-h]
```

Description

`viprcli logout` command logs you out of ViPR Controller CLI. The command is silent on exit.

Once logged out you must authenticate to run the CLI commands again.

Options

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 25

viprcli meter commands

This chapter contains the following topics:

- viprcli meter 282

viprcli meter

Return performance statistics for ViPR Controller in a specified time range.

Syntax

```
        viprcli meter
        -year|y <year>
        -month|mon <month>
        -day|d <day>
        -hour|hr <hour>
        [-minute|min <minute>]
        [-format|f {xml|json}]
        [-hostname|hn <hostname>]
        [-port|po
    <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        [-h]
```

Description

`viprcli meter` command returns performance statistics for ViPR Controller in a specified time range. This command generates the statistics for the whole ViPR Controller system. There is no way to limit it to a particular tenant or project. This is intended for the admin user.

Options

year|y

A four digit year, such as 2013. This is a mandatory parameter.

month|mon

A number from 1 to 12, indicating the month. For example, 6 is June. This is a mandatory parameter.

day|d

A number from 1 to 31 indicating the day of the month. This is a mandatory parameter.

hour|hr

A number from 1 to 23. This is a mandatory parameter.

minute|min

The default is to deliver metrics for all 60 minutes of the specified hour. This is an optional parameter.

format|f

This specified the format of the command output. This is an optional parameter. The default is JSON format.

common arguments

This operation also takes the Common Parameters on page 24.

Example 99

```
# viprcli meter -year 2013 -month 9 -day 21 -hour 12
{
   "stats": [
      {
         "allocated_capacity": 2560,
         "bandwidth_in": null,
         "bandwidth_out": null,
         "idle_time_counter": null,
         "io_time_counter": null,
         "kbytes_transferred": null,
         "native_guid": "ISILON+
6805ca00acee0d63a6508020ba69b3849f37
+FILESYSTEM+/ifs/sos/
ProviderTenant_fs1_033d0821-a618-4c7b-bdf7-35ef13b90fd4",
         "object_count": null,
         "project_id": "urn:storageos:
Project:bfaf8b4b-b363-41b7-9f76-7ced92efa9a6:",
         "provisioned_capacity": 20971520,
         "queue_length": null,
         "read_hit_ios": null,
         "read_ios": null,
         "real_size": null,
         "resource_id": "urn:storageos:
FileShare:033d0821-a618-4c7b-bdf7-35ef13b90fd4:",
         "service_type": "file",
         "snapshot_capacity": 0,
         "snapshot_count": 0,
         "system_metadata_size": null,
         "tenant_id": "urn:storageos:
TenantOrg:dbe23c11-4d96-4fa5-a617-01c8906eef9d:",
         "time_collected": 1379766354202,
         "time_measured": 1379766354202,
         "total_ios": null,
         "user_id": null,
         "user_metadata_size": null,
         "user_size": null,
         "virtual_pool_id": "urn:storageos:
VirtualPool:587ba016-bb54-42fc-b710-795577f6abd9:",
         "write_hit_ios": null,
         "write_ios": null
      }
   ]
}
```

CHAPTER 26

viprcli monitor commands

This chapter contains the following topics:

- viprcli monitor 286

viprcli monitor

Get system events for a given time period.

Syntax

```
viprcli monitor
-year|y <year>
-month|mon <month>
-day|d <day>
-hour|hr <hour>
[-minute|min <minute>]
[-format|f {xml|json}]
[-hostname|hn <hostname>]
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
[-h]
```

Description

`viprcli monitor` command helps the user to get system events for a given time period.

Options

year|y

A four digit year, such as 2013. This is a mandatory parameter.

month|mon

A number from 1 to 12, indicating the month. For example, 6 is June. This is a mandatory parameter.

day|d

A number from 1 to 31 indicating the day of the month. This is a mandatory parameter.

hour|hr

A number from 1 to 23. This is a mandatory parameter.

minute|min

The default is to deliver metrics for all 60 minutes of the specified hour. This is an optional parameter.

format|f

This specified the format of the command output. This is an optional parameter. The default is JSON format.

common arguments

This operation also takes the Common Parameters on page 24.

Example 100

Example 100 (continued)

```
viprcli monitor -year 2013 -month 1 -day 17 -hour 12 -minute 34
{
   "events": [
      {
         "description": "Recurring: Mount request from 192.168.99.7
on 192.168.96.132 for /ifs/vipr/urn:storageos:FileShare
:956f68f2-390f-46c9-80ff-bfec312c21c8: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.99.7, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSha
re:956f68f2-390f-46c9-80ff-bfec312c21c8:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358425999
      },
      {
         "description": "Recurring: Mount request from
192.168.99.107 on 192.168.96.132 for /ifs/vipr/urn:storageos:FileSha
re:14a621ed-f4b9-49a0-9ebf-f085d77dc2ce: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.99.107, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileS
hare:14a621ed-f4b9-49a0-9ebf-f085d77dc2ce:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358426021
      },
      {
         "description": "Recurring: Mount request from
192.168.98.75 on 192.168.96.132 for /ifs/vipr/urn:storageos:FileShar
e:6d4b4495-8403-4ff9-90dd-128ae055721d: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.98.75, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSh
are:6d4b4495-8403-4ff9-90dd-128ae055721d:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358426037
      },
      {
         "description": "Recurring: Mount request from
192.168.180.83 on 192.168.96.132 for /ifs/vipr/
urn:storageos:FileShar
e:289011dd-a8d1-4959-9c12-9833412ba724: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.180.83, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSh
are:289011dd-a8d1-4959-9c12-9833412ba724:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
```

Example 100 (continued)

```
         "tenantId": "",
         "timeOccurred": 1358426055
      },
      {
         "description": "Mount request from 192.168.99.7 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:956f68f2-3
90f-46c9-80ff-bfec312c21c8: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.99.7, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSha
re:956f68f2-390f-46c9-80ff-bfec312c21c8:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358425998
      },
      {
         "description": "Mount request from 192.168.99.7 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:956f68f2-3
90f-46c9-80ff-bfec312c21c8: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.99.7, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSha
re:956f68f2-390f-46c9-80ff-bfec312c21c8:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358425999
      },
      {
         "description": "Mount request from 192.168.99.107 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:14a621ed
-f4b9-49a0-9ebf-f085d77dc2ce: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.99.107, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileS
hare:14a621ed-f4b9-49a0-9ebf-f085d77dc2ce:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358426021
      },
      {
         "description": "Mount request from 192.168.99.107 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:14a621ed
-f4b9-49a0-9ebf-f085d77dc2ce: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.99.107, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileS
hare:14a621ed-f4b9-49a0-9ebf-f085d77dc2ce:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
```

Example 100 (continued)

```
         "timeOccurred": 1358426021
      },
      {
         "description": "Mount request from 192.168.180.83 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:289011dd-
a8d1-4959-9c12-9833412ba724: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.180.83, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSh
are:289011dd-a8d1-4959-9c12-9833412ba724:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358426024
      },
      {
         "description": "Mount request from 192.168.98.75 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:6d4b4495-
8403-4ff9-90dd-128ae055721d: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.98.75, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSh
are:6d4b4495-8403-4ff9-90dd-128ae055721d:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358426036
      },
      {
         "description": "Mount request from 192.168.98.75 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:6d4b4495-
8403-4ff9-90dd-128ae055721d: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.98.75, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSh
are:6d4b4495-8403-4ff9-90dd-128ae055721d:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358426037
      },
      {
         "description": "Mount request from 192.168.180.83 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:289011dd-
a8d1-4959-9c12-9833412ba724: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.180.83, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSh
are:289011dd-a8d1-4959-9c12-9833412ba724:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358426055
```

Example 100 (continued)

```
      },
      {
         "description": "Mount request from 192.168.99.7 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:956f68f2-3
90f-46c9-80ff-bfec312c21c8: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.99.7, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSha
re:956f68f2-390f-46c9-80ff-bfec312c21c8:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358426059
      },
      {
         "description": "Mount request from 192.168.99.7 on
192.168.96.132 for /ifs/vipr/urn:storageos:FileShare:956f68f2-3
90f-46c9-80ff-bfec312c21c8: failed with errno: 2",
         "eventId": "400130001",
         "eventSource": "urn:storageos:StorageSystem:18ed2117-
c852-438c-9b79-e0b2505e380f:",
         "eventType": "ArrayGeneric",
         "extensions": "{client=192.168.99.7, devid=1.0,
host=192.168.96.132, lnn=1.0, path=/ifs/vipr/urn:storageos:FileSha
re:956f68f2-390f-46c9-80ff-bfec312c21c8:, val=0.0}",
         "recordType": "Alert",
         "severity": "information",
         "tenantId": "",
         "timeOccurred": 1358426059
      }
   ]
}
```

CHAPTER 27

viprcli network commands

This chapter contains the following topics:

- viprcli network create 292
- viprcli network list 293
- viprcli network show 294
- viprcli network update 296
- viprcli network assign 297
- viprcli network delete 298
- viprcli network endpoint add 298
- viprcli network endpoint remove 300
- viprcli network register 301
- viprcli network deregister 302

viprcli network create

Create a network with the specified parameters.

Syntax

```
            viprcli network create
            -name|n <networkname>
            -transport_type|t {FC | IP | Ethernet |
        ScaleIO}
            [-varrays
        [<varrayname>....]
            [-endpoints <endpoints>
        [<endpoints>...]]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli network create` command creates a network with the specified parameters. This command is silent on success.

Options

name|n

The name of network to be created. This is a mandatory parameter.

transport_type|t

The network protocol. One of the following: FC, IP, Ethernet, or ScaleIO. This is a mandatory parameter.

varrays

The names of one or more existing virtual arrays to be associated with the network. This is an optional parameter.

endpoints

Client endpoints to be added to the network. For an iSCSI host, the endpoint is an IQN. For an Ethernet host, the endpoint is an IP address. For a fibre channel host, the endpoint is a WWN. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

```
# viprcli network create -name mynetwork
          -varrays myvarray -network_type FC
```

Use the list command to see the network created.

```
# viprcli network list -v
[
   {
      "creation_time": 1377506473610,
      "discovered": false,
      "endpoints": [],
      "endpoints_discovered": [],
      "id": "urn:storageos:Network:
fca8b776-c07f-4ff1-a6c6-9a93cf5b5a8b:",
      "inactive": false,
      "link": {
         "href": "/vdc/networks/urn:storageos:Network:
fca8b776-c07f-4ff1-a6c6-9a93cf5b5a8b:",
         "rel": "self"
      },
      "name": "mynetwork",
      "tags": [],
      "transport_type": "FC",
      "varray": {
         "id": "urn:storageos:VirtualArray:
f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
         "link": {
            "href": "/vdc/varrays/urn:storageos:VirtualArray:
f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
            "rel": "self"
         }
      }
   }
]
```

viprcli network list

List networks in a virtual array.

Syntax

```
viprcli network list
[-varray|va <varrayname>]
[{-verbose|-long}]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli network list` command displays the list of networks in a virtual array. If you specify the -v switch, ViPR Controller displays a more detailed list of networks in the virtual array.

Options

varray|va

The name of a valid ViPR Controller virtual array. This is an optional parameter.

{verbose|v| long|l}

These options provide additional information. These are optional parameters.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 101

```
# viprcli network list -va myvarray -v
[
   {
      "creation_time": 1377506473610,
      "discovered": false,
      "endpoints": [],
      "endpoints_discovered": [],
      "id": "urn:storageos:Network:fca8b776-c07f-4ff1-
a6c6-9a93cf5b5a8b:",
      "inactive": false,
      "link": {
         "href": "/vdc/networks/urn:storageos:Network:fca8b776-
c07f-4ff1-a6c6-9a93cf5b5a8b:",
         "rel": "self"
      },
      "name": "mynetwork",
      "tags": [],
      "transport_type": "FC",
      "varray": {
         "id":
"urn:storageos:VirtualArray:f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
         "link": {
            "href": "/vdc/varrays/
urn:storageos:VirtualArray:f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
            "rel": "self"
         }
      }
   }
]
```

viprcli network show

Return the details of a network associated with the virtual array.

Syntax

```
            viprcli network show
            -name|n <networkname>
            [-xml]
            [-hostname|hn <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
```

```
              [-h]
```

Description

`viprcli network show` command returns the details of a network associated with the virtual array.

Options

name|n

The name of an existing network. This is a mandatory parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 102

```
# viprcli network show
             -n mynetwork -va myvarray
{
   "creation_time": 1377506473610,
   "discovered": false,
   "endpoints": [],
   "endpoints_discovered": [],
   "id": "urn:storageos:Network:fca8b776-c07f-4ff1-
a6c6-9a93cf5b5a8b:",
   "inactive": false,
   "link": {
      "href": "/vdc/networks/urn:storageos:Network:
       fca8b776-c07f-4ff1-a6c6-9a93cf5b5a8b:",
      "rel": "self"
   },
   "name": "mynetwork",
   "tags": [],
   "transport_type": "FC",
   "varray": {
      "id": "urn:storageos:VirtualArray:
       f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
      "link": {
         "href": "/vdc/varrays/urn:storageos:VirtualArray:
         f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
         "rel": "self"
 }
```

viprcli network update

Update a network.

Syntax

```
viprcli network update
-name|n <networkname>
[-newname|nn <newname>
[-varray_add|vaadd <varray_add>
[<varray_add>...]]
[-varray_remove|varemove
<varray_remove> [<varray_remove>...]]
[-endpoint_add|epadd
<endpoint_add> [<endpoint_add>...]]
[-endpoint_remove|epremove
<endpoint_remove> [<endpoint_remove>...]]
[-hostname|hn
<hostname>]
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookie_file>]
[-h]
```

Description

`viprcli network update` command updates the specified network. This operation allows you to change the varray to which the network belongs, the label assigned to the network, or the end points in a network.

Options

name|n

The name of an existing network to update. This is a mandatory parameter.

newname|nn

The new name to update. This is an optional parameter.

varray_add|vaadd

List of virtual arrays to be associated to the network. This is an optional parameter.

varray_remove|varemove

List of virtual arrays to be dissociated from the network. This is an optional parameter.

endpoint_add|epadd

List of endpoints to be associated to the network. This is an optional parameter.

endpoint_remove|epremove

List of endpoints to be dissociated from the network. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 103

```
# viprcli network update -n mynetwork2 -vaadd myvarray -nn filearray
```

Use the list command to view the results.

```
# viprcli network list -va filearray
  NAME        ENDPOINTS
  mynetwork
  mynetwork2
```

viprcli network assign

Assign a virtual array to a network.

Syntax

```
            viprcli network assign
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n <networkname>
            [-varray|va <varrayname>]
```

Description

`viprcli network assign` command assigns a virtual array to a network.

Options

name|n

The name of an existing network. This is a mandatory parameter.

varray|va

The name of a virtual storage array. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli network delete

Delete a network associated with the specified virtual array.

Syntax

```
viprcli network delete
-name|n <networkname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli network delete` command deletes a network from the specified virtual array. This command is silent on success.

Options

name|n

The name of an existing network. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 104

```
# viprcli network delete -n mynetwork2
```

Use the list command to view the results.

```
# viprcli network list
  NAME        ENDPOINTS
  mynetwork
```

viprcli network endpoint add

Add an endpoint to a network.

Syntax

```
viprcli network endpoint add
[-h]
[-hostname|hn
```

```
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n <networkname>
            -endpoint|e <endpoint>
```

Description

`viprcli network endpoint add` command adds an endpoint to the specifiednetwork.

Options

name|n

The name of an existing network. This is a mandatory parameter.

endpoint|e

Client endpoint to be added to the network. For an iSCSI host, the endpoint is an IQN. For an Ethernet host, the endpoint is an IP address. For a fibre channel host, the endpoint is a WWN. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 105

Fibre Channel endpoint

```
# viprcli network endpoint add -name mynetwork
          -endpoint 11:00:00:00:c9:5c:90:47
{
   "endpoints": {
      "endpoints": "11:00:00:00:C9:5C:90:47"
   },
   "id":
"urn:storageos:Network:f83aaf2e-46a4-400b-8f63-2a344131e2fa:",
   "inactive": "false",
   "name": "mynetwork",
   "VirtualArray": "urn:storageos:VirtualArray:
   87078d8b-504f-4613-b1c1-39554257c653:",
   "TransportType": "FC"
}
]
```

Example 106

IP endpoint

```
# viprcli network endpoint add -name mynetwork -endpoint 192.168.1.2
```

Example 107

iSCSI endpoint

```
# viprcli network endpoint add -name mynetwork
          -endpoint iqn.2013-01.com.abc.hostname
```

viprcli network endpoint remove

Remove an endpoint from a network.

Syntax

```
            viprcli network endpoint remove
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n <networkname>
            -endpoint|e <endpoint>
```

Description

`viprcli network endpoint remove` command removes an endpoint from the specified network.

Options

name|n

The name of an existing network. This is a mandatory parameter.

endpoint|e

Client endpoint to be removed from the network. For an iSCSI host, the endpoint is an IQN. For an Ethernet host, the endpoint is an IP address. For a fibre channel host, the endpoint is a WWN. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 108

IP endpoint

```
# viprcli network endpoint remove -name mynetwork -endpoint
192.168.1.1
```

Example 109

iSCSI endpoint

```
# viprcli network endpoint remove -name mynetwork
          -endpoint iqn.2013-01.com.abc.hostname
```

Example 110

```
# viprcli network endpoint remove -name mynetwork
         -endpoint 11:00:00:00:c9:5c:90:47
{
   "endpoints": null,
   "id":
"urn:storageos:Network:f83aaf2e-46a4-400b-8f63-2a344131e2fa:",
   "inactive": "false",
   "name": "mynetwork",
   "varray": "urn:storageos:VirtualArray:
   87078d8b-504f-4613-b1c1-39554257c653:",
   "transport_type": "FC"
}
```

viprcli network register

Register a network.

Syntax

```
viprcli network register
-name|n <networkname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli network register` command registers a network. This command is silent on success.

Options

name|n

The name of an existing network. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 111

```
# viprcli network register -n mynetwork
```

viprcli network deregister

Unregister a network.

Syntax

```
viprcli network deregister
-name|n <networkname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli network deregister` command unregisters a network.

Options

name|n

The name of an existing network. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 112

```
# viprcli network deregister -n mynetwork
```

CHAPTER 28

viprcli networksystem commands

This chapter contains the following topics:

- viprcli networksystem aliases add 304
- viprcli networksystem aliases remove 304
- viprcli networksystem aliases show 305
- viprcli networksystem aliases update 306
- viprcli networksystem create 307
- viprcli networksystem discover 308
- viprcli networksystem list 310
- viprcli networksystem list-connections 311
- viprcli networksystem show 312
- viprcli networksystem delete 313
- viprcli networksystem register 314
- viprcli networksystem deregister 315

viprcli networksystem aliases add

Add aliases to a network system.

Syntax

```
viprcli networksystem aliases add
       -aliases|als <wwn_aliases>
   [<wwn_aliases> ...]
       -name|n <networkname>
       -fabricid|fid
   <fabricid>
       [-hostname|hn
   <hostname>]
       [-port|po <port_number>]
       [-portui|pu
   <ui_port_number>]
       [-cookiefile|cf
   <cookie_file>]
       [-help|h]
```

Description

`viprcli networksystem aliases add` adds an alias to a network system.

Options

name|n

Name of the network system. This is a mandatory parameter.

aliases|als

The WWN address and the alias name. For example, wwn,alias1 wwn,alias2. This is a mandatory parameter.

fabricid|fid

The fabric name or ID. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the common arguments list.

viprcli networksystem aliases remove

Remove aliases from a network system.

Syntax

```
viprcli networksystem aliases remove
       -aliases|als <wwn_aliases>
   [<wwn_aliases> ...]
       -name|n <networkname>
       -fabricid|fid
   <fabricid>
       [-hostname|hn
```

```
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-help|h]
```

Description

`viprcli networksystem aliases remove` removes an alias from a network system.

Options

name|n

Name of the network system. This is a mandatory parameter.

aliases|als

The WWN address and the alias name. For example, wwn,alias1 wwn,alias2. This is a mandatory parameter.

fabricid|fid

The fabric name or ID. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the common arguments list.

viprcli networksystem aliases show

Show the aliases for a network system.

Syntax

```
viprcli networksystem aliases show
       -name|n <networkname>
       -fabricid|fid <fabricid>
       [-xml]
       [-hostname|hn
   <hostname>]
       [-port|po <port_number>]
       [-portui|pu
   <ui_port_number>]
       [-cookiefile|cf
   <cookie_file>]
       [-help|h]
```

Description

`viprcli networksystem aliases show` shows the aliases for a network system.

Options

name|n

Name of the network system. This is a mandatory parameter.

fabricid|fid

The fabric name or ID. This is a mandatory parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the common arguments list.

viprcli networksystem aliases update

Update aliases for a network system.

Syntax

```
viprcli networksystem aliases update
       -alias|als <wwn_aliases>
   [<wwn_aliases> ...]
       -name|n <networkname>
       -oldwwn|ow
   <old_wwn_address>
       -newwwn|nw
   <new_wwn_address>
       -fabricid|fid <fabricid>
       [-hostname|hn
   <hostname>]
       [-port|po <port_number>]
       [-portui|pu
   <ui_port_number>]
       [-cookiefile|cf
   <cookie_file>]
       [-help|h]
```

Description

`viprcli networksystem aliases update` updates an alias for a network system.

Options

name|n

Name of the network system. This is a mandatory parameter.

alias|als

The WWN address and the alias name. For example, wwn,alias1 wwn,alias2. This is a mandatory parameter.

oldwwn|ow

The old WWN address. This is a mandatory parameter.

newwwn|nw

The new WWN address. This is a mandatory parameter.

fabricid|fid

The fabric name or ID. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the common arguments list.

viprcli networksystem create

Create a network system in ViPR Controller.

Syntax

```
viprcli networksystem create
-name|n <networksystemname>
-type|t {brocade | cisco} <networksystemtype>
-deviceip|dip <deviceip>
-deviceport|dp <port>
-user <username>
[-smisip <smisipaddress>]
[-smisport <smisport>]
[-smisuser <smisuser>]
[-smisssl]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli networksystem create` command adds a network system (switch) to ViPR Controller and executes a discovery against that switch using the parameters provided. This command is silent on success.

Options

name|n

The name of the network system you want to create. This is a mandatory parameter.

type|t

Two types are supported: brocade and cisco. This is a mandatory parameter.

deviceip|dip

The fully qualified domain name or IP address of the network system you want to create. This is a mandatory parameter.

deviceport|dp

The port number of the network system you want to create. This is a mandatory parameter.

user

A user credentials of the network system. This is a mandatory parameter.

smisip

The IP address of the SMI-S provider. This is an optional parameter.

smisport

The SMI-S provider port, typically 5989. This is an optional parameter.

smisuser

A name with credentials on the SMI-S provider. This is an optional parameter.

smisssl

If this option is included, ViPR Controller will try to discover the switch on a secure port (5589) over SSL. If this option is not included, ViPR Controller will try to discover the switch on port 5588. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 113

```
# viprcli networksystem create -name brocade
-type brocade -dip 192.168.1.1 -dp 5989 -user admin
-smisip 192.168.1.1 -smisport 5989 -smisuser Administrator -smisssl
Enter password of the network system:
Retype password:
Enter password of the SMIS for network system:
Retype password:
```

Example 114

```
# viprcli networksystem create -n ciscomds -t mds
-dip 198.168.1.2 -dp 22 -user admin
Enter password of the network system:
Retype password:
```

Use the list command to view the result.

```
# viprcli networksystem list
  NAME
  brocade
  ciscomds
```

viprcli networksystem discover

Discover a network system (swtich) from ViPR Controller.

Syntax

```
viprcli networksystem discover
-name|n <networksystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
```

```
             [-cookiefile|cf <cookie_file>]
             [-h]
```

Description

`viprcli networksystem discover` command initiates a discovery on a network system (switch). The command refreshes the port information for a discovered Brocade or Cisco switch. This allows changes made to port information made through switch management software or SAN element managers to be reflected in ViPR Controller.

Options

name|n

The name of the network system you want to discover. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 115

```
# viprcli networksystem discover -n cisco
{
   "link": {
      "href": "/zone/network-systems/
             urn:storageos:NetworkSystem:0d6945b0-ebbc-4b3f-b502-
cd54faf915b1:
             /tasks/07f7b523-35c3-48aa-ba76-a8cc4152e8f2",
      "rel": "self"
   },
   "op_id": "07f7b523-35c3-48aa-ba76-a8cc4152e8f2",
   "resource": {
      "id": "urn:storageos:NetworkSystem:0d6945b0-ebbc-4b3f-b502-
cd54faf915b1:",
      "link": {
         "href": "/zone/network-systems
              /urn:storageos:NetworkSystem:0d6945b0-ebbc-4b3f-b502-
cd54faf915b1:",
         "rel": "self"
      },
      "name": "cisco"
   },
   "start_time": 1373902082859,
   "state": "pending"
}
```

viprcli networksystem list

List all network system objects in ViPR Controller.

Syntax

```
viprcli networksystem list
[{-verbose|v <verbose> | -long|l <long>]}
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli networksystem list` command displays the names of the network systems that have been added to ViPR Controller. You can display additional information about the network systems using the -l and -v switches.

Options

{verbose|v |long|l}

These options provide additional information. Use either verbose or long and not both.

common arguments

This operation also takes the Common Parameters on page 24.

Example 116

```
# viprcli networksystem list -v
[
   {
      "detailed_status_message": "Retrieved 29 connections from
device urn:storageos:NetworkSystem:0d6945b0-ebbc-4b3f-b502-
cd54faf915b1: at 7/12/13 7:17 PM",
      "id": "urn:storageos:NetworkSystem:0d6945b0-ebbc-4b3f-b502-
cd54faf915b1:",
      "inactive": false,
      "ip_address": "192.168.222.107",
      "job_discovery_status": "COMPLETE",
      "job_metering_status": "CREATED",
      "last_discovery_run_time": 1373656644653,
      "last_metering_run_time": 0,
      "link": {
         "href": "/zone/network-systems/urn:storageos:NetworkSystem:
0d6945b0-ebbc-4b3f-b502-cd54faf915b1:",
         "rel": "self"
      },
      "name": "cisco",
      "native_guid": "MDS+192.168.222.107+22",
      "next_discovery_run_time": 1373660204481,
      "next_metering_run_time": 0,
      "port_number": 22,
      "registration_status": "REGISTERED",
```

Example 116 (continued)

```
        "smis_port_number": 22,
        "smis_provider_ip": "192.168.222.107",
        "smis_use_ssl": true,
        "smis_user_name": "admin",
        "system_type": "mds",
        "uptime": "12 days, 8 hours, 26 minutes, 59 seconds",
        "user_name": "admin",
        "version": "5.0(4b)"
    }
]
```

viprcli networksystem list-connections

Display the worldwide names of all the ports managed by a ViPR Controller network system.

Syntax

```
viprcli networksystem list-connections
-name|n <networksystemname>
[{-verbose|v <verbose> | -long|l <long>]}
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli networksystem list-connections` command displays the worldwide names of all the ports managed by a ViPR Controller network system with the following details:

- Farbric ID
- Remote Port Name
- Remote Node Name

Options

name|n

The name of the network system list-connections you want to display.

{verbose|v |long|l}

These options provide additional information. Use either verbose or long and not both.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 117

Example 117 (continued)

```
# viprcli networksystem list-connections -n cisco
  FABRIC_ID REMOTE_PORT_NAME        REMOTE_NODE_NAME
  3173      20:00:00:25:B5:26:05:01 20:00:00:25:B5:26:00:00
  3173      20:00:00:25:B5:26:05:03 20:00:00:25:B5:26:00:01
  3173      20:00:00:25:B5:26:05:05 20:00:00:25:B5:26:00:02
  3173      20:00:00:25:B5:26:05:07 20:00:00:25:B5:26:00:03
  3173      20:00:00:25:B5:26:05:09 20:00:00:25:B5:26:00:04
  3173      20:00:00:25:B5:26:05:0B 20:00:00:25:B5:26:00:05
  3173      20:00:00:25:B5:26:05:0D 20:00:00:25:B5:26:00:06
  3173      20:00:00:25:B5:26:05:0F 20:00:00:25:B5:26:00:07
  3173      20:00:00:25:B5:26:05:11 20:00:00:25:B5:26:00:08
  3173      20:00:00:25:B5:26:05:13 20:00:00:25:B5:26:00:09
  3173      20:00:00:25:B5:63:FF:01 20:00:00:25:B5:63:00:00
  3173      20:41:00:05:73:D7:47:00 2C:65:00:05:73:D7:47:01
  3173      20:42:00:05:73:D7:47:00 2C:65:00:05:73:D7:47:01
  3173      50:06:01:65:46:E0:71:42 50:06:01:60:C6:E0:71:42
  3173      50:06:01:6D:46:E0:71:42 50:06:01:60:C6:E0:71:42
  3175      20:43:00:05:73:D7:47:00 2C:67:00:05:73:D7:47:01
  3175      20:44:00:05:73:D7:47:00 2C:67:00:05:73:D7:47:01
  3175      50:06:01:67:46:E0:71:42 50:06:01:60:C6:E0:71:42
  3175      50:06:01:6F:46:E0:71:42 50:06:01:60:C6:E0:71:42
  3177      20:45:00:05:73:D7:47:00 2C:69:00:05:73:D7:47:01
  3177      20:46:00:05:73:D7:47:00 2C:69:00:05:73:D7:47:01
  3177      50:06:01:65:46:E4:71:42 50:06:01:60:C6:E0:71:42
  3177      50:06:01:6D:46:E4:71:42 50:06:01:60:C6:E0:71:42
  3180      20:48:54:7F:EE:D6:37:00 2C:6C:54:7F:EE:D6:37:01
  3180      50:06:01:62:46:E0:71:42 50:06:01:60:C6:E0:71:42
  3180      50:06:01:6A:46:E0:71:42 50:06:01:60:C6:E0:71:42
  3181      20:48:00:05:73:D7:47:00 2C:6D:00:05:73:D7:47:01
  3181      50:06:01:63:46:E0:71:42 50:06:01:60:C6:E0:71:42
  3181      50:06:01:6B:46:E0:71:42 50:06:01:60:C6:E0:71:42
```

viprcli networksystem show

Display a verbose listing of all parameters of a specific network system.

Syntax

```
 viprcli networksystem show
-name|n <networksystem>
[-xml]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli networksystem show` command includes all of the information provided during the `viprcli networksystem create` operation that built the network system.

Options

name|n

A valid network system name. This is a mandatory parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

common arguments

This operation also takes the Common Parameters on page 24.

Example 118

```
# viprcli networksystem show -n cisco
{
   "detailed_status_message": "Retrieved 29 connections from device
   urn:storageos:NetworkSystem:0d6945b0-ebbc-4b3f-b502-
cd54faf915b1:
   at 7/12/13 7:17 PM",
   "id": "urn:storageos:NetworkSystem:
   0d6945b0-ebbc-4b3f-b502-cd54faf915b1:",
   "inactive": false,
   "ip_address": "192.168.222.107",
   "job_discovery_status": "COMPLETE",
   "job_metering_status": "CREATED",
   "last_discovery_run_time": 1373656644653,
   "last_metering_run_time": 0,
   "link": {
      "href": "/vdc/network-systems/urn:
      storageos:NetworkSystem:0d6945b0-ebbc-4b3f-b502-
cd54faf915b1:",
      "rel": "self"
   },
   "name": "cisco",
   "native_guid": "MDS+192.168.222.107+22",
   "next_discovery_run_time": 1373660204481,
   "next_metering_run_time": 0,
   "port_number": 22,
   "registration_status": "REGISTERED",
   "smis_port_number": 22,
   "smis_provider_ip": "192.168.222.107",
   "smis_use_ssl": true,
   "smis_user_name": "admin",
   "system_type": "mds",
   "uptime": "12 days, 8 hours, 26 minutes, 59 seconds",
   "user_name": "admin",
   "version": "5.0(4b)"
}
```

viprcli networksystem delete

Delete a network system object from ViPR Controller.

Syntax

```
viprcli networksystem delete
-name|n <networksystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
```

```
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli networksystem delete` command removes a network system (switch) from ViPR Controller. This command is silent on success.

Options

name|n

The name of the network system you want to delete. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 119

```
# viprcli networksystem delete -name brocade1
```

Use the list command to view the result.

```
# viprcli networksystem list
  NAME
  cisco
```

viprcli networksystem register

Register a network system.

Syntax

```
            viprcli networksystem register
            -name|n <networksystemname>
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli networksystem register` command allows you to register a network system. This command is silent on success.

Options

name|n

The name of the network system you want to register. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 120

```
# viprcli networksystem register -n brocade

# viprcli networksystem list
  NAME
  brocade
```

viprcli networksystem deregister

Unregister a network system.

Syntax

```
viprcli networksystem deregister
-name|n <networksystemname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli networksystem deregister` command allows you to unregister a network system. This command is silent on success.

Options

name|n

The name of the network system you want to unregister. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 121

```
# viprcli networksystem deregister -n brocade
```

CHAPTER 29

viprcli objectuser commands

This chapter contains the following topics:

- viprcli objectuser create_secretkey 318

viprcli objectuser create_secretkey

Creates a secret key.

Syntax

```
        viprcli objectuser create_secretkey
        [-h]
        [-hostname|hn <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        -storagesystem|st
    <storagesystem>
        -objectuser|ob
    <objectuser>
        [-autogenerate|autogen]
```

Description

The `viprcli objectuser create_secretkey` command creates a secret key.

Options

storagesystem|st

Name of the storage system. This is a mandatory parameter.

objectuser|ob

Object user ID. This is a mandatory parameter.

autogenerate|autogen

Run in synchronous mode. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 122

```
viprcli objectuser create_secretkey -st ECS -objectuser mbuser
```

CHAPTER 30

viprcli order commands

This chapter contains the following topics:

- viprcli order list 320
- viprcli order show 320
- viprcli order show-execution 323

viprcli order list

List all orders.

Syntax

```
viprcli order list
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli order list` command lists all orders.

Options

common arguments

This operation also takes the Common Parameters on page 24.

Example 123

```
# viprcli order list
  ID
ORDERNUMBER STATUS
  urn:storageos:Order:22a40e65-7b84-4b12-8c49-bb2e3a5e337e:
3           ERROR
  urn:storageos:Order:66b8a570-5f50-451a-9d94-eb4f477d5209:
4           APPROVAL
  urn:storageos:Order:7758f4ee-4d72-467f-a810-4cb8be2266f5:
1           ERROR
  urn:storageos:Order:8bf2a5bd-d019-4317-80dd-a515ae4c640f:
2           ERROR
  urn:storageos:Order:ed72e258-ec80-4fad-85b3-d084d61291cf:
5           APPROVAL
```

viprcli order show

Show details about a specific order.

Syntax

```
viprcli order show
-uri|id <uri>
[-waitForExecution]
[-xml <xml>]
[-hostname|hn
<hostname>]
[-cookiefile|cf
```

```
        <cookie_file>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-h]
```

Description

`viprcli order show` command shows the details for a specific order.

Options

-uri|id

URI or order Id of the service request. This is a mandatory parameter.

-waitForExecution

This option is used to force ViPR to wait for the order to be processed before details are displayed.

-xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 124

```
# viprcli order show -id urn:storageos:Order:
029d203e-3c28-42db-992a-b0e0ee6c0508:vdc1
{
  "createdDate": 1435604333502,
   "dateCompleted": 1435604394645,
   "execution": {
      "href": "/api/orders/urn:storageos:Order:
029d203e-3c28-42db-992a-b0e0ee6c0508:vdc1/execution",
      "id": "urn:storageos:Order:029d203e-3c28-42db-992a-
b0e0ee6c0508:vdc1"
   },
   "executionWindow": null,
   "id": "urn:storageos:Order:029d203e-3c28-42db-992a-
b0e0ee6c0508:vdc1",
   "inactive": false,
   "link": {
      "href": "/api/orders/urn:storageos:Order:
029d203e-3c28-42db-992a-b0e0ee6c0508:vdc1",
      "rel": "self"
   },
   "message": "",
   "orderNumber": "3",
   "parameters": [
      {
         "encrypted": false,
         "friendly_value": "my_varray",
         "label": "Virtual Array",
         "user_input": true,
         "value": "\"urn:storageos:VirtualArray:e2661eb6-15f3-40f5-
b753-08eca26f68c3:vdc1\""
```

Example 124 (continued)

```
      },
      {
         "encrypted": false,
         "friendly_value": "my_vpool",
         "label": "Virtual Pool",
         "user_input": true,
         "value": "\"urn:storageos:VirtualPool:389cee33-beba-4c33-
b818-82db0b83928a:vdc1\""
      },
      {
         "encrypted": false,
         "friendly_value": "myproj",
         "label": "Project",
         "user_input": true,
         "value": "\"urn:storageos:Project:3dbb5faa-06dc-40b2-
a775-5af0e7c46dea:global\""
      },
      {
         "encrypted": false,
         "friendly_value": "vol_success",
         "label": "Name",
         "user_input": true,
         "value": "\"vol_success\""
      },
      {
         "encrypted": false,
         "friendly_value": "",
         "label": "Consistency Group",
         "user_input": true,
         "value": "\"\""
      },
      {
        "encrypted": false,
         "friendly_value": "1",
         "label": "Number Of Volumes",
         "user_input": true,
         "value": "\"1\""
      },
      {
         "encrypted": false,
         "friendly_value": "1",
         "label": "Size (GB)",
         "user_input": true,
         "value": "\"1\""
      }
   ],
   "service": {
      "href": "/api/services/urn:storageos:CatalogService:c01dbe47-
cc20-4a10-8015-6dff979ab971:vdc1",
      "id": "urn:storageos:CatalogService:c01dbe47-
cc20-4a10-8015-6dff979ab971:vdc1"
   },
   "status": "SUCCESS",
   "submittedBy": "root",
   "summary": "Create Block Volume",
   "tags": {
      "tag": []
   }

}
```

viprcli order show-execution

Show execution details for an order.

Syntax

```
viprcli order show-execution
-uri|id <uri>
[-xml <xml>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli order show-execution` command shows the execution details for the specific order.

Options

-uri|id

URI or order Id of the service request. This is a mandatory parameter.

-xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 125

```
# viprcli order show-execution -id urn:storageos:Order:
22a40e65-7b84-4b12-8c49-bb2e3a5e337e:
{
   "affectedResources": [],
   "currentTask": "",
   "endDate": "Jan 21, 2014 12:13:46 PM",
   "executionLogs": [
      {
         "date": "Jan 21, 2014 12:13:45 PM",
         "level": "ERROR",
         "message": "Error 12000: An error occurred while executing
the job. Failed to delete volumes [urn:storageos:Volume:3494b85f-
cb65-4e95-ac1b-ad933ffc02cd:], Op: DELETE VOLUME",
         "phase": "EXECUTE",
         "stackTrace":
"com.emc.vipr.client.exceptions.ServiceErrorException: Error 12000:
An error occurred while executing the job. Failed to delete volumes
[urn:storageos:Volume:3494b85f-cb65-4e95-ac1b-ad933ffc02cd:], Op:
DELETE VOLUME\n\tat
com.emc.vipr.client.core.impl.TaskUtil.checkForError(TaskUtil.java:
73)\n\tat com.emc.vipr.client.Task.waitFor(Task.java:137)\n\tat
com.emc.sa.service.vipr.tasks.WaitForTask.executeTask(WaitForTask.ja
```

Example 125 (continued)

```
va:20)\n\tat
com.emc.sa.service.vipr.tasks.WaitForTask.executeTask(WaitForTask.ja
va:5)\n\tat
com.emc.sa.engine.ExecutionUtils.execute(ExecutionUtils.java:63)\n
\tat com.emc.sa.engine.ExecutionUtils.execute(ExecutionUtils.java:
54)\n\tat
com.emc.sa.service.vipr.ViPRExecutionUtils.execute(ViPRExecutionUtil
s.java:28)\n\tat
com.emc.sa.service.vipr.block.BlockStorageUtils.deactivateVolume(Blo
ckStorageUtils.java:258)\n\tat
com.emc.sa.service.vipr.block.BlockStorageUtils.removeVolume(BlockSt
orageUtils.java:248)\n\tat
com.emc.sa.service.vipr.block.BlockStorageUtils.removeVolumes(BlockS
torageUtils.java:240)\n\tat
com.emc.sa.service.vipr.block.RemoveBlockStorageService.execute(Remo
veBlockStorageService.java:23)\n\tat
com.emc.sa.engine.ExecutionEngineImpl.execute(ExecutionEngineImpl.ja
va:191)\n\tat
com.emc.sa.engine.ExecutionEngineImpl.runService(ExecutionEngineImpl
.java:125)\n\tat
com.emc.sa.engine.ExecutionEngineImpl.executeOrder(ExecutionEngineIm
pl.java:68)\n\tat
com.emc.sa.engine.ExecutionEngineDispatcher.processOrder(ExecutionEn
gineDispatcher.java:43)\n\tat
com.emc.sa.engine.ExecutionEngineDispatcher
$Consumer.consumeItem(ExecutionEngineDispatcher.java:64)\n\tat
com.emc.sa.engine.ExecutionEngineDispatcher
$Consumer.consumeItem(ExecutionEngineDispatcher.java:58)\n\tat
com.emc.storageos.coordinator.client.service.impl.DistributedQueueCo
nsumer$1.run(DistributedQueueConsumer.java:83)\n\tat
java.util.concurrent.ThreadPoolExecutor
$Worker.runTask(ThreadPoolExecutor.java:895)\n\tat
java.util.concurrent.ThreadPoolExecutor
$Worker.run(ThreadPoolExecutor.java:918)\n\tat
java.lang.Thread.run(Thread.java:662)\n"
      }
   ],
   "executionStatus": "FAILED",
   "executionTasks": [
      {
         "date": "Jan 21, 2014 12:13:35 PM",
         "detail": "Volume: urn:storageos:Volume:3494b85f-cb65-4e95-
ac1b-ad933ffc02cd:",
         "elapsed": 115,
         "level": "INFO",
         "message": "Get Block Volume",
         "phase": "PRECHECK"
      },
      {
         "date": "Jan 21, 2014 12:13:35 PM",
         "detail": "Volume: urn:storageos:Volume:3494b85f-cb65-4e95-
ac1b-ad933ffc02cd:",
         "elapsed": 102,
         "level": "INFO",
         "message": "Get Exports For Volume",
         "phase": "EXECUTE"
      },
      {
         "date": "Jan 21, 2014 12:13:35 PM",
         "detail": "Volume: urn:storageos:Volume:3494b85f-cb65-4e95-
ac1b-ad933ffc02cd:",
         "elapsed": 68,
         "level": "INFO",
         "message": "Get Active Snapshots For Volume",
         "phase": "EXECUTE"
```

Example 125 (continued)

```
      },
      {
         "date": "Jan 21, 2014 12:13:35 PM",
         "detail": "Volume: urn:storageos:Volume:3494b85f-cb65-4e95-
ac1b-ad933ffc02cd:",
         "elapsed": 65,
         "level": "INFO",
         "message": "Get Active Continuous Copies For Volume",
         "phase": "EXECUTE"
      },
      {
         "date": "Jan 21, 2014 12:13:45 PM",
         "detail": "Volume: urn:storageos:Volume:3494b85f-cb65-4e95-
ac1b-ad933ffc02cd:",
         "elapsed": 10143,
         "level": "ERROR",
         "message": "Deactivate Volume",
         "phase": "EXECUTE",
         "stackTrace":
"com.emc.vipr.client.exceptions.ServiceErrorException: Error 12000:
An error occurred while executing the job. Failed to delete volumes
[urn:storageos:Volume:3494b85f-cb65-4e95-ac1b-ad933ffc02cd:], Op:
DELETE VOLUME\n\tat
com.emc.vipr.client.core.impl.TaskUtil.checkForError(TaskUtil.java:
73)\n\tat com.emc.vipr.client.Task.waitFor(Task.java:137)\n\tat
com.emc.sa.service.vipr.tasks.WaitForTask.executeTask(WaitForTask.ja
va:20)\n\tat
com.emc.sa.service.vipr.tasks.WaitForTask.executeTask(WaitForTask.ja
va:5)\n\tat
com.emc.sa.engine.ExecutionUtils.execute(ExecutionUtils.java:63)\n
\tat com.emc.sa.engine.ExecutionUtils.execute(ExecutionUtils.java:
54)\n\tat
com.emc.sa.service.vipr.ViPRExecutionUtils.execute(ViPRExecutionUtil
s.java:28)\n\tat
com.emc.sa.service.vipr.block.BlockStorageUtils.deactivateVolume(Blo
ckStorageUtils.java:258)\n\tat
com.emc.sa.service.vipr.block.BlockStorageUtils.removeVolume(BlockSt
orageUtils.java:248)\n\tat
com.emc.sa.service.vipr.block.BlockStorageUtils.removeVolumes(BlockS
torageUtils.java:240)\n\tat
com.emc.sa.service.vipr.block.RemoveBlockStorageService.execute(Remo
veBlockStorageService.java:23)\n\tat
com.emc.sa.engine.ExecutionEngineImpl.execute(ExecutionEngineImpl.ja
va:191)\n\tat
com.emc.sa.engine.ExecutionEngineImpl.runService(ExecutionEngineImpl
.java:125)\n\tat
com.emc.sa.engine.ExecutionEngineImpl.executeOrder(ExecutionEngineIm
pl.java:68)\n\tat
com.emc.sa.engine.ExecutionEngineDispatcher.processOrder(ExecutionEn
gineDispatcher.java:43)\n\tat
com.emc.sa.engine.ExecutionEngineDispatcher
$Consumer.consumeItem(ExecutionEngineDispatcher.java:64)\n\tat
com.emc.sa.engine.ExecutionEngineDispatcher
$Consumer.consumeItem(ExecutionEngineDispatcher.java:58)\n\tat
com.emc.storageos.coordinator.client.service.impl.DistributedQueueCo
nsumer$1.run(DistributedQueueConsumer.java:83)\n\tat
java.util.concurrent.ThreadPoolExecutor
$Worker.runTask(ThreadPoolExecutor.java:895)\n\tat
java.util.concurrent.ThreadPoolExecutor
$Worker.run(ThreadPoolExecutor.java:918)\n\tat
java.lang.Thread.run(Thread.java:662)\n"
      }
   ],
   "startDate": "Jan 21, 2014 12:13:35 PM"
}
```

Example 125 (continued)

CHAPTER 31

viprcli project commands

This chapter contains the following topics:

- viprcli project create 328
- viprcli project list 329
- viprcli project show 330
- viprcli project update 331
- viprcli project delete 332
- viprcli project get-acl 333
- viprcli project update-acl 334
- viprcli project tag 335

viprcli project create

Create a project.

Syntax

```
viprcli project create
-name|n <projectname>
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli project create` command creates a new project. A project is a resource organization abstraction that maps resources to applications, virtual data centers, departments, or other entities meaningful to the user. Users can create their own projects, and they can associate multiple resources from different services with them. Resources from one project can be shared between users under the same tenant.

Options

name|n

A valid project name. This is a mandatory parameter.

tenant|tn

A valid ViPR Controller tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 126

```
# viprcli project create -n IND
```

Use list command to view the results.

```
# viprcli project list
  NAME
  HDD
  IND
```

viprcli project list

Get a list of all projects belonging to a specified tenant.

Syntax

```
            viprcli project list
            [-tenant|tn
        <tenantname>]
            [-verbose|v]
            [-long|l]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

The viprcli project list command displays only the projects that belong to the tenant to which your user identity belongs, by default. You can provide another tenant's name to see that tenant's projects.

Options

tenant|tn

The name of a valid ViPR Controller tenant. If a tenant name is not specified, projects for the current user's parent tenant are displayed. This is an optional parameter. This is an optional parameter.

verbose|long|l|v

These options provide additional information. Use either of these, but not both.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 127

```
viprcli project list
    NAME
    myProject
```

Example 128

```
viprcli project list -v
[
   {
      "id": "urn:storageos:Project:
```

Example 128 (continued)

```
        539ce2ff-c007-41de-acfb-514b33dfe35c:",
        "inactive": false,
        "link": {
           "href": "/projects/urn:storageos:Project:
           539ce2ff-c007-41de-acfb-514b33dfe35c:",
           "rel": "self"
        },
        "name": "myProject",
        "owner": "root",
        "tenant": {
           "id": "urn:storageos:TenantOrg:
           bbbb303b-b6dd-4cd7-9cd0-c6769201781e:",
           "link": {
              "href": "/tenants/urn:storageos:TenantOrg:
              bbbb303b-b6dd-4cd7-9cd0-c6769201781e:",
              "rel": "self"
           }
        }
    }
]
```

viprcli project show

Show details of project.

Syntax

```
viprcli project show
-name|n <projectname>
[-tenant|tn <tenantname>]
[-xml]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli project show` command displays extended information for a specific project including the project owner, its ViPR Controller UID, and information about the tenant to which the project belongs.

Options

name|n

A valid ViPR Controller project name. This is a mandatory parameter.

tenant|tn

A valid ViPR Controller tenant name. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

common arguments

This operation also takes the Common Parameters on page 24.

Example 129

```
# viprcli project show -n myProject

{
   "id": "urn:storageos:Project:539ce2ff-c007-41de-
acfb-514b33dfe35c:",
   "inactive": false,
   "link": {
      "href": "/projects/urn:storageos:
      Project:539ce2ff-c007-41de-acfb-514b33dfe35c:",
      "rel": "self"
   },
   "name": "myProject",
   "owner": "root",
   "tenant": {
      "id": "urn:storageos:TenantOrg:
      bbbb303b-b6dd-4cd7-9cd0-c6769201781e:",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:
          bbbb303b-b6dd-4cd7-9cd0-c6769201781e:",
         "rel": "self"
      }
   }
}
```

viprcli project update

Update a project.

Syntax

```
viprcli project update
-name|n <projectname>
[-newname|np <newprojectname>]
[-newowner|o <newowner>]
[-quota_enable|qe <quota_enable> {true | false}]
[-quota_capacity|qc <quota_capacity>]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli project update` command updates the project with the specified details. This command is silent on success.

Options

name|n

A valid ViPR Controller existing project name.

newname|np

A valid new ViPR Controller project name.

newowner|o

New owner of the project.

quota_enable|qe

Enable or disable quota. Possible values are: true or false. The default is false. This is an optional parameter.

quota_capacity|qc

Quota capacity in GB. Quota reserves the maximum limit for allocated storage space for the specified tenant user. This is an optional parameter and used only when you set the `quota_enable` to true.

tenant|tn

A valid ViPR Controller tenant name.

common arguments

This operation also takes the Common Parameters on page 24.

Example 130

```
# viprcli project update -n HD -np HDD
```

Use the list command to view results.

```
# viprcli project list
  NAME
  HDD
```

viprcli project delete

Delete a project.

Syntax

```
viprcli project delete
-name|n <projectname>
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli project delete` command deletes a project only if no volumes or file shares are provisioned under that project. This command is silent on success.

Options

name|n

A valid ViPR Controller project name. This is a mandatory parameter.

tenant|tn

A valid ViPR Controller tenant name.

common arguments

This operation also takes the Common Parameters on page 24.

Example 131

```
# viprcli project delete -n BLOCK_PROJECT
```

Use the list command to view the result.

```
# viprcli project list
  NAME
  FACILITIES
  HR
  Marketing
  myviprproject
```

viprcli project get-acl

Return the project ACL details.

Syntax

```
viprcli project get-acl
-name|n <projectname>
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli project get-acl` command returns the project ACL details.

Options

name|n

A valid ViPR Controller existing project name. This is a mandatory parameter.

tenant|tn

A valid ViPR Controller tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 132

```
 # viprcli project get-acl -n HDD
{
   "acl": []
}
```

viprcli project update-acl

Update the project ACL details.

Syntax

```
viprcli project update-acl
[-portui|pu <ui_port_number>]
-name|n <projectname>
-operation|op <operation> {add | remove}
-privilege|pe <privilege> {all | backup | own}
[-subjectid|sub <subjectid>]
[-group|gp <subjectid>]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli project update-acl` command updates the project ACL details. This command is silent on success.

Options

name|n

A valid ViPR Controller existing project name. This is a mandatory parameter.

operation|op

ACL operation to perform on the project. You can either add or remove the ACL. This is a mandatory parameter.

privilege|pe

The privilege to be assigned. The possible privileges are: all, backup, or own. This is a mandatory parameter.

subjectid|sub

The subject ID specifies a user, typically a name in Active Directory. This is an optional parameter.

group|gp

The group specifies a user group, typically a name in Active Directory. This is a optional parameter.

tenant|tn

A valid ViPR Controller tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 133

```
# viprcli project update-acl -n HDD -op add -pe all -sub <user>
```

viprcli project tag

Add or delete a tag name for the specified project.

Syntax

```
viprcli project tag
-name|n <projectname>
[-add <addtags> [<addtags...>]]
[-remove <removetags> [<removetags...>]]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli project tag` command add or deletes a tag name for the specified project.

Options

name|n

A valid project name. This is a mandatory parameter.

tenant|tn

A valid ViPR Controller tenant name. This is an optional parameter.

add

Add one or more tag names to the ViPR Controller project name. This is an optional parameter. You can either use add or remove and not both.

remove

Delete one or more tag names to the ViPR Controller project name. This is an optional parameter. You can either use add or remove and not both.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 32

viprcli protectionsystem commands

This chapter contains the following topics:

- viprcli protectionsystem create 338
- viprcli protectionsystem discover 339
- viprcli protectionsystem list 340
- viprcli protectionsystem show 341
- viprcli protectionsystem update 343
- viprcli protectionsystem connectivity 345
- viprcli protectionsystem delete 346

viprcli protectionsystem create

Add a protection system to ViPR Controller.

Syntax

```
viprcli protectionsystem create
-name|n <protectionsystemname>
deviceip|dip <deviceip>
deviceport|dp <deviceport>
username|un <username>
[-type|T <protectionsystemtype>]
[-registration_mode|rm <registration_mode>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli protectionsystem create` command creates a protection system. This command is silent on success. A protection system is a separately installed datacenter asset dedicated to protecting data. A protection system does not expose storage of its own. RecoverPoint is the only protection system currently supported by ViPR Controller.

When a protection system is created, it is immediately discovered. You do not need to call `viprcli protectionsystem discover` command for a newly created protection system.

Options

name|n

The name of a protection system. This field appears in the ViPR Controller portal. This is a mandatory parameter.

deviceip|dip

The IP address of a protection system. This is a mandatory parameter.

deviceport|dp

Specify 7225 for RecoverPoint systems. This is the HTTPS protocol port for communicating with the RecoverPoint API. This is a mandatory parameter.

username|un

A user id for accessing the data protection system. This is a mandatory parameter.

registration_mode|rm

Registration Mode of a storage system. This is an optional parameter. Possible values include:

- SYSTEM
- SYSTEM_AND_PORTS
- SYSTEM_AND_COMPONENTS

type|t

Currently the only valid value is `rp`. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 134

```
# viprcli protectionsystem create -n chartsystem
  -dip 192.168.1.1 -dp 7225 -un root
Enter password of the protection system:
Retype password:
```

Use list command to view the results.

```
# viprcli protectionsystem list
  NAME        SYSTEM_TYPE IP_ADDRESS   PORT_NUMBER
  chartsystem rp          192.168.1.1 7225
```

viprcli protectionsystem discover

Re-discover a protection system.

Syntax

```
viprcli protectionsystem discover
[-name|n <protectionsystemname>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli protectionsystem discover` command refreshes the information about protection systems in your data center. This call is specifically for protection systems that have been previously created and discovered by ViPR Controller, but may have been modified or upgraded. This command is silent on success.

Options

name|n

The name of a protection system to be discovered. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 135

Example 135

```
# viprcli protectionsystem discover
```

Example 136

```
# viprcli protectionsystem discover -n chartsystem
```

viprcli protectionsystem list

Display a list of protection systems established in ViPR Controller.

Syntax

```
viprcli protectionsystem list
[-type|t <protectionsystemtype>]
[{-verbose|v |-long|l}]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli protectionsystem list` command displays information about all protection systems in ViPR Controller. The -long and -verbose options supply additional information about the protection systems.

Options

type|t

Currently the only valid value is `rp`. This is an optional parameter.

{ verbose|v |-long|l }

These options provide additional information. Use either of these, but not both.

common arguments

This operation also takes the Common Parameters on page 24.

Example 137

```
# viprcli protectionsystem list
  NAME        SYSTEM_TYPE IP_ADDRESS   PORT_NUMBER
  chartsystem rp          192.168.98.10 7225
```

viprcli protectionsystem show

Display the detailed information about a protection system.

Syntax

```
            viprcli protectionsystem show
            -name|n <psname>
            [-type|t
        <protectionsystemtype>]
            [-xml]
            [-hostname|hn
        <hostname>]
            [-cookiefile|cf
        <cookie_file>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-h]
```

Description

`viprcli protectionsystem show` command displays an extended information about a ViPR Controller protection system. RecoverPoint is currently the only protection system supported by ViPR Controller. The name of the protection system must be supplied as an argument to the viprcli call.

Options

name|n

The name of the protection system. This a mandatory parameter.

type|t

The protection system type. The default and only supported value is `rp`. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 138

```
viprcli protectionsystem show -name rp_west1
{
   "clusters": [
      {
         "cluster_id": "0x478b98e918d15478",
         "cluster_name": "NewYork"
      },
      {
         "cluster_id": "0x22a9431c63b4fe45",
```

Example 138 (continued)


```
            "cluster_name": "Hopkinton"
         },
         {
            "cluster_id": "0x49d637f2b520686d",
            "cluster_name": "Syracuse"
         },
         {
            "cluster_id": "0xc916995fdc011a9",
            "cluster_name": "Moncton"
         },
         {
            "cluster_id": "0x6d4760c01291090f",
            "cluster_name": "Boston"
         }
      ],
      "compatibility_status": "COMPATIBLE",
      "creation_time": 1436042070825,
      "global": false,
      "id": "urn:storageos:ProtectionSystem:215489d1-
d601-4ba3-872d-0cfa309c7a89:vd
c1",
      "inactive": false,
      "installation_id": "Boston_Hopkinton_Moncton_NewYork_Syracuse",
      "internal": false,
      "ip_address": "10.247.169.16",
      "job_discovery_status": "COMPLETE",
      "job_metering_status": "CREATED",
      "last_discovery_run_time": 1436042096372,
      "last_discovery_status_message": "Discovery completed
successfully for Protec
tion System: urn:storageos:ProtectionSystem:215489d1-
d601-4ba3-872d-0cfa309c7a89
:vdc1",
      "last_metering_run_time": 0,
      "link": {
         "href": "/vdc/protection-systems/
urn:storageos:ProtectionSystem:215489d1-d
601-4ba3-872d-0cfa309c7a89:vdc1",
         "rel": "self"
      },
      "major_version": "4.1.SP1.P1(h.167)",
      "name": "rp_west1",
      "native_guid": "RP+Boston_Hopkinton_Moncton_NewYork_Syracuse",
      "next_discovery_run_time": 1436045671491,
      "next_metering_run_time": 0,
      "port_number": 7225,
      "registration_status": "REGISTERED",
      "remote": false,
      "success_discovery_time": 1436042096372,
      "success_metering_time": 0,
      "system_type": "rp",
      "tags": [],
      "user_name": "admin",
      "vdc": {
         "id":
"urn:storageos:VirtualDataCenter:ebd8a6f7-41db-4f0d-8e8e-014b8badcc7
5:vdc1",
         "link": {
            "href": "/vdc/
urn:storageos:VirtualDataCenter:ebd8a6f7-41db-4f0d-8e8e-0
14b8badcc75:vdc1",
            "rel": "self"
         }
      }
}
```

Example 138 (continued)

viprcli protectionsystem update

Modify the parameters of a ViPR Controller protection system.

Syntax

```
    viprcli protectionsystem update
    -name|n <psname>
    [-deviceip|dip
<deviceip>]
    [-deviceport|dp
<deviceport>]
    [-clusterid|cid
<clusterid>]
    [-varray_add|vaadd <varray_add>
[varray_add> ...]]
    [-varray_remove|varemove
<varray_remove> [<varray_remove> ...]]
    [-username|un
<username>]
    [-type|t
<protectionsystemtype>]
    [-hostname <hostname>]
    [-cookiefile|f
<cookie_file>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-h]
```

Description

`viprcli protectionsystem update` command modifies one of these protection system parameters.

- device ip
- device port
- user name
- cluster id
- virtual array to which the RecoverPoint cluster is associated
- the type of protection system.

Note

Currently, protection system type only has one valid value: rp. The value cannot be changed in this version of ViPR Controller.

Options

name|n

The name of the protection system to be updated. This is a mandatory parameter.

deviceip|dip

The IP address of a protection system. This is an optional parameter.

deviceport|dp

Specify 7225 for RecoverPoint systems. This is the HTTPS protocol port for communicating with the RecoverPoint API. This is an optional parameter.

username|u

A user id for accessing the data protection system. This is an optional parameter.

type|t

Currently the only valid value is `rp`. This is an optional parameter.

clusterid|cid

The protection system cluster ID or cluster name. This is an optional parameter.

varray_add|vaadd

List of new virtual arrays to be associated with the cluster. If this field contains any entries for a specific RecoverPoint cluster, then the virtual array for this cluster is considered to be explicitly assigned, and only those virtual arrays will be allowed to use the RecoverPoint cluster. This is an optional parameter.

Note

To add more than one virtual array, separate the ID of each virtual array with a space.

varray_remove|varemove

List of virtual arrays to be dis-associated from the cluster. This is an optional parameter.

Note

To remove more than one virtual array, separate the ID of each virtual array with a space.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 139

```
# viprcli protectionsystem update -cf /root/rootcookie29669 -hostname 10.247.101.40 -n
rp_protection_system -cid Syracuse -varray_remove urn:storageos:VirtualArray:e5cfc34d-
e850-48a0-bbd2-67b94f7be816:vdc1 -varray_add urn:storageos:VirtualArray:c763d1ca-a94b-4efb-
ae78-d5176b587c53:vdc1
```

viprcli protectionsystem connectivity

Display information about a protection system and its associated storage.

Syntax

```
viprcli protectionsystem connectivity
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-name|n <psname>]
```

Description

`viprcli protectionsystem connectivity` command displays the following information.

Table 6 Description of command output

Field	Description
protection_site	The RecoverPoint system in the RecoverPoint cluster.
protection_system id	The ID is the protection system within ViPR Controller.
protection_system link	The resource that returns information on this protection system, and its rel value.
protection_system href	The resource that returns information on this protection system.
protection_system rel	The relationship of the listed resource to the linked object or document, typically "self".
name	This name appears in the ViPR Controller portal.

Options

name|n

The name of protection system to display. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 140

Example 140 (continued)

```
# viprcli protectionsystem connectivity -n chartsystem
{
   "protection_site": [],
   "protection_system": {
      "id": "urn:storageos:ProtectionSystem:80e0b83f-f72f-482b-
af8e-3f0ae0063ac5:",
      "link": {
         "href": "/vdc/protection-systems/
urn:storageos:ProtectionSystem:80e0b83f-f72f-482b-
af8e-3f0ae0063ac5:",
         "rel": "self"
      },
      "name": "chartsystem"
   }
}
```

viprcli protectionsystem delete

Delete a protection system from ViPR Controller

Syntax

```
viprcli protectionsystem delete
-name|n <psname>
[-type|t <protectionsystemtype>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli protectionsystem delete` command removes a previously-created protection system from ViPR Controller. RecoverPoint is the only protection system currently supported by ViPR Controller. This command is silent on success.

Options

name|n

The name of protection system to delete. This is a mandatory parameter.

type|t

Currently the only valid value is `rp`. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 141

Example 141 (continued)

```
# viprcli protectionsystem delete -name chartsystem
```

CHAPTER 33

viprcli quotadirectory commands

This chapter contains the following topics:

- viprcli quotadirectory create 350
- viprcli quotadirectory delete 351
- viprcli quotadirectory list 352
- viprcli quotadirectory show 353
- viprcli quotadirectory update 354

viprcli quotadirectory create

Create a quota directory.

Syntax

```
        viprcli quotadirectory create
        [-h]
        [-hostname <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        -name|n
    <computesystemname>
        -filesystem|fs
    <filesystem>
        -project|pr
    <projectname>
        [-tenant|tn <tenantname>]
        [-size|s <size>]
        [-oplock|oplk <oplock>]
        [-securitystyle|secsy
    <securitystyle>]
        [-advisorylimit|advlmt
    <advisorylimit>]
        [-softlimit|softlmt
    <softlimit>]
        [-graceperiod|grace
    <graceperiod>]
        [-synchronous|sync]
        [-synctimeout|syncto]
```

Description

The viprcli quotadirectory create command creates a quota directory.

Options

name|n

Name of the quota directory. This is a mandatory parameter.

filesystem|fs

Name of the file system. This is a mandatory parameter.

project|pr

Name of the project. This is a mandatory parameter.

tenant|tn

Name of the tenant. This is an optional parameter.

size|s

Size of the quota directory. This is an optional parameter.

oplock|oplk

The oplock for the quota directory. This is an optional parameter.

securitystyle|secsy

The security style of the quota directory. This is an optional parameter.

advisorylimit|advlmt

Advisory limit in percentage for the file system. This is an optional parameter.

softlmit|softlmt

Soft limit in percentage for the file system. This is an optional parameter.

graceperiod|grace

Grace period in days for the soft limit. This is an optional parameter.

synchronous|sync

Synchronous quota directory create. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli quotadirectory delete

Delete a quota directory.

Syntax

```
    viprcli quotadirectory delete
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n <name>
    -filesystem|fs
<filesystem>
    -project|pr
<projectname>
    [-tenant|tn
<tenantname>]
    [-forcedelete|fd]
    [-synchronous|sync]
    [-synctimeout|syncto]
```

Description

The viprcli quotadirectory delete command deletes a quota directory.

Options

name|n

The name of the quota directory. This is a mandatory parameter.

filesystem|fs

The name of the file system. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

forcedelete|fd

Force delete the quota directory. This is an optional parameter.

synchronous|sync

Perform a synchronous quota directory delete. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli quotadirectory list

List a quota directory.

Syntax

```
viprcli quotadirectory list
-filesystem|fs <filesystem>
-project|pr <projectname>
-verbose|v
-long|l
-tenant|tn <tenantname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli quotadirectory list command lists a quota directory.

Options

filesystem|fs

The name of the file system.

project|pr

The name of the project.

verbose|v

Lists the quota directories with details.

long|l

Lists the quota directories with details in a table.

tenant|tn

The name of the tenant.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli quotadirectory show

Show a quota directory.

Syntax

```
viprcli quotadirectory show
-name|n <computesystemname>
-filesystem|fs <filesystem>
-project|pr <projectname>
-xml
-tenant|tn <tenantname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli quotadirectory show command shows a quota directory.

Options

name|n

The name of the quota directory.

filesystem|fs

The name of the file system.

project|pr

The name of the project.

xml

The XML response.

tenant|tn

The name of the tenant.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli quotadirectory update

Update a quota directory.

Syntax

```
    viprcli quotadirectory update
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -filesystem|fs
<filesystem>
    -name|n
<computesystemname>
    -project|pr
<projectname>
    [-tenant|tn
<tenantname>]
    [-size|s <size>]
    [-oplock|oplk <oplock>]
    [-securitystyle|secsy
<securitystyle>]
    [-advisorylimit|advlmt
<advisorylimit>]
    [-softlimit|softlmt
<softlimit>]
    [-graceperiod|grace
<graceperiod>]
```

Description

The viprcli quotadirectory update command updates a quota directory.

Options

filesystem|fs

The name of the file system. This is a mandatory parameter.

name|n

The name of the quota directory. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

size|s

The size of the quota directory. This is an optional parameter.

oplock|oplk

The oplock for the quota directory. This is an optional parameter.

securitystyle|secsy

The security style of the quota directory. This is an optional parameter.

advisorylimit|advlmt

Advisory limit in percentage for the file system. This is an optional parameter.

softlimit|softlmt

Soft limit in percentage for the file system. This is an optional parameter.

graceperiod|grace

Grace period in days for the soft limit. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 142

```
viprcli quotadirectory update -n qd1 -fs fs4 -pr Project1 -advlmt
20 -size 512MB -grace 2
```

CHAPTER 34

viprcli sanfabrics commands

This chapter contains the following topics:

- viprcli sanfabrics activate 358
- viprcli sanfabrics get-sanzone 358
- viprcli sanfabrics list 359
- viprcli sanfabrics list-sanzones 360
- viprcli sanfabrics show 360
- viprcli sanfabrics update 361

viprcli sanfabrics activate

Activates a VSAN or fabric.

Syntax

```
viprcli sanfabrics activate
-name|n <computesystemname>
-fabricid|fid <fabricid>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli sanfabrics activate command activates a VSAN or fabric.

Options

name|n

The name of the network system.

fabricid|fid

The name of the VSAN or fabric.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli sanfabrics get-sanzone

Get the SAN zone for a VSAN or fabric.

Syntax

```
viprcli sanfabrics get-sanzone
-name|n <computesystemname>
-fabricid|fid <fabricid>
-sanzonename|zn <san_zone_name>
-xml
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli sanfabrics get-sanzone command gets the SAN zone for a VSAN or fabric.

Options

name|n

The name of the network system.

fabricid|fid

The name of the VSAN or fabric.

sanzonename|zn

The name of the SAN zone.

xml

The XML response.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli sanfabrics list

List a VSAN or fabric.

Syntax

```
viprcli sanfabrics list
-name|n <computesystemname>
-verbose|v
-long|l
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli sanfabrics list command lists a VSAN or fabric.

Options

name|n

The name of the network system.

verbose|v

List SAN zones of a fabric or VSAN.

long|l

List SAN zones of a fabric or VSAN.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli sanfabrics list-sanzones

List the SAN zones for a VSAN or fabric.

Syntax

```
    viprcli sanfabrics list-sanzones
    -name|n
<computesystemname>
    -fabricid|fid
<fabricid>
    [-verbose|v | -long|l]
    [-excludealiases|exal]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

The viprcli sanfabrics list-sanzones command lists the SAN zones for a VSAN or fabric.

Options

name|n

The name of the network system. This is a mandatory parameter.

fabricid|fid

The name of the VSAN or fabric. This is a mandatory parameter.

{verbose|v | long|l}

List SAN zones of a fabric or VSAN. You can use either verbose or long, but not both. This is an optional parameter.

excludealiases|exal

Exclude the aliases in the list. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli sanfabrics show

Show a VSAN or fabric.

Syntax

```
    viprcli sanfabrics show
    -name|n <computesystemname>
    -fabricid|fid <fabricid>
```

```
-xml
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli sanfabrics show command shows a VSAN or fabric.

Options

name|n

The name of the network system.

fabricid|fid

The name of the VSAN or fabric.

xml

The XML response.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli sanfabrics update

Updates a VSAN or fabric.

Syntax

```
viprcli sanfabrics update
-name|n <computesystemname>
-fabricid|fid <fabricid>
-sanzonename|zn <san_zone_name>
-operation|op <san-zone-operation>
-add_wwpn <add_wwpn> [<add_wwpn> ...]
-remove_wwpn|awp <remove_wwpn> [<remove_wwpn> ...]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

The viprcli sanfabrics update command updates a VSAN or fabric.

Options

name|n

The name of the network system.

fabricid|fid

The name of the VSAN or fabric.

sanzonename|zn

The name of the SAN zone.

operation|op

The SAN zone operation.

remove_wwpn|awp

A list of WWPN addresses of a SAN zone, for example:wwpn1 wwpn2.

add_wwpn

A list of WWPN addresses of a SAN zone, for example:wwpn1 wwpn2.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 35

viprcli schedule-policy commands

This chapter contains the following topics:

- viprcli schedule-policy create 364
- viprcli schedule-policy delete 365
- viprcli schedule-policy list 366
- viprcli schedule-policy show 367
- viprcli schedule-policy update 367

viprcli schedule-policy create

Creates a policy, which defines regularly scheduled intervals for when the ViPR Controller creates snapshots of file systems. The command also defines the retention period, defining how long to keep a snapshot before deleting the snapshot.

Syntax

```
    viprcli schedule-policy create
    [-h]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -policyname|polname
<policyname>
    -policytype|poltype
<policytype>
    -schedulefrequency|schedfreq
{days,weeks,months}
    -schedulerepeat|schedrep
<schedulerepeat>
    -scheduletime|schedtim
<scheduletime>
    [-scheduleDayOfWeek|scheddow
<scheddow>]
    [-scheduleDayOfMonth|scheddom
<scheddom>]
    [-expiretype|exptype
{hours,days,weeks,months,never}]]
    [-expirevalue|expval
<expirevalue>]
    -tenant|tn <tenantname>
```

Description

The `viprcli schedule-policy create` command creates a policy, which defines regularly scheduled intervals for when the ViPR Controller creates snapshots of file systems. The `schedule-policy create` command also defines the retention period, defining how long to keep a snapshot before deleting the snapshot.

Options

policyname|polname

Name of the policy, for example, policy1. This is a mandatory parameter.

policytype|poltype

Type of the policy, for example, snapshot. This is a mandatory parameter.

schedulefrequency|schedfreq

Schedule frequency of the policy, for example, days or weeks. This is a mandatory parameter.

schedulerepeat|schedrep

Schedule repeat for the policy, for example, 1. This is a mandatory parameter.

scheduletime|schedtim

Schedule time for the policy, for example, 10:20 AM. This is a mandatory parameter.

scheduleDayOfWeek|scheddow

Scheduled day of week. This is an optional parameter.

scheduleDayOfMonth|scheddom

Scheduled day of month. This is an optional parameter.

expiretype|exptype

Snapshot schedule expire type. This is an optional parameter.

expirevalue|expval

Snapshot schedule expire value. This is an optional parameter.

tenant|tn

Name of tenant. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

```
viprcli schedule-policy create -polname policy1 -poltype snapshot -
schedfreq days -schedrep 1 -schedtim 2 -tn "Provider Tenant"
```

viprcli schedule-policy delete

Deletes a schedule policy.

Syntax

```
            viprcli schedule-policy delete
            [-h]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -tenant|tn <tenantname>
            -policyname|polnm
        <policy_name>
```

Description

The `viprcli schedule-policy delete` command deletes a schedule policy.

Options

policyname|polnm

The policy name, for example, policy1. This is a mandatory parameter.

tenant|tn

The name of a tenant. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 143

```
viprcli schedule-policy delete -polnm policy4 -tn "Provider Tenant"
```

viprcli schedule-policy list

Lists the schedule policies.

Syntax

```
            viprcli schedule-policy list
            [-h]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cf <cookiefile>]
            -tenant|tn <tenantname>
```

Description

The `viprcli schedule-policy list` command lists the schedule policies.

Options

tenant|tn

Name of tenant. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 144

```
viprcli schedule-policy list -tn "Provider Tenant"
NAME
policy4
policy3
Weekly_Snap_Policy12
policy1
Weekly_Snap_Policy16
policy2
Daily_Snap_Policy1
Daily_Snap_Policy4
Weekly_Snap_Policy10
```

viprcli schedule-policy show

Presents the schedule policy settings.

Syntax

```
viprcli schedule-policy show
[-h]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-tenant|tn <tenantname>
-policyname|polnm
<policyname>
```

Description

The `viprcli schedule-policy show` command presents the schedule policy settings.

Options

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

policyname|polnm

The policy name, for example, policy1.

common arguments

This operation also takes the Common Parameters on page 24.

Example 145

```
viprcli schedule-policy show -tn "Provider Tenant" -polnm policy1
```

viprcli schedule-policy update

Edits a schedule policy.

Syntax

```
viprcli schedule-policy update
[-h]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
```

```
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-policyname|polname
        <policyname>]
            [-newpolicyname|newpolname
        <newpolicyname>]
            [-policytype|poltype
        <policytype>]
            [-schedulefrequency|schedfreq
        {days,weeks,months]
            [-schedulerepeat|schedrep
        <schedulerepeat>
            [-scheduletime|schedtim
        <scheduletime>]
            [-scheduleDayOfWeek|scheddow
        <scheduledow>]
            [-scheduleDayOfMonth|scheddom
        <scheduledom>]
            [-expiretype|exptype
        {hours,days,weeks,months,never}]
            [-expirevalue|expval
        <expirevalue>]
            -tenant|tn <tenantname>
```

Description

The `viprcli schedule-policy update` command edits a schedule policy.

Options

policyname|polname

The policy name, for example, policy1. This is a mandatory parameter.

newpolicyname|newpolname

New name for the policy. This is an optional parameter.

policytype|poltype

Type of the policy. This is an optional parameter.

schedfrequency|schedfreq

Schedule frequency of the policy, for example, days or weeks. This is an optional parameter.

schedulerepeat|schedrep

Schedule repeat for the policy, for example, 1. This is an optional parameter.

scheduletime|schedtim

Schedule time for the policy, for example, 10:20 AM. This is an optional parameter.

scheduleDayOfWeek|scheddow

Scheduled day of the week. This is an optional parameter.

scheduleDayOfMonth|scheddom

Scheduled day of month. This is an optional parameter.

expiretype|exptype

Snapshot schedule expire type. This is an optional parameter.

expireval|expval

Snapshot schedule expire value. This is an optional parameter.

tenant|tn

The name of a tenant. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 146

```
viprcli schedule-policy update -polname policynewagain6 -tn
"Provider Tenant" -schedulerepeat 4 -scheduletime "10:42" -expval 4
```

Example 147

```
viprcli schedule-policy update -polname policy8 -tn "Provider
Tenant" -expval 4
```

Example 148

```
viprcli schedule-policy update -polname policy8 -tn "Provider
Tenant" -schefreq weeks
```

Example 149

```
viprcli snapshot snapshot-policy-update -polname policy8 -tn
"Provider Tenant" -schefreq weeks -scheddow Tuesday
```

CHAPTER 36

viprcli scheduled_event commands

This chapter contains the following topics:

- viprcli scheduled_event create 372
- viprcli scheduled_event get 374
- viprcli scheduled_event cancel 375
- viprcli scheduled_event delete 375
- viprcli scheduled_event update 375

viprcli scheduled_event create

Use this command to create one of the following orders with a scheduled event:

- Create Block Snapshot
- Create Block Full Copy
- Create a Filesystem Snapshot

To use this feature you will need:

- Ids for the following components:
  - Service catalog
  - Tenant
  - Project
- To create a file with the scheduled event parameters in JSON format (see below for details).

Command syntax

```
vipr scheduled_event create
[-f <file path and name.]
```

f

The path and name of the file containing the scheduled event parameters. This is a mandatory field.

Command example:

```
vipr scheduled_event create -f ./createScheduledEventScript
```

Create scheduled event file

The create scheduled event file should be created in JSON format as demonstrated below:

```
 { "scheduleInfo": {
    "hourOfDay": 12,
    "minuteOfHour": 30,
    "durationLength": 3600,
    "cycleType": "DAILY",
    "cycleFrequency": 1,
    "sectionsInCycle": [

    ],
    "startDate": "2016-10-05",
    "reoccurrence": 10,
    "endDate": null,
    "dateExceptions": null
  },
  "orderCreateParam": {
    "parameters": [
      {
        "label": "volumes",
```

```
        "value": "\"urn:storageos:Volume:5eeb3d7e-bf9f-4afc-
bfec-1fdf9f70afd0:vdc1\"",
        "userInput": true,
        "encrypted": false
      },

      {
        "label": "name",
        "value": "\"test-cli-{datetime}\"",
        "userInput": true,
        "encrypted": false
      },
      {
        "label": "project",
        "value": "\"urn:storageos:Project:ab5a1ef9-96b2-4631-
ab79-232337b21368:global\"",
        "userInput": true,
        "encrypted": false
      },
      {
        "label": "readOnly",
        "value": "\"false\"",
        "userInput": true,
        "encrypted": false
      },
      {
        "label": "type",
        "value": "\"local\"",
        "userInput": true,
        "encrypted": false
      }
    ]
    "additionalScheduleInfo":5,
    "catalogService": "urn:storageos:CatalogService:
6ff0ed89-1aad-4fff-8461-c9c3bfa82bb9:vdc1",
    "tenantId": "urn:storageos:TenantOrg:f735c88c-
f59c-45ed-9b06-6779d24fb7b7:global",
    "scheduledEventId": null,
    "scheduledTime": null
  }
}
```

Note the following when entering the `scheduleInfo` parameters:

Parameter	Description
hourOfDay and minuteOfHour	The time of day (UTC) the scheduled event will run.
durationLength	If the order is unable to execute at the scheduled time, ViPR Controller will continue to attempt to run the order for 3600 seconds after the set time. This parameter cannot be changed.
cycleType	How often to run the service order. Options are: Hourly, Daily, Weekly, Monthly
cycleFrequency	How often to run the order during the time defined in the cycleType , for example 1 Day would be daily, 2 Days, would schedule it to run every other day.
sectionsInCycle	If you entered a Weekly, or Monthly cycleType, then you can enter which day of the week, or month to schedule

Parameter	Description
	the event. Weekly values are 1 - 7, 1 being Sunday and 7 being Saturday. Monthly values are 1 - 31.
startDate	The date and time to run the first scheduled event. Format is: yyyy-mm-dd for example 2016-10-5
reoccurence	If you want a recurring scheduled event that stops after a certain number of cycles, enter the number of recurrences to schedule here. Once you have reached the given number of recurrences, no more snapshots or copies will be taken for this order. A zero value will run the scheduled event indefinitely.
endDate	To set an end date for when the schedule events will stop. Format is: yyyy-mm-dd for example 2016-10-5
dateExceptions	To list one or more dates when the service will not be run. Format is: yyyy-mm-dd, yyyy-mm-dd for example 2016-11-30, 2016-12-31 In this example, the order would not run on 2016-11-30 or 2016-12-31.
additionalScheduleInfo	The maximum number of snapshots or copies to keep. Once the retention is met, the oldest snapshot or copy is removed before the new one is created. It is recommended that you do not manually remove any of the snapshots or copies being managed by the ViPR Controller scheduler.

Note

When working with block full copy orders, if you have set the order to create multiple copies, Automatic Expiration applies to all the copies created for the order for example: If you set the order to create 3 full copies, and you set automatic expiration to keep 5 full copies, ViPR Controller, will maintain 15 (3x5) full copies, and delete the three oldest copies after the retention is met.

viprcli scheduled_event get

Use this command to view the parameters of a scheduled event.

To use this feature you will need the schedule event ID.

Syntax

```
vipr scheduled_event get
[-id <file path and name.]
```

id

The schedule event ID. This is a mandatory field.

Command example:

```
vipr scheduled_event get -id urn:storageos:ScheduledEvent:
33d3e6fa-0abd-4a93-a126-5f29e771e810:vdc1
```

viprcli scheduled_event cancel

Use this command to cancel the scheduled event for an order.

To use this feature you will need the schedule event ID.

Syntax

```
vipr scheduled_event cancel
[-id <file path and name.]
```

id

The schedule event ID. This is a mandatory field.

Command example:

```
vipr scheduled_event cancel -id urn:storageos:ScheduledEvent:
33d3e6fa-0abd-4a93-a126-5f29e771e810:vdc1
```

viprcli scheduled_event delete

Use this command to delete a scheduled event.

To use this feature you will need the schedule event ID.

Syntax

```
vipr scheduled_event delete
[-id <file path and name.]
```

id

The schedule event ID. This is a mandatory field.

Command example:

```
vipr scheduled_event delete -id urn:storageos:ScheduledEvent:
33d3e6fa-0abd-4a93-a126-5f29e771e810:vdc1
```

viprcli scheduled_event update

Use this command to make changes to a scheduled event.

To use this feature you will need:

- The schedule event ID
- To create a file with the scheduled event parameters in JSON format (see below for details).

Command syntax

```
vipr scheduled_event update
[-id <file path and name.]
[-f <file path and name.]
```

id

The schedule event ID. This is a mandatory field.

f

The path and name of the update file containing the scheduled event parameters. This is a mandatory field.

Command example:

```
vipr scheduled_event update -id urn:storageos:ScheduledEvent:
33d3e6fa-0abd-4a93-a126-5f29e771e810:vdc1 -f ./
updateScheduledEventScript
```

Update scheduled event file

The update scheduled event file should be created in JSON format as demonstrated below:

```
{
  "additionalScheduleInfo":5,
  "scheduleInfo": {
    "hourOfDay": 0,
    "minuteOfHour": 0,
    "durationLength": 3600,
    "cycleType": "DAILY",
    "cycleFrequency": 8,
    "sectionsInCycle": [

    ],
    "startDate": "2016-11-05",
    "reoccurrence": 12,
    "endDate": null,
    "dateExceptions": null
  }
}
```

Note the following when entering the `scheduleInfo` parameters:

Parameter	Description
hourOfDay	The time of day (UTC) the scheduled event will run.
minuteOfHour	The time of day (UTC) the scheduled event will run.
durationLength	If the order is unable to execute at the scheduled time, ViPR Controller will continue to attempt to run the order for 3600 seconds after the set time. This parameter cannot be changed.
cycleType	How often to run the service order. Options are: Hourly, Daily, Weekly, Monthly

Parameter	Description
cycleFrequency	How often to run the order during the time defined in the cycleType , for example 1 Day would be daily, 2 Days, would schedule it to run every other day.
sectionsInCycle	If you entered a Weekly, or Monthly cycleType, then you can enter which day of the week, or month to schedule the event. Weekly values are 1 - 7, 1 being Sunday and 7 being Saturday. Monthly values are 1 - 31.
startDate	The date and time to run the first scheduled event. Format is: yyyy-mm-dd for example 2016-10-5
reoccurence	If you want a recurring scheduled event that stops after a certain number of cycles, enter the number of recurrences to schedule here. Once you have reached the given number of recurrences, no more snapshots or copies will be taken for this order. A zero value will run the scheduled event indefinitely.
endDate	To set an end date for when the schedule events will stop. Format is: yyyy-mm-dd for example 2016-10-5
dateExceptions	To list one or more dates when the service will not be run. Format is: yyyy-mm-dd, yyyy-mm-dd for example 2016-11-30, 2016-12-31 In this example, the order would not run on 2016-11-30 or 2016-12-31.
additionalScheduleInfo	The maximum number of snapshots or copies to keep. Once the retention is met, the oldest snapshot or copy is removed before the new one is created. It is recommended that you do not manually remove any of the snapshots or copies being managed by the ViPR Controller scheduler.

Note

When working with block, or application full copy orders, if you have set the order to create multiple copies, Automatic Expiration applies to all the copies created for the order for example: If you set the order to create 3 full copies, and you set automatic expiration to keep 5 full copies, ViPR Controller, will maintain 15 (3x5) full copies, and delete the three oldest copies after the retention is met.

CHAPTER 37

viprcli snapshot commands

This chapter contains the following topics:

- viprcli snapshot activate 380
- viprcli snapshot create 381
- viprcli snapshot delete 384
- viprcli snapshot delete-acl 385
- viprcli snapshot export-file 386
- viprcli snapshot export-rule 388
- viprcli snapshot import-to-vplex 390
- viprcli snapshot list 391
- viprcli snapshot list-acl 393
- viprcli snapshot restore 394
- viprcli snapshot resync 396
- viprcli snapshot share-acl 397
- viprcli snapshot show 398
- viprcli snapshot show-exports 400
- viprcli snapshot show-shares 401
- viprcli snapshot tasks 402
- viprcli snapshot tag 404
- viprcli snapshot unexport-file 405
- viprcli snapshot expand 406

viprcli snapshot activate

Activate a snapshot and establish the synchronization between the source volume or consistency group and the target snapshot.

Syntax

```
viprcli snapshot activate
    [-h]
    [-hostname|hn <hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n <snapshotname>
    [-tenant|tn <tenantname>
    -project|pr <project>
    [-consistencygroup|cg
<consistencygroup>]
    [-volume|vol
<volumename>]
    [-synchronous|sync]
    [-synctimeout|syncto]
```

Description

`viprcli snapshot activate` command activates a snapshot, establishing the synchronization between the source volume or consistency group and the target snapshot. The activate command is only used on for block devices like VNX Block and VMAX.

Options

name|n

The name of a valid ViPR Controller snapshot. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

consistencygroup|cg

The name of a consistency group in ViPR Controller project. This is an optional parameter.

volume|vol

The name of a valid volume in ViPR Controller. This is an optional parameter.

synchronous|sync

Perform the snapshot creation synchronously. If this argument is not specified, the snapshot creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 150

Create a snapshot of a volume with inactivate= true. In this example, the snapshot is created, then set in an active state with the viprcli snapshot activate command. Once the snapshot is activated, the user can restore the volume from the snapshot. An inactive snapshot cannot be restored.

```
# viprcli snapshot create -tn TENANT
-pr IMG_PROJECT -vol  IMG_VOL  -name SNAP1 -inactive true

# viprcli snapshot activate -tn TENANT
-pr IMG_PROJECT -vol  IMG_VOL  -name SNAP1

# viprcli snapshot show  -tn TENANT
-pr IMG_PROJECT -vol  IMG_VOL -name SNAP1

# viprcli snapshot restore -tn TENANT
-pr IMG_PROJECT -vol  IMG_VOL -name SNAP1
```

viprcli snapshot create

Create a snapshot of a specified file system, volume, or consistency group.

Syntax

```
            viprcli snapshot create
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n <snapshotname>
            [-tenant|tn <tenantname>
            -project|pr
        <projectname>
            [-inactive|ci {TRUE | FALSE}]
            [-type|t <type>
            [-readonly|ro
        <true,false>
            [-synchronous|sync]
            [-synctimeout|syncto]
            [-filesystem|fs
        <filesystemname>]
            [-volume|vol
        <volumename>]
            [-consistencygroup|cg
```

```
        <consistencygroup>]
```

Description

`viprcli snapshot create` command requires a filesystem, volume, or consistency group name, and the label of the snapshot. This command creates a snapshot on the array. It is valid for all supported arrays. This command is silent on success.

Options

name|n

The name of a valid ViPR Controller snapshot. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

inactive|ci

You must activate the snapshot before you restore the block snapshot for a volume. This is an optional parameter.

type|t

Bookmark of specific type, such as rp (Recover Point). This is an optional parameter.

readonly|ro

Creates a snapshot in read-only mode when set to true. This is an optional parameter.

synchronous|sync

Perform the snapshot creation synchronously. If this argument is not specified, the snapshot creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

filesystem|fs

The name of a valid filesystem in ViPR Controller. This is an optional parameter.

volume|vol

The name of a valid volume in ViPR Controller. This is an optional parameter.

consistencygroup|cg

The name of a consistency group in ViPR Controller project. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 151

```
# viprcli snapshot create -n snap2_volume2
-project vipr_project -vol my_volume2
```

Use the list command to view results.

```
# viprcli snapshot list -pr vipr_project -vol my_volume2
  NAME
  snap1_volume2,snap1_volume2
  snap2_volume2,snap2_volume2
```

Example 152

-Create Snapshot for SRDF source volume

```
# viprcli snapshot create -n snap2 -pr sanity -vol mysrdfvol
```

Use the list command to view results.

```
# viprcli snapshot list -l -pr sanity -vol mysrdfvol
  NAME                      IS_SYNC_ACTIVE
WWN
                                       TAGS
  snap2     True                 6000097000019570150553303033443 2
```

Example 153

-Create Snapshot for SRDF target volume

```
# viprcli snapshot create -n snapr2 -pr sanity -vol mysrdfvol-
target-myvarray -sync
```

Use the list command to view results.

```
# viprcli snapshot list -l -pr sanity -vol mysrdfvol-target-myvarray
  NAME                      IS_SYNC_ACTIVE
WWN
                                     TAGS
  snapr2    True                 6000097000019570157353303034323 5
```

viprcli snapshot delete

Delete the specified snapshot.

Syntax

```
viprcli snapshot delete
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n <snapshotname>
    [-tenant|tn <tenantname>
    -project|pr <project>
    [-filesystem|fs
<filesystemname>]
    [-volume|vol
<volumename>]
    [-consistencygroup|cg
<consistencygroup>]
    [-synchronous|sync]
    [-synctimeout|syncto]
```

Description

`viprcli snapshot delete` command deletes the specified snapshot.

Options

name|n

The name of a valid ViPR Controller snapshot. This is a mandatory parameter.

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

filesystem|fs

The name of a valid file system in ViPR Controller. This is an optional parameter.

volume|v

The name of a valid volume in ViPR Controller. This is an optional parameter.

consistencygroup|cg

The name of a consistency group in ViPR Controller project. This is an optional parameter.

synchronous|sync

Perform the snapshot creation synchronously. If this argument is not specified, the snapshot creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 154

```
viprcli snapshot list -pr bob_project -vol my_volume2
  NAME
  snap1_volume2,snap1_volume2
  snap2_volume2,snap2_volume2
standalone:~ # viprcli snapshot delete -n snap1_volume2 -pr
bob_project -vol my_volume2
standalone:~ # viprcli snapshot list -pr bob_project -vol my_volume2
  NAME
  snap2_volume2,snap2_volume2
```

viprcli snapshot delete-acl

Delete all the access control lists from a snapshot share.

Syntax

```
            viprcli snapshot delete-acl
            [-fsname|fsn
        <filesystem_name>]
            [-sname|sn
        <snapshot_name>]
            [-share|sh
        <share_name>]
            [-project|pr
        <project_name>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

The `viprcli snapshot delete-acl` command deletes all the access control lists from a snapshot share.

Options

fsname|fsn

Name of the file system from which the snapshot is taken. This is a mandatory parameter.

sname|sn

Name of snapshot. This is a mandatory parameter.

share|sh

Name of share. This is a mandatory parameter.

project|pr

Name of project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

The following example is used to delete all ACLs on a snapshot share:

```
viprcli snapshot delete-acl -fsname netcproshare1 -sname snap-
netcproshare1 -share snapcifsnetpro1 -project netpro
```

viprcli snapshot export-file

Export a file system snapshot.

Syntax

```
    viprcli snapshot export-file
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n <snapshotname>
    -filesystem|fs
<filesystemname>
    [-tenant|tn
<tenantname>
    -project|pr <project>
    -protocol|pl {NFS | CIFS}
    [-security|sec
<securitytype>]
    [-permission|pe
<permission>]
    [-rootuser|ru
<root_user>]
    [-endpoints|ep
<endpoint>]
    [-share|sh <sharename>]
    [-description|desc
<description>]
    [-subdir <subdir>]
    [-permission_type|pt
<permissiontype> {allow | deny}]
    [-synchronous|sync]
```

```
              [-synctimeout|syncto]
```

Description

`viprcli snapshot export-file` command is used only for exporting file system snapshots.

Note

You can only export file system snapshots of VNX File arrays using this command. Exporting an Isilon file system snapshot using this command shows an error message. This is because, a file system snapshot is automatically exported by default when you export a file system of an Isilon array using the `viprcli filesystem export` command.

Options

name|n

The name of a valid ViPR Controller snapshot. This is mandatory parameter.

filesystem|fs

The name of a valid filesystem in ViPR Controller. This is mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is mandatory parameter.

protocol|pl

File protocol - one of NFS , NFSv4 or CIFS. This is mandatory parameter.

security|sec

Security Type. Choose one of sys, krb5, krb5i, or krb5p. This is an optional parameter.

permission|pe

Filesystem access permissions, one of ro, rw, or root. This is an optional parameter.

rootuser|ru

Root user mapping for anonymous accesses, (nobody or root). This is an optional parameter.

endpoints|ep

A list of client endpoints in ip, net, or netgroup format. For an iSCSI host, the endpoint is an IQN. For an Ethernet host, the endpoint is an IP address. For a fibre channel host, the endpoint is a WWN. This is an optional parameter.

share|sh

The name of a Windows share. This parameter should be used for CIFS exports only. This is an optional parameter.

description|desc

A string description of a Windows share. This is an optional parameter.

subdir

Name of the sub directory. This is an optional parameter.

permission_type|pt

Type of permission of the SMB share. The valid values are allow or deny. The default is allow. This is an optional parameter.

synchronous|sync

Synchronous filesystem export. By default, CLI commands are asynchronous. The -synchronous option waits until the fileshare snapshot export has been successfully completed before returning control to the shell. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 155

NFS share exported to a UNIX box using NFS protocol

```
# viprcli snapshot export-file -n TEST_SNAPSHOT1
-tn TEST_TENANT11 -pr TEST_PROJECT11 -fs TEST_ISILON_FILESHARE11
-ep 192.168.66.65 -pro NFS -ru root -perm rw -t sys
```

Example 156

CIFS share exported to a Windows box using CIFS protocol

```
# viprcli snapshot export-file -n TEST_SNAPSHOT1
-tn TEST_TENANT11 -pr TEST_PROJECT11 -fs TEST_ISILON_FILESHARE11
-share <Windows share name> -description 'Description String'
```

viprcli snapshot export-rule

Add, update, and delete export rules for file system snapshots. .

Syntax

```
            viprcli snapshot export-rule
            -name|n <snapshotname>
            -operation|op
        <exportruleoperation>
            -filesystem|fs
        <filesystemname>
```

```
            -securityflavor|sec
        <securityflavor>
            -project|pr <project>
            [-roothosts|rhosts <roothosts>
        [<roothosts>...]]
            [-readonlyhosts|rohosts
        <readonlyhosts> [<readonlyhosts>...]]
            [-readwritehosts|rwhosts
        <readwritehosts> [<readwritehosts>...]]
            [-tenant|tn
        <tenantname>]
            [-user|u <user>]
            [-subdirectory|subDir
        <subdirectory>]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli snapshot export-rule` command is used to add, update, and delete export rules for file system snapshots.

Options

name|n

The name of a valid ViPR Controller snapshot. This is a mandatory parameter.

operation|op

The name of the export rule operation. This is a mandatory parameter.

filesystem|fs

The name of a valid filesystem in ViPR Controller. This is a mandatory parameter.

securityflavor|sec

The name of a security flavor. This is a mandatory parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

roothosts|rhosts

The names of root hosts. This is an optional parameter.

readonlyhosts|rohosts

The names of read-only hosts. This is an optional parameter.

readwritehosts|rwhosts

The names of read-write hosts. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

user|u

The name of a user. This is an optional parameter.

subdirectory|subDir

Name of the sub directory. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 157

```
#
```

viprcli snapshot import-to-vplex

Import a VPLEX snapshot into VPLEX as a VPLEX volume.

Syntax

```
            viprcli snapshot import-to-vplex
            -name|n <snapshotname>
            -project|pr <project>
            [-tenant|tn
        <tenantname>]
            [-volume|vol
        <volumename>]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli snapshot import-to-vplex` command imports a VPLEX snapshot into VPLEX as a VPLEX volume. The command is silent on completion.

Options

name|n

The name of a valid ViPR Controller snapshot. This is a mandatory parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

volume|vol

The name of a valid volume in ViPR Controller. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 158

```
# viprcli snapshot import-to-vplex -name "KrisVPLEXSnap-1" -project
"KrisViPRCG" -volume "KrisVPLEX-2"
```

viprcli snapshot list

Display the list of snapshots under a given file system, volume, or consistency group name.

Syntax

```
            viprcli snapshot list
            -project|pr <project>
            [-filesystem|fs
        <filesystemname>]
            [-volume|vol
        <volumename>]
            [-consistencygroup|cg
        <consistencygroup>]
            [-tenant|tn
        <tenantname>]
            [{-verbose|v <verbose> | -long|l
        <long>}]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli snapshot list` command displays the list of snapshots under a give filesystem, volume, or consistency group name. If the verbose option is given, complete details for the snapshots are returned.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

filesystem|fs

The name of a valid filesystem in ViPR Controller. This is an optional parameter.

volume|vol

The name of a valid volume in ViPR Controller. This is an optional parameter.

consistencygroup|cg

The name of a consistencygroup in ViPR Controller project. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

{verbose|v | long|l}

These options provide additional information. Use either verbose or long.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 159

```
# viprcli snapshot list -pr my_project -vol my_volume2
  NAME
  snap1_volume2,snap1_volume2
```

Example 160

```
# viprcli snapshot list -pr my_project -vol my_volume2 -v
[
   {
      "creation_time": 1364926697177,
      "device_label": "1363807836865740",
      "id": "urn:storageos:BlockSnapshot:c43f31dd-dd5d-4b90-bf02-
a1f9bd865599:",
      "inactive": false,
      "is_sync_active": true,
      "link": {
         "href": "/block/snapshots/
urn:storageos:BlockSnapshot:c43f31dd-dd5d-4b90-bf02-a1f9bd865599:",
         "rel": "self"
      },
      "name": "snap1_volume2",
      "native_id": "1363807836865740",
      "neighborhood": {
         "id": "urn:storageos:varray:
202e2d28-6bad-48ea-82f9-977e018c7ae7:",
         "link": {
            "href": "/vdc/varray/urn:storageos:varray:
202e2d28-6bad-48ea-82f9-977e018c7ae7:",
            "rel": "self"
         }
      },
      "parent": {
         "id": "urn:storageos:Volume:bbd31a9a-34d8-44ad-
ab97-1cd24a0aac6f:",
         "link": {
            "href": "/block/volumes/
urn:storageos:Volume:bbd31a9a-34d8-44ad-ab97-1cd24a0aac6f:",
```

Example 160 (continued)

```
            "rel": "self"
         },
         "name": "snap1_volume2"
      },
      "protocols": [
         "iSCSI"
      ],
      "source_native_id": "1363807836865740",
      "storage_controller":
"urn:storageos:StorageSystem:d3fa8d0d-209d-440b-805e-1fd87c006073:",
      "wwn": "60000970000195900701533030323532"
   }
]
```

viprcli snapshot list-acl

Provides the list of access control entries on a CIFs snapshot share.

Syntax

```
viprcli snapshot list-acl
    [-fsname|fsn
<filesystem_name>]
    [-sname|sn
<snapshot_name>]
    [-share|sh
<share_name>]
    [-project|pr
<project_name>
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

The `viprcli snapshot list-acl` command provides the list of access control entries on a CIFs snapshot share.

Options

fsname|fsn

Name of the filesystem from which the snapshot is taken. This is a mandatory parameter.

sname|sn

Name of snapshot. This is a mandatory parameter.

share|sh

Name of share. This is a mandatory parameter.

project|pr

Name of project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

The following example is used to list the access control entries on a CIFs snapshot share.

```
viprcli snapshot list-acl -fsname netcproshare1 -sname snap-
netcproshare1 -project netpro -share snapcifsnetpro1
ERRORTYPE SNAPSHOT_ID PERMISSION SHARE_NAME USER GROUP
None urn:storageos:Snapshot:857fa072-adfa-4a5f-
b2b5-511639723a96:vdc1 Read snapcifsnetpro1 testfile
```

viprcli snapshot restore

Restore a volume or file system snapshot, overwriting the target file system, volume, or consistency group.

Syntax

```
    viprcli snapshot restore
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n <snapshotname>
    [-tenant|tn
<tenantname>]
    -project|pr <project>
    [-filesystem|fs
<filesystemname>]
    [-volume|vol
<volumename>]
    [-consistencygroup|cg
<consistencygroup>]
    [-synchronous|sync]
    [-syncdirection|syncdir
<syncdirection>]
    [-synchtimeout|syncto]
```

Description

`viprcli snapshot restore` command restores a volume or filesystem snapshot, overwriting the target file system or volume.

Options

name|n

The name of a valid ViPR Controller snapshot. This is mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is mandatory parameter.

project|pr

The name of a valid ViPR Controller project. This is mandatory parameter.

filesystem|fs

The name of a valid filesystem in ViPR Controller. This is an optional parameter.

volume|vol

The name of a valid volume in ViPR Controller. This is an optional parameter.

consistencygroup|cg

The name of a consistency group in ViPR Controller project. This is an optional parameter.

synchronous|sync

Perform the snapshot creation synchronously. If this argument is not specified, the snapshot creation will be performed asynchronously. This is an optional parameter.

syncdirection|syncdir

Specifies the sync direction for snapshot restores. Allowable values are:

- SOURCE_TO_TARGET
- TARGET_TO_SOURCE
- NONE

This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 161

Create a snapshot of a volume with the default value inactivate= FALSE. In this example, the user checks the inactivate state by calling `viprcli snapshot show` command. Once the user is sure the snapshot is activate, the user can restore the volume from the snapshot. An inactive snapshot cannot be restored.

```
# viprcli snapshot create -tn IMG_TENANT
-pr IMG_PROJECT -vol  IMG_VOL  -name SNAP1

# viprcli snapshot show  -tn TENANT -pr IMG_PROJECT
-vol  IMG_VOL -name SNAP1

# viprcli snapshot restore -tn TENANT
-pr IMG_PROJECT -vol  IMG_VOL -name SNAP1
```

Example 162

Create a snapshot of a volume with inactivate= true. In this example, the snapshot is created, then set in an active state with the `viprcli snapshot activate` command. Once the snapshot is activated, the user can restore the volume from the snapshot. An inactive snapshot cannot be restored.

```
# viprcli snapshot create -tn TENANT
-pr IMG_PROJECT -vol  IMG_VOL  -name SNAP1 -inactive true

# viprcli snapshot activate -tn TENANT
-pr IMG_PROJECT -vol  IMG_VOL  -name SNAP1

# viprcli snapshot show  -tn TENANT
-pr IMG_PROJECT -vol  IMG_VOL -name SNAP1

# viprcli snapshot restore -tn TENANT
-pr IMG_PROJECT -vol  IMG_VOL -name SNAP1
```

viprcli snapshot resync

Resynchronize a snapshot and its volume or consistency group for XtremIO arrays.

Syntax

```
viprcli snapshot resyn
viprcli snapshot restore
      [-h]
      [-hostname|hn
 <hostname>]
      [-port|po
 <port_number>]
      [-portui|pu
 <ui_port_number>]
      [-cookiefile|cf
 <cookie_file>]
      -name|n
 <snapshotname>
      [-tenant|tn
 <tenantname>]
      -project|pr
 <project>
      [-volume|vol
 <volumename>]
      [-consistencygroup|cg
 <consistencygroup>]
      [-synchronous|sync]
      [-synctimeout|syncto]
```

Description

`viprcli snapshot resync` updates an existing snapshot with all changes made to a volume or consistency group.

Options

name|n

Name of snapshot. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is mandatory parameter.

volume|vol

The name of a valid volume in ViPR Controller. This is an optional parameter.

consistencygroup|cg

The name of a consistency group in a ViPR Controller project. This is an optional parameter.

synchronous|sync

Perform the snapshot resync synchronously. If this argument is not specified, this command is performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

In this example, `viprcli snapshot resync` is updating snap1 with all changes made to volume1 in proj.

```
viprcli snapshot resync -name snap1 -pr proj -volume volume1
```

viprcli snapshot share-acl

Used to set an access control list (ACL) on a snapshot share.

Syntax

```
        viprcli snapshot share-acl
        [-fsname|fsn
    <filesystem_name>]
        [-sname|sn
    <snapshot_name>]
        [-share|sh
    <share_name>]
        [-operation
    <operation_name>]
        [-project|pr
    <project_name>]
```

```
            [-user <user_name>[
            [-permission
        <permission_name>[
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

The `viprcli snapshot share-acl` command sets an access control list (ACL) on a snapshot share.

Options

fsname|fsn

Name of the filesystem from which the snapshot is taken. This is a mandatory parameter.

sname|sn

Name of snapshot. This is a mandatory parameter.

operation|op

ACL operation: Add, or Delete. This is a mandatory parameter.

project|pr

Name of project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

The following example is used to set an ACL on a snapshot share:

```
viprcli snapshot share-acl -fsname netcproshare1 -sname snap-
netcproshare1 -share snapcifsnetpro1 -operation add -project netpro
-user testfile -permission Read
```

viprcli snapshot show

Show the snapshot details for the given snapshot name. Specify a file system, volume, or consistency group name (but only one at a time).

Syntax

```
            viprcli snapshot show
            -name|n <snapshotname>
            -project|pr <project>
            [-filesystem|fs <filesystemname>]
            [-volume|vol <volumename>]
            [-consistencygroup|cg <consistencygroup>]
            [-tenant|tn <tenantname>
            [-xml]
```

```
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli snapshot show` command returns the details for the given snapshot name. Specify a file system, volume, or consistency group name (but only one at a time).

Options

name|n

The name of a valid ViPR Controller snapshot. This is a mandatory parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

filesystem|fs

The name of a valid filesystem in ViPR Controller. This is an optional parameter.

volume|vol

The name of a valid volume in ViPR Controller. This is an optional parameter.

consistencygroup|cg

The name of a consistency group in ViPR Controller project. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 163

```
# viprcli snapshot show -pr bob_project
-vol my_volume2 -n snap1_volume2
{
   "creation_time": 1364926697177,
   "device_label": "1363807836865740",
   "id": "urn:storageos:BlockSnapshot:
   c43f31dd-dd5d-4b90-bf02-a1f9bd865599:",
   "inactive": false,
   "is_sync_active": true,
   "link": {
      "href": "/block/snapshots/urn:storageos:BlockSnapshot:
      c43f31dd-dd5d-4b90-bf02-a1f9bd865599:",
```

Example 163 (continued)

```
        "rel": "self"
    },
    "name": "snap1_volume2",
    "native_id": "1363807836865740",
    "neighborhood": {
        "id": "urn:storageos:varray:
        202e2d28-6bad-48ea-82f9-977e018c7ae7:",
        "link": {
            "href": "/vdc/varray/urn:storageos:
            varray:202e2d28-6bad-48ea-82f9-977e018c7ae7:",
            "rel": "self"
        }
    },
    "parent": {
        "id": "urn:storageos:Volume:
         bbd31a9a-34d8-44ad-ab97-1cd24a0aac6f:",
        "link": {
            "href": "/block/volumes/urn:storageos:Volume:
             bbd31a9a-34d8-44ad-ab97-1cd24a0aac6f:",
            "rel": "self"
        },
        "name": "snap1_volume2"
    },
    "protocols": [
        "iSCSI"
    ],
    "source_native_id": "1363807836865740",
    "storage_controller": "urn:storageos:StorageSystem:
      d3fa8d0d-209d-440b-805e-1fd87c006073:",
    "wwn": "60000970000195900701533030323532"
}
```

viprcli snapshot show-exports

Shows the details of the exports associated with a specific snapshot. You can specify a subdirectory name or show file system exports for all directories.

Syntax

```
viprcli snapshot show-exports
-name|n <snapshotname>
-project|pr <project>
-filesystem|fs
<filesystemname>
    [-subDir <subdirectory>]
    [-allDir]
    [-tenant|tn <tenantname>
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

`viprcli snapshot show-exports` command returns the export details of the given snapshot name.

Options

name|n

The name of a valid ViPR Controller snapshot. This is a mandatory parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

filesystem|fs

The name of a valid filesystem in ViPR Controller for the snapshot. This is a mandatory parameter.

subDIr

Show the file system exports of the specified sub directory. This is an optional parameter.

allDir

Show the file system exports for all directories. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

```
#
```

viprcli snapshot show-shares

Shows the details of the file share associated with a specific snapshot.

Syntax

```
            viprcli snapshot show-shares
            -name|n <snapshotname>
            -project|pr <project>
            -share|sh <sharename>
            [-tenant|tn <tenantname>
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
```

```
            [-h]
```

Description

`viprcli snapshot show-shares` command returns the details of the given file share for the specified snapshot name.

Options

name|n

The name of a valid ViPR Controller snapshot. This is a mandatory parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

share|sh

The name of a valid file share in ViPR Controller for the snapshot. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

```
#
```

viprcli snapshot tasks

Check the status of asynchronous tasks.

Syntax

```
            viprcli snapshot tasks
            [-name|n <snapshotname>]
            -project|pr <project>
            [-filesystem|fs
        <filesystemname>]
            [-volume|vol
        <volumename>]
            [-consistencygroup|cg
        <consistencygroup>]
            [-tenant|tn <tenantname>
            [-ip <opid>]
            [-verbose|v]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
```

```
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli snapshot tasks` command returns the status of asynchronous snapshot tasks.

Options

project|pr

Name of a valid ViPR Controller project. This is a mandatory parameter.

name|n

Name of a valid ViPR Controller snapshot. This is an optional parameter.

filesystem|fs

Name of a valid filesystem in ViPR Controller. This is an optional parameter.

volume|vol

Name of a valid volume in ViPR Controller. This is an optional parameter.

consistencygroup|cg

Name of a consistencygroup in ViPR Controller project. This is an optional parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

verbose|v

This option provides additional information. This is an optional parameter.

id

Operation ID. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 164

```
 # viprcli snapshot tasks -vol VOLUME_1 -tn VNX_TENANT11 -pr VNX_PROJECT11
Name                                    Capacity            Protocols
1cdb6083-d091-4c3d-b41c-a65a468ad873    VOLUME_SNAPSHOT_1   ready
58366824-fce7-485c-976b-bb2253c9dda1    VOLUME_SNAPSHOT_3   ready
6d785835-fbdc-49a1-b2a9-d8d4fb899f18    VOLUME_SNAPSHOT_2   ready
d9d57dea-4444-4894-bcdb-7bd8fce29dd6    VOLUME_SNAPSHOT_4   ready
```

Example 165

Example 165 (continued)

This example command shows how to retrieve snapshot task information for a given volume with specific snapshot name.

```
viprcli snapshot tasks -vol VOLUME_1 -tn VNX_TENANT11 -pr VNX_PROJECT11 -name
VOLUME_SNAPSHOT_1
Name                                    Capacity            Protocols
1cdb6083-d091-4c3d-b41c-a65a468ad873    VOLUME_SNAPSHOT_1   ready
bdbf82bc-5736-45e7-81d6-8de2fe30013f    VOLUME_SNAPSHOT_1   ready
```

viprcli snapshot tag

Add or delete a tag name for the specified snapshot.

Syntax

```
    viprcli snapshot tag
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n <snapshotname>
    [-tenant|tn
<tenantname>]
    -project|pr <project>
    [-filesystem|fs <filesystemname>
| -volume|vol <volumename> | -consistencygroup|cg
<consistencygroup>]
    [-add <addtags>
[<addtags...>]]
    [-remove <removetags>
[<removetags...>]]
```

Description

`viprcli snapshot tag` command add or deletes a tag name for the specified snapshot.

Options

name|n

Name of a valid ViPR Controller snapshot. This is a mandatory parameter.

tenant|tn

A valid ViPR Controller tenant name. This is an optional parameter.

project|pr

Name of a valid ViPR Controller project. This is a mandatory parameter.

filesystem|fs

Name of a valid filesystem in ViPR Controller. This is an optional parameter.

volume|vol

Name of a valid volume in ViPR Controller. This is an optional parameter.

consistencygroup|cg

Name of a consistencygroup in ViPR Controller project. This is an optional parameter.

add

Add one or more tag names to the ViPR Controller snapshot name. This is an optional parameter. You can either use add or remove and not both.

remove

Delete one or more tag names to the ViPR Controller snapshot name. This is an optional parameter. You can either use add or remove and not both.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli snapshot unexport-file

Unexport a file system snapshot.

Syntax

```
        viprcli snapshot unexport-file
        [-h]
        [-hostname|hn
    <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        -name|n <snapshotname>
        -filesystem|fs
    <filesystemname>
        [-tenant|tn <tenantname>
        -project|pr <project>
        -protocol|pl {NFS | CIFS}
        [-share|sh <sharename>]
        [-synchronous|sync]
        [-synctimeout|syncto]
```

Description

`viprcli snapshot unexport-file` command unexports a file system snapshot.

Note

You can only unexport a file system snapshots of VNX File arrays using this command. Unexporting an Isilon file system snapshot using this command shows an error message. This is because, a file system snapshot is automatically unexported by default when you export a file system of an Isilon array using the `viprcli filesystem unexport` command.

Options

name|n

The name of a valid ViPR Controller snapshot. This is mandatory parameter.

filesystem|fs

The name of a valid filesystem in ViPR Controller. This is mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is mandatory parameter.

protocol|pl

File protocol - one of NFS , NFSv4 or CIFS. This is mandatory parameter.

share|sh

The name of a Windows share. This parameter should be used for CIFS exports only. This is an optional parameter.

synchronous|sync

Synchronous filesystem export. By default, CLI commands are asynchronous. The -synchronous option waits until the fileshare snapshot export has been successfully completed before returning control to the shell. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 166

```
# viprcli snapshot unexport-file -n TEST_SNAPSHOT1
-pr TEST_PROJECT11 -fs TEST_ISILON_FILESHARE11 -pl NFS -ru root -pe
ro -t sys
```

viprcli snapshot expand

Expand the size of a snapshot.

Syntax

```
            viprcli snapshot expand
            [-h]
            [-hostname|hn
        <hostname>]
```

```
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n <snapshotname>
            [-tenant|tn
        <tenantname>
            -project|pr <projectname>
            -size|size<snapshotsize[kKmMgGtT]>
            [-synchronous|sync]
            [-synctimeout|syncto]
            [-volume|vol <volumename>
            -consistencygroup|cg <consistencygroup>]
```

Description

`viprcli snapshot expand` command expands the size of a snapshot. This command applicable for XtremIO snapshots only.

Note

Expand block snapshot is supported only for XtremeIO. You can only expand the snapshot size equal to the source volume size.

Options

name|n

The name of a valid ViPR Controller snapshot. This is mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

projectname|pr

Name of the project. This is a mandatory parameter.

size|s

New size of blocksnapshot:{number}[unit]. A size suffix for K for kilobytes, M for megabytes, G for gigabytes, T for terabytes is optional. Default unit is bytes.

synchronous|sync

Synchronous block snapshot expansion operation. By default, CLI commands are asynchronous. The synchronous option waits until the block snapshot expansion operation is completes successfully before returning control to the shell. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

volumename|vol

Name of the volume.This is an optional paramater.

consistencygroup|cg

Name of the consistency group. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 167

Expand non-CG snapshot

```
# viprcli snapshot expand  -name XIO_snap_expand_2
-project COP-32230 -size 2GB -vol
Test_XIO_Expand_vol1
```

Example 168

Expand CG snapshot

```
# viprcli snapshot expand  -name Snap1_XIO_RO-1
-project COP-32230 -size 2GB
-consistencygroup COP -32230
```

CHAPTER 38

viprcli snapshotsession commands

This chapter contains the following topics:

- viprcli snapshotsession create 410
- viprcli snapshotsession deactivate 411
- viprcli snapshotsession linktarget 412
- viprcli snapshotsession list 413
- viprcli snapshotsession relinktargets 414
- viprcli snapshotsession restore 415
- viprcli snapshotsession show 416
- viprcli snapshotsession unlinktargets 418

viprcli snapshotsession create

Creates a snapshot session.

Syntax

```
viprcli snapshotsession create
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n <name>
    [-tenant|tn <tenantname>
    -project|pr
<projectname>
    [-count|ct]
    [-targetname|tgn
<target_name>]
    [-copymode|cm
<copy,nocopy>]
    [-volume|vol
<volumename>]
    [-consistencygroup|cg
<consistencygroup>]
```

Description

The `viprcli snapshotsession create` command creates a snapshot session.

Options

name|n

Name of a snapshot session. This is a mandatory parameter.

tenant|tn

Name of a tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

count|ct

Number of target volumes. This is an optional parameter.

targetname|tgn

The target name. This is an optional parameter.

copymode|cm

Specifies whether to create in copy or nocopy mode. This is an optional parameter.

volume|vol

Name of a volume. This is an optional parameter.

consistencygroup|cg

Name of a consistency group. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 169

```
# viprcli snapshotsession create -n snapshotVOLNeww3 -pr Project1 -
vol VolumeNew-1 -target_name tgh1 -ct 2 -cm nocopy
```

viprcli snapshotsession deactivate

Deactivates a snapshot session.

Syntax

```
            viprcli snapshotsession deactivate
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n <name>
            [-tenant|tn <tenantname>
            -project|pr
        <projectname>
            [-volume|vol
        <volumename>]
            [-consistencygroup|cg
        <consistencygroup>]
```

Description

The `viprcli snapshotsession deactivate` command deactivates a snapshot session.

Options

name|n

Name of a snapshot session. This is a mandatory parameter.

tenant|tn

Name of a tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

volume|vol

Name of a volume. This is an optional parameter.

consistencygroup|cg

The name of a consistency group. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 170

```
# viprcli snapshotsession deactivate -n snapshotVOLNew1 -pr
Project1 -vol VolumeNew-1
```

viprcli snapshotsession linktarget

Links a target to a snapshot session.

Syntax

```
    viprcli snapshotsession linktarget
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n <name>
    [-tenant|tn <tenantname>
    -project|pr
<projectname>
    -count|ct
    -targetname|tgn
<target_name>
    -copymode|cm
<copy,nocopy>
    [-volume|vol
<volumename>]
    [-consistencygroup|cg
<consistencygroup>]
```

Description

The `viprcli snapshotsession linktarget` command links a target to a snapshot session.

Options

name|n

The name of a snapshot session. This is a mandatory parameter.

tenant|tn

The name of a tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

count|ct

Number of target volumes. This is a mandatory parameter.

targetname|tgn

The target name. This is a mandatory parameter.

copymode|cm

Specifies whether to create in copy or nocopy mode. This is a mandatory parameter.

volume|vol

Name of a volume. This is an optional parameter.

consistencygroup|cg

The name of a consistency group. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 171

```
# viprcli snapshotsession linktarget -n snapshot3 -pr Project1 -vol
VolumeNew-2 -ct 1 -target_name linkingtarget1 -cm copy
```

viprcli snapshotsession list

Lists snapshot sessions.

Syntax

```
            viprcli snapshotsession list
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-tenant|tn <tenantname>
            -project|pr
        <projectname>
            [-volume|vol
        <volumename>]
            [-consistencygroup|cg
        <consistencygroup>]
```

Description

The `viprcli snapshotsession list` command lists snapshot sessions.

Options

tenant|tn

Name of tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

volume|vol

Name of a volume. This is an optional parameter.

consistencygroup|cg

Name of a consistency group. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 172

```
# viprcli snapshotsession list -pr Project1 -volume VolumeNew-1
o/p:
NAME
snapshotVOLNew1
snapshotVOLNeww2
snapshotVOLNew2
snapshotVOLNeww
snapshotVOLNeww3
```

viprcli snapshotsession relinktargets

Relinks a target to a snapshot session.

Syntax

```
        viprcli snapshotsession relinktargets
        [-h]
        [-hostname <hostname>]
        [-port|po
    <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        -name|n <name>
        [-tenant|tn <tenantname>
        -project|pr
    <projectname>
        -volume|vol <volumename>
        -targetvolumes|tgnames TARGET_NAMES
    [TARGET_NAMES ...]
```

Description

The `viprcli snapshotsession relinktargets` command relinks a target to a snapshot session.

Options

name|n

The name of a snapshot session. This is a mandatory parameter.

tenant|tn

The name of a tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

volume|vol

Name of a volume. This is a mandatory parameter.

targetvolumes|tgnames

List of target volumes. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 173

```
# viprcli snapshotsession relinktargets -n snapshot3 -pr Project1 -
vol VolumeNew-2 -id urn:storageos:BlockSnapshot:4515f753-
b1b2-4444-8176-bb002f058bc3:vdc1
```

viprcli snapshotsession restore

Restores a snapshot session.

Syntax

```
            viprcli snapshotsession restore
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n <name>
            [-tenant|tn <tenantname>
            -project|pr
        <projectname>
            [-volume|vol
        <volumename>]
            [-consistencygroup|cg
```

```
        <consistencygroup>]
```

Description

The `viprcli snapshotsession restore` command restores a snapshot session.

Options

name|n

Name of a snapshot session. This is a mandatory parameter.

tenant|tn

Name of a tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

volume|vol

Name of a volume. This is an optional parameter.

consistencygroup|cg

Name of a consistency group. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 174

```
# viprcli snapshotsession restore -n snapshotVOLNeww2 -pr Project1 -
vol VolumeNew-1
```

viprcli snapshotsession show

Shows a snapshot session.

Syntax

```
            viprcli snapshotsession show
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|n <name>
            [-tenant|tn <tenantname>
            -project|pr
        <projectname>
            [-volume|vol
```

```
        <volumename>]
            [-consistencygroup|cg
        <consistencygroup>]
```

Description

The `viprcli snapshotsession show` command shows a snapshot session.

Options

name|n

Name of a snapshot session. This is a mandatory parameter.

tenant|tn

Name of a tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

volume|vol

Name of a volume. This is an optional parameter.

consistencygroup|cg

Name of a consistency group. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 175

```
# viprcli snapshotsession show -n SNAPSess2 -pr Project1 -cg cg2
{
"consistency_group": {
"id": "urn:storageos:BlockConsistencyGroup:
47e7d1c4-44f5-462b-8b5b-418f8278e313:vdc1",
"link": {
"href": "/block/consistency-groups/
urn:storageos:BlockConsistencyGroup:
47e7d1c4-44f5-462b-8b5b-418f8278e313:vdc1",
"rel": "self"
}
},
"creation_time": 1453096392617,
"global": false,
"id": "urn:storageos:BlockSnapshotSession:7013c78a-bb61-4c3a-
a1b4-6f616098624a:vdc1",
"inactive": false,
"internal": false,
"link": {
"href": "/block/snapshot-sessions/
urn:storageos:BlockSnapshotSession:7013c78a-bb61-4c3a-
a1b4-6f616098624a:vdc1",
"rel": "self"
},
"linked_targets": [
{
"id": "urn:storageos:BlockSnapshot:f376c2e2-
cf84-48c2-87bb-8c9211e6015a:vdc1",
```

Example 175 (continued)

```
"link": {
"href": "/block/snapshots/urn:storageos:BlockSnapshot:f376c2e2-
cf84-48c2-87bb-8c9211e6015a:vdc1",
"rel": "self"
}
},
{
"id": "urn:storageos:BlockSnapshot:34124da5-
b32f-4c49-967e-326d8789c65e:vdc1",
"link": {
"href": "/block/snapshots/urn:storageos:BlockSnapshot:34124da5-
b32f-4c49-967e-326d8789c65e:vdc1",
"rel": "self"
}
}
],
"name": "SNAPSess2",
"project": {
"id": "urn:storageos:Project:
6b0cc96f-3bd1-48df-8c48-30a0c61c676b:global",
"link": {
"href": "/projects/urn:storageos:Project:
6b0cc96f-3bd1-48df-8c48-30a0c61c676b:global",
"rel": "self"
}
},
"protocols": [],
"remote": false,
"session_label": "SNAPSess2",
"tags": [],
"vdc": {
"id": "urn:storageos:VirtualDataCenter:5f9ed4da-
b7f1-4488-9078-72368efc9923:vdc1",
"link": {
"href": "/vdc/urn:storageos:VirtualDataCenter:5f9ed4da-
b7f1-4488-9078-72368efc9923:vdc1",
"rel": "self"
}
}
}
DEACTIVATE :
```

viprcli snapshotsession unlinktargets

Unlinks a target from a snapshot session.

Syntax

```
            viprcli snapshotsession unlinktargets
[-h]
[-hostname <hostname>]
[-port|po        <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-name|n <name>]
[-tenant|tn <tenantname>]
[-project|pr        <projectname>]
[-targetvolumes        <target_name>, <target_name>:delete]
[-volume|vol        <volumename>]
```

```
[-consistencygroup|cg        <consistencygroup>]
```

Description

The `viprcli snapshotsession unlinktargets` command unlinks a target from a snapshot session.

Options

name|n

The name of a snapshot session. This is a mandatory parameter.

tenant|tn

The name of a tenant. This is an optional parameter.

project|pr

Name of project. This is a mandatory parameter.

targetvolumes

List the target volumes to unlink using the following format:
`-targetvolumes <target_name> <target_name>`
Or to unlink and delete the target volume using the following format.
`-targetvolumes <target_name>:delete`(See examples below)
This is a mandatory parameter.

volume|vol

Name of a volume. This is an optional parameter.

consistencygroup|cg

The name of a consistency group. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 176 Unlink the target volume

```
viprcli snapshotsession unlinktargets -name volume1 -volume VOL1 -
project Project1 -targetvolumes TG1
```

Example 177 Unlink and delete the target volume

```
viprcli snapshotsession unlinktargets -name volume1 -volume VOL1 -
project Project1 -targetvolumes TG1:delete TG2
```

CHAPTER 39

viprcli storagepool commands

This chapter contains the following topics:

- viprcli storagepool delete 422
- viprcli storagepool deregister 423
- viprcli storagepool list 424
- viprcli storagepool register 425
- viprcli storagepool show 426
- viprcli storagepool update 428

viprcli storagepool delete

Deletes a storage pool.

Syntax

```
            viprcli storagepool delete
            -name <poolname>

            -type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleio | xtremio | ddmc | ibmxiv | vnxe | datadomain |
ecs}}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            [-serialnumber|sn
        <serialnumber>]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storagepool delete` command deletes a specified storage pool.

Options

name

The storage pool name. This is a mandatory parameter.

type|t

Storage system type. One of the following: isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, datadomain, or ecs. Specify either a storage system name or serial number for the type and not both. This is a mandatory parameter.

storagesystem|ss

The storage system where the target storage pool resides. Specify either a storage system or serial number for the type and not both.

serialnumber|sn

The serial number of the storage system. Specify either a storage system or serial number for the type and not both.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli storagepool deregister

Remove the registered storage pool. An unregistered storage pool cannot be used for storage within ViPR Controller.

Syntax

```
            viprcli storagepool deregister
            -name|n <poolname>

            -type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleio | xtremio | ddmc | ibmxiv | vnxe | ecs |
datadomain}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storagepool deregister` command removes the registered storage pool. An unregistered storage pool cannot be used for storage within ViPR Controller. This command is silent on success.

Options

name|n

The name of the storage pool you want to unregister. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is a mandatory parameter.

storagesystem|ss |serialnumber|sn

The storage system name or serial number on which the target storage pool resides. Use either a storage system name or serial number and not both.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 178

Example 178 (continued)

```
# viprcli storagepool deregister -name 'First Pool' -type vnxblock -
sn APM00121202922
```

viprcli storagepool list

Return a list of all storage pools for a specified storage system.

Syntax

```
            viprcli storagepool list

            -type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleio | xtremio | ddmc | ibmxiv | vnxe | ecs |
datadomain}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            [{-verbose|v | -long|l}]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storagepool list` command returns a list of all storage pools for a specified storage system.

Options

type|t

Storage system type. One of the following: isilon, vnxblock, vnxfile, vmax, netapp, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, netappc, vnxe, ecs, or datadomain. Specify either a storage system name or serial number for the type and not both. This is a mandatory parameter.

storagesystem|ss

The storage system where the target storage pool resides. Specify either a storage system or serial number for the type and not both.

serialnumber|sn

The serial number of the storage system. Specify either a storage system or serial number for the type and not both.

{verbose|v | long|l}

These options offer additional information for this operation. Use either verbose or long and not both.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 179

```
viprcli storagepool list -ss VNXFILE1
"urn:storageos:StoragePool:cd4e18a0-578f-4952-b3aa-f44b1cb6234d:"
"urn:storageos:StoragePool:d6dd22f3-9ccc-4cee-a053-20e7ba5d3083:"
```

viprcli storagepool register

Register a storage pool with ViPR Controller.

Syntax

```
            viprcli storagepool register

            -type {isilon | vnxblock | vnxfile |
        vmax | netapp | netappc |vplex | hds | openstack | scaleio
| xtremio | ddmc |
        ibmxiv | vnxe | ecs | datadomain}
            {-serialnumber|sn
        <serialnumber>}
            [-name|n <poolname>]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storagepool register` command registers only those storage pools which were manually unregistered by the ViPR Controller administrator. Storage pools are registered by default when they are discovered in ViPR Controller. This command is silent on success.

Options

type|t

Storage system type. One of the following: isilon, vnxblock, vnxfile, vmax, netapp, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, netappc, vnxe, ecs, or datadomain. Specify either a storage system name or serial number for the type and not both. This is a mandatory parameter.

serialnumber|sn

The serial number of the storage system. This is a mandatory parameter.

name|n

The name of the storage pool you want to register. If this parameter is omitted, all pools on the array specified by the serial number argument are registered. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 180

```
# viprcli storagepool register -type vnxfile
          -sn APM00112803071 -name 'Pool 0'
```

Use the list command to view the result.

```
# viprcli storagepool list -sn APM00112803071 -type vnxfile

POOL_NAME         REGISTRATION STATUS    FREE CAPACITY  TOTAL
CAPACITY
Pool 0            REGISTERED
clarsas_archive   REGISTERED
```

viprcli storagepool show

Return a detailed listing for a particular storage pool.

Syntax

```
            viprcli storagepool show
            -name|n <storagepoolname>

            type|t {isilon | vnxblock | vnxfile |
        vmax | netapp | netappc |vplex | hds | openstack | scaleio
| xtremio | ddmc |
        ibmxiv | vnxe | ecs | datadomain}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            [-xml]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storagepool show` command returns the detailed information which includes the size and the percentage of the pool that is in use.

Options

name|n

The name of the storage pool. This is a mandatory parameter.

type|t

Storage system type. One of the following: isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. Specify either a storage system name or serial number for the type and not both. This is a mandatory parameter.

storagesystem|ss

The storage system where the target storage pool resides. Specify either a storage system or serial number for the type and not both.

serialnumber|sn

The serial number of the storage system. Specify either a storage system or serial number for the type and not both.

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 181

```
# viprcli storagepool show -n 'Pool 0' -t vnxblock -ss  CLARIION
+APM00112803075
{
   "vpool_set": [
      "urn:vipr:VirtualPool:476f9335-b646-47cf-8d6b-5c9cb16fa96e:"
   ],
   "free_gb": 37,
   "id": "urn:vipr:StoragePool:61ddd6b0-5495-41ae-a786-
d0c15db8569d:",
   "inactive": false,
   "job_discovery_status": "CREATED",
   "job_metering_status": "CREATED",
   "last_discovery_run_time": 0,
   "last_metering_run_time": 0,
   "link": {
      "href": "/vdc/storage-systems/
               urn:vipr:StorageSystem:
5e83cc32-7071-4045-9828-8de480b47c2b:/
               storage-pools/
               urn:vipr:StoragePool:61ddd6b0-5495-41ae-a786-
d0c15db8569d:",
      "rel": "self"
   },
   "maximum_thick_volume_size": 34603008,
   "maximum_thin_volume_size": 17179869184,
   "minimum_thick_volume_size": 1,
   "minimum_thin_volume_size": 1,
   "name": "CLARIION+APM00112803075+POOL+U+Pool 0",
   "native_guid": "CLARIION+APM00112803075+POOL+U+Pool 0",
   "next_discovery_run_time": 0,
   "next_metering_run_time": 0,
   "operational_status": "READY",
```

Example 181 (continued)

```
   "percent_subscribed": 106,
   "percent_used": 90,
   "pool_name": "Pool 0",
   "protocols": [
      "FC",
      "ISCSI"
   ],
   "registration_status": "REGISTERED",
   "storage_system": {
      "id": "urn:vipr:StorageSystem:
5e83cc32-7071-4045-9828-8de480b47c2b:",
      "link": {
         "href": "/vdc/storage-systems/
                  urn:vipr:StorageSystem:
5e83cc32-7071-4045-9828-8de480b47c2b:",
         "rel": "self"
      }
   },
   "subscribed_gb": 381,
   "supported_volume_types": "THIN_AND_THICK",
   "usable_gb": 360,
   "used_gb": 323
}
```

viprcli storagepool update

Modify the parameters of a storage pool.

Syntax

```
            viprcli storagepool update
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -type

{isilon,vnxblock,vnxfile,vmax,netapp,vplex,hds,openstack,scaleio,xtr
emio,ddmc,ibmxiv,netappc,vnxe,ecs,datadomain}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            [-name|n
        <storagepoolname>]
            [-vaadd <varray>
        [<varray_add>...]]
            [-varem <varray>
        [<varray_remove>...]]
            [-maxresources|mr
        <maxresources>]
            [-maxpoolutilization|mpu
        <maxpoolutilization>]
            [-maxthinpoolsubscription|mtps
        <maxthinpoolsubscription>]
            [-volumetype
```

```
        <volumetype>]
```

Description

`viprcli storagepool update` command allows you to associate the storage pool to one or more virtual arrays and remove the pool from one or more virtual arrays. This command is silent on success.

Options

type|t

Storage system type. One of the following: isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. Specify either a storage system name or serial number for the specified type and not both. This is a mandatory parameter.

storagesystem|ss

The storage system where the target storage pool resides. Specify either a storage system or serial number for the specified type and not both. This is an optional parameter.

serialnumber|sn

The serial number of the storage system. Specify either a storage system or serial number for the specified type and not both. This is an optional parameter.

name|n

The name of the storage pool. This is an optional parameter.

vaadd

Add the storage pool to a virtual array. This virtual array will subsequently appear under TAGGED_VARRAYS when you call the viprcli storageport list -l operation. This is an optional parameter.

varem

Remove the storage pool from a virtual array. The specified virtual array will be removed from the TAGGED_VARRAYS list when you call the operation viprcli storageport list -l. This is an optional parameter.

maxresources|mr

The maximum number of resources in the storage system. This is an optional parameter.

maxpoolutilization|mpu

The maximum pool utilization in the storage system. This is an optional parameter.

maxthinpoolsubcription|mtps

The maximum number of thin pool subscription. This is an optional parameter.

volumetype

Type of volume. One of the following values: THIN_AND_THICK, THICK_ONLY, THIN_ONLY. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

```
# viprcli storagepool update -type vnxfile
          -name 'Pool 0' -sn APM00112803071

# viprcli storagepool list -sn APM00112803071
-type vnxfile -l
POOL_NAME       REGISTRATION_STATUS FREE TOTAL VPOOL SET   TAGGED
VARRAYS
Pool 0          REGISTERED                     vnxfile-thick-raid10-
vpool
clarsas_archive REGISTERED
```

CHAPTER 40

viprcli storageport commands

This chapter contains the following topics:

- viprcli storageport create 432
- viprcli storageport delete 433
- viprcli storageport deregister 434
- viprcli storageport list 435
- viprcli storageport register 436
- viprcli storageport show 438
- viprcli storageport update 440

viprcli storageport create

Creates a storage port.

Syntax

```
        viprcli storageport create
        -st|storagetype {openstack}
        -portname|pn PORTNAME
        -pid|portid PORTID
        -transporttype|tt
    <transporttype>
        {-storagesystem|ss
    <storagesystemname> | -serialnumber|sn <serialnumber>}
        [-serialnumber|sn
    <serialnumber>]
        [-hostname|hn
    <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        [-h]
```

Description

`viprcli storageport create` command creates a storage port.

Options

st|storagetype

The type of the storage system. The only valid value is `openstack`. This is a mandatory parameter.

portname|pn

The name of the storage port. This is a mandatory parameter.

pid|portid

The storage port network ID. This is a mandatory parameter.

transporttype|tt

The transport protocol type: FC | IP | Ethernet. This is a mandatory parameter.

storagesystem|ss

The storage system where the target storage pool resides. Specify either a storage system or serial number for the type and not both.

serialnumber|sn

The serial number of the storage system. Specify either a storage system or serial number for the type and not both.

Common Arguments

This operation also takes the Common Parameters on page 24.

Create a storage port using the

ViPR Controller
CLI

1. List the storage systems using `viprcli storagesystem list`. Note the last three digits of the serial number of the storage system to which the storage port needs to be added.
2. Create the storage ports using
   `viprcli storageport create -portname|pn` *`portname`* `-pid` *`wwn_of_the_port`* `-transporttype|tt` *`transport_type`* `-systemtype|st` *`storage_type`* `-serialnumber|sn` *`serialnumber`*
3. List the storage ports for the storage system to ensure that the new storage port created successfully using
   `viprcli storageport list -sn` *`serialnumber`* `-t` *`storage_type`*

viprcli storageport delete

Deletes a storage port.

Syntax

```
            viprcli storageport delete
            -name|n <storageportname>
            -type|t <storagesystemtype>
            {-storagesystem|ss <storagesystemname> | -serialnumber|
sn <serialnumber>}
            [-serialnumber|sn <serialnumber>]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli storageport delete` command deletes a specified storage port.

Options

name

The storage port name. This is a mandatory parameter.

type|t

The type of the storage system. This is a mandatory parameter.

storagesystem|ss

The storage system where the target storage pool resides. Specify either a storage system or serial number for the type and not both.

serialnumber|sn

The serial number of the storage system. Specify either a storage system or serial number for the type and not both.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli storageport deregister

Unregister a storage port.

Syntax

```
            viprcli storageport deregister
            -name|n
        <storageportname>
            -type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleio | xtremio | ddmc | ibmxiv | vnxe | ecs |
datadomain}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storageport deregister` command unregisters a storage port. Storage ports are registered automatically when they are discovered. This command is silent on success.

Options

name|n

The storage port name to be registered. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is a mandatory parameter.

{storagesystem| ss | serialnumber|sn}

The storage system name or serial number on which the storage pool resides. One of these parameters is required.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 182

Example 182 (continued)

```
# viprcli storageport deregister -t isilon -n ViPRPort
-sn 6805ca00acee0d63a6508020ba69b3849f37
```

Use the list command to view results.

```
# viprcli storageport list -t isilon
-sn 805ca00acee0d63a6508020ba69b3849f37
PORT_NAME TRANSPORT_TYPE NETWORK_NAME PORT_NETWORK_ID
REGISTRATION_STATUS
ViPRPort  IP             ipzone       ViPRPort        UNREGISTERED
```

viprcli storageport list

Return the list of all storage ports for a specified storage system.

Syntax

```
            viprcli storageport list
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            -type|t {isilon, vnxblock, vnxfile,
        vmax}
            [-metrics|m]
            [{-verbose|v | -long|l}]
```

Description

`viprcli storageport list` command displays information about storage ports associated with a particular storage system. Storage ports are discovered - you cannot create a storage port object with the ViPR Controller command line interface.

Options

storagesystem|ss

The storage system where the target storage pool resides. Either this parameter or the serial number parameter are required. You do not need both storage system and serial number.

serialnumber|sn

The serial number of the storage system. Either this parameter or the storage system parameter are required.

type|t

Storage system type. One of the following: isilon, vnxblock, vnxfile, vmax.

verbose|v

List storage port with details. This is an optional parameter.

metrics|m

List storage port metrics. This is an optional parameter.

long|l

List storage port in table with details. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 183

```
# viprcli storageport list -type vnxblock
-ss CLARIION+APM12310359509

PORT    TRANSPORT NETWORK      PORT NETWORK
REGISTRATION
NAME    TYPE      NAME         ID                        STATUS

SP_A:0  FC        VSAN_1337    50:00:09:82:FD:64:8F:59   REGISTERED
SP_A:1  FC        VSAN_1337    50:00:09:82:8C:4D:4B:B3   REGISTERED
SP_A:3  FC        EMP_network  50:00:09:82:A4:5F:FC:23   REGISTERED
SP_A:3  FC                     50:00:09:82:AC:B3:06:B6   REGISTERED
SP_B:0  FC        VSAN_1337    50:00:09:82:98:72:33:CB   REGISTERED
SP_B:1  FC        VSAN_1337    50:00:09:82:B5:67:16:C7   REGISTERED
```

viprcli storageport register

Register a storage port with ViPR Controller.

Syntax

```
            viprcli storageport register
            -transporttype|tt {FC | Ethernet | IP
        | ScaleIO}

            -type|t {isilon | vnxblock | vnxfile |
        vmax | netapp | netappc |vplex | hds | openstack | scaleio
| xtremio | ddmc |
        ibmxiv | vnxe | ecs | datadomain}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|Sn <serialnumber>}
            [-name|n
        <storageportname>]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
```

```
              [-h]
```

Description

`viprcli storageport register` command registers a storage port that was de-registered manually by a ViPR Controller administrator. Storage Ports are registered automatically when they are discovered. This command is silent on success.

Options

transporttype|tt

One of FC , Ethernet , ScaleIO, or IP. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is a mandatory parameter.

{storagesystem|ss |serialnumber|sn}

The storage system name or serial number on which the storage pool resides. One of these parameters is required.

name|n

The storage port name to be registered. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 184

```
# viprcli storageport register -t isilon -tt IP
-sn 6805ca00acee0d63a6508020ba69b3849f37
```

Use the list command to view results.

```
# viprcli storageport list -t isilon
-sn 6805ca00acee0d63a6508020ba69b3849f37
PORT_NAME TRANSPORT_TYPE NETWORK_NAME PORT_NETWORK_ID
REGISTRATION_STATUS
ViPRPort  IP             ipzone       ViPRPort        REGISTERED
```

viprcli storageport show

Return detailed listing for a particular storage port.

Syntax

```
            viprcli storageport show
            -name|n
        <storageportname>

            -type|t {isilon | vnxblock | vnxfile |
        vmax | netapp | netappc |vplex | hds | openstack | scaleio
| xtremio | ddmc |
        ibmxiv | vnxe | datadomain}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            [-xml]
            [-hostname|hn
        <hostname>]
            [-port}po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storageport show` command displays an extended list of information about a particular storage port. By default, the output is in JSON format, but you can specify the -xml switch to display the information in XML format.

Options

name|n

The world wide name (WWN) of a storage port. This is a mandatory parameter.

type|t

Storage system type. One of the following: isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, or datadomain. This is a mandatory parameter.

storagesystem|ss

The storage system where the target storage pool resides. Either this parameter or the serialnumber parameter are required. You do not need both storagesystem and serialnumber.

serialnumber|sn

The serial number of the storage system. Either this parameter or the storagesystem parameter are required.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 185

```
 # viprcli storageport show -n ViPRPort -t isilon
-sn 805ca00acee0d63a6508020ba69b3849f37
{
   "creation_time": 1379849239457,
   "id": "urn:storageos:StoragePort:e2c67f63-c3b0-4935-9f61-
f98745c29765:",
   "inactive": false,
   "link": {
      "href": "/vdc/storage-systems/urn:storageos:StorageSystem:
       fddb1447-0915-4ae9-a259-09d4927fc794:/storage-ports/urn:
       storageos:StoragePort:e2c67f63-c3b0-4935-9f61-f98745c29765:",
      "rel": "self"
   },
   "name": "lglw6111",
   "native_guid": "ISILON+6805ca00acee0d63a6508020ba69b3849f37+PORT
+
    ViPRPort",
   "network": {
      "id": "urn:storageos:Network:8f70bb3f-6e76-4388-
a8fa-9e061e0ff47e:",
      "link": {
         "href": "/vdc/networks/urn:storageos:Network:
          8f70bb3f-6e76-4388-a8fa-9e061e0ff47e:",
         "rel": "self"
      }
   },
   "operational_status": "UNKNOWN",
   "port_group": "ViPRPort",
   "port_name": "ViPRPort",
   "port_network_id": "ViPRPort",
   "port_type": "frontend",
   "registration_status": "REGISTERED",
   "storage_system": {
      "id": "urn:storageos:StorageSystem:
      fddb1447-0915-4ae9-a259-09d4927fc794:",
      "link": {
         "href": "/vdc/storage-systems/urn:storageos:StorageSystem:
         fddb1447-0915-4ae9-a259-09d4927fc794:",
         "rel": "self"
      }
   },
   "tags": [],
   "transport_type": "IP"
}
```

Example 186

```
# viprcli storageport show -n  SP_A:0 -t vnxblock
-sn APM00121301070
{
   "creation_time": 1379686724356,
   "id": "urn:storageos:StoragePort:
400f4741-91f9-4d98-897a-0aced0795bf8:",
   "inactive": false,
   "link": {
      "href": "/vdc/storage-systems/urn:storageos:StorageSystem:
```

Example 186 (continued)

```
      9726ee49-c706-496f-8be0-6851a1fb14e2:/storage-ports/urn:
      storageos:StoragePort:400f4741-91f9-4d98-897a-0aced0795bf8:",
      "rel": "self"
   },
   "name": "CLARIION+APM00121301070+PORT+50:06:01:60:47:20:25:F6",
   "native_guid": "CLARIION+APM00121301070+PORT+
   50:06:01:60:47:20:25:F6",
   "network": {
      "id": "urn:storageos:Network:3f0b780d-43fc-4acf-a05f-
d0ed08e06d9e:",
      "link": {
         "href": "/vdc/networks/urn:storageos:Network:
         3f0b780d-43fc-4acf-a05f-d0ed08e06d9e:",
         "rel": "self"
      }
   },
   "operational_status": "OK",
   "port_group": "SP_A",
   "port_name": "SP_A:0",
   "port_network_id": "50:06:01:60:47:20:25:F6",
   "port_speed_gbps": 4,
   "port_type": "frontend",
   "registration_status": "REGISTERED",
   "storage_system": {
      "id": "urn:storageos:StorageSystem:
      9726ee49-c706-496f-8be0-6851a1fb14e2:",
      "link": {
         "href": "/vdc/storage-systems/urn:storageos:StorageSystem:
         9726ee49-c706-496f-8be0-6851a1fb14e2:",
         "rel": "self"
      }
   },
   "tags": [],
   "transport_type": "FC"
}
```

viprcli storageport update

Update the storage system for a registered storage port.

Syntax

```
            viprcli storageport update
            -transporttype|tt {FC | Ethernet | IP |
        ScaleIO}

            -type|t {isilon | vnxblock | vnxfile |
        vmax | netapp | netappc |vplex | hds | openstack | scaleio
| xtremio | ddmc |
        ibmxiv | vnxe | ecs | datadomain}
            [-varray_add|vaadd
        <varray_add>[
            [-varray_remove|varemove
        <varray_remove>[
            [-network|nw <network>[
            [{-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}]
            [-portname|pn
        <storageportname>]
```

```
            [-port_network_id|pnwid
        <port_networkd_id>]
            [-group|g <portgroup>]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storageport update` command does not work with storage ports that are discovered. This command is silent on success.

Options

transporttype|tt

One of FC , Ethernet , IP, or ScaleIO. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is a mandatory parameter.

varray_add|vaadd

List of a virtual arrays to be associated to the storage port. This is an optional parameter.

varray_remove|varemove

List of a virtual arrays to be dissociated from the storage port. This is an optional parameter.

network|nw

The name of a network to which this port is physically connected. This is a mandatory parameter.

{storagesystem|ss | serialnumber|sn}

The storage system name or serial number on which the storage pool resides. Either one of these parameters is required.

portname|pn

The storage port name to be updated. This is an optional parameter.

port_network_id/pnwid

The new storage port network identifier (WWPN). This is an optional parameter.

group|g

The group to be processed. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 187

Example 187 (continued)

```
# viprcli storageport update -t isilon -nw ipzone -va UK
-tt IP -sn 6805ca00acee0d63a6508020ba69b3849f37
```

CHAPTER 41

viprcli storageportgroup commands

This chapter contains the following topics:

- viprcli storageportgroup create 444
- viprcli storageportgroup delete 445
- viprcli storageportgroup deregister 446
- viprcli storageportgroup list 447
- viprcli storageportgroup register 448
- viprcli storageportgroup show 449

viprcli storageportgroup create

Creates a storage port.

Syntax

```
            viprcli storageportgroup create
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
{vmax|-t PORTGROUPNAME|-n STORAGEPORTS|-sp <STORAGEPORTS...>}

            [-serialnumber|sn
        <serialnumber>]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storageportgroup create` command adds a storage port group to storagesystem.

Options

port|port_number

The port number of the ViPR. This is a mandatory parameter.

portui|ui_port_number|po

https port number of ViPR Portal UI. This is a mandatory parameter.

cookiefile|cf

Full name of the cookiefile

storagesystem|ss

Name of the storage system or storage sytem where the port group belongs.. Specify either a storage system or serial number for the type and not both.

serialnumber|sn

The serial number of the storage system. Specify either a storage system or serial number for the type and not both.

vmax|t

Type of storage system.

PORTGROUPNAME|portgroupname

Name of the storage port group.

STORAGEPORTS|sp

Storage port members.

Common Arguments

This operation also takes the Common Parameters on page 24.

Create a storage port using the

ViPR Controller

CLI

1. List the storage systems using `viprcli storagesystem list`. Note the last three digits of the serial number of the storage system to which the storage port needs to be added.
2. Create the storage port groups using
   `viprcli storageportgroup create -n PG_HR -sn 197 -t vmax -sp FA-2D:6 FA-1D:6`
3. List the storage port groups for the storage system to ensure that the new storage portgroup created successfully using
   `viprcli storageportgroup list -sn 197 -t vmax`

viprcli storageportgroup delete

Deletes a registered storage portgroup.

Syntax

```
            viprcli storageportgroup delete
            -name|n <storageportname>
            -type|t <storagesystemtype>
            {-storagesystem|ss <storagesystemname> | -serialnumber|
sn <serialnumber>}
            [-serialnumber|sn <serialnumber>]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli storageportgroup delete` command deletes a registered storage portgroup.

Options

name

The hostname or IPv4 address or IPv6 address in quotes and brackets of ViPR. This is a mandatory parameter.

type|t

The type of the storage system is applicable for VMAX storage system only. This is a mandatory parameter.

portgroupname|n

Name of the Storageportgroup.

storagesystem|ss

Name of the storage system.

serialnumber|sn

The serial number of the storage system. Specify either a storage system or serial number for the type and not both.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli storageportgroup deregister

Unregister a storage portgroup.

Syntax

```
            viprcli storageportgroup deregister
            -name|n
        <storageportname>
            -type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleio | xtremio | ddmc | ibmxiv | vnxe | ecs |
datadomain}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storageport deregister` command deregisters a storage portgroup.

Options

portgroupname|n

The storage port name to be registered. This is a mandatory parameter.

hostname|hostname

Hostname or IPv4 address or IPv6 address inside quotes and brackets of ViPR.

port|port_number

Port number of ViPR

ui_port_number|portui

https port number of the ViPR Portal UI.

coookiefile|cf

Full name of cookiefile.

type|t

The storage type is applicable for VMAX storage system only . This is a mandatory parameter.

{storagesystem| ss | serialnumber|sn}

The storage system name or serial number on which the storage pool resides. One of these parameters is required.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 188

```
#viprcli storageportgroup deregister -n PG_HR -sn 197 -t vmax
```

Use the list command to view results.

viprcli storageportgroup list

List storageportgroup for a storagesystem.

Syntax

```
            viprcli storageportgroup list
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            -type|t {isilon, vnxblock, vnxfile,
        vmax}
            [-metrics|m]
            [{-verbose|v | -long|l}]
```

Description

`viprcli storageportgroup list` command displays information about storage ports groups associated with a particular storage system. Storage ports groups are discovered - you cannot create a storage port object with the ViPR Controller command line interface.

Options

storagesystem|ss

Type of the storage system.

serialnumber|sn

The serial number of the storage system. Either this parameter or the storage system parameter are required.

type|t

Storage system type. Applicable for VMAX system only.

verbose|v

List storage portgroup with details. This is an optional parameter.

long|l

List storage portgroup in table with details. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 189

```
# viprcli storageportgroup list -sn 197 -t vmax
```

viprcli storageportgroup register

Register a storage portgroup with ViPR Controller.

Syntax

```
            viprcli storageportgroup register
            -type|t vmax
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|Sn <serialnumber>}
            [-name|n
        <storageportname>]
            [-hostname|hn <hostname>]
            [-h]
```

Description

`viprcli storageportgroup register` command registers a storage portgroup.

Options

type|t

Applicable for vmax system only. This is a mandatory parameter.

{storagesystem|ss |serialnumber|sn}

The storage system name or serial number on which the storage pool resides. One of these parameters is required.

portgroupname|n

The storage portgroup name to be registered. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 190

```
#viprcli storageportgroup register -n PG_HR -sn 197 -t vmax
```

Use the list command to view results.

viprcli storageportgroup show

Show details of a storage portgroup.

Syntax

```
            viprcli storageportgroup show
            -name|n
        <storageportname>
            -type|t {isilon | vnxblock | vnxfile |
        vmax | netapp | netappc |vplex | hds | openstack | scaleio
| xtremio | ddmc |
        ibmxiv | vnxe | datadomain}
            {-storagesystem|ss
        <storagesystemname> | -serialnumber|sn <serialnumber>}
            [-xml]
            [-hostname|hn
        <hostname>]
            [-port}po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storageportgroup show` command displays an extended list of information about a particular storage portgroup.

Options

name|n

Name (fully qualified domain name) or IPv4 address 9i.e.192.0.2.0) or IPv6 address in quotes and brackets (i.e "[2001:db8::1]") of ViPR

port number|-po

Port number of ViPR.

portui|pu

https port number of ViPR Portal UI.

type|t

Storage system type. Applicable for VMAX storage system type only. This is a mandatory parameter.

storagesystem|ss

The storage system where the target storage pool resides. Either this parameter or the serialnumber parameter are required. You do not need both storagesystem and serialnumber.

serialnumber|sn

The serial number of the storage system. Either this parameter or the storagesystem parameter are required.

verbose|v

Lists the storageportgroup with details.

long|l

Lists Storageportgroup in table with details.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 191

```
#viprcli storageportgroup show -n PG_HR -sn 197 -t vmax
```

Example 192

CHAPTER 42

viprcli storageprovider commands

This chapter contains the following topics:

- viprcli storageprovider list 452
- viprcli storageprovider create 452
- viprcli storageprovider delete 454
- viprcli storageprovider scan 455
- viprcli storageprovider show 455
- viprcli storageprovider update 456

viprcli storageprovider list

List a storage provider.

Syntax

```
            viprcli storageprovider list
            [-interface|if <interface>
        {hicommand | smis | vplex | cinder, scaleioapi | ddmc |
ibmxiv |
        xtremio}]
            [-v|verbose]
            [-l|long]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storageprovider list` command lists a storage provider.

Options

interface|if

The interface of the storage provider. Valid options include `hicommand`, `smis`, `vplex`, `cinder`, `scaleioapi`, `ddmc`, `xtremio`, and `ibmxiv` .

verbose|v

Lists the details of the storage provider.

long|l

Lists the details in a table of the storage provider.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli storageprovider create

Create a storage provider.

Syntax

```
viprcli storageprovider create
[name|n        <storage_provider_name>
-provip|providerip <provider        ipaddress>
-provport <provider        port>
-user|u <username>
-interface|if <interface>        {hicommand | smis | vplex |
cinder, scaleioapi | ddmc | ibmxiv | xtremio}

[-hostname|hn        <hostname>]
```

```
[-usessl|ssl]
[-secondary_username <Secondary        Username>]
[-element_manager_url <Element Manager URL>]
[-sio_cli <cli_path>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-h]
```

Description

`viprcli storageprovider create` command creates a storage provider.

Options

name|n

The name of the storage provider. This is a mandatory parameter.

provip|providerip

The IP address of the storage provider. This is a mandatory parameter.

Note

For a storage provider whose `interface` is set to `scaleioapi`, this is the IP address of the ScaleIO Gateway host.

provport

The port number of the storage provider. This is a mandatory parameter.

Note

For a storage provider whose `interface` is set to `scaleioapi`, this is the port used to access the ScaleIO REST API service.

- With SSL enabled, the default is 443.
- With SSL disabled, the default is 80.

user|u

The username of the storage provider. This is a mandatory parameter.

Note

For ScaleIO, this is a user that can access the Primary MDM.

interface|if

The interface of the storage provider. This is a mandatory parameter. Possible types are: hicommand, smis, vplex, cinder, scaleioapi, ddmc, xtremio, and ibmxiv. If ibmxiv is the interface given, you will also need to enter the following two options:

- hyperScaleHost — IP address of the IBM XIV Hyper Scale Manager host.

- hyperScalePort — Hyper Scale Manager REST Server Port. Default is 8443.

```
viprcli storageprovider update -n xiv -provip 10.247.23.203 -
provport 5989 -user admin -if ibmxiv -hyperScaleHost
10.247.23.203 -hyperScalePort 8443 -secondary_username
```

usessl|ssl

Sets the use of SSL. This is an optional parameter.

secondary_username

Specify a secondary username to be used. This is an optional parameter.

element_manager_url

The URL of the element manager. This is an optional parameter.

sio_cli

The path to the ScaleIO CLI. This is an optional parameter.

Note

This parameter is no longer used.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli storageprovider delete

Delete a storage provider.

Syntax

```
viprcli storageprovider delete
[-name|n <storage_provider_name>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli storageprovider delete` command deletes a storage provider.

Options

name|n

The name of the storage provider. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli storageprovider scan

Scan a storage provider.

Syntax

```
viprcli storageprovider scan
[-hostname|hn
<hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookie_file>]
[-h]
```

Description

`viprcli storageprovider scan` command scans a storage provider.

Note

Rediscovery of a standalone ScaleIO storage provider discovers new SDC hosts or removes deleted SDC hosts, but it does not update theScaleIO initiators. To rediscover the ScaleIO initiators, rediscover the storage systems associated with the ScaleIO storage provider. See . viprcli storagesystem discover on page 465

Options

common arguments

This operation also takes the Common Parameters on page 24.

viprcli storageprovider show

Show a storage provider.

Syntax

```
viprcli storageprovider show
[-name|n <storage_provider_name>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-xml]
[-h]
```

Description

`viprcli storageprovider show` command shows a storage provider.

Options

name|n

The name of the storage provider. This is a mandatory parameter.

xml

Displays the output in XML format.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli storageprovider update

Update a storage provider.

Syntax

```
            viprcli storageprovider update
[-name|n        <storage_provider_name>]
[-newname        <storage_provider_name>]
[-provip|providerip <provider        ipaddress>]
[-provport <provider        port>>]

  [-user|u <username>]
[-interface|if        <interface>]
[-ssl|usessl ]
[-secondary_username <Secondary        Username>]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-usessl|ssl]
[-h]
```

Description

`viprcli storageprovider update` command updates a storage provider.

Options

name|n

The name of the storage provider. This is a mandatory parameter.

newname

The new name of the storage provider. This is a mandatory parameter.

provip|providerip

The IP address of the storage provider. This is a mandatory parameter.

Note

For a storage provider whose `interface` is set to `scaleioapi`, this is the IP address of the ScaleIO Gateway host.

---

provport

The port number of the storage provider. This is a mandatory parameter.

Note

For a storage provider whose `interface` is set to `scaleioapi`, this is the port used to access the ScaleIO REST API service.

- With SSL enabled, the default is 443.
- With SSL disabled, the default is 80.

user|u

The username of the storage provider. This is a mandatory parameter.

interface|if

The interface of the storage provider. This is a mandatory parameter. Possible types are: hicommand, smis, vplex, cinder, scaleioapi, ddmc, xtremio, and ibmxiv. If ibmxiv is the interface given, you will also need to enter the following two options:

- hyperScaleHost — IP address of the IBM XIV Hyper Scale Manager host.
- hyperScalePort — Hyper Scale Manager REST Server Port. Default is 8443.

```
viprcli storageprovider update -n xiv -provip 10.247.23.203 -
provport 5989 -user admin -if ibmxiv -hyperScaleHost
10.247.23.203 -hyperScalePort 8443 -secondary_username
```

usessl|ssl

Sets the use of SSL.

secondary_username

Specify a secondary username to be used.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 43

viprcli storagesystem commands

This chapter contains the following topics:

- viprcli storagesystem connectivity 460
- viprcli storagesystem create 461
- viprcli storagesystem delete 463
- viprcli storagesystem deregister 464
- viprcli storagesystem discover 465
- viprcli storagesystem discover_arrayaffinity 466
- viprcli storagesystem discover_unmanagedfilesystems 467
- viprcli storagesystem discover_unmanagedvolumes 468
- viprcli storagesystem get_unmanagedfilesystems 469
- viprcli storagesystem get_unmanagedvolumes 470
- viprcli storagesystem list 472
- viprcli storagesystem register 475
- viprcli storagesystem show 476
- viprcli storagesystem show-unmanagedexportmask 478
- viprcli storagesystem update 479

viprcli storagesystem connectivity

Returns connectivity information for a storage system.

Syntax

```
            viprcli storagesystem connectivity
            -serialnumber|sn
        <serialnumber>
            [-type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleioapi | xtremio | ddmc | ibmxiv | vnxe | datadomain
| ecs}]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli storagesystem connectivity` command lists switches and arrays to which a storage system is connected.

Options

serialnumber|sn

The serial number of a ViPR Controller storage system. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleioapi, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is an optional parameter. But the type must be specified along with either a name or serial number.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 193

```
# viprcli storagesystem connectivity -t vnxblock -sn APM00121202922
{
   "storage_connectivity": []
}
```

viprcli storagesystem create

Create a storage system object.

Syntax

```
            viprcli storagesystem create
            -name|n
        <storagesystemame>
            -type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
hds | openstack | ecs
        | ibmxiv | vnxe}
            [-deviceip|dip
        <deviceip>]
            [-deviceport|dp
        <deviceport>]
            [-user|u <username>]
            [-serialnumber|sn
        <serialnumber>]
            [-smisip <smisip>]
            [-smisport|sp
        <smisport>]
            [-smisuser|su
        <smisuser>]
            [-if|interface
        <interface>]
            [-usessl|ssl <sslstate>]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli storagesystem create` command creates and registers a storage system object accessible to the user. A storage system is an array or other supported storage device. This command is silent on success.

If you discover VMAX or VNXblock through SMI-S, all arrays that the SMI-S Provider is aware of get discovered and registered, and all the storage pools on those arrays are immediately available to users.

Options

name|n

The name of a storage system that has been discovered with the `viprcli storagesystem discover` operation. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, hds, openstack, ecs, ibmxiv or vnxe. This is a mandatory parameter.

deviceip|dip

The IP address of the storage system. The device IP address is required to create a storage system of type vnxfile, netapp, or isilon. This is an optional parameter.

deviceport|dp

The port of the storage system. The device port number is required to create a storage system of type vnxfile, netapp, or isilon.This is an optional parameter.

user|u

The user name to use to authenticate on the storage system. The user name is required to create a storage system of type vnxfile, netapp, or isilon. This is an optional parameter.

serialnumber|sn

The serial number of the storage system. This is an optional parameter.

smisip

The IP address of an SMI-S provider. The SMI-S IP address is required to create a storage system of type vnxblock or vmax. This is an optional parameter.

smisport|sp

The communications port for the SMI-S provider. The SMI-S port is required to create a storage system of type vnxblock or vmax. The default is port 5989, which uses ssl. You must specify the -ssl parameter for the default port. This is an optional parameter.

smisuser|su

A valid SMI-S user. The SMI-S user is required to create a storage system of type vnxblock or vmax.

usessl|ssl

Specify this parameter to communicate with over SSL. If you do not specify this parameter, secure sockets are not used. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 194

```
# viprcli storagesystem create -n RPSystem -t vnxblock
-smisip 192.168.182.223 -sp 5989 -sn APM00120701350 -su admin  -ssl
Enter password of the SMIS provider:
Retype password:
```

Use list command to view the results.

```
# viprcli storagesystem list
```

Name	Provider Name	System Type	Serial Number
CLARIION+APM00111701806	RPSystem	vnxblock	APM00111701206

Example 194 (continued)

CLARIION+APM00120701240	RPSystem	vnxblock	APM00120701840
CLARIION+APM00120701245	RPSystem	vnxblock	APM00120701845
SYMMETRIX+000194900372	RPSystem	vmax	000194900472
SYMMETRIX+000194900373	RPSystem	vmax	000194900573
SYMMETRIX+000194900374	RPSystem	vmax	000194900674

viprcli storagesystem delete

Delete a storage system.

Syntax

```
            viprcli storagesystem delete
            -name|n
        <storagesystemame>
            [-type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleioapi | xtremio | ddmc | ibmxiv | vnxe | ecs |
datadomain}]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli storagesystem delete` command deletes the specified storage system. This command is silent on success.

Options

name|n

The name of a storage system to be deleted. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleioapi, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is an optional parameter. But the type must be specified along with either a name or serial number.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 195

Example 195 (continued)

```
# viprcli storagesystem delete -n CLARIION+APM00121301070 -t
vnxblock
```

Use list command to view the results.

```
 # viprcli storagesystem list
  NAME                    PROVIDER_NAME SYSTEM_TYPE
SERIAL_NUMBER
   SS01                                  isilon
6805ca00acee0d63a6508020ba69b3849f37
```

viprcli storagesystem deregister

Unregister the specified storage system.

Syntax

```
            viprcli storagesystem deregister
            -name
        <storagesystemname>
            [-type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleioapi | xtremio | ddmc | ibmxiv | vnxe | datadomain
| ecs}]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli storagesystem deregister` command unregisters the specified storage system. This command is silent on success.

Options

name|n

The name of a storage system that has been discovered with the `viprcli storagesystem discover` operation. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleioapi, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is an optional parameter. But the type must be specified along with either a name or serial number.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 196

```
# viprcli storagesystem deregister -n CLARIION+APM00120400480 -t
vnxblock
```

viprcli storagesystem discover

Discover the storage systems, associated storage pools, and storage ports.

Syntax

```
            viprcli storagesystem discover
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [{-name|n <name> |
        serialnumber|sn <serialnumber>]
            [-type|t <storagesystemtype>

{isilon,vnxblock,vnxfile,vmax,netapp,,vplex,hds,openstack,scaleioapi
,xtremio,ddmc,ibmxiv,netappc,vnxe,ecs,datadomain}]
            [-all|a]
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

`viprcli storagesystem discover` command discovers storage systems, the associated storage pools, and the storage ports. This command is silent on success.

Options

name|n

The name of a storage system that has not been discovered with the `viprcli storagesystem discover` operation. This is an optional parameter. Use either a serial number or name and not both.

serialnumber|sn

The serial number of the storage system that has not been discovered with the `viprcli storagesystem discover` operation. This is an optional parameter. Use either a serial number or name and not both.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleioapi, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain.. This is an optional

parameter. But the type must be specified along with either a name or serial number.

all|a

Discover all objects. This is an optional parameter.

synchronous|sync

Execute in synchronous mode. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 197

```
# viprcli storagesystem discover -n CLARIION+APM00120400480 -t
vnxblock
```

Example 198

```
# viprcli storagesystem discover -sn APM00120400480 -t vnxblock
```

Example 199

```
# viprcli storagesystem discover -all
```

viprcli storagesystem discover_arrayaffinity

Discover the host mapping to VMAX, VNX for Block, Unity, and XtremIO storage systems for the given storage system.

If the storage system is managed by a storage provider, host/array affinity will be discovered on all the storage systems managed by the storage provider.

Syntax

```
viprcli storagesystem discover_arrayaffinity
[-type|t <storagesystemtype> {vnxblock,vmax,xtremio} unity]
[-n]
```

Options

type|t

One of vnxblock, vmax, or xtremio.

name|n

The name of a storage system.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

```
./viprcli storagesystem discover_arrayaffinity -t unity  -n
losav164.lss.emc.com
```

viprcli storagesystem discover_unmanagedfilesystems

Discover the unmanaged filesystem by storage system.

Syntax

```
        viprcli storagesystem
discover_unmanagedfilesystems
        [{-name|n <storagesystemame> |
    serialnumber|sn <serialnumber>]
        [-type|t <storagesystemtype>
    {isilon | vnxfile | netapp | netappc | vnxe | datadomain}]
        [-all|a]
        [-hostname|hn
    <hostname>]
        [-port|po
    <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        [-h]
```

Description

`viprcli storagesystem discover_unmanagedfilesystems` command discovers the unmanaged filesystems by storage system.

Options

type|t

One of isilon, vnxfile, netapp, netappc, vnxe, or datadomain. This is an optional parameter. But the type must be specified along with either a name or serial number.

name|n

The name of a storage system that has not been discovered with the viprcli storagesystem discover operation. This is an optional parameter. Use either a serial number or name and not both.

serialnumber|sn

The serial number of the storage system that has not been discovered with the viprcli storagesystem discover operation. This is an optional parameter. Use either a serial number or name and not both.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 200

```
# viprcli storagesystem discover_unmanagedfilesystems -t vnxfile -
sn APM00121600480
```

Example 201

```
# viprcli storagesystem discover_unmanagedfilesystems -t vnxfile -n
CLARIION+APM00121212922
```

viprcli storagesystem discover_unmanagedvolumes

Discover the unmanaged volumes by storage system.

Syntax

```
            viprcli storagesystem discover_unmanagedvolumes
            [{-name|n <storagesystemame> |
        serialnumber|sn <serialnumber>]

            [-type|t <storagesystemtype>
        {vnxblock | vmax | vplex | hds | scaleio | xtremio | ibmxiv
| vnxe}}]
            [-all|a]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli storagesystem discover_unmanagedvolumes` command discovers the unmanaged volumes by storage system.

Options

type|t

One of vnxblock, vmax, vplex, hds, scaleio, xtremio, ibmxiv, or vnxe}. This is an optional parameter. But the type must be specified along with either a name or serial number.

name|n

The name of a storage system that has not been discovered with the viprcli storagesystem discover operation. This is an optional parameter. Use either a serial number or name and not both.

serialnumber|sn

The serial number of the storage system that has not been discovered with the viprcli storagesystem discover operation. This is an optional parameter. Use either a serial number or name and not both.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 202

```
# viprcli storagesystem discover_unmanagedvolumes -t vnxblock -sn
APM00120400480
```

Example 203

```
# viprcli storagesystem discover_unmanagedvolumes -t vnxblock -n
CLARIION+APM00121202922
```

viprcli storagesystem get_unmanagedfilesystems

Return the unmanaged file systems of specified storage system.

Syntax

```
            viprcli storagesystem get_unmanagedfilesystems
            [{-name|n <storagesystemame> |
        serialnumber|sn <serialnumber>]
            -type|t <storagesystemtype>
        {isilon | vnxfile | netapp | netappc | vnxe | datadomain}
            [-all|a]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
```

```
            [-h]
```

Description

`viprcli storagesystem get_unmanagedfilesystems` command returns the unmanaged file systems of the specified storage system. This command is silent on success.

Options

type|t

One of isilon, vnxfile, netapp, netappc, vnxe or datadomain}. This is an mandatory parameter. The type must be specified along with either a name or serial number.

name|n

The name of a storage system that has not been discovered with the viprcli storagesystem discover operation. This is an optional parameter. Use either a serial number or name and not both.

serialnumber|sn

The serial number of the storage system that has not been discovered with the viprcli storagesystem discover operation. This is an optional parameter. Use either a serial number or name and not both.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 204

```
# viprcli storagesystem get_unmanagedfilesystems -t vnxfile -sn
APM00121400480
```

Example 205

```
# viprcli storagesystem get_unmanagedfilesystems -t vnxfile -n
CLARIION+APM00121212922
```

viprcli storagesystem get_unmanagedvolumes

Return the unmanaged volumes of specified storage system.

Syntax

```
            viprcli storagesystem get_unmanagedvolumes
            [{-name|n <storagesystemame> |
        serialnumber|sn <serialnumber>]
```

```
            -type|t <storagesystemtype>
        {vnxblock | vmax | vplex | hds | scaleio | xtremio | ibmxiv
| vnxe}
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli storagesystem get_unmanagedvolumes` command returns the unmanaged volumes of the specified storage system. This command is silent on success.

Options

type|t

One of vnxblock, vmax, vplex, hds, scaleio, xtremio, ibmxiv, or vnxe. This is a mandatory parameter and must be specified along with either a name or serial number.

name|n

The name of a storage system that has not been discovered with the viprcli storagesystem discover operation. This is an optional parameter. Use either a serial number or name and not both.

serialnumber|sn

The serial number of the storage system that has not been discovered with the viprcli storagesystem discover operation. This is an optional parameter. Use either a serial number or name and not both.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 206

```
 # viprcli storagesystem get_unmanagedvolumes -t vnxblock -sn
APM00120400480
```

Example 207

```
# viprcli storagesystem get_unmanagedvolumes -t vnxblock -n
CLARIION+APM00121202922
```

viprcli storagesystem list

Display the list of all storage systems.

Syntax

```
            viprcli storagesystem list
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-type|t <storagesystemtype>

{isilon,vnxblock,vnxfile,vmax,netapp,vplex,hds,openstack,scaleio,xtr
emio,ddmc,ibmxiv,netappc,vnxe,ecs,datadomain}]
            [-verbose|v]
            [-long|l]
```

Description

`viprcli storagesystem list` command displays the list of all storage systems accessible to the user.

Options

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleioapi, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is an optional parameter. But the type must be specified along with either a name or serial number.

verbose|v

Lists storage systems with details. This is an optional parameter. The verbose listing of storage systems displays a detailed listing of all the registered systems. Use either verbose or long.

long|l

Lists storage systems in a large table. This is an optional parameter. Use either verbose or long.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 208

```
# viprcli storagesystem list -l
NAME                    PROVIDER_NAME SYSTEM_TYPE SERIAL_NUMBER
IP_ADDRESS  PORT_NUMBER REGISTRATION_STATUS SMIS_USE_SSL
JOB_DISCOVERY_STATUS
CLARIION+APM00120400480vnx_test      vnxblock    APM00120400480
```

Example 208 (continued)

```
192.168.1.2 5989        REGISTERED          True         COMPLETE
CLARIION+APM00121202922vnx-block-dev vnxblock    APM00121202922
192.168.1.4 5988        REGISTERED          False        COMPLETE
```

Example 209

```
# viprcli storagesystem list -v
[
   {
      "active_provider_uri": {
         "id": "urn:storageos:SMISProvider:
        5df2f9a1-8d4e-4493-8dab-9040a2df8a9b:",
         "link": {
            "href": "/vdc/smis-providers/urn:storageos:
            SMISProvider:5df2f9a1-8d4e-4493-8dab-9040a2df8a9b:",
            "rel": "self"
         }
      },
      "assocociated_systems": [],
      "async_actions": [
         "CreateGroupReplica",
         "CreateElementReplica"
      ],
      "compatibility_status": "COMPATIBLE",
      "creation_time": 1378302052454,
      "export_masks": [],
      "firmware_version": "05.32.000.5.206",
      "id": "urn:storageos:StorageSystem:
      3b8145df-70a9-40d0-b9ef-373405f131e4:",
      "inactive": false,
      "ip_address": "192.168.99.25",
      "job_discovery_status": "COMPLETE",
      "job_metering_status": "COMPLETE",
      "last_discovery_run_time": 1378403795012,
      "last_discovery_status_message":
      "Discovery completed successfully for Storage System:
      urn:storageos:StorageSystem:
      3b8145df-70a9-40d0-b9ef-373405f131e4:",
      "last_metering_run_time": 1378404330370,
      "link": {
         "href": "/vdc/storage-systems/urn:
         storageos:StorageSystem:3b8145df-70a9-40d0-
b9ef-373405f131e4:",
         "rel": "self"
      },
      "max_resources": -1,
      "model": "Rack Mounted VNX7500",
      "name": "CLARIION+APM00120400480",
      "native_guid": "CLARIION+APM00120400480",
      "next_discovery_run_time": 1378407301442,
      "next_metering_run_time": 1378407923574,
      "num_resources": 0,
      "protocols": [],
      "provider_name": "vnx_test",
      "reachable": true,
      "registration_status": "REGISTERED",
      "secondary_ips": [],
      "serial_number": "APM00120400480",
      "smis_port_number": 5989,
      "smis_provider_ip": "192.168.99.25",
      "smis_providers": [
```

Example 209 (continued)


```
         {
            "id": "urn:storageos:SMISProvider:
             5df2f9a1-8d4e-4493-8dab-9040a2df8a9b:",
            "link": {
               "href": "/vdc/smis-providers/urn:
                storageos:SMISProvider:
                5df2f9a1-8d4e-4493-8dab-9040a2df8a9b:",
               "rel": "self"
            }
         }
      ],
      "smis_use_ssl": true,
      "smis_user_name": "admin",
      "supported_provisioning_type": "THIN_AND_THICK",
      "system_type": "vnxblock",
      "tags": []
   },
   {
      "active_provider_uri": {
         "id": "urn:storageos:SMISProvider:
         f41c2a65-8290-40b6-9071-e95d3423e339:",
         "link": {
            "href": "/vdc/smis-providers/urn:storageos:
            SMISProvider:f41c2a65-8290-40b6-9071-e95d3423e339:",
            "rel": "self"
         }
      },
      "assocociated_systems": [],
      "async_actions": [
         "CreateGroupReplica",
         "CreateElementReplica"
      ],
      "compatibility_status": "COMPATIBLE",
      "creation_time": 1378294707198,
      "export_masks": [],
      "firmware_version": "05.32.000.5.206",
      "id": "urn:storageos:StorageSystem:
       fbdba394-d13f-40ba-aaf5-87a864022fd2:",
      "inactive": false,
      "ip_address": "192.168.99.24",
      "job_discovery_status": "COMPLETE",
      "job_metering_status": "COMPLETE",
      "last_discovery_run_time": 1378403846395,
      "last_discovery_status_message":
      "Discovery completed successfully for Storage System:
      urn:storageos:StorageSystem:fbdba394-d13f-40ba-
aaf5-87a864022fd2:",
      "last_metering_run_time": 1378404327430,
      "link": {
         "href": "/vdc/storage-systems/urn:storageos:
         StorageSystem:fbdba394-d13f-40ba-aaf5-87a864022fd2:",
         "rel": "self"
      },
      "max_resources": -1,
      "model": "Rack Mounted VNX5300",
      "name": "CLARIION+APM00121202922",
      "native_guid": "CLARIION+APM00121202922",
      "next_discovery_run_time": 1378407332830,
      "next_metering_run_time": 1378407923553,
      "num_resources": 0,
      "protocols": [],
      "provider_name": "vnx-block-dev",
      "reachable": true,
      "registration_status": "REGISTERED",
      "secondary_ips": [],
```

Example 209 (continued)

```
      "serial_number": "APM00121202922",
      "smis_port_number": 5988,
      "smis_provider_ip": 192.168.99.24",
      "smis_providers": [
         {
            "id": "urn:storageos:SMISProvider:
           f41c2a65-8290-40b6-9071-e95d3423e339:",
            "link": {
               "href": "/vdc/smis-providers/urn:storageos:
              SMISProvider:f41c2a65-8290-40b6-9071-e95d3423e339:",
               "rel": "self"
            }
         }
      ],
      "smis_use_ssl": false,
      "smis_user_name": "admin",
      "supported_provisioning_type": "THIN_AND_THICK",
      "system_type": "vnxblock",
      "tags": []
   }
]
```

viprcli storagesystem register

Register the specified storage system.

Syntax

```
            viprcli storagesystem register
            -name
        <storagesystemname>
            [-type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleioapi | xtremio | ddmc | ibmxiv | vnxe | datadomain
| ecs}]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli storagesystem register` command registers the specified storage system accessible to the user. This command is silent on success.

Options

name|n

The name of a storage system that has been discovered with the `viprcli storagesystem discover` operation. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleioapi, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is an optional parameter. But the type must be specified along with either a name or serial number.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 210

```
# viprcli storagesystem register -n CLARIION+APM00120400480 -t
vnxblock
```

viprcli storagesystem show

Return the details for a specified storage system type.

Syntax

```
            viprcli storagesystem show
            [{-name <storagesystemname> |
        -sn <serialnumber>}]
            [-type|t <storagesystemtype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleioapi | xtremio | ddmc | ibmxiv | vnxe | datadomain
| ecs}]
            [-xml]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli storagesystem show` command displays the details for a specified storage system accessible to the user.

Options

name|n

The storage system name. You must specify either a storage system name or a serial number. This is an optional parameter.

serialnumber|sn

The serial number of the storage system. You must specify either a storage system name or a serial number. This is an optional parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleioapi, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is an optional parameter. But the type must be specified along with either a name or serial number.

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 211

```
viprcli storagesystem show -t vmax -n SYMMETRIX+000196801612
{
   "active_provider_uri": {
      "id":
"urn:storageos:StorageProvider:efe2ba3f-3dc4-48e8-9c0f-325b3325f2d5:
vdc1",
      "link": {
         "href": "/vdc/smis-providers/
urn:storageos:StorageProvider:efe2ba3f-3dc4-48e8-9c0f-325b3325f2d5:v
dc1",
         "rel": "self"
      }
   },
   "associated_systems": [],
   "async_actions": [
      "CreateElementReplica",
      "CreateGroupReplica"
   ],
   "average_port_metrics": -1.0,
   "compatibility_status": "COMPATIBLE",
   "connected_systems": [],
   "creation_time": 1471006976580,
   "export_masks": [],
   "firmware_version": "5977.916.878",
   "global": false,
   "id": "urn:storageos:StorageSystem:256e89c9-d48c-49f2-
a30f-47cd69a1dfd4:vdc1",
   "inactive": false,
   "internal": false,
   "job_arrayaffinity_status": "COMPLETE",
   "job_discovery_status": "COMPLETE",
   "job_metering_status": "CREATED",
   "last_arrayaffinity_run_time": 1474766191947,
   "last_arrayaffinity_status_message": "Array Affinity Discovery
completed successfully for Storage System:
urn:storageos:StorageSystem:256e89c9-d48c-49f2-
a30f-47cd69a1dfd4:vdc1",
   "last_discovery_run_time": 1474296243478,
   "last_discovery_status_message": "Unknown Status",
   "last_metering_run_time": 0,
   "link": {
      "href": "/vdc/storage-systems/urn:storageos:StorageSystem:
256e89c9-d48c-49f2-a30f-47cd69a1dfd4:vdc1",
      "rel": "self"
   },
  "max_resources": -1,
   "model": "VMAX100K",
   "name": "SYMMETRIX+000196801612",
```

Example 211 (continued)

```
   "native_guid": "SYMMETRIX+000196801612",
  "next_arrayaffinity_run_time": 1474766743789,
   "next_discovery_run_time": 1474296353396,
   "next_metering_run_time": 0,
   "num_resources": 5,
   "protocols": [],
   "reachable": true,
   "registration_status": "REGISTERED",
   "remote": false,
   "secondary_ips": [],
   "serial_number": "000196801612",
   "smis_port_number": 5989,
   "smis_provider_ip": "lglw9071.lss.emc.com",
   "smis_providers": [
      {
         "id":
"urn:storageos:StorageProvider:efe2ba3f-3dc4-48e8-9c0f-325b3325f2d5:
vdc1",
         "link": {
            "href": "/vdc/smis-providers/
urn:storageos:StorageProvider:efe2ba3f-3dc4-48e8-9c0f-325b3325f2d5:v
dc1",
            "rel": "self"
         }
      }
   ],
 "smis_use_ssl": true,
   "smis_user_name": "admin",
   "success_arrayaffinity_time": 1474766191947,
   "success_discovery_time": 1474296243478,
   "success_metering_time": 0,
   "supported_provisioning_type": "THIN",
   "supported_replication_types": [],
   "system_type": "vmax",
   "tags": [],
   "vdc": {
      "id": "urn:storageos:VirtualDataCenter:d4251c37-6982-492a-
b9d9-8a1f84b1cd52:vdc1",
      "link": {
         "href": "/vdc/
urn:storageos:VirtualDataCenter:d4251c37-6982-492a-
b9d9-8a1f84b1cd52:vdc1",
         "rel": "self"
      }
   }
}
```

viprcli storagesystem show-unmanagedexportmask

Displays the details of a specified unmanaged export mask.

Syntax

```
            viprcli storagesystem show-unmanagedexportmask
            -id <id>
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
```

```
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-h]
```

Description

`viprcli storagesystem list` command displays the details of a specified unmanaged export mask.

Options

id

The ID of the unmanaged export mask. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 212

viprcli storagesystem update

Update the storage system.

Syntax

```
            viprcli storagesystem update
            [-h]
            [-hostname|hn
        <viprhostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <name>
            -type|t <storagesystemtype>

{isilon,vnxblock,vnxfile,vmax,netapp,vplex,hds,openstack,scaleio,xtr
emio,ddmc,ibmxiv,netappc,vnxe,ecs,datadomain}
            [-maxresources|mr
        <maxresources>]
            [-newname|nn
        <newstoragesystemname>]
            [newipaddress|ndip
        <newipaddress>]
            [-newport|ndp
        <newport>]
            [-newsusername|nun
        <newusername>]
```

Description

`viprcli storagesystem update` command updates a storage system object accessible to the user. This command is silent on success.

Note

Update requires at least one of these parameters to be specified along with the mandatory parameters: -newname, -newport, -newipaddress, -newusername, or -maxresources .

Options

name|n

The name of a storage system that has been discovered with the `viprcli storagesystem discover` operation. This is a mandatory parameter.

type|t

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleioapi, xtremio, ddmc, ibmxiv, vnxe, ecs, or datadomain. This is an optional parameter. But the type must be specified along with either a name or serial number.

maxresources|mr

The maximum number of resources in the storage system. This is an optional parameter.

newname|nn

The new name for the storage system. This is an optional parameter.

newipaddress|ndip

The new IP address of the storage system. This is an optional parameter.

newport|ndp

The new communications port for the storage system. This is an optional parameter.

newusername|nun

The new user name to use to authenticate on the storage system. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 213

```
# viprcli storagesystem update -n CLARIION+APM00120400480 -t
vnxblock -nn CLARIION+APM00120400482
```

CHAPTER 44

viprcli system commands

This chapter contains the following topics:

- viprcli system add-license 483
- viprcli system add-site 483
- viprcli system cluster-ipinfo 484
- viprcli system cluster-ipreconfig 485
- viprcli system cluster-poweroff 487
- viprcli system cluster-recovery 488
- viprcli system cluster-recovery-status 488
- viprcli system connectemc-ftps 489
- viprcli system connectemc-smtp 490
- viprcli system create-backup 491
- viprcli system db-consistency-check 492
- viprcli system db-consistency-check-cancel 492
- viprcli system db-consistency-check-status 493
- viprcli system dbrepair-status 493
- viprcli system delete-backup 494
- viprcli system delete-site 495
- viprcli system delete-sites 495
- viprcli system delete-task 496
- viprcli system disable-update-check 496
- viprcli system download-backup 497
- viprcli system failover-site 498
- viprcli system get-alerts 498
- viprcli system get-cluster-state 500
- viprcli system get-diagnostics 501
- viprcli system get-esrsconfig 505
- viprcli system get-health 505
- viprcli system get-license 506
- viprcli system get-log-level 507
- viprcli system get-logs 512
- viprcli system get-properties 514
- viprcli system get-properties-metadata 525
- viprcli system get-stats 527
- viprcli system get-storage 530
- viprcli system get-target-version 531
- viprcli system install-image 532
- viprcli system ipreconfig-status 533
- viprcli system list-backup 534
- viprcli system list-external-backup 534
- viprcli system list-sites 536
- viprcli system pause-site 537
- viprcli system pause-sites 538
- viprcli system pull-backup 538
- viprcli system pull-backup-cancel 539
- viprcli system query-backup 540

- viprcli system query-backup-info 540
- viprcli system reboot-node 541
- viprcli system remove-image 542
- viprcli system reset-properties 543
- viprcli system restart-service 544
- viprcli system restore-backup 545
- viprcli system restore-backup-status 546
- viprcli system resume-site 547
- viprcli system retry-site 547
- viprcli system send-alert 548
- viprcli system send-heartbeat 550
- viprcli system send-registration 550
- viprcli system set-log-level 551
- viprcli system set-properties 552
- viprcli system show-site 555
- viprcli system site-error 556
- viprcli system site-time 557
- viprcli system skip-setup 557
- viprcli system switchover-site 558
- viprcli system update-cluster 559
- viprcli system update-site 560
- viprcli system upload 561
- viprcli system upload-backup 561
- viprcli system upload-backup-status 562

viprcli system add-license

Add a license file.

Syntax

```
viprcli system add-license
-licensefile|lf <licensefile>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli system add-license` command adds a license file.

Options

licensefile|lf

Name of the license file. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system add-site

Adds a ViPR Controller standby site when configuring ViPR Controller for disaster recovery.

Syntax

```
viprcli system add-site
[-h]
[-hostname <hostname>]
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-name|n
-standbyipaddress|stip
[-description|desc]
-user|u
```

Description

The `viprcli system add-site` command adds a ViPR Controller standby site when configuring ViPR Controller for disaster recovery..

Options

name|n

Name of the site. This is a mandatory parameter.

standbyipaddress|stip

Public IP address of the site. This is a mandatory parameter.

description|desc

Description of site. This is an optional parameter.

user|u

Username for the site. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 214

```
viprcli system add-site -n standby2 -stip 10.247.142.224 -u root
{
"description": "",
"global": null,
"name": "standby2",
"remote": null,
"state": "STANDBY_ADDING",
"tags": [],
"uuid": "b94efdc0-ba06-11e5-b4f9-172b559b2d7a",
"vdc": null,
"vdc_id": "vdc1",
"vip": "10.247.142.224"
}
```

viprcli system cluster-ipinfo

Get the IPs of the ViPR Controller nodes for Hyper-V and non-vApp platforms in a single Virtual Data Center (VDC) configuration.

Syntax

```
            viprcli system cluster-ipinfo
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system cluster-ipinfo` command gets the IPs of the ViPR Controller nodes for Hyper-V and non-vApp platforms.

Note

This command is for single Virtual Data Center (VDC) configurations only. Multiple Virtual Data Center (VDC) configurations are not supported.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 215

```
#viprcli system cluster-ipinfo
{
   "ipv4_setting": {
      "network_addrs": {
         "network_addr": [
            "192.168.0.215",
            "192.168.0.216",
            "192.168.0.217"
         ]
      },
      "network_gateway": "192.168.96.1",
      "network_netmask": "255.255.248.0",
      "network_vip": "192.168.0.214"
   },
   "ipv6_setting": {
      "network_addrs": {
         "network_addr": [
            "::0",
            "::0",
            "::0"
         ]
      },
      "network_gateway6": "::0",
      "network_prefix_length": "64",
      "network_vip6": "::0"
   }
}
```

viprcli system cluster-ipreconfig

Reconfigures the IPs of the ViPR Controller nodes for Hyper-V and non-vApp platforms in a single Virtual Data Center (VDC) configuration.

Syntax

```
            viprcli system cluster-ipreconfig
            -network_vip|nvip
        <network_vip>
            -network_addrs|naddr [network_addrs
        [network_addrs ... 11]]>
            -network_mask|nmask
        <network_mask>
            -network_gateway|ng
```

```
        <network_gateway>
            -network_vip6|nvip6
        <network_vip6>
            -network_addrs6|naddr6 [network_addrs6
        [network_addrs6 ... 11]]>
            -prefix_length|npl [1-128]
            -network_gateway6|ng6
        <network_gateway6>
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system cluster-ipreconfig` command reconfigures the IPs of the ViPR Controller nodes for Hyper-V and non-vApp platforms. ViPR Controller nodes must be stable.

Note

This command is for single Virtual Data Center (VDC) configurations only. Multiple Virtual Data Center (VDC) configurations are not supported.

Options

-network_vip|nvip

The IPv4 address that the ViPR Controller UI, REST API, and CLI clients use to access ViPR Controller. This address must be different from the IPv4 address of the ViPR Controller servers. This is a mandatory parameter.

-network_addrs|naddr

IPv4 addresses of the ViPR Controller servers.
If you are configuring an IPv4 network, each ViPR Controller node requires a unique, static IPv4 address in the subnet defined by the netmask.

If you are not configuring settings for an IPv4 network, provide a default value of 0.0.0.0 for each server, which disables IPv4.

This is a mandatory parameter.

-network_mask|nmask

Netmask for the IPv4 network. This is a mandatory parameter.

-network_gateway|ng

IP address of the default IPv4 gateway.
If you are not configuring settings for an IPv4 network, provide a default value of 0.0.0.0 which disables IPv4.

This is a mandatory parameter.

-network_vip6|nvip6

The IPv6 address that the ViPR Controller UI, REST API, and CLI clients use to access ViPR Controller. This address must be different from the IPv6 address of the ViPR Controller servers.
This is a mandatory parameter.

-network_addrs6|naddr6

IPv6 addresses of the ViPR Controller servers.
. If you are configuring an IPv6 network, each ViPR Controller node requires a unique, static IPv6 address in the subnet defined by the netmask.

If you are not configuring an IPv6 network, provide a default value of ::0 for each server, which disables IPv6.

This is a mandatory parameter.

-prefix_length|npl

The IPv6 prefix length. Valid values are 1 - 128. This is a mandatory parameter.

-network_gateway6|ng6

Network gateway IPv6.
If you are not configuring settings for an IPv6 network, provide a default value of ::0 which disables IPv6.

This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system cluster-poweroff

Power off the cluster.

Syntax

```
    viprcli system cluster-poweroff
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-portui|pu
<ui_port_number>]
    [-h]
```

Description

`viprcli system cluster-poweroff` command powers off the cluster.

Options

Common Arguments

This operation also takes the Common Parameters on page 24.

```
# viprcli system cluster-poweroff
```

viprcli system cluster-recovery

Triggers ViPR Controller node recovery for ViPR Controller installations on Hyper-V and on VMware without vApp.

Syntax

```
viprcli system cluster-recovery
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
[-h]
```

Description

`viprcli system cluster-recovery` command triggers ViPR Controller node recovery for ViPR Controller installations on Hyper-V and on VMware without vApp.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 216

```
# viprcli system cluster-recovery
Error: Failed to cluster recovery
Reason:'HTTP code: 500, Server Error [ViPR internal server error.
Error details: Trigger node recovery failed. Cause: Cluster is
stable and no need to do node recovery.]'
```

viprcli system cluster-recovery-status

Gets the status of ViPR Controller node recovery for ViPR Controller installations on Hyper-V and on VMware without vApp.

Syntax

```
viprcli system cluster-recovery-status
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
```

```
            [-h]
```

Description

`viprcli system cluster-recovery-status` command gets the status of ViPR Controller node recovery for ViPR Controller installations on Hyper-V and on VMware without vApp.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 217

```
# viprcli system cluster-recovery-status
{
   "status": "NOT_STARTED"
}
```

viprcli system connectemc-ftps

Configure ConnectEMC service with FTPS details.

Syntax

```
            viprcli system connectemc-ftps
            -ftpserver|fsvr
        <ftpserver>
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system connectemc-ftps` command configures the ConnectEMC service with FTPS host name details.

Note

The system automatically reboots after configuring FTPS successfully. Wait for the system to reboot and re-login with your credentials.

Options

ftpserver|fsvr

FTP server details. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 218

```
# viprcli system connectemc-ftps -fsvr corp.mydomain.com
# Broadcast message from root (Mon Jan 12 13:27:30 2014):
The system is going down for reboot NOW!
```

viprcli system connectemc-smtp

Configure ConnectEMC service with SMTP details.

Syntax

```
            viprcli system connectemc-smtp
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -primaryemail|pm
        <primaryemail>
            -smtpserver|sms
        <smtpserver>
            -senderemail|se
        <senderemail>
```

Description

`viprcli system connectemc-smtp` command configures the ConnectEMC service with SMTP details.

Note

The system automatically reboots after configuring SMTP successfully. Wait for the system to reboot and re-login with your credentials.

Options

primaryemail|pm

Primary email ID to contact. This is a mandatory parameter.

smtpserver|sms

SMTP server details. This is a mandatory parameter.

senderemail|se

Sender email ID. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 219

```
# viprcli system connectemc-smtp -sms mydomain.com -pm
user@mydomain.com -se sender@senderdomain.com
# Broadcast message from root (Mon Jan 12 13:32:30 2014):
The system is going down for reboot NOW!
```

viprcli system create-backup

Create a system backup.

Syntax

```
            viprcli system create-backup
            -name|n <backup_name>
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-force]
            [-help|h]
```

Description

The `viprcli system create-backup` command creates a system backup.

Options

name|n

The name of the system backup. This is a mandatory parameter.

Note

An underscore ("_") character is not allowed in the system backup name.

force

Force the creation of a system backup.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system db-consistency-check

Use to trigger a database consistency check.

Syntax

```
    viprcli system db-consistency-check
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
```

Description

The `viprcli system db-consistency-check` command triggers a database consistency check.

Options

common arguments

This operation also takes the Common Parameters on page 24.

viprcli system db-consistency-check-cancel

Use to cancel a database consistency check.

Syntax

```
    viprcli system db-consistency-check-cancel
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
```

Description

The `viprcli system db-consistency-check-cancel` command cancels a database consistency check.

Options

common arguments

This operation also takes the Common Parameters on page 24.

viprcli system db-consistency-check-status

Use to view the status of a database consistency check.

Syntax

```
            viprcli system db-consistency-check-status
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
```

Description

The `viprcli system db-consistency-check-status` command allows you to view the status of a database consistency check.

Options

common arguments

This operation also takes the Common Parameters on page 24.

Example 220

```
viprcli system db-consistency-check-status
{
"end_time": 1452802389405,
"inconsistency_count": 0,
"progress": 100,
"start_time": 1452802387324,
"status": "SUCCESS",
"working_point": "StorageOS/ScopedLabelDbIndex/ScopedTagPrefixIndex"
}
```

viprcli system dbrepair-status

Gets the status of ViPR Controller database repair.

Syntax

```
            viprcli system dbrepair-status
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
```

```
        <cookiefile>]
            [-h]
```

Description

`viprcli system dbrepair-status` command gets the status of ViPR Controller database repair.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 221

```
# viprcli system dbrepair-status
{
   "last_completion_time": "2015-07-01T19:04:32.131Z",
   "progress": "100",
   "start_time": "2015-07-01T18:24:42.025Z",
   "status": "SUCCESS"
}
```

viprcli system delete-backup

Deletes a system backup.

Syntax

```
viprcli system delete-backup
-name|n <backup_name>
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-help|h]
```

Description

The `viprcli system delete-backup` command deletes a system backup.

Options

name|n

The name of the system backup. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system delete-site

Deletes a standby site.

Syntax

```
    viprcli system delete-site
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    -name|n NAME
```

Description

The viprcli system delete-site command deletes a standby site.

Options

name|n

Name of the site. This is a mandatory parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli system delete-sites

Deletes multiple standby sites.

Syntax

```
    viprcli system delete-sites
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -names|ns NAMES [NAMES ...]
```

Description

The viprcli system delete-sites command deletes multiple standby sites.

Options

names|ns

Names of the sites. This is a mandatory parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli system delete-task

Delete a task.

Syntax

```
viprcli system delete-task
[-h]
[-hostname|hn
<hostname>]
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookie_file>]
-taskid|tid <taskid>
```

Description

The viprcli system delete-task command deletes a task. You can use this command when a ViPR Controller task remains in a pending state for two or more days.

Options

taskid|tid

The ID of the task to be deleted. This is a mandatory parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

The following example demonstrates how to delete a task.

```
viprcli system delete-task -tid urn:storageos:Task:f363bf0e-
a589-4fc0-ae62-cd5b2bee266e:vdc1
```

viprcli system disable-update-check

Disable the update check.

Syntax

```
viprcli system disable-update-check
[-hostname|hn <hostname>]
```

```
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookiefile>]
            [-h]
```

Description

`viprcli system disable-update-check` command disables update check.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system download-backup

Downloads a system backup.

Syntax

```
viprcli system download-backup
-name|n <backup_name>
-filepath|fp <download_filepath>
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-help|h]
```

Description

The `viprcli system download-backup` command downloads a system backup.

Options

name|n

The name of the system backup. This is a mandatory parameter.

filepath|fp

The download filepath of the system backup. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system failover-site

Use to perform a failover from a standby site.

Syntax

```
viprcli system failover-site
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -name|n NAME
```

Description

The `viprcli system failover-site` command provides the ability to perform a failover from a site.

Options

name|n

Name of the site. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system get-alerts

Retrieve alerts and save it in a file name of your choice for troubleshooting analysis.

Syntax

```
viprcli system get-alerts
    -filepath|fp <filepath>
    [-severity|sv severity]
    [-start|st <start>]
    [-end|en <end>]
    [-node <nodeID>]
    [-nodename <nodename>]
    [-format|fm <format> { json | xml
| text/plain }]
    [-maxcount|mc
<maxcount>]
    [-regular|regex <regular
expression>]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
```

```
        <cookiefile>]
            [-h]
```

Description

`viprcli system get-alerts` command retrieves and saves the output with the specified file name. This information can be used for troubleshooting activities.

Options

filepath|fp

The name and path of the output file. This is a mandatory parameter.

severity|sv

The minimum alert severity level to display. The default is 7. Valid values are from 0 (FATAL), 4 (ERROR), 5 (WARN), 7 (INFO), 8 (DEBUG), and 9 (TRACE).

start|st

Return entries logged no earlier that this day and time. The day and time should be specified in this format: yyyy-mm-dd_hh:mm:ss.

end|en

Return entries logged no later that this day and time. The day and time should be specified in this format: yyyy-mm-dd_hh:mm:ss.

nodeid

The ViPR Controller node id. You can use either the nodeid or the nodename, you cannot use both.

nodename

The ViPR Controller node name. You can only either the nodeid or the nodename, you cannot use both.

maxcount|mc

Maximum number of log messages to retrieve. This may return more than maximum, if there are more messages with same date as the maximum message.

format|fm

Information is returned in JSON format by default. This option can be set to json, xml or text/plain format.

regular|regex

Regular expression. Sequence of characters that forms a search pattern.

For example, {M,N} - Denotes the minimum M and the maximum N match count.

For example, [...] - Denotes a set of possible character matches.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 222

Example 222 (continued)

```
viprcli system get-log-level
```


```
viprcli system get-log-level
```


```
#viprcli system get-alerts -fp example_file
```

Use the `vi` command to view the file content.

```
#viprcli vi example_file
2013-09-2 10:31:04,000 localhost ALERTS WARN NTP: [UNCONFIGURED].
2013-09-2 10:46:04,000 localhost ALERTS WARN NTP: [UNCONFIGURED].
2013-09-2 11:01:04,000 localhost ALERTS WARN NTP: [UNCONFIGURED].
2013-09-2 11:16:04,000 localhost ALERTS WARN NTP: [UNCONFIGURED].
2013-09-2 11:31:04,000 localhost ALERTS WARN NTP: [UNCONFIGURED].
2013-09-2 11:46:04,000 localhost ALERTS WARN NTP: [UNCONFIGURED].
2013-09-2 12:01:04,000 localhost ALERTS WARN NTP: [UNCONFIGURED].
2013-09-2 12:16:04,000 localhost ALERTS WARN NTP: [UNCONFIGURED].
2013-09-2 12:31:04,000 localhost ALERTS WARN NTP: [UNCONFIGURED].
```

viprcli system get-cluster-state

viprcli includes a utility that returns status information for the ViPR Controller cluster. The cluster state must report as STABLE before you attempt to upgrade to a new version of ViPR Controller.

Syntax

```
viprcli system get-cluster-state
       [-h]
       [-hostname|hn <hostname>]
       [-port|po
   <port_number>]
       [-portui|pu
   <ui_port_number>]
       [-cookiefile|cf
   <cookiefile>]
       [-force|f]
```

Description

viprcli system get-cluster-state returns information about the condition of the current cluster and a list of ViPR Controller versions, as follows:

- CLUSTER_STATE - STABLE, SYNCING, DEGRADED, and UPGRADING.
- CURRENT_VERSION - The currently running ViPR Controller version.
- AVAILABLE_VERSIONS - Bootable versions available in the local ViPR Controller repository.
- REMOVABLE_VERSIONS - older versions that can be deleted from your ViPR Controller cluster.

- NEW_VERSIONS - New versions are available for downloading from a remote repository.

Options

force|f

Force viprcli to show all removable versions. By default, `viprcli system get-cluster-state` command shows the removable versions only when the number of available versions in the local repository reaches maximum of 4. The -f switch displays all removable versions even if the number of available (bootable) images is less than 4.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 223

```
# viprcli system get-cluster-state

STATE_INFORMATION
  CLUSTER_STATE CURRENT_VERSION  AVAILABLE_VERSIONS
REMOVABLE_VERSIONS
  STABLE        vipr-1.1.0.6.277 vipr-1.1.0.6.252    None
                                 vipr-1.1.0.6.262
                                 vipr-1.1.0.6.277
   NEW_VERSIONS
   vipr-1.1.0.7.462
   vipr-1.1.0.7.461
   vipr-1.1.0.7.460
   vipr-1.1.0.7.459
```

viprcli system get-diagnostics

Returns the diagnostics for the specified nodes.

Syntax

```
            viprcli system get-diagnostics
            [-node <nodeid>]
            [-nodename <nodename>]
            [-verbose|v <verbose>]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system get-diagnostics` returns diagnostics for the specified nodes.

Options

node

The ViPR Controller node id. You can use either the nodeid or the nodename, you cannot use both.

nodename

The ViPR Controller node name. You can only either the nodeid or the nodename, you cannot use both.

Verbose|v

List the information with more details.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 224

```
# viprcli system get-diagnostics -v
{
   "node_diagnostics_list": {
      "node_diagnostics": [
         {
            "ip": "192.168.98.224",
            "node_id": "syssvc-1",
            "tests": {
               "test": [
                  {
                     "name": "Network interface",
                     "status": "OK"
                  },
                  {
                     "name": "Network routing",
                     "status": "OK"
                  },
                  {
                     "name": "DNS",
                     "status": "OK"
                  },
                  {
                     "name": "NTP",
                     "status": "OK"
                  },
                  {
                     "name": "Remote repository",
                     "status": "CONFIGURED,UNREACHABLE"
                  },
                  {
                     "name": "Disk space",
                     "status": "ZK_FUL"
                  },
                  {
                     "name": "Memory usage",
                     "status": "OK"
                  },
                  {
                     "name": "Cpu usage",
                     "status": "OK"
                  },
                  {
                     "name": "Nodes connectivity",
```

Example 224 (continued)

```
                    "status": "REACHABLE"
                },
                {
                    "name": "Network vip",
                    "status": "CONFIGURED, REACHABLE"
                },
                {
                    "name": "Firewall",
                    "status": "CONFIGURED, RUNNING"
                },
                {
                    "name": "IP subnets",
                    "status": "SAME"
                }
            ]
        }
    },
    {
        "ip": "192.168.98.238",
        "node_id": "syssvc-2",
        "tests": {
            "test": [
                {
                    "name": "Network interface",
                    "status": "OK"
                },
                {
                    "name": "Network routing",
                    "status": "OK"
                },
                {
                    "name": "DNS",
                    "status": "OK"
                },
                {
                    "name": "NTP",
                    "status": "OK"
                },
                {
                    "name": "Remote repository",
                    "status": "CONFIGURED,UNREACHABLE"
                },
                {
                    "name": "Disk space",
                    "status": "ZK_FUL"
                },
                {
                    "name": "Memory usage",
                    "status": "OK"
                },
                {
                    "name": "Cpu usage",
                    "status": "OK"
                },
                {
                    "name": "Nodes connectivity",
                    "status": "REACHABLE"
                },
                {
                    "name": "Network vip",
                    "status": "CONFIGURED, REACHABLE"
                },
                {
                    "name": "Firewall",
                    "status": "CONFIGURED, RUNNING"
```

Example 224 (continued)


```
                    },
                    {
                       "name": "IP subnets",
                       "status": "SAME"
                    }
                 ]
              }
           },
           {
              "ip": "192.168.98.254",
              "node_id": "syssvc-3",
              "tests": {
                 "test": [
                    {
                       "name": "Network interface",
                       "status": "OK"
                    },
                    {
                       "name": "Network routing",
                       "status": "OK"
                    },
                    {
                       "name": "DNS",
                       "status": "OK"
                    },
                    {
                       "name": "NTP",
                       "status": "OK"
                    },
                    {
                       "name": "Remote repository",
                       "status": "CONFIGURED,UNREACHABLE"
                    },
                    {
                       "name": "Disk space",
                       "status": "ZK_FUL"
                    },
                    {
                       "name": "Memory usage",
                       "status": "OK"
                    },
                    {
                       "name": "Cpu usage",
                       "status": "OK"
                    },
                    {
                       "name": "Nodes connectivity",
                       "status": "REACHABLE"
                    },
                    {
                       "name": "Network vip",
                       "status": "CONFIGURED, REACHABLE"
                    },
                    {
                       "name": "Firewall",
                       "status": "CONFIGURED, RUNNING"
                    },
                    {
                       "name": "IP subnets",
                       "status": "SAME"
                    }
                 ]
              }
           }
        ]
```

Example 224 (continued)

```
    }
}
```

viprcli system get-esrsconfig

Returns the EMC Secure Remote Support (ESRS) configuration details.

Syntax

```
viprcli system get-esrsconfig
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

viprcli system get-esrsconfig returns the ESRS configuration details.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 225

```
 # viprcli system get-esrsconfig
{
   "ip_address": "192.168.1.2",
   "model-name": "ViPR-GM",
   "serial-no": "VG9SSD7YX7200G-3"
}
```

viprcli system get-health

Return the health of nodes and services.

Syntax

```
viprcli system get-health
[-nodeid|id <nodeID>]
[-nodename <nodename>]
[-hostname|hn <hostname>]
```

```
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

viprcli system get-health returns the health of nodes and services.

Options

nodeid

The ViPR Controller node id. You can use either the nodeid or the nodename, you cannot use both.

nodename

The ViPR Controller node name. You can only either the nodeid or the nodename, you cannot use both

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 226

viprcli system get-license

Return the license details.

Syntax

```
            viprcli system get-license
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system get-license` command displays the content of a license file.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 227

Example 227 (continued)

```
# viprcli system get-license
{'license_feature': [{'product': 'abcd', 'licensed_axm':
'true', 'site_id': 'UNKNOWN', 'notice': 'ACTIVATED TO SiteID:
PTA08AUG2008229174', 'issued_date': '03/01/2013', 'version': '1.0',
'license_id_indicator': 'U', 'model': 'ViPR_Controller',
'expired_ind': 'false', 'serial': 'AV123456678VA',
'storage_capacity': '5607509301657600'}, {'product': 'AB34568PWA7',
'licensed_ind': 'true', 'site_id': 'UNKNOWN', 'noticce':
'ACTIVATED TO SiteID: PTA08AUG2008229174', 'issued_date':
'03/01/2013',
'version': '1.0', 'license_id_indicator': 'U', 'model':
'ViPR_Object',
'expired_ind': 'false', 'serial': 'AV123456678VA',
'storage_capacity':
'5607509301657600'}], 'license_text':
'##################################
#############################\r\n# EMC License File\r\n# Activation
Date:
  Mar 01, 2013 11:41:34 AM\r\n# Activated By:  xxxx xxxx\r\n#
Type:UNSERVED\r
\n#######################################################
```

viprcli system get-log-level

Return the logging levels.

Syntax

```
            viprcli system get-log-level
            [-logs|lg <logs>
        [<logs>...]]
            [-nodes|nds <nodes>
        [<nodes>...]]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system get-log-level` returns the logging levels for the nodes and services with severity information.

Options

logs|lg

One of the following:

- dbsvc
- authsvc
- sasvc

- apisvc
- controllersvc
- geodbsvc
- geosvc
- coordinatorsvc
- portalsvc
- syssvc

nodes|nds

The name of the nodes. Nodes are named according to the following pattern:

```
syssvc-node*n*
```

Where *n* is the number of the node. For example, for a 3-node ViPR Controller cluster, the nodes are named vipr1 , vipr2 , and vipr3.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 228

Default

```
# viprcli system get-log-level
{
   "levels": [
      {
         "node": "vipr1",
         "service": "dbsvc",
         "severity": "INFO"
      },
      {
         "node": "vipr1",
         "service": "authsvc",
         "severity": "INFO"
      },
      {
         "node": "vipr1",
         "service": "sasvc",
         "severity": "INFO"
      },
      {
         "node": "vipr1",
         "service": "apisvc",
         "severity": "INFO"
      },
      {
         "node": "vipr1",
         "service": "controllersvc",
         "severity": "INFO"
      },
      {
         "node": "vipr1",
         "service": "geodbsvc",
         "severity": "INFO"
      },
```

Example 228 (continued)

```
        {
           "node": "vipr1",
           "service": "geosvc",
           "severity": "INFO"
        },
        {
           "node": "vipr1",
           "service": "coordinatorsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr1",
           "service": "portalsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr1",
           "service": "syssvc",
           "severity": "INFO"
        },
        {
           "node": "vipr3",
           "service": "dbsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr3",
           "service": "authsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr3",
           "service": "sasvc",
           "severity": "INFO"
        },
        {
           "node": "vipr3",
           "service": "apisvc",
           "severity": "INFO"
        },
        {
           "node": "vipr3",
           "service": "controllersvc",
           "severity": "INFO"
        },
        {
           "node": "vipr3",
           "service": "geodbsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr3",
           "service": "geosvc",
           "severity": "INFO"
        },
        {
           "node": "vipr3",
           "service": "coordinatorsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr3",
           "service": "portalsvc",
           "severity": "INFO"
        },
```

Example 228 (continued)

```
        {
           "node": "vipr3",
           "service": "syssvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "dbsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "authsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "sasvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "apisvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "controllersvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "geodbsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "geosvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "coordinatorsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "portalsvc",
           "severity": "INFO"
        },
        {
           "node": "vipr2",
           "service": "syssvc",
           "severity": "INFO"
        }
    ]
}
```

Example 229

Reset to Error

Using `viprcli system set-log-level` command.

Example 229 (continued)

```
# viprcli system get-log-level
{
   "levels": [
      {
         "node": "standalone",
         "service": "apisvc",
         "severity": "ERROR"
      },
      {
         "node": "standalone",
         "service": "authsvc",
         "severity": "ERROR"
      },
      {
         "node": "standalone",
         "service": "controllersvc",
         "severity": "ERROR"
      },
      {
         "node": "standalone",
         "service": "geodbsvc",
         "severity": "ERROR"
      },
      {
         "node": "standalone",
         "service": "geosvc",
         "severity": "ERROR"
      },
      {
         "node": "standalone",
         "service": "dbsvc",
         "severity": "ERROR"
      },
      {
         "node": "standalone",
         "service": "objcontrolsvc",
         "severity": "ERROR"
      },
      {
         "node": "standalone",
         "service": "portalsvc",
         "severity": "ERROR"
      },
      {
         "node": "standalone",
         "service": "sasvc",
         "severity": "ERROR"
      },
      {
         "node": "standalone",
         "service": "syssvc",
         "severity": "ERROR"
      }
   ]
}
```

viprcli system get-logs

Log information is generated by the ViPR Controller appliance starting at initial startup and initialization, and can be used for troubleshooting and other system management activities.

Syntax

```
    viprcli system get-logs
    -filepath|fp
<filepath_and_filename>
    [-log|lg <logname>]
    [-severity|sv
<severity>]
    [-start|st
<startdate_time>]
    [-end|en
<enddate_time>]
    [-node <node_id>]
    [-nodename <node_name>]
    [-format|fm <format> {json |
xml}]
    [-maxcount|mc
<maxcount>]
    [-regular|regex <regular
expression>]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-h]
```

Description

The output from the viprcli system get-logs operation can be JSON output or XML based output. You can choose the output type by specifying the -xml option. The information retrieved can be extensive, and must be saved in a local file.

Options

filepath|fp

The name and path of the output file. This is a mandatory parameter.

log|lg

One of the following:

- vasasvc
- cassvc
- coordinatorsvc
- dbsvc
- apisvc
- controllersvc
- syssvc

severity|sv

The minimum log severity level to display. The default is 7. Valid values are from 0 (FATAL), 4 (ERROR), 5 (WARN), 7 (INFO), 8 (DEBUG), and 9 (TRACE).

start|st

Return entries logged no earlier that this day and time. The day and time should be specified in this format: yyyy-mm-dd_hh:mm:ss or milliseconds.

end|en

Return entries logged no later that this day and time. The day and time should be specified in this format: yyyy-mm-dd_hh:mm:ss or milliseconds.

nodeid

The ViPR Controller node id. You can use either the nodeid or the nodename, you cannot use both.

nodename

The ViPR Controller node name. You can only use either the nodeid or the nodename, you cannot use both.

maxcount|mc

Maximum number of log messages to retrieve. This may return more than maximum, if there are more messages with same date as the maximum message.

format|fm

Information is returned in JSON format by default. This option can be set to json or xml format.

regular|regex

Regular expression. Sequence of characters that forms a search pattern. For example:

- {M,N} - Denotes the minimum M and the maximum N match count.
- [...] - Denotes a set of possible character matches.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 230

```
# viprcli system get-logs -fp /tmp/LogFile1.log -fm xml
```

Use vi command to view the log file.

```
# vi /tmp/LogFile1.log
<log>
        <node>          vipr1</node>
        <severity>      INFO</severity>
        <thread>        qtp1306283693-7317</thread>
        <message>       Auth headers: Auth Token: no - Proxy Token:
no - Basic Auth: no</message>
```

Example 230 (continued)

```
        <service>       syssvc</service>
        <time>          2015-06-16 17:30:31,959</time>
        <line>          77</line>
        <class>         RequestAuditFilter</class>
</log>
<log>
        <node>          vipr1</node>
        <severity>      INFO</severity>
        <thread>        Thread-2</thread>
        <message>       Removing engine: db05af2c-27c4-4f49-
bf4a-5d73d99a92dc</message>
        <service>       sasvc</service>
        <time>          2015-06-16 17:30:33,132</time>
        <line>          152</line>
        <class>         ExecutionEngineMonitor</class>
</log>
```

viprcli system get-properties

Return the system component properties as a key-value pair.

Syntax

```
viprcli system get-properties
 [-type|t {ovf,config,mutated,obsolete,secrets,all}]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli system get-properties` returns system component properties as a key-value pair.

Options

type|t

The configuration property type. Valid options include `ovf`, `config`, `mutated`, `obsolete`, `secrets`, and `all`.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 231 Get all system properties

```
# viprcli system get-properties
{
   "properties": {
      "entry": [
```

Example 231 Get all system properties (continued)

```
        {
           "key": "backup_external_location_password",
           "value": ""
        },
        {
           "key": "backup_external_location_url",
           "value": ""
        },
        {
           "key": "backup_external_location_username",
           "value": ""
        },
        {
           "key": "backup_max_used_disk_percentage",
           "value": "30"
        },
        {
           "key": "backup_scheduler_copies_to_keep",
           "value": "5"
        },
        {
           "key": "backup_scheduler_enable",
           "value": "true"
        },
        {
           "key": "backup_scheduler_interval",
           "value": "1day"
        },
        {
           "key": "backup_scheduler_time",
           "value": "100"
        },
        {
           "key": "backup_threshold_disk_percentage",
           "value": "80"
        },
        {
           "key": "certificate_version",
           "value": "0"
        },
        {
           "key": "compute_aix_version",
           "value": "6.1.0"
        },
        {
           "key": "compute_aixvio_version",
           "value": "2.0.0"
        },
        {
           "key": "compute_redhat_linux_version",
           "value": "5.9"
        },
        {
           "key": "compute_suse_linux_version",
           "value": "11"
        },
        {
           "key": "compute_vmware_vcenter_version",
           "value": "5.0"
        },
        {
           "key": "compute_windows_version",
           "value": "6.0.6002"
        },
        {
```

Example 231 Get all system properties (continued)

```
            "key": "config_version",
            "value": "1427340337241"
         },
         {
            "key": "controller_brocade_firmware_version",
            "value": "11.2.1"
         },
         {
            "key": "controller_compute_discovery_interval",
            "value": "3600"
         },
         {
            "key": "controller_cs_discovery_interval",
            "value": "86400"
         },
         {
            "key": "controller_datadomain_firmware_version",
            "value": "5.5.0.0"
         },
         {
            "key": "controller_ddmc_firmware_version",
            "value": "1.2.1.0"
         },
         {
            "key": "controller_discovery_core_pool_size",
            "value": "3"
         },
         {
            "key": "controller_discovery_interval",
            "value": "3600"
         },
         {
            "key": "controller_enable_autodiscovery",
            "value": "true"
         },
         {
            "key": "controller_enable_autoscan",
            "value": "true"
         },
         {
            "key": "controller_enable_metering",
            "value": "true"
         },
         {
            "key": "controller_enable_monitoring",
            "value": "true"
         },
         {
            "key": "controller_hicommand_provider_version",
            "value": "7.4"
         },
         {
            "key": "controller_ibmxiv_firmware_version",
            "value": "10.2.4"
         },
         {
            "key": "controller_ibmxiv_provider_version",
            "value": "10.2.4.0"
         },
         {
            "key": "controller_isilon_firmware_version",
            "value": "7.0.2.0"
         },
         {
            "key": "controller_max_pool_utilization_percentage",
```

Example 231 Get all system properties (continued)

```
            "value": "75"
         },
         {
            "key":
"controller_max_thin_pool_subscription_percentage",
            "value": "300"
         },
         {
            "key": "controller_mds_communication_timeout",
            "value": "300"
         },
         {
            "key": "controller_mds_firmware_version",
            "value": "5.0(1a)"
         },
         {
            "key": "controller_metering_interval",
            "value": "3600"
         },
         {
            "key": "controller_netapp_firmware_version",
            "value": "8.1.1"
         },
         {
            "key": "controller_ns_discovery_interval",
            "value": "3600"
         },
         {
            "key": "controller_ns_zone_refresh_always",
            "value": "true"
         },
         {
            "key": "controller_rp_firmware_version",
            "value": "4.0"
         },
         {
            "key": "controller_scaleio_firmware_version",
            "value": "1.21.0.20"
         },
         {
            "key": "controller_scan_interval",
            "value": "600"
         },
         {
            "key": "controller_smis_provider_version",
            "value": "4.6.1.1"
         },
         {
            "key": "controller_ucs_firmware_version",
            "value": "2.0"
         },
         {
            "key": "controller_vmax_firmware_version",
            "value": "5876.221"
         },
         {
            "key": "controller_vnxblock_firmware_version",
            "value": "05.32"
         },
         {
            "key": "controller_vnxfile_firmware_version",
            "value": "7.1.71"
         },
         {
            "key": "controller_vplex_firmware_version",
```

Example 231 Get all system properties (continued)

```
            "value": "5.2"
         },
         {
            "key": "controller_xtremio_firmware_version",
            "value": "2.4.0.20"
         },
         {
            "key": "image_server_address",
            "value": ""
         },
         {
            "key": "image_server_encpassword",
            "value": ""
         },
         {
            "key": "image_server_http_port",
            "value": "44491"
         },
         {
            "key": "image_server_image_directory",
            "value": "images/"
         },
         {
            "key": "image_server_image_import_timeout",
            "value": "1800"
         },
         {
            "key": "image_server_job_polling_interval",
            "value": "60"
         },
         {
            "key": "image_server_os_install_timeout",
            "value": "3600"
         },
         {
            "key": "image_server_os_network_ip",
            "value": ""
         },
         {
            "key": "image_server_ssh_port",
            "value": "22"
         },
         {
            "key": "image_server_ssh_timeout",
            "value": "20"
         },
         {
            "key": "image_server_tftpboot_directory",
            "value": "/opt/tftpboot/"
         },
         {
            "key": "image_server_username",
            "value": "root"
         },
         {
            "key": "internal_api_key_algo",
            "value": "sha-1"
         },
         {
            "key": "network_gateway",
            "value": "10.247.39.1"
         },
         {
            "key": "network_gateway6",
            "value": "::0"
```

Example 231 Get all system properties (continued)

```
        },
        {
           "key": "network_nameservers",
           "value": "10.254.66.23,10.254.66.24"
        },
        {
           "key": "network_netmask",
           "value": "255.255.255.0"
        },
        {
           "key": "network_ntpservers",
           "value": "10.254.140.22"
        },
        {
           "key": "network_prefix_length",
           "value": "64"
        },
        {
           "key": "network_standalone_ipaddr",
           "value": "10.247.39.169"
        },
        {
           "key": "network_standalone_ipaddr6",
           "value": "::0"
        },
        {
           "key": "network_vip",
           "value": "0.0.0.0"
        },
        {
           "key": "network_vip6",
           "value": "::0"
        },
        {
           "key": "node_count",
           "value": "1"
        },
        {
           "key": "password_change_interval",
           "value": "60"
        },
        {
           "key": "password_changed_character",
           "value": "2"
        },
        {
           "key": "password_dictionary_rule",
           "value": "no"
        },
        {
           "key": "password_expire_days",
           "value": "0"
        },
        {
           "key": "password_history_rule",
           "value": "3"
        },
        {
           "key": "password_lowercase_alphabet",
           "value": "1"
        },
        {
           "key": "password_min_length",
           "value": "8"
        },
```

Example 231 Get all system properties (continued)

```
        {
           "key": "password_numeric_character",
           "value": "1"
        },
        {
           "key": "password_repeating_character",
           "value": "3"
        },
        {
           "key": "password_special_character",
           "value": "1"
        },
        {
           "key": "password_uppercase_alphabet",
           "value": "1"
        },
        {
           "key": "root_id_dsa",
           "value": "********"
        },
        {
           "key": "root_id_ecdsa",
           "value": "********"
        },
        {
           "key": "root_id_rsa",
           "value": "********"
        },
        {
           "key": "ssh_config_version",
           "value": "1427340200219"
        },
        {
           "key": "ssh_host_dsa_key",
           "value": "********"
        },
        {
           "key": "ssh_host_ecdsa_key",
           "value": "********"
        },
        {
           "key": "ssh_host_rsa_key",
           "value": "********"
        },
        {
           "key": "storageos_id_dsa",
           "value": "********"
        },
        {
           "key": "storageos_id_ecdsa",
           "value": "********"
        },
        {
           "key": "storageos_id_rsa",
           "value": "********"
        },
        {
           "key": "svcuser_id_dsa",
           "value": "********"
        },
        {
           "key": "svcuser_id_ecdsa",
           "value": "********"
        },
        {
```

Example 231 Get all system properties (continued)

```
            "key": "svcuser_id_rsa",
            "value": "********"
        },
        {
            "key": "system_connectemc_encrypt",
            "value": "yes"
        },
        {
            "key": "system_connectemc_ftps_fepfolder",
            "value": "Default"
        },
        {
            "key": "system_connectemc_ftps_hostname",
            "value": "corpusfep3.emc.com"
        },
        {
            "key": "system_connectemc_ftps_password",
            "value": "Default"
        },
        {
            "key": "system_connectemc_ftps_port",
            "value": "990"
        },
        {
            "key": "system_connectemc_ftps_username",
            "value": "Default"
        },
        {
            "key": "system_connectemc_smtp_authtype",
            "value": "None"
        },
        {
            "key": "system_connectemc_smtp_emcto",
            "value": "emailalert@emc.com"
        },
        {
            "key": "system_connectemc_smtp_enabletls",
            "value": "no"
        },
        {
            "key": "system_connectemc_smtp_enabletlscert",
            "value": "no"
        },
        {
            "key": "system_connectemc_smtp_from",
            "value": ""
        },
        {
            "key": "system_connectemc_smtp_password",
            "value": ""
        },
        {
            "key": "system_connectemc_smtp_port",
            "value": "0"
        },
        {
            "key": "system_connectemc_smtp_server",
            "value": ""
        },
        {
            "key": "system_connectemc_smtp_to",
            "value": ""
        },
        {
            "key": "system_connectemc_smtp_username",
```

Example 231 Get all system properties (continued)


```
            "value": ""
         },
         {
            "key": "system_connectemc_transport",
            "value": "None"
         },
         {
            "key": "system_enable_firewall",
            "value": "yes"
         },
         {
            "key": "system_login_banner",
            "value": "You are accessing ViPR. By using this system
you consent to the owning organization's terms and conditions."
         },
         {
            "key": "system_permit_root_console",
            "value": "no"
         },
         {
            "key": "system_permit_root_ssh",
            "value": "no"
         },
         {
            "key": "system_proxyuser_encpassword",
            "value": "AUvFbY+4ueYC6kQJaVLEVn4="
         },
         {
            "key": "system_root_authorizedkeys2",
            "value": ""
         },
         {
            "key": "system_root_encpassword",
            "value": "$6$JPjDj/4RsQiB2y1y$GdrNX5qfCh.rhCx/
X8ras2Q2RMyVUF/5I92SzEDzblO.LYqf7DdX7Zv.3jcNsGKwN.UUS6882121DO/
HVwUJ4."
         },
         {
            "key": "system_root_expiry_date",
            "value": ""
         },
         {
            "key": "system_svcuser_authorizedkeys2",
            "value": ""
         },
         {
            "key": "system_svcuser_encpassword",
            "value": "$6$LYJU22kpKYrKR3ZR$djQL0caLuM/
Y0Zu4PoG0aSSpdqrL85dkqFNSvKqOmslCTU7N0mI4iTu10foFqug9dVr7D2Y7VmAFpoR
C07E9C0"
         },
         {
            "key": "system_svcuser_expiry_date",
            "value": ""
         },
         {
            "key": "system_sysmonitor_encpassword",
            "value": "$6$6CDtFLu14q8/MH2o
$RTy7c9bFkje.lXGv0d4.29TPMixdRel92JnjLGJWLTFe0paihOF/
PshQoYK7ln5LfcBujPSU7VAwmuJWE.mkb."
         },
         {
            "key": "system_update_check_frequency_hours",
            "value": "24"
         },
```

Example 231 Get all system properties (continued)

```
         {
            "key": "system_update_password",
            "value": ""
         },
         {
            "key": "system_update_proxy",
            "value": ""
         },
         {
            "key": "system_update_repo",
            "value": "https://colu.emc.com/soap/rpc"
         },
         {
            "key": "system_update_username",
            "value": ""
         },
         {
            "key": "task_clean_interval",
            "value": "60"
         },
         {
            "key": "task_ttl",
            "value": "10080"
         },
         {
            "key": "vasa_base_url",
            "value": ""
         },
         {
            "key":
"vcenter_das_admissionControlPolicy_hostFailureLevel_failoverLevel",
            "value": "3"
         },
         {
            "key": "vcenter_das_admissionControlPolicy_policy",
            "value": "resourcePercentage"
         },
         {
            "key":
"vcenter_das_admissionControlPolicy_resourcePercentage_cpuPercentage
",
            "value": "25"
         },
         {
            "key":
"vcenter_das_admissionControlPolicy_resourcePercentage_memoryPercent
age",
            "value": "75"
         },
         {
            "key": "vcenter_das_admissionControl_enabled",
            "value": "true"
         },
         {
            "key": "vcenter_das_enabled",
            "value": "true"
         },
         {
            "key": "vcenter_das_hostMonitoring_enabled",
            "value": "true"
         },
         {
            "key": "vcenter_das_vmSettings_isolationResponse",
            "value": "powerOff"
         },
```

Example 231 Get all system properties (continued)

```
            {
               "key": "vcenter_das_vmSettings_restartPriority",
               "value": "high"
            },
            {
               "key": "vcenter_das_vmSettings_vmMonitoring_monitoring",
               "value": "vmAndAppMonitoring"
            },
            {
               "key":
"vcenter_das_vmSettings_vmMonitoring_sensitivity",
               "value": "medium"
            },
            {
               "key": "vcenter_drs_automationLevel",
               "value": "fullyAutomated"
            },
            {
               "key": "vcenter_drs_dpm_dpmThreshold",
               "value": "3"
            },
            {
               "key": "vcenter_drs_dpm_enabled",
               "value": "true"
            },
            {
               "key": "vcenter_drs_dpm_powerManagement",
               "value": "automated"
            },
            {
               "key": "vcenter_drs_enabled",
               "value": "true"
            },
            {
               "key": "vcenter_drs_migrationThreshold",
               "value": "3"
            },
            {
               "key": "vcenter_host_operation_timeout",
               "value": "450"
            },
            {
               "key": "vcenter_operation_timeout",
               "value": "150"
            },
            {
               "key": "vcenter_swapfile_policy",
               "value": "vmDirectory"
            }
         ]
      }
}
```

viprcli system get-properties-metadata

Return the metadata information for system properties.

Syntax

```
            viprcli system get-properties-metadata
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system get-properties-metadata` command returns the metadata information for system properties.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 232

```
# viprcli system get-properties-metadata | more
{
   "metadata": {
      "entry": [
         {
            "key": "config_version",
            "value": {
               "controlNodeOnly": "false",
               "description": "Internal configuration version
number.",
               "hidden": "true",
               "label": "Configuration version number",
               "type": "uint64",
               "userConfigurable": "false",
               "userMutable": "false",
               "value": "0"
            }
         },
         {
            "key": "controller_discovery_interval",
            "value": {
               "advanced": "true",
               "controlNodeOnly": "true",
               "description": "Number of seconds between discovery
              operations of storage systems.",
               "label": "Storage Systems",
               "rebootRequired": "true",
               "reconfigRequired": "false",
```

Example 232 (continued)


```
                "tag": "Discovery",
                "type": "uint32",
                "userConfigurable": "false",
                "userMutable": "true",
                "value": "3600"
             }
          },
          {
             "key": "controller_enable_autodiscovery",
             "value": {
                "advanced": "true",
                "allowedValues": [
                   "true",
                   "false"
                ],
                "controlNodeOnly": "true",
                "description": "Indicates whether auto-discovery is
                enabled.",
                "label": "Enable Auto-Discovery",
                "rebootRequired": "true",
                "reconfigRequired": "false",
                "tag": "Discovery",
                "type": "string",
                "userConfigurable": "false",
                "userMutable": "true",
                "value": "true"
             }
          },
          {
             "key": "controller_enable_autoscan",
             "value": {
                "advanced": "true",
                "allowedValues": [
                   "true",
                   "false"
                ],
                "controlNodeOnly": "true",
                "description": "Indicates whether auto-scan of
                 SMI-S providers is enabled.",
                "label": "Enable Auto-Scan",
                "rebootRequired": "true",
                "reconfigRequired": "false",
                "tag": "Discovery",
                "type": "string",
                "userConfigurable": "false",
                "userMutable": "true",
                "value": "true"
             }},
          {
             "key": "system_root_authorizedkeys2",
             "value": {
                "advanced": "true",
                "controlNodeOnly": "false",
                "description": "Authorized public SSH keys for the
local 'root' account.",
                "label": "SSH Key (root)",
                "maxLen": "8192",
                "rebootRequired": "false",
                "reconfigRequired": "true",
                "tag": "Security",
                "type": "string",
                "userConfigurable": "false",
                "userMutable": "true",
                "value": "ssh-dss AAAAB3NzaC1kc3MAAACBAMa1xQiTbZY+
                 LnfqVGq/uWi7u382tKS+OePsKOpKzW61CaiLCDza2s9ZM7g
```

Example 232 (continued)

```
                 JErPxLI4hc5st1Y03YnndtK1yHlMn9IlTeh3nfC4nrmKobEdwtq9

DiLYluZXLtS1brrBxcPt8viMSPKvAc5gzyzleF5jrxKRfi4fwUKUAP1/

XYY6pAAAAFQCK0mGCn60PmABaVsOoMAMIw2cZNQAAAIBRUUYm5cZsc/

i1rQj0bYIEDynuZofxuEF9Oh6ubMJVdDsqJQJEknkPvMVQxig9Jke
                lQp8l0arL4FczLJ/nRLAN8eSbSsm7mYiQ3eKAluDRGK472hLMa3J
                 /nfgf1cdLmmNeifjBP+VCyXpO+vwtmE2eivzKOyNZUK+
                 QpB6ovM3YbwAAAIEAuT4vXRb68MJ
+IdRZPshMrmaLGFlb5ZdTnXTchp
                 /Ay5KAsM2XuoYBpiFJsiNp7gPFtWv/vLyGYFMNCBrF/
                 oPigRgEgu09z5DMZpo2JsWg025g55qbJjyz3gYb+SoBJrw4DTaGh
                 SDDOnnOsr88lbKOFKSEN3uR0IS+jsySrBpztXY=bourne.dsa"
            }},}}
```

viprcli system get-stats

Return the system statistics.

Syntax

```
            viprcli system get-stats
            [-node <node>]
            [-nodename <nodename>]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system get-stats` returns the complete system statistics.

Options

node

The ViPR Controller node id. You can use either the nodeid or the nodename, you cannot use both.

nodename

The ViPR Controller node name. You can only either the nodeid or the nodename, you cannot use both.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 233

Example 233 (continued)

```
# viprcli system get-stats
{
   "node_stats_list": {
      "node_stats": [
         {
            "data_disk_stats": {
               "data_available_kb": "488585380",
               "data_used_kb": "1259900",
               "root_available_kb": "8352992",
               "root_used_kb": "1432256"
            },
            "disk_io_stats_list": {
               "disk_io_stats": [
                  {
                     "avg_svctime": "0.08",
                     "avg_wait": "0.16",
                     "disk_id": "sdb",
                     "reads_per_second": "0.0",
                     "sectors_read_per_second": "0.04",
                     "sectors_write_per_second": "12.29",
                     "util_perc": "0.01",
                     "writes_per_second": "0.79"
                  },
                  {
                     "avg_svctime": "2.27",
                     "avg_wait": "4.25",
                     "disk_id": "sda",
                     "reads_per_second": "0.01",
                     "sectors_read_per_second": "1.61",
                     "sectors_write_per_second": "0.0",
                     "util_perc": "0.0",
                     "writes_per_second": "0.0"
                  },
                  {
                     "avg_svctime": "0.4",
                     "avg_wait": "0.69",
                     "disk_id": "sdc",
                     "reads_per_second": "0.01",
                     "sectors_read_per_second": "0.63",
                     "sectors_write_per_second": "34.66",
                     "util_perc": "0.09",
                     "writes_per_second": "2.27"
                  }
               ]
            },
            "ip": "192.168.98.224",
            "load_avg": {
               "load_avg_last_fifteen_minutes": "0.4",
               "load_avg_last_five_minutes": "0.52",
               "load_avg_last_one_minute": "0.79"
            },
            "memory": {
               "free_memory_in_kb": "116348",
               "memory_buffers_in_kb": "149784",
               "total_memory_in_kb": "4057564"
            },
            "node_id": "syssvc-1",
            "service_stats_list": {
               "service_stats": [
                  {
                     "command": "java",
                     "file_descriptors_ctr": "167",
                     "name": "apisvc",
                     "status": {
                        "active_threads_ctr": "52",
```

Example 233 (continued)


```
                "resident_mem_in_pages": "48440",
                "start_time_seconds": "21",
                "total_program_size_in_pages": "235312",
                "total_uptime_seconds": "2725",
                "virtual_mem_size_in_bytes": "963837952",
                "virtual_mem_size_in_pages": "48440"
            }
        },
        {
            "command": "java",
            "file_descriptors_ctr": "158",
            "name": "authsvc",
            "status": {
                "active_threads_ctr": "41",
                "resident_mem_in_pages": "34543",
                "start_time_seconds": "21",
                "total_program_size_in_pages": "214823",
                "total_uptime_seconds": "2646",
                "virtual_mem_size_in_bytes": "879915008",
                "virtual_mem_size_in_pages": "34543"
            }
        },
        {
            "command": "java",
            "file_descriptors_ctr": "159",
            "name": "controllersvc",
            "status": {
                "active_threads_ctr": "518",
                "resident_mem_in_pages": "117642",
                "start_time_seconds": "21",
                "total_program_size_in_pages": "492750",
                "total_uptime_seconds": "2863",
                "virtual_mem_size_in_bytes": "2018304000",
                "virtual_mem_size_in_pages": "117642"
            }
        },
        {
            "command": "java",
            "file_descriptors_ctr": "88",
            "name": "coordinatorsvc",
            "status": {
                "active_threads_ctr": "39",
                "resident_mem_in_pages": "28977",
                "start_time_seconds": "21",
                "total_program_size_in_pages": "211625",
                "total_uptime_seconds": "701",
                "virtual_mem_size_in_bytes": "866816000",
                "virtual_mem_size_in_pages": "28977"
            }
        },
        {
            "command": "java",
            "file_descriptors_ctr": "467",
            "name": "dbsvc",
            "status": {
                "active_threads_ctr": "215",
                "resident_mem_in_pages": "319095",
                "start_time_seconds": "21",
                "total_program_size_in_pages": "531452",
                "total_uptime_seconds": "25302",
                "virtual_mem_size_in_bytes": "2176827392",
                "virtual_mem_size_in_pages": "319095"
            }
        },
        {
```

Example 233 (continued)

```
                    "command": "java",
                    "file_descriptors_ctr": "172",
                    "name": "syssvc",
                    "status": {
                       "active_threads_ctr": "108",
                       "resident_mem_in_pages": "60056",
                       "start_time_seconds": "21",
                       "total_program_size_in_pages": "649493",
                       "total_uptime_seconds": "2795",
                       "virtual_mem_size_in_bytes": "2660323328",
                       "virtual_mem_size_in_pages": "60056"
                    }
                 },
                 {
                    "command": "java",
                    "file_descriptors_ctr": "109",
                    "name": "vasasvc",
                    "status": {
                       "active_threads_ctr": "26",
                       "resident_mem_in_pages": "20133",
                       "start_time_seconds": "21",
                       "total_program_size_in_pages": "337455",
                       "total_uptime_seconds": "579",
                       "virtual_mem_size_in_bytes": "1382215680",
                       "virtual_mem_size_in_pages": "20133"
                    }
                 }
              ]
```

viprcli system get-storage

Return the statistical information of the underlying storage type in kilo bytes.

Syntax

```
viprcli system get-storage
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli system get-storage` returns the statistics of the underlying storage types in kilo bytes.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 234

Example 234 (continued)

```
 # viprcli system get-storage
{
   "controller": {
      "block_managed_capacity_kb": "636672.0",
      "file_managed_capacity_kb": "0.0",
      "free_managed_capacity_kb": "5.7235582366E11"
   },
   "object": {
      "capacity_kb": "0.0"
   }
}
```

viprcli system get-target-version

Displays the current version of ViPR Controller.

Syntax

```
viprcli system get-target-version
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

viprcli system get-target-version displays the current version of ViPR Controller.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 235

```
# viprcli system get-target-version
 TARGET_VERSION
 vipr-1.1.0.7.331
```

viprcli system install-image

Install a new ViPR Controller image from the remote ViPR Controller repository.

Syntax

```
viprcli system install-image
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-force|f]
    -v <target_version>
```

Description

`viprcli system install-image` command installs the specified target version from a remote repository. The version to install must appear in the list of NEW versions when you run `viprcli system get-cluster-state`. This command is silent on success.

Note

To install an image from a local repository use `viprcli system upload` command.

Options

-force|f

Force installs an older image. This is an optional parameter.

-v

The target version of ViPR Controller to install. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 236

Run the install image command for the version of your choice.

```
# viprcli system install-image vipr-1.1.0.7.462
```

viprcli system ipreconfig-status

Get the status of the reconfiguration of the IPs of the ViPR Controller nodes for Hyper-V and non-vApp platforms in a single Virtual Data Center (VDC) configuration.

Syntax

```
viprcli system ipreconfig-status
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
[-h]
```

Description

`viprcli system ipreconfig-status` command gets the status of the reconfiguration of the IPs of the ViPR Controller nodes for Hyper-V and non-vApp platforms.

Note

This command is for single Virtual Data Center (VDC) configurations only. Multiple Virtual Data Center (VDC) configurations are not supported.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 237 (continued)

```
# viprcli system ipreconfig-status
No IP Reconfigure Status
```

viprcli system list-backup

Lists a system backup.

Syntax

```
viprcli system list-backup
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-verbose|v]
[-help|h]
```

Description

The `viprcli system list-backup` command lists a system backup.

Options

verbose|v

List the system backups, with details.

Note

The verbose output is not always different than the non-verbose output.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system list-external-backup

Use to view a list of external backups.

Syntax

```
viprcli system
list-external-backup
[-h]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
```

Description

The `viprcli system list-external-backup` command allows you to view a list of external backups.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 238

```
viprcli system list-external-backup
{ "backups": [ "3node-1-3-3-42d59320-c4a1-11e5-6367-
c50bc875e6e3.zip", "aaa-3-3-c5d99f70-c599-11e5-
cb41-1a6665efa303.zip", "helen-0129-3-3-1a16ca30-c62c-11e5-6367-
c50bc875e6e3.zip", "helen-1-3-3-1a16ca30-c62c-11e5-6367-
c50bc875e6e3.zip", "helen123-3-3-c5d99f70-c599-11e5-
cb41-1a6665efa303.zip", "hyperv-more-data-3-3-cedeffe0-c4d4-11e5-
acd2-f845ebd4c8a3.zip", "hyperv-rare-data-3-2-cedeffe0-c4d4-11e5-
acd2-f845ebd4c8a3.zip", "hyperv-rare-data-3-3-8a81d5f0-
c4a9-11e5-57dd-82c694771b63.zip", "hyperv-rare-data-3-3-cedeffe0-
c4d4-11e5-acd2-f845ebd4c8a3.zip", "mmm-3-3-c5d99f70-c599-11e5-
cb41-1a6665efa303.zip", "new-0128-3-3-fda07580-c582-11e5-6367-
c50bc875e6e3.zip", "vipr-2.4.1.0.220-3-20160124050000-3-3.zip",
"vipr-2.5.0.0.34e3605-3-20160129022128-3-3-1a16ca30-c62c-11e5-6367-
c50bc875e6e3.zip",
"vipr-2.5.0.0.34e3605-3-20160129023914-3-3-1a16ca30-c62c-11e5-6367-
c50bc875e6e3.zip",
"vipr-2.5.0.0.34e3605-3-20160129093604-3-3-1a16ca30-c62c-11e5-6367-
c50bc875e6e3.zip",
"vipr-2.5.0.0.34e3605-3-20160130010001-3-3-1a16ca30-c62c-11e5-6367-
c50bc875e6e3.zip",
"vipr-2.5.0.0.34e3605-3-20160131010002-3-3-1a16ca30-c62c-11e5-6367-
c50bc875e6e3.zip",
"vipr-2.5.0.0.34e3605-3-20160201010002-3-3-1a16ca30-c62c-11e5-6367-
c50bc875e6e3.zip",
"vipr-2.5.0.0.8a83749-3-20160127030901-3-3-42d59320-c4a1-11e5-6367-
c50bc875e6e3.zip",
"vipr-2.5.0.0.8a83749-3-20160127041144-3-3-8a81d5f0-
c4a9-11e5-57dd-82c694771b63.zip",
"vipr-2.5.0.0.8a83749-3-20160127091736-3-3-cedeffe0-c4d4-11e5-acd2-
f845ebd4c8a3.zip", "vipr-2.5.0.0.8a83749-3-20160127100033-3-3-
cedeffe0-c4d4-11e5-acd2-f845ebd4c8a3.zip",
"vipr-2.5.0.0.8a83749-3-20160128010002-3-3-42d59320-c4a1-11e5-6367-
c50bc875e6e3.zip", "vipr-2.5.0.0.8a83749-3-20160128010002-3-3-
cedeffe0-c4d4-11e5-acd2-f845ebd4c8a3.zip",
"vipr-2.5.0.0.8a83749-3-20160128030718-3-3-411998f0-c56a-11e5-acd2-
f845ebd4c8a3.zip", "vipr-2.5.0.0.8a83749-3-20160128033424-3-3-
cedeffe0-c4d4-11e5-acd2-f845ebd4c8a3.zip",
"vipr-2.5.0.0.b84c538-3-20160128060428-3-2-fda07580-c582-11e5-6367-
c50bc875e6e3.zip", "vipr-2.5.0.0.ec51e5d-3-20160128084644-3-3-
c5d99f70-c599-11e5-cb41-1a6665efa303.zip",
"vipr-2.5.0.0.ec51e5d-3-20160128095539-3-3-533f59f0-c5a3-11e5-6367-
c50bc875e6e3.zip",
"vipr-2.5.0.0.ec51e5d-3-20160129010002-3-3-533f59f0-c5a3-11e5-6367-
c50bc875e6e3.zip", "vipr-2.5.0.0.ec51e5d-3-20160129010002-3-3-
c5d99f70-c599-11e5-cb41-1a6665efa303.zip",
"vipr-2.5.0.0.ec51e5d-3-20160130010002-3-3-c5d99f70-c599-11e5-
cb41-1a6665efa303.zip", "vipr-2.5.0.0.ec51e5d-3-20160131010001-3-3-
c5d99f70-c599-11e5-cb41-1a6665efa303.zip",
"vipr-2.5.0.0.ec51e5d-3-20160201010002-3-3-c5d99f70-c599-11e5-
cb41-1a6665efa303.zip", "vmware-3-3-cedeffe0-c4d4-11e5-acd2-
f845ebd4c8a3.zip" ] }
```

viprcli system list-sites

Lists the ViPR Controller standby sites configured for ViPR Controller disaster recovery.

Syntax

```
            viprcli system list-site
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-verbose|v]
            [-long|l]
```

Description

The `viprcli system list-sites` command lists the ViPR Controller standby sites configured for ViPR Controller disaster recovery.

Options

verbose|v

List sites with details. This is an optional parameter.

long|l

List sites in table. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 239

```
viprcli system list-sites
vipr1:/opt/storageos/cli # viprcli system list-sites
NAME
standby2
standby3
vipr1:/opt/storageos/cli # viprcli system list-sites -l
NAME VIP STATE
standby2 10.247.142.211 STANDBY_SYNCED
standby3 10.247.142.205 ACTIVE
vipr1:/opt/storageos/cli # viprcli system list-sites -v
[
{
"description": "stand by 2",
"global": null,
"name": "standby2",
"remote": null,
"state": "STANDBY_SYNCED",
"tags": [],
"uuid": "257afbb0-bab2-11e5-b4f9-172b1b96685c",
```

Example 239 (continued)

```
"vdc": null,
"vdc_id": "vdc1",
"vip": "10.247.142.211"
},
{
"description": "",
"global": null,
"name": "standby3",
"remote": null,
"state": "ACTIVE",
"tags": [],
"uuid": "d35222d0-ba9f-11e5-b4f9-172bed838f97",
"vdc": null,
"vdc_id": "vdc1",
"vip": "10.247.142.205"
}
]
```

viprcli system pause-site

Use to pause a standby site.

Syntax

```
viprcli system pause-site
[-h]
[-hostname <hostname>]
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-name|n NAME
```

Description

The `viprcli system pause-site` command pauses a standby sites.

Options

name|n

Name of the site. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system pause-sites

Use to pause multiple standby sites.

Syntax

```
        viprcli system pause-sites
        [-h]
        [-hostname <hostname>]
        [-port|po
    <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        -names|ns [NAMES...]
```

Description

The `viprcli system pause-sites` command pauses multiple standby sites.

Options

names|ns

Names of the sites. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 240

```
viprcli system pause-sites -ns standby2 standby3
```

viprcli system pull-backup

Use to pull a specific backup.

Syntax

```
        viprcli system pull-backup
        [-h]
        [-hostname <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        -name|n <backup name>
```

```
            [-force]
```

Description

The `viprcli system pull-backup` command allows you to pull a specific backup.

Options

name|n

Name of the backup. This is a mandatory parameter.

force

Pull backup forcibly. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 241

```
viprcli system pull-backup -name vmware-3-3-cedeffe0-c4d4-11e5-acd2-
f845ebd4c8a3.zip
```

viprcli system pull-backup-cancel

Use to cancel the pull of a specific backup.

Syntax

```
            viprcli system
     pull-backup-cancel
            [-h]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
```

Description

The `viprcli system pull-backup-cancel` command allows you to cancel the pull of a specific backup.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system query-backup

Use to view information about a ViPR backup set.

Syntax

```
viprcli system query-backup
[-h]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-name|n <backup name>
```

Description

The `viprcli system query-backup` command provides information about a ViPR backup set.

Options

name|n

Name of the backup. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system query-backup-info

Use to view information about a ViPR backup set.

Syntax

```
viprcli system
query-backup-info
[-h]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-name|n <backup name>
[-islocal|isl]
```

Description

The `viprcli system query-backup-info` command allows you to view information about a ViPR backup set.

Options

name|n

Name of the backup. This is a mandatory parameter.

islocal|isl

Specifies whether the backup is on a local or an external server. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system reboot-node

Reboot a specific node.

Syntax

```
    viprcli system reboot-node
    id <node_id>
    nodename <node_name>
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-portui|pu
<ui_port_number>]
    [-h]
```

Description

`viprcli system reboot-node` command re-boots the specified node.

The user must accept the confirmation to re-boot the node.

Options

id

The ViPR Controller node id. You can use either the nodeid or the nodename, you cannot use both.

nodename

The ViPR Controller node name. You can use either the nodeid or the nodename, you cannot use both.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 242

Example 242 (continued)

```
# viprcli system reboot-node -id vasasvc-node3
Do you really want to reboot node:vasasvc-node3(y/n)?:
n
```

viprcli system remove-image

Removes a ViPR Controller image from your local repository.

Syntax

```
viprcli system remove-image
-v <version_to_remove>
[-f]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

The version to remove must appear in the list of removable versions when you run viprcli system get-cluster-state.

```
# viprcli system get-cluster-state

STATE_INFORMATION
CLUSTER_STATE          STABLE
CURRENT_VERSION        vipr-1.1.0.6.277
AVAILABLE_VERSIONS     vipr-1.1.0.6.277, vipr-1.1.0.6.278,
vipr-1.1.0.5.123
REMOVABLE_VERSIONS     vipr-1.1.0.5.123
```

Options

-v

The version of ViPR Controller to install. This is a mandatory parameter.

-f

The -f option allows you to remove images that are listed under AVAILABLE_VERSIONS when you run the command viprcli system get-cluster-state. Without this option you can only remove those images listed as REMOVABLE_VERSIONS. The ViPR Controller cluster's currently running version cannot be removed.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 243

Example 243

```
# viprcli system remove-image -v vipr-1.1.0.6.277 -f
# viprcli system get-cluster-state

STATE_INFORMATION
  CLUSTER_STATE CURRENT_VERSION  AVAILABLE_VERSIONS
REMOVABLE_VERSIONS
  STABLE        vipr-1.1.0.6.278 vipr-1.1.0.6.278   None
```

viprcli system reset-properties

Reset the specified key-value pair to the default value.

Syntax

```
viprcli system reset-properties
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
[-force|f <force option>]
-propertyfile|pf <propertyfile>
```

Description

`viprcli system reset-properties` resets the specified key-value pair to the default value for the specified keys in the properties file. The properties file to reset should contain only the keys and not any values associated with it. This command is silent on success.

Options

force|f

Force option. This is an optional parameter.

propertyfile|pf

Name of the property file from which to reset the properties. This is a mandatory parameter.
You can edit the properties file to include only those keys for which you want to reset to default.

Note

In this sample `props.cfg` file here you can observe only keys are mentioned and not values for the key.

```
# vi props.cfg
system_update_proxy
system_update_username
```

```
~
~
~
:qw
```

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 244

```
viprcli system reset-properties -pf props.cfg
```

Use the get-properties command to view the results.

viprcli system restart-service

Restart a specified service.

Syntax

```
            viprcli system restart-service
            id <nodeid>
            nodename <nodename>
            -servicename|svc
        <servicename>
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system restart-service` command restarts a specified service.

The user must accept the confirmation to restart the service.

Options

id

The ViPR Controller node id. You can use either the id or the nodename, you cannot use both.

nodename

The ViPR Controller node name. You can use either the nodeid or the nodename, you cannot use both.

servicename|svc

One of the following services in ViPR Controller:

- vasasvc
- cassvc
- coordinatorsvc
- dbsvc
- apisvc
- controllersvc
- syssvc

This is a mandatory argument.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 245

```
# viprcli system restart-service -id vasasv-node1 -svc vasasvc
Do you really want to restart service:vasasvc in node: vasasv-
node1(y/n)?:
n
```

viprcli system restore-backup

Use to restore a backup set.

Syntax

```
            viprcli system restore-backup
            [-h]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <backup name>
            -islocal|isl <islocal>
```

Description

The `viprcli system restore-backup` command restores a backup set.

Options

name|n

Name of the backup. This is a mandatory parameter.

islocal|isl

Indicates whether the backup is on a local or an external server. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 246

```
viprcli system restore-backup -name abc.zip -islocal false
```

viprcli system restore-backup-status

Use to view the status of a backup set restore operation.

Syntax

```
        viprcli system
restore-backup-status
        [-h]
        [-hostname <hostname>]
        [-port|po <port_number>]
        [-portui|pu
   <ui_port_number>]
        [-cookiefile|cf
   <cookiefile>]
        -name|n <backup name>
        -islocal|isl <islocal>
```

Description

The `viprcli system restore-backup-status` command allows you to view the status of a backup set restore operation.

Options

name|n

Name of the backup. This is a mandatory parameter.

islocal|isl

Indicates whether the backup is on a local or an external server. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 247

```
viprcli system restore-backup-status -name abc.zip -islocal false
```

viprcli system resume-site

Use to resume a standby site.

Syntax

```
viprcli system resume-site
[-h]
[-hostname <hostname>]
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-name|n NAME
```

Description

The `viprcli system resume-site` command resumes a standby site.

Options

name|n

Name of the site. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system retry-site

Use to perform a retry operation on a standby site.

Syntax

```
viprcli system retry-site
[-h]
[-hostname <hostname>]
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-name|n NAME
```

Description

The `viprcli system retry-site` command provides the ability to perform a retry operation on a standby site.

Options

name|n

Name of the site. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system send-alert

Send an alert to SYR database.

Syntax

```
viprcli system send-alerts
[-severity|sv <severity>]
[-start|st <starttime>]
[-end|en <endtime>]
[-node|nd nodeID]
[-format|fm <format> { json | xml | native | padded }]
[-maxcount|mc <maxcount>]
[-source|src <targetversion>]
[-eventid|eid <event_id>]
[-message|msg <message>]
[-contact|ct <contact>]
[-log|lg <logname>]
[-regular|regex <regular expression>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli system send-alert` sends an alerts to SYR database.

Options

severity|sv

The minimum alert severity level to display. The default is 7. Valid values are from 0 (FATAL), 4 (ERROR), 5 (WARN), 7 (INFO), 8 (DEBUG), and 9 (TRACE).

start|st

Return entries logged no earlier that this day and time. The day and time should be specified in this format: yyyy-mm-dd_hh:mm:ss.

end|en

Return entries logged no later that this day and time. The day and time should be specified in this format: yyyy-mm-dd_hh:mm:ss.

node|nd

The name of the node. Nodes are named according to the following pattern:

```
syssvc-node*n*
```

Where *n* is the number of the node.

For example, for a 3-node ViPR Controller configuration, the nodes are named syssvc-node1 , syssvc-node2 , and syssvc-node3.

maxcount|mc

Maximum number of log messages to retrieve. This may return more than maximum, if there are more messages with same date as the maximum message.

format|fm

Information is returned in JSON format by default. This option can be set to json, xml, native, or padded format.

regular|regex

Regular expression. Sequence of characters that forms a search pattern.

For example, {M,N} - Denotes the minimum M and the maximum N match count.

For example, [...] - Denotes a set of possible character matches.

source|src

Target version.

eventid|eid

Event ID of the system event.

message|msg

Event message.

contact|ct

Contact information.

log|lg

Name of the log.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 248

```
 # viprcli system send-alert
{'op_id': '1f80e511-d77c-4446-8497-9f116b3b12fb', 'resource':
{'id': 'urn:storageos:SysEvent:a56bfb85-7df9-4bd3-b73e-
b529ad1407e6:'},
'description': 'Send alert event', 'state': 'pending'}
```

viprcli system send-heartbeat

Send the system heart beat to indicate that the system is in working condition.

Syntax

```
viprcli system send-heartbeat
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli system send-heartbeat` sends the system heart beat indicating that the system is in working condition. This command is silent on success.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system send-registration

Send system registration details.

Syntax

```
viprcli system send-registration
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli system send-registration` sends the system registration details. This command is silent on success.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 249

Example 249 (continued)

```
# viprcli system send-registration
#
```

viprcli system set-log-level

Configure the logging level.

Syntax

```
            viprcli system set-log-level
            [-severity|sv <severity>]
            [-logs|lg <logs> [<logs>
        ...] Logs Name]
            [-nodes|nds <nodes> [<nodes>
        ...] Nodes]
            [-expiretime|ext
        <expiretime>]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli system set-log-level` command configures the logging levels. This command is silent on success.

Options

severity|sv

The minimum severity level to set. The default is 7. Valid values are from 0 (FATAL), 4 (ERROR), 5 (WARN), 7 (INFO), 8 (DEBUG), and 9 (TRACE). This is an optional parameter.

logs|lg

One of the following:

- vasasvc
- cassvc
- coordinatorsvc
- dbsvc
- apisvc
- controllersvc
- syssvc

nodes|nds

The name of the node. Nodes are named according to the following pattern:

```
syssvc-node*n*
```

Where *n* is the number of the node. For example, for a 3-node Software Defined Storage cluster, the nodes are named syssvc-node1 , syssvc-node2 , and syssvc-node3.

expiretime|ext

Log level expiration time, in minutes.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 250

```
 # viprcli system set-log-level -lg apisvc controllersvc
coordinatorsvc sasvc portalsvc -sv 8 -ext 30
```

viprcli system set-properties

Update the key-value pair with the specified values in the properties file.

Syntax

```
            viprcli system set-properties
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
            [-propertyfile|pf
        <propertyfile>]
            [-propertyname|pn
        <propertyname>]
            [-propertyvaluefile|pvf
        <propertyvaluefile>]
```

Description

`viprcli system set-properties` updates the key-value properties with specified values in the properties file. You can use viprcli system get-properties to get the full list of key-value properties that you can set. In addition, viprcli system get-properties-metadata returns the full list of properties and their metadata, such as description, data type, and values.

Note

Do not mix propertyfile|pf parameter with propertyname|pn and propertyvaluefile|pvf parameters when you run the command. Use either one of the options as described in this section.

Options

propertyfile|pf

Name of the property file from which to set the properties. This is an optional parameter. You can edit the properties file to include only those keys for which you want to update the values.

Note

In this sample `props.cfg` file here you can observe both keys and their values are mentioned as a pair.

```
# vi props.cfg
system_update_proxy=update_proxy
system_update_username=update_username
~
~
~
~
~
~
~
:qw
```

propertyname|pn

Name of the property. This is an optional parameter.

Note

Use this propertyname|pn parameter only in conjunction with propertyvaluefile|pf parameter. This option is usefully especially when there is a multi-line property value to be set. However, it can be used for a single-line property value setting as well.

propertyvaluefile|pvf

Path of the property file which contains the multi-line value to be set. This is an optional parameter. For example, the `ssl_cert.pem` file contains a multi-line value as show here:


```
-----BEGIN PRIVATE KEY-----MIIEowIBAAKCAQEAwOFcPlfxn5Zj3VPk
+1vJ5qVPi5siT9Ok6aQ5ZYRtwXir+axbfOltNCuh6TtwDTVp+9X/
0ERpg2Qp0tc20qkPwaT/
+oYKHlhpQuVX4lQIDAQABAoIBAG8TGNj6MjZSGA6EWlHY4Gayxuge32aqblywFxa
VYtfu7IlvzWvSL7mI3z
+9gJ7NsrmBncAkPKhuz57NAUOIOHvrYO5EOOsDCBdtmfvY9chObe4bgIk6b0kFg/
8XH2lJJRSYomv66fskY
+8qWpTWhIsCuGoZQffIsMtZtYno1mwkJb0m8r6st9EJO/
MzcoI1dEafQSGFmtm90GAQdgXtWKvAK72yDSm2
+2BgfVwdvqJvwqSw5EHVVmvjRMwjzb+/
Yxjl4jCmTlX5VXxl8BXLlWCfXYZk3BzfQHsxXPsaMOP9OfaowlB
+dgktcOmrhp323A9qJ7GM00TP6LHDcywQKBgQC6IFOl33YkKjaX2kRZTRV8xNa5a
```

```
Bk/lI2CvA1rKt3jlndb
+5GmuEvRsXHjbb28pbRWwNmlRcVB0vkQKBgH9NMSC8hlhSVPhN1uHGsORXvGxnl3
pTFaaIuJCRFUCMIncbP
-----END PRIVATE KEY----------BEGIN CERTIFICATE-----
MIIEZzCCA0+gAwIBAgIQUS735fP
+V/GflmPdU+RXCPLUx4qzGUp0ZXtUpigbKaMtAQ34Awf0OM/bzz
+VUtXdNdoBGaATdzp8clcpcIoiMO7Cqog
+jOWeWnVfTWvKAs9492MivWVui/IJqmH
+ABhQzMblLMOtBQ25kVsYsFRZlOPjIKvzNTFh2/qDJaJ
+oJ2TADEnzoEaIQ65hXIIh5X6hgoeWGlC5VfiRTgMBAAGjgY4wgYswHwYDVR0RBB
gwFoIUbGdsdzIwNjIub
+BAIwADA7BgNVHR8ENDAyMDCgLqAshipodHRwOi8vY3JsLnRoYXd0ZS5jb20vVGh
hd3RlVHJpYWxTU0xDQS
+LUAgPG4lzruf07rdwL2HK6uqtRghF7cFVfBisBtMW8lsrzqw8XADi
+ej2YAQluXkdWXtT6C5j0rrpye8Qx
+y7SZzh0i4QXiRaC3SYPQp2MsWfSzKipnq8updnwf82oYCfPsrVc9gU1en1Wrthr
x1REAz0DAYYCERuM174
-----END CERTIFICATE
```

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 251

```
# viprcli system set-properties -pf props.cfg
```

Use the get-properties command to view the results.

Example 252

```
# viprcli system set-properties -pn system_ssl_cert_pem -pvf
ssl_cert.pem
```

Use the get-properties command to view the results.

Example 253

```
./viprcli system set-properties -propertyname strong_ciphers -
propertyvaluefile propvalue
```

You can adjust the ViPR Controller installation to use SHA-256 encryption with this command where the file, `propvalue`, contains the new value to be set.

You will need to create a property values file with a comma-separated list of the encryption algorithms you want ViPR Controller to support, for example,

```
TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA
```

Example 253 (continued)

.

Note

Both AES-128 and SHA-256 encryption algorithms are supported by default.

---

viprcli system show-site

Shows site details.

Syntax

```
viprcli system show-site
[-h]
[-hostname <hostname>]
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
-name|n
```

Description

The `viprcli system show-site` command shows site details.

Options

name|n

Name of the site. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 254

```
viprcli system show-site -n "Default Active Site"
{
"global": null,
"name": "Default Active Site",
"remote": null,
"state": "ACTIVE",
"tags": [],
"uuid": "bb2607b0-ba06-11e5-b4f9-172b1b96685c",
"vdc": null,
"vdc_id": "vdc1",
"vip": "10.247.142.211"
}
```

viprcli system site-error

Use to query the latest error message for a specific standby site.

Syntax

```
            viprcli system site-error
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n
```

Description

The `viprcli system site-error` command queries the latest error message for a specific standby site.

Options

name|n

Name of the site. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 255

```
viprcli system site-error -n standby4
{'creationTime': 1452754387978, 'errorMessage': 'Failed to resume
standby. Standby site is still in RESUMING state after 20 minutes',
'serviceCodeName': 'SYS_DR_RESUME_STANDBY_TIMEOUT', 'serviceCode':
308}
viprcli system pause-sites -ns standby2 standby3
viprcli system show-site -n standby2
{
"description": "",
"global": null,
"name": "standby2",
"remote": null,
"state": "STANDBY_PAUSING",
"tags": [],
"uuid": "b94efdc0-ba06-11e5-b4f9-172b559b2d7a",
"vdc": null,
"vdc_id": "vdc1",
"vip": "10.247.142.224"
}
vipr1:/opt/storageos/cli # viprcli system show-site -n standby3
{
"description": "",
"global": null,
"name": "standby3",
"remote": null,
```

Example 255 (continued)

```
"state": "STANDBY_PAUSING",
"tags": [],
"uuid": "b9716ce0-ba95-11e5-b4f9-172bed838f97",
"vdc": null,
"vdc_id": "vdc1",
"vip": "10.247.142.205"
}
```

viprcli system site-time

Use to query the transition times for a specific standby site.

Syntax

```
    viprcli system site-time
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -name|n NAME
```

Description

The `viprcli system site-time` command queries the transition times for a specific standby site.

Options

name|n

Name of the site. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system skip-setup

Enables execution of ViPR Controller UI catalog services without initial setup.

Syntax

```
    viprcli system skip-setup
    [-hostname <hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
```

```
        <cookiefile>]
            [-h]
```

Description

`viprcli system skip-setup` command enables execution of ViPR Controller UI catalog services without initial setup.

Options

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

```
# viprcli system skip-setup
Enter password of the Proxy User:
Retype password:
```

viprcli system switchover-site

Use to switch over to a target new active site.

Syntax

```
            viprcli system switchover-site
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n NAME
```

Description

The `viprcli system switchover-site` command provides the ability to switch over to a target new active site.

Options

name|n

Name of the site. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system update-cluster

Upgrades ViPR Controller to a newer version.

Syntax

```
viprcli system update-cluster
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-force|f]
    -version|v
<target_version>
```

Description

`viprcli system update-cluster` command updates to the specified version. The version to which you are updating your cluster must appear in the list of available versions when you run `viprcli system get-cluster-state`.

Note

Before performing an upgrade, be sure to review the pre-upgrade planning, and post-upgrade requirements and steps described in the "Upgrading ViPR Controller," chapter of the *ViPR Controller Installation, Upgrade, and Maintenance Guide*, which is available from the ViPR Controller Product Documentation Index.

Options

force|f

Version numbers are not verified before the update is executed. This is an optional parameter.

version|v

The version of ViPR Controller to install. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 256

```
viprcli system update-cluster -v vipr-1.1.0.6.278

Broadcast message from root (Thu Jul 18 14:41:28 2013):
The system is going down for reboot NOW!

### After Reboot ###
```

Example 256 (continued)

viprcli system update-site

Updates a site on the system.

Syntax

```
            viprcli system update-site
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n
            [-newname <name>]
            [-description|desc]
```

Description

The `viprcli system add-site` command updates a site on the system.

Options

name|n

Name of the site. This is a mandatory parameter.

newname

New name of storage provider. This is an optional parameter.

description|desc

Description of site. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 257

```
viprcli system update-site -n standby2 -desc "New desc"
viprcli system update-site -n standby2 -newname standby4
viprcli system show-site -n standby4
{
"description": "New desc",
"global": null,
"name": "standby4",
"remote": null,
"state": "STANDBY_SYNCED",
"tags": [],
"uuid": "b94efdc0-ba06-11e5-b4f9-172b559b2d7a",
"vdc": null,
"vdc_id": "vdc1",
```

Example 257 (continued)

```
"vip": "10.247.142.224"
}
```

viprcli system upload

Install a ViPR Controller image file from a local repository.

Syntax

```
viprcli system upload
-imagefile|if
<image_file>
[-port|po
<port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookiefile>]
[-h]
```

Description

`viprcli system upload` command installs a ViPR Controller image file only from a local repository or folder. `img` is the only supported image file type. This command is silent on success.

Note

To install from a remote repository use `viprcli system install-image` command.

Options

-imagefile|if

The complete path of the ViPR Controller image file to install. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system upload-backup

Use to upload a backup

Syntax

```
viprcli system upload-backup
[-h]
```

```
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <backup name>
```

Description

The `viprcli system upload-backup` command uploads a backup.

Options

name|n

Name of the backup. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli system upload-backup-status

Use to view the status of an upload backup operation.

Syntax

```
            viprcli system
    upload-backup-status
            [-h]
            [-hostname <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <backup name>
```

Description

The `viprcli system upload-backup-status` command allows you to view the status of an upload backup operation.

Options

name|n

Name of the backup. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 258

Example 258 (continued)

```
viprcli system upload-backup-status -name kumar1
{
"allItems": {
"backupName": "kumar1",
"errorCode": "FTP_NOT_CONFIGURED",
"progress": "0",
"status": "FAILED"
},
"backup_name": "kumar1",
"error_code": "FTP_NOT_CONFIGURED",
"progress": 0,
"status": "FAILED"
}
```

CHAPTER 45

viprcli task commands

This chapter contains the following topics:

- viprcli task 566

viprcli task

Check the status of an asynchronous task. Use this command to roll back or resume an order for a task that has been suspended.

Syntax

```
            viprcli task
            [-h]
[-hostname <hostname>]
[-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-task_id|tid <task_id>]{rollback,resume}
```

Description

The `viprcli task` command lists task objects for respective domain objects. for example, if you issue a CLI command to run migration, you can run `viprcli volume tasks` to obtain a list of tasks associated with the volume(s) being migrated. Then you can use the `viprcli task` command with the `{resume|rollback}` arguments on any task that is in the `suspend_no_error` state.

Options

hostname|hn

The ViPR Controller hostname (fully qualified domain name) or IPv4 address. For example, 192.0.2.0 or a ViPR Controller IPv6 address inside quotes and brackets. For example, "[2001:db8::1]"

port|po

The ViPR Controller port number.

portui|pu

The https port number of ViPR Controller Portal UI.

cookiefile|cf

The ViPR Controller cookie file full name.

task_id|tid

An operation ID for a domain's task object. You need the task ID if you want to resume or roll back a task that is in the `suspended_no_error` state. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 259

Example 259 (continued)

```
viprcli task -h -hostname <hostname> -port <port_number> -portui
<ui_port_number> -cf <cookiefile> -tid <task_id>
{rollback,resume}
```

CHAPTER 46

viprcli tenant commands

This chapter contains the following topics:

- viprcli tenant add-attribute 570
- viprcli tenant add-group 570
- viprcli tenant add-namespace 571
- viprcli tenant add-role 572
- viprcli tenant create 573
- viprcli tenant delete-role 575
- viprcli tenant delete 576
- viprcli tenant get-clusters 577
- viprcli tenant get-hosts 578
- viprcli tenant get-role 578
- viprcli tenant get-vcenters 579
- viprcli tenant list 580
- viprcli tenant list-object-namespaces 581
- viprcli tenant remove-attribute 582
- viprcli tenant show 582
- viprcli tenant show-object-namespaces 583
- viprcli tenant update-quota 584

viprcli tenant add-attribute

Map an Active Directory attribute to a ViPR Controller tenant.

Syntax

```
viprcli tenant add-attribute
-domain <domain>
[-name|n <tenantname>]
[-key <keyname>]
[-value <value>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli tenant add-attribute` command is used to assign an Active Directory attribute to a ViPR Controller tenant.

Options

domain

The domain of the Active Directory user. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is a mandatory parameter.

key

An Active Directory key to map to the tenant. This is an optional parameter.

value

The value in Active Directory corresponding to the specified Key. This is an optional parameter.

name|n

The name of a tenant. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli tenant add-group

Map an Active Directory user or group to a ViPR Controller tenant.

Syntax

```
viprcli tenant add-group
-group <group>
-domain <domain>
[-name|n <tenantname>]
[-hostname|hn <hostname>]
```

```
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookiefile>]
            [-h]
```

Description

`viprcli tenant add-group` command is used to assign an Active Directory user or group to a tenant. Do this by mapping the user or group field in your Active Directory to the current tenant.

Options

domain

The domain of the Active Directory user. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is a mandatory parameter.

group

An Active Directory group to map to the tenant. This is a mandatory parameter.

name|n

The name of a tenant. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli tenant add-namespace

Add an Elastic Cloud Storage namespace to a tenant.

Syntax

```
            viprcli tenant create
            -name|n <tenantname>
            -namespace <namespace>
            -description
        <description>
            [-hostname|hn <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli tenant add-namespace` command maps a new Elastic Cloud Storage namespace to the tenant. Namespaces are used in environments where ViPR Controller is managing object storage. A ViPR Controller user that is a member of this tenant must also be a user in the Elastic Cloud Storage namespace being added, in order to create a bucket as a member of this ViPR Controller tenant.

Options

name|n

A valid tenant name. This is a mandatory parameter.

namespace

The name of the namespace to map to the tenant. This is a mandatory parameter.

description

The description of the namespace. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 260

```
# viprcli tenant add-namespace -name "Provider Tenant" -description
"New name" -namespace esc
```

viprcli tenant add-role

Assigns a new role to a user in a tenant.

Syntax

```
            viprcli tenant add-role
            -name|n <tenantname>
            role|r {TENANT_ADMIN PROJECT_ADMIN}
        [{TENANT_ADMIN, PROJECT_ADMIN} ...]

            {[-group|g <group>] | [-subject-id|sb
        <subjectid>]}
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli tenant add-role` command adds a specified role to a user in a tenant while maintaining all previous role assignments.

Options

name|n

A valid ViPR Controller tenant name. This is a mandatory parameter.

role|r

One or more of the following:

- TENANT_ADMIN
- PROJECT_ADMIN

This is a mandatory parameter.

group|g

The group argument specifies a named group of users. Typically, this is the name of a group in Active Directory. You must specify either a subject ID or a group and not both.

subject-id|sb

The Active Directory ID. You must specify either a subject ID or a group and not both.

common arguments

This operation also takes the Common Parameters on page 24.

Example 261

```
viprcli tenant add-role -r PROJECT_ADMIN -subject-id testuser -n
"Provider Tenant"
```

viprcli tenant create

Create a subtenant with the given parameters.

Syntax

```
            viprcli tenant create
            -name|n <tenantname>
            -domain <domain>
            [-key <keyname>]
            [-value <value>]
            [-namespace <namespace>]
            [-hostname|hn <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli tenant create` command creates a subtenant. Subtenants are used heavily in environments where ViPR Controller is managing object storage. The users mapped to a subtenant are imported from Active Directory.

Note

Subtenants can be created only under provider tenant (root) and cannot be created under user created tenants.

Options

name|n

A valid tenant name. This is a mandatory parameter.

domain

The Active Directory user domain. This is a mandatory parameter.

key

An Active Directory key to map to the tenant. This is an optional parameter.

value

The value in Active Directory corresponding to the specified Key. This is an optional parameter.

namespace

The name of the namespace to map to the tenant. This parameter is used where ViPR Controller is managing object storage. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 262

```
# viprcli tenant create -name CLI -key OU
-value ASD -domain <domain>

# viprcli tenant show -name CLI
{
   "attribute": [
      {
         "key": "ou",
         "value": "cli-ind"
      }
   ],
   "id": "urn:storageos:TenantOrg:8a5d0eab-03e2-470f-8f35-
c4c37e0fbf1a:",
   "inactive": false,
   "link": {
      "href": "/tenants/urn:storageos:TenantOrg:
      8a5d0eab-03e2-470f-8f35-c4c37e0fbf1a:",
      "rel": "self"
   },
   "name": "CLI",
   "parent_tenant": {
      "id": "urn:storageos:TenantOrg:
       0823b1ef-5132-4278-a7e4-28053e643b2b:",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:
         0823b1ef-5132-4278-a7e4-28053e643b2b:",
         "rel": "self"
      }
   }
}
```

viprcli tenant delete-role

Remove a role from a user or group of users.

Syntax

```
    viprcli tenant delete-role
    -name|n <tenantname>
    role|r {TENANT_ADMIN,
PROJECT_ADMIN}
    [{-group|g <group>] |
[-subject-id|sb <subjectid>}]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-h]
```

Description

`viprcli tenant delete-role` command removes a role assignment from an active directory user or group assigned to a tenant. The role can be either **TENANT_ADMIN** or **PROJECT_ADMIN**. This command is silent on success.

Options

name|n

A valid ViPR Controller tenant name. This is a mandatory parameter.

role|r

This is a mandatory parameter. The possible values are:

- TENANT_ADMIN
- PROJECT_ADMIN

group|g

The group argument specifies a named group of users. Typically, this is the name of a group in Active Directory. You must specify either a subject ID or a group. This is an optional parameter.

subject-id|sb

The subject ID specifies a user, typically a name in Active Directory. You must specify either a subject ID or a group. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 263

Example 263 (continued)

```
# viprcli tenant delete-role -name 'Provider Tenant' -role
PROJECT_ADMIN
          -sb employee@sanity.local
# viprcli tenant get-role -name 'Provider Tenant'
{
   "link": {
      "href": "/tenants/urn:storageos:TenantOrg:
eda8a3de-b440-4e56-978b-30d31ff3a82d:/role-assignments",
      "rel": "self"
   },
   "role_assignments": [
      {
         "role": [
            "TENANT_ADMIN",
            "TENANT_APPROVER",
            "PROJECT_ADMIN"
         ],
         "subject_id": "emp01@sanity.local"
      },
      {
         "role": [
            "TENANT_ADMIN",
            "TENANT_APPROVER",
            "PROJECT_ADMIN"
         ],
         "subject_id": "emp02@sanity.local"
      },
      {
         "role": [
            "TENANT_ADMIN"
         ],
         "subject_id": "root"
      },
      {
         "role": [
            "TENANT_ADMIN"
         ],
         "subject_id": "employee@sanity.local"
      }
   ]
}
```

viprcli tenant delete

Delete a tenant with the given name.

Syntax

```
viprcli tenant delete
-name <tenantname>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli tenant delete` command deletes a tenant and is silent on success.

Options

name|n

A valid tenant name.

common arguments

This operation also takes the Common Parameters on page 24.

Example 264

```
# viprcli tenant delete -n tenant1

// This command is silent on success
```

viprcli tenant get-clusters

Return the list of clusters in a tenant.

Syntax

```
viprcli tenant get-clusters
[-tenant|tn <tenantname>]
[{-verbose|v <verbose> | -long|l <long>}]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli tenant get-clusters` command returns the list of clusters in a tenant.

Options

tenant|tn

The name of a tenant. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is an optional parameter.

{verbose|v |long|l}

These options display additional information about clusters. Use only either verbose or long and not both.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli tenant get-hosts

Return the list of hosts in a tenant.

Syntax

```
viprcli tenant get-hosts
[-tenant|tn <tenantname>]
[{-verbose|v <verbose> | -long|l <long>}]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli tenant get-hosts` command returns the list of hosts in a tenant.

Options

tenant|tn

The name of a tenant. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is an optional parameter.

{verbose|v |long|l}

These options display additional information about host. Use only either verbose or long and not both.

common arguments

This operation also takes the Common Parameters on page 24.

Example 265

```
# viprcli tenant get-hosts
  NAME
  vipr_host.example.mycompany.com
  viprhost20
```

viprcli tenant get-role

Retrieve the users with roles assigned to a tenant.

Syntax

```
viprcli tenant get-role
-name|n <tenantname>
[-xml <xml>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
```

```
            [-cookiefile|cf <cookiefile>]
            [-h]
```

Description

`viprcli tenant get-role` command lists each user with roles assigned to the current tenant, and their role assignments. If a user has not been assigned any role then that user will not be listed. Only PROJECT_ADMIN and TENANT_ADMIN roles are listed for each user - other roles assigned to those users are not displayed.

Options

name|n

A valid ViPR Controller tenant name.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

common arguments

This operation also takes the Common Parameters on page 24.

Example 266

```
viprcli tenant get-role -n "Provider Tenant"

{'role_assignments': [{'subject_id': 'jordab', 'role':
['TENANT_ADMIN', 'PROJECT_ADMIN']},
{'subject_id': 'mquinn', 'role': ['TENANT_ADMIN']}],
'link': {'href': '/tenants/urn:storageos:TenantOrg:5a0e8577-
ff0c-4e68-93dd-64594928f370:/role-assignments',
'rel': 'self'}}
```

viprcli tenant get-vcenters

Return the list of vCenters in a tenant.

Syntax

```
            viprcli tenant get-vcenters
            [-tenant|tn <tenantname>]
            [{-verbose|v <verbose> | -long|l <long>}]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookiefile>]
            [-h]
```

Description

`viprcli tenant get-vcenters` command returns the list of vCenters in a tenant.

Options

tenant|tn

The name of a tenant. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is an optional parameter.

{verbose|v |long|l}

These options display additional information about vCenters. Use only either verbose or long and not both.

common arguments

This operation also takes the Common Parameters on page 24.

Example 267

```
# viprcli tenant get-vcenters
  NAME
  viprvcenter.example.mycompany.com
```

viprcli tenant list

Return a list of tenants and subtenants.

Syntax

```
viprcli tenant list
[-verbose|v]
[-long|l]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli tenant list` command displays information about the current user's tenant. A tenant represents an organization operating within the ViPR Controller environment. Tenants are created in the system for the purposes of strict security isolation. As an example, imagine Company1 and Company2 are two tenants sharing the same infrastructure. Security isolation between the tenants ensures that no one from Company1 can know of or affect anything in Company2's storage and vice versa.

Options

{ verbose|v |long|l}

These options provide additional information. Use either verbose or long and not both.

common arguments

This operation also takes the Common Parameters on page 24.

Example 268

```
# viprcli tenant list
NAME                                    ENTERPRISE-SUFFIX
tenant2                                 tenant2.com
tenant_011707425144079                  tenant_011707425144079.com
webstorage
webstorage.example.mycompany.com
```

viprcli tenant list-object-namespaces

Returns a list of object namespaces.

Syntax

```
            viprcli tenant list-object-namespaces
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -namespacestoragesystem|nsstsystem
        <namespacestoragesystem>
```

Description

The `viprcli tenant list-object-namespaces` command returns a list of object namespaces.

Options

namespacestoragesystem|nsstsystem

Specifies the name of the namespace storage system. This is a mandatory parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 269

```
viprcli tenant list-object-namespaces -nsstsystem ECS
```

viprcli tenant remove-attribute

Remove an Active Directory attribute from a ViPR Controller tenant.

Syntax

```
viprcli tenant remove-attribute
[-name|n <tenantname>]
-key <keyname>
-value <value>
-domain <domain>
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli tenant remove-attribute` command deletes an Active Directory attribute from a ViPR Controller tenant.

Options

key

An Active Directory key to map to the tenant. This is a mandatory parameter.

value

The value in Active Directory corresponding to the specified Key. This is a mandatory parameter.

domain

The domain of the Active Directory user. This is a mandatory parameter.

name|n

The name of a tenant. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli tenant show

Display the detailed information of a tenant.

Syntax

```
viprcli tenant show
[name|n <tenantname>]
[-xml]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
```

```
            [-cookiefile|cf <cookiefile>]
            [-h]
```

Description

`viprcli tenant show` command displays extended information for a specified tenant, such as its activity status, and its enterprise suffix.

Options

name|n

A valid ViPR Controller tenant name. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

common arguments

This operation also takes the Common Parameters on page 24.

Example 270

```
# viprcli tenant show -n tenant1
{
   "enterprise-suffix": [
      "tenant.com"
   ]
   "id": "urn:storageos:TenantOrg:
40329560-826d-4c7a-9d1f-710ad2b7f17c:"
   "inactive": false
   "name": "tenant1"
   "parent-tenant":
"urn:storageos:TenantOrg:c89743b1-9400-4ef8-8cf2-4b84cbfeaf70:
}
```

viprcli tenant show-object-namespaces

Displays object namespaces.

Syntax

```
            viprcli tenant show-object-namespaces
            [-h]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -namespacestoragesystem|nsstsystem
        <namespacestoragesystem>
```

```
            -namespace
```

Description

`viprcli tenant show-object-namespaces` command displays object namespaces.

Options

namespacestoragesystem|nsstsystem

Specifies the name of the namespace storage system. This is a mandatory parameter.

namespace

Specifies the name of the namespace. This is a mandatory parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 271

```
viprcli tenant tenant show-object-namespaces -nsstsystem ECS -
namespace test
```

viprcli tenant update-quota

Update the tenant with quota information.

Syntax

```
            viprcli tenant update-quota
            [-tenant|tn <tenantname>]
            [-quota_enable|qe <quota_enable> {true | false}]
            [-quota_capacity|qc <quota_capacity>]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookiefile>]
            [-h]
```

Description

`viprcli tenant update-quota` command updates the specified tenant with specified quota information.

Options

tenant|tn

The name of a tenant. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is an optional parameter.

quota_enable|qe

Enable or disable quota. Possible values are: true or false. The default is false. This is an optional parameter.

quota_capacity|qc

Quota capacity in GB. Quota reserves the maximum limit for allocated storage space for the specified tenant user. This is an optional parameter and used only when you set the `quota_enable` to true.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 47

viprcli varray commands

This chapter contains the following topics:

- viprcli varray create 588
- viprcli varray get-acl 588
- viprcli varray list 589
- viprcli varray show 591
- viprcli varray allow 592
- viprcli varray disallow 593
- viprcli varray update 594
- viprcli varray list-storage-ports 594
- viprcli varray list-storage-portsgroup 595
- viprcli varray delete 596

viprcli varray create

Create a virtual array.

Syntax

```
viprcli varray create
-name|n <varrayname>
[-autosanzoning <autosanzoning>] {true | false}
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli varray create` command creates a virtual array with the given name.

Options

name|n

The name of a valid virtual array. This is a mandatory parameter.

autosanzoning

Boolean to allow automatic SAN zoning. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 272

```
viprcli varray create -n my_varray
{
   "id": "urn:storageos:VirtualArray:6e0eb32f-2091-4fd1-
ac83-6c046be72ba9:",
   "inactive": "false",
   "name": "my_varray"
}
```

viprcli varray get-acl

List tenants that have ACL privileges assigned for this virtual array.

Syntax

```
viprcli varray get-acl
-name|n <varrayname>
[-hostname|hn <hostname>]
```

```
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli varray get-acl` command lists the tenants that have USE privileges for the current virtual array.

Options

name|n

A valid ViPR Controller virtual array name. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 273

```
# viprcli varray get-acl -n my_varray
  TENANTNAME PRIVILEGE

# viprcli varray allow -tenant
'sanity.local tenant' -name my_varray

# viprcli varray get-acl -n my_varray
  TENANTNAME          PRIVILEGE
  sanity.local tenant USE

# viprcli varray disallow -tenant
'sanity.local tenant' -name my_varray

# viprcli varray get-acl -n my_varray
  TENANTNAME PRIVILEGE
```

viprcli varray list

Lists the virtual arrays.

Syntax

```
            viprcli varray list
            [{-verbose|-long}]
            [-vdcname|vn <vdc_name>]
            [-tenant|tn <tenant>]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli varray list` command lists the virtual arrays in the virtual data center.

Options

{verbose|v |long|l}

These arguments display additional information about the virtual arrays of the specified type. These are optional parameters. Use only either verbose or long.

vdcname|vn

The name of the virtual data center. Returns the current ViPR instance or local ViPR instance.

tenant|tn

The name of the tenant. If user has permission on the input tenant, returns the varrays to which the tenant has access. If user does not have permission on the input tenant, an insufficient permission exception will be thrown.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 274

```
# viprcli varray list -v
[
   {
      "auto_san_zoning": true,
      "creation_time": 1377075603323,
      "id":
"urn:storageos:VirtualArray:f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
      "inactive": false,
      "link": {
         "href": "/vdc/varrays/urn:storageos:VirtualArray:
         f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
         "rel": "self"
      },
      "name": "myvarray",
      "tags": []
   },
   {
      "auto_san_zoning": true,
      "creation_time": 1377267401756,
      "id": "urn:storageos:VirtualArray:
      43373b71-2a29-4c88-b83d-aa85cce2765b:",
      "inactive": false,
      "link": {
         "href": "/vdc/varrays/urn:storageos:VirtualArray:
        43373b71-2a29-4c88-b83d-aa85cce2765b:",
         "rel": "self"
      },
      "name": "filearray",
      "tags": []
   },
   {
      "auto_san_zoning": false,
      "creation_time": 1376916869921,
      "id": "urn:storageos:VirtualArray:
       501ed39e-834c-4dc0-baad-d16a5174482d:",
      "inactive": false,
      "link": {
```

Example 274 (continued)

```
            "href": "/vdc/varrays/urn:storageos:VirtualArray:
            501ed39e-834c-4dc0-baad-d16a5174482d:",
            "rel": "self"
        },
        "name": "my_varray2",
        "tags": []
    }

]
```

viprcli varray show

Display the details about a virtual array.

Syntax

```
viprcli varray show
-name|n <varrayname>
[-xml]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli varray show` command displays the name, inactivity flag and internal ViPR Controller id of a virtual array.

Options

name|n

The name of the virtual array. This is a mandatory parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 275

```
# viprcli varray show -n my_varray
{
   "auto_san_zoning": true,
   "creation_time": 1377075603323,
   "id": "urn:storageos:VirtualArray:
    f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
```

Example 275 (continued)

```
    "inactive": false,
    "link": {
       "href": "/vdc/varrays/urn:storageos:VirtualArray:
       f2db8dcc-6e3a-466e-9893-34ed6693ba5d:",
       "rel": "self"
    },
    "name": "my_varray",
    "tags": []
}
```

viprcli varray allow

Allow a specific tenant to use a virtual array.

Syntax

```
viprcli varray allow
-name|n <varrayname>
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli varray allow` command allows a specific tenant to use a virtual array.

Options

name|n

The name of a valid virtual array. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 276

```
# viprcli varray create -n my_varray

# viprcli varray allow -name my_varray
```

viprcli varray disallow

Remove the USE privilege from a tenant for the specified virtual array.

Syntax

```
viprcli varray disallow
-name|n <varrayname>
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli varray disallow` command prevents a tenant from using a virtual array. This operation removes the USE access privilege from the specified tenant.

Options

name|n

The name of a valid virtual array. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 277

```
# viprcli varray get-acl -n my_varray
  TENANTNAME PRIVILEGE

# viprcli varray allow -tenant 'sanity.local tenant' -name my_varray

# viprcli varray get-acl -n my_varray
  TENANTNAME          PRIVILEGE
  sanity.local tenant USE

# viprcli varray disallow -tenant 'sanity.local tenant' -name
my_varray

# viprcli varray get-acl -n my_varray
  TENANTNAME PRIVILEGE
```

viprcli varray update

Update the virtual array.

Syntax

```
viprcli varray update
-name|n <varrayname>
[-devregistered|reg
<devregistered>]
[-autosanzoning
<autosanzoning>]
[-hostname|hn
<hostname>]
[-port|po <port_number>]
[-portui|pu
<ui_port_number>]
[-cookiefile|cf
<cookie_file>]
[-h]
```

Description

`viprcli varray update` command updates the virtual array.

Options

name|n

The name of a valid virtual array. This is a mandatory parameter.

devregistered|reg

Boolean. Indicates if the device is registered to the virtual array. This is an optional parameter.

autosanzoning

Boolean to allow automatic SAN zoning. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

viprcli varray list-storage-ports

List implicitly associated storage ports of a virtual array.

Syntax

```
viprcli varray list-storage-ports
-name|n <varrayname>
[-network_connectivity <network_connectivity>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
```

```
            [-h]
```

Description

`viprcli varray list-storage-ports` command lists implicitly associated storage port information.

Options

name|n

The name of a valid virtual array. This is a mandatory parameter.

network_connectivity

Boolean to list implicitly associated storage ports. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 278

```
viprcli varray list-storage-ports -n MY_VARRAY
 PORT_NAME PORT_GROUP PORT_NETWORK_ID
TRANSPORT_TYPE STORAGE_SYSTEM
  FA-8E:0   FA-8E      50:00:02:53:00:17:85:1C
FC             SYMMETRIX+000195701573
  FA-8E:1   FA-8E      50:00:02:53:00:17:85:1D
FC             SYMMETRIX+000195701573
  SP_A:10   SP_A       iqn.1994-04.com.emc:cx.apm00121500018.a10
IP             CLARIION+APM00121500018
  SP_A:11   SP_A       iqn.1994-04.com.emc:cx.apm00121500018.a11
IP             CLARIION+APM00121500018
  SP_A:4    SP_A       50:06:02:54:4E:A0:59:B0
FC             CLARIION+APM00121500018
  SP_A:8    SP_A       iqn.1994-04.com.org:cx.apm00112400115.a8
IP             CLARIION+APM00121500018
  SP_A:9    SP_A       iqn.1994-04.com.org:cx.apm00112400115.a9
IP             CLARIION+APM00121500018
```

viprcli varray list-storage-portsgroup

List implicitly associated storage portsgroup of a virtual array.

Syntax

```
            viprcli varray list-storage-portsgroup
            -name|n <varrayname>
            [-network_connectivity <network_connectivity>]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
```

```
            [-h]
```

Description

`viprcli varray list-storage-portsgroup` command lists implicitly associated storage port group information.

Options

name|n

The name of a valid virtual array. This is a mandatory parameter.

network_connectivity

Boolean to list implicitly associated storage ports. This is an optional parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 279

```
viprcli varray list-storage-portsgroup -n MY_VARRAY
 PORT_NAME PORT_GROUP PORT_NETWORK_ID
TRANSPORT_TYPE STORAGE_SYSTEM
  FA-8E:0   FA-8E      50:00:02:53:00:17:85:1C
FC             SYMMETRIX+000195701573
  FA-8E:1   FA-8E      50:00:02:53:00:17:85:1D
FC             SYMMETRIX+000195701573
  SP_A:10   SP_A       iqn.1994-04.com.emc:cx.apm00121500018.a10
IP             CLARIION+APM00121500018
  SP_A:11   SP_A       iqn.1994-04.com.emc:cx.apm00121500018.a11
IP             CLARIION+APM00121500018
  SP_A:4    SP_A       50:06:02:54:4E:A0:59:B0
FC             CLARIION+APM00121500018
  SP_A:8    SP_A       iqn.1994-04.com.org:cx.apm00112400115.a8
IP             CLARIION+APM00121500018
  SP_A:9    SP_A       iqn.1994-04.com.org:cx.apm00112400115.a9
IP             CLARIION+APM00121500018
```

viprcli varray delete

Deletes a virtual array.

Syntax

```
            viprcli varray delete
            -name|n <varrayname>
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli varray delete` command deletes a virtual array.

Options

name|n

The name of a valid virtual array. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

Example 280

```
# viprcli varray delete -n test
"varray test: Deleted"
```

CHAPTER 48

viprcli vcenter commands

This chapter contains the following topics:

- viprcli vcenter create 600
- viprcli vcenter delete 601
- viprcli vcenter discover 602
- viprcli vcenter get-clusters 603
- viprcli vcenter get-datacenters 603
- viprcli vcenter get-hosts 604
- viprcli vcenter list 605
- viprcli vcenter show 605
- viprcli vcenter tasks 607
- viprcli vcenter update 608

viprcli vcenter create

Adds a vCenter to the ViPR Controller physical assets.

Syntax

```
viprcli vcenter create
    -name|n <vcentername>
    -vcenter_ip|vcip
<vcenter_ip>
    -vcenter_port|vcpo
<vcenterport>
    -user|u <username>
    [-tenant|tn <tenant>]
    [-cascade_tenancy|cascade <true |
false>]
    [-osversion|ov
<osversion>]
    [-usessl|ssl]
    [-hostname <hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookie_file>]
    [-h]
```

Description

`viprcli vcenter create` command creates a vCenter. This command is silent on success.

Options

name|n

The vCenter host name. This is a mandatory parameter.

vcenter_ip|vcip

The IP address of the vCenter. This is a mandatory parameter.

vcenter_port|vcpo

The port number of the vCenter. This is a mandatory parameter.

user|u

The user name of the vCenter host. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

cascade_tenancy|cascade

If `true`, assigns the vCenter, and all its resources, (datacenters, clusters, and hosts) to the same tenant. If you choose to cascade tenancy, then you can only assign the vCenter to one tenant.
If `false`, assigns the vCenter to be shared across different tenants. If you did not enable Cascade Tenancy select the tenants with which the vCenter will be shared.

Note

If you assign the vCenter to multiple tenants, you will need to assign each datacenters to one of the vCenter tenants.

This is an optional parameter.

osversion|ov

The operating system version number of host. This is an optional parameter.

usessl|ssl

Use SSL or not. To enable SSL, add the -usessl or -ssl option to the command line. If this parameter is not in the command line, then SSL is not used. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 281

```
 # viprcli vcenter create -n examplevcenter -vcip 192.168.1.2 -vcpo
608 -u admin
Enter password of the VCenter:
```

viprcli vcenter delete

Delete a vCenter.

Syntax

```
viprcli vcenter delete
-name|n <vcentername>
[-tenant|tn <tenant>]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vcenter delete` command deletes the specified vCenter. This command is silent on success.

Options

name|n

The vCenter host name. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 282

```
# viprcli vcenter delete -n examplevcenter
```

Use the list command view results.

```
# viprcli vcenter list
  NAME
  viprvm.example.mydomain.com
```

viprcli vcenter discover

Discover a vCenter.

Syntax

```
viprcli vcenter discover
[-name|n <vcentername>]
[-tenant|tn <tenantname>]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-xml]
[-h]
```

Description

`viprcli vcenter discover` command discovers a vCenter.

Options

name|n

The name of the vCenter. This is a mandatory parameter.

tenant|tn

The name of the tenant.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenter get-clusters

Display the clusters of a vCenter.

Syntax

```
viprcli vcenter get-clusters
-name|n <vcenterhostname>
[{-verbose|v <verbose> | long|l <long>}]
[-tenant|tn <tenant>]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vcenter get-clusters` command displays the clusters of a vCenter.

Options

name|n

The name of vCenter. This is a mandatory option.

{verbose|v | long|l}

These options provide additional information. Use either verbose or long.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenter get-datacenters

Display the datacenters of a vCenter.

Syntax

```
viprcli vcenter get-datacenters
-name|n <vcenterhostname>
[{-verbose|v <verbose> | long|l <long>}]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vcenter get-datacenters` command displays the datacenters of a vCenter.

Options

name|n

The name of vCenter. This is a mandatory option.

{verbose|v | long|l}

These options provide additional information. Use either verbose or long.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenter get-hosts

Display the hosts of a vCenter.

Syntax

```
viprcli vcenter get-hosts
-name|n <vcenterhostname>
[{-verbose|v <verbose> | long|l <long>}]
[-tenant|tn <tenant>]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vcenter get-hosts` command displays the hosts of a vCenter.

Options

name|n

The name of vCenter. This is a mandatory option.

{verbose|v | long|l}

These options provide additional information. Use either verbose or long.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 283

```
# viprcli vcenter get-hosts -n viprvm.example.mydomain.com
  NAME
  host_01.example.mydomain.com
```

viprcli vcenter list

Display the list of vCenters.

Syntax

```
viprcli vcenter list
[-verbose|v <verbose>]
[-long|l <long>]
[-tenant|tn <tenant>]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

viprcli vcenter list displays the list of vCenters.

Options

tenant|tn

The name of the tenant. This is an optional parameter.

verbose|v | long|l

These options provide additional information. Use either verbose or long and not both.

common arguments

This operation also takes the Common Parameters on page 24.

Example 284

```
# viprcli vcenter list
  NAME
  examplevcenter
  viprvm.example.mydomain.com
```

viprcli vcenter show

Return the vCenter details.

Syntax

```
viprcli vcenter show
-name|n <vcenterhostname>
[-xml <xmlresponse>]
[-tenant|tn <tenantname>]
[-hostname <hostname>]
```

```
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli vcenter show` command returns the vCenter details.

Options

name|n

The ViPR Controller host name. This is a mandatory parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

tenant|tn

Name of the tenant. If a tenant name is not provided, the attribute is assigned to the Provider Tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 285

```
# viprcli vcenter show -n viprvm.example.mydomain.com
{
   "compatibility_status": "COMPATIBLE",
   "creation_time": 1379686793615,
   "id":
"urn:storageos:Vcenter:b693f040-4462-4641-9ca9-0eddae42289c:",
   "inactive": false,
   "ip_address": "viprvm.example.mydomain.com",
   "job_discovery_status": "COMPLETE",
   "job_metering_status": "CREATED",
   "last_discovery_run_time": 1379854187234,
   "last_discovery_status_message": "",
   "last_metering_run_time": 0,
   "link": {
      "href": "/compute/vcenters/urn:storageos:Vcenter:
       b693f040-4462-4641-9ca9-0eddae42289c:",
      "rel": "self"
   },
   "name": "viprvm.example.mydomain.com",
   "native_guid": "",
   "next_discovery_run_time": 0,
   "next_metering_run_time": 0,
   "port_number": 443,
   "registration_status": "REGISTERED",
   "tags": [],
   "tenant": {
      "id": "urn:storageos:TenantOrg:
      dbe23c11-4d96-4fa5-a617-01c8906eef9d:",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:
          dbe23c11-4d96-4fa5-a617-01c8906eef9d:",
         "rel": "self"
      }
```

Example 285 (continued)

```
    },
    "use_ssl": false,
    "user_name": "administrator"
}
```

viprcli vcenter tasks

Check the tasks of a vCenter.

Syntax

```
viprcli vcenter tasks
[-name|n <vcentername>]
[-tenant|tn <tenantname>]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-id <opid>]
[-verbose|v]
[-h]
```

Description

`viprcli vcenter tasks` command checks the tasks of a vCenter.

Options

name|n

The name of the vCenter. This is a mandatory parameter.

tenant|tn

The name of the tenant.

id

The operation ID of a vCenter.

verbose|v

List all the tasks of a vCenter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenter update

Allows you to update the properties of a vCenter that has been added to the ViPR Controller physical asstets.

Syntax

```
        viprcli vcenter update
        -name|n <vcentername>
        -vcenter_ip|vcip
    <vcenter_ip>
        -vcenter_port|vcpo
    <vcenterport>
        -user|u <username>
        [-tenant|tn <tenant>]
        [-cascade_tenancy|cascade <Y|N>]
        [-osversion|ov
    <osversion>]
        [-usessl|ssl
    <SSL_state>]
        [-hostname <hostname>]
        [-port|po <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookie_file>]
        [-h]
```

Description

`viprcli vcenter update` Allows you to update the properties of a vCenter that has been added to the ViPR Controller physical asstets.

Options

name|n

The vCenter host name. This is a mandatory parameter.

vcenter_ip|vcip

The IP address of the vCenter. This is a mandatory parameter.

vcenter_port|vcpo

The port number of the vCenter. This is a mandatory parameter.

user|u

The user name of the vCenter host. This is a mandatory parameter.

tenant|tn

The name of the tenant.

cascade_tenancy|cascade

If yes, assigns the vCenter, and all its resources, (datacenters, clusters, and hosts) to the same tenant. If you choose to cascade tenancy, then you can only assign the vCenter to one tenant.
If no, assigns the vCenter to be shared across different tenants. If you did not enable Cascade Tenancy select the tenants with which the vCenter will be shared.

Note

If you assign the vCenter to multiple tenants, you will need to assign each datacenters to one of the vCenter tenants.

osversion|ov

The operating system version number of host. This is an optional parameter.

usessl|ssl

Use SSL or not. Possible values are: yes or no. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 49

viprcli vcenterdatacenter commands

This chapter contains the following topics:

- viprcli vcenterdatacenter create 612
- viprcli vcenterdatacenter create-cluster 612
- viprcli vcenterdatacenter delete 613
- viprcli vcenterdatacenter get-clusters 614
- viprcli vcenterdatacenter get-hosts 614
- viprcli vcenterdatacenter list 615
- viprcli vcenterdatacenter show 616
- viprcli vcenterdatacenter update-cluster 616

viprcli vcenterdatacenter create

Create a vCenter datacenter.

Syntax

```
viprcli vcenterdatacenter create
-name|n <vcenterdatacentername>
-vcenter <vcentername>
[-tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vcenterdatacenter create` command creates a vCenter datacenter. This command is silent on success.

Options

name|n

The vCenter datacenter name. This is a mandatory parameter.

vcenter|vc

The name of the vCenter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenterdatacenter create-cluster

Create a vCenter datacenter cluster.

Syntax

```
viprcli vcenterdatacenter create-cluster
-name|n <vcenterdatacentername>
-vcenter <vcentername>
-cluster <cluster>
[-tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vcenterdatacenter create-cluster` command creates a vCenter datacenter cluster. This command is silent on success.

Options

name|n

The vCenter datacenter name. This is a mandatory parameter.

vcenter|vc

The name of the vCenter. This is a mandatory parameter.

cluster

The name of the cluster. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenterdatacenter delete

Delete a vCenter datacenter.

Syntax

```
viprcli vcenterdatacenter delete
-name|n <vcenterdatacentername>
-vcenter|vc <vcentername>
[-tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vcenterdatacenter delete` command deletes a vCenter datacenter. This command is silent on success.

Options

name|n

The vCenter datacenter name. This is a mandatory parameter.

vcenter|vc

The name of the vCenter. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenterdatacenter get-clusters

Display the clusters of a vCenter datacenter.

Syntax

```
viprcli vcenterdatacenter get-clusters
-name|n <vcenteratacentername>
-vcenter <vcentername>
[{-verbose|v <verbose> | -long|l <long>}]
[-tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vcenterdatacenter get-clusters` command displays the clusters of a vCenter datacenter.

Options

name|n

The vCenter datacenter name. This is a mandatory parameter.

vcenter

The name of vCenter. This is a mandatory parameter.

{verbose|v | long|l}

These options provide additional information. Use either verbose or long.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenterdatacenter get-hosts

Display the hosts of a vCenter datacenter.

Syntax

```
viprcli vcenterdatacenter get-hosts
-name|n <vcenteratacentername>
-vcenter <vcentername>
[{-verbose|v <verbose> | -long|l <long>}]
[-tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
```

```
            [-h]
```

Description

`viprcli vcenterdatacenter get-hosts` command displays the hosts of a vCenter datacenter.

Options

name|n

The vCenter datacenter name. This is a mandatory parameter.

vcenter

The name of vCenter. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

{verbose|v | long|l}

These options provide additional information. Use either verbose or long.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenterdatacenter list

Display the list of datacenters in a vCenter.

Syntax

```
            viprcli vcenterdatacenter list
            -vcenter <vcentername>
            [-verbose|v <verbose>]
            [-long|l <long>]
            [-tenant|tn <tenant>]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli vcenterdatacenter list` command displays the list of datacenter in a vCenter.

Options

vcenter

The name of vCenter. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenterdatacenter show

Return the datacenter details of a vCenter.

Syntax

```
viprcli vcenterdatacenter show
-name|n <vcenteratacentername>
-vcenter <vcentername>
[-xml <xmlresponse>]
[-tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vcenterdatacenter show` command returns the datacenter details of a vCenter.

Options

name|n

The vCenter datacenter name. This is a mandatory parameter.

vcenter

The name of vCenter. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vcenterdatacenter update-cluster

Updates a vCenter datacenter cluster.

Syntax

```
viprcli vcenterdatacenter update-cluster
-name|n <vcenterdatacentername>
-vcenter <vcentername>
-cluster <cluster>
[-tenant|tn <tenant>]
```

```
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu <ui_port_number>]
            [-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli vcenterdatacenter update-cluster` command updates a vCenter datacenter cluster. This command is silent on success.

Options

name|n

The vCenter datacenter name. This is a mandatory parameter.

vcenter|vc

The name of the vCenter. This is a mandatory parameter.

cluster

The name of the cluster. This is a mandatory parameter.

tenant|tn

The name of the tenant. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 50

viprcli vdc commands

This chapter contains the following topics:

- viprcli vdc add 620
- viprcli vdc delete 620
- viprcli vdc disconnect 621
- viprcli vdc list 621
- viprcli vdc reconnect 622
- viprcli vdc show 622
- viprcli vdc update 623

viprcli vdc add

Add a vdc.

Syntax

```
viprcli vdc add
[-name|n <vdcname>]
[-certfile|cf <certificate_file>]
[-key|k <key> ]
[-endpoint|ep <endpoint>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vdc add` command adds a vdc.

Options

name|n

The name of the vdc. This is a mandatory parameter.

certfile|cf

The path of the file containing the certificate to be added. This is a mandatory parameter.

key|k

The secure key of the vdc to be added. This is a mandatory parameter.

endpoint|ep

The virtual IP address of the vdc to be added. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vdc delete

Delete a vdc.

Syntax

```
viprcli vdc delete
[-name|n <viprhostname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vdc delete` command deletes a vdc.

Options

name|n

The name of the vdc. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vdc disconnect

Disconnect a vdc.

Syntax

```
viprcli vdc disconnect
[-name|n <viprhostname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vdc disconnect` command disconnects a vdc.

Options

name|n

The name of the vdc. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vdc list

List a vdc.

Syntax

```
viprcli vdc list
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-v|verbose]
[-l|long]
[-h]
```

Description

`viprcli vdc list` command lists the details of a vdc.

Options

verbose|v

Lists the details of the virtual data center.

long|l

Lists the details in a table of the virtual data center.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vdc reconnect

Reconnect a vdc.

Syntax

```
viprcli vdc reconnect
[-name|n <viprhostname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli vdc reconnect` command reconnects a vdc.

Options

name|n

The name of the vdc. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vdc show

Show a vdc.

Syntax

```
viprcli vdc show
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-xml]
```

```
            [-h]
```

Description

`viprcli vdc show` command shows a vdc.

Options

xml

The xml response.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli vdc update

Update a vdc.

Syntax

```
viprcli vdc update
[-name|n <viprhostname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-newname|nw <newname>]
[-description|desc <description>]
[-h]
```

Description

`viprcli vdc update` command updates a vdc.

Options

name|n

The name of the vdc. This is a mandatory parameter.

newname|nw

The new name for the vdc to be updated.

description|desc

The description of the virtual data center.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 51

viprcli vnasserver commands

This chapter contains the following topics:

- viprcli vnasserver assign 626
- viprcli vnasserver list 626
- viprcli vnasserver show 627
- viprcli vnasserver unassign 628

viprcli vnasserver assign

Assigns a vNAS server to a ViPR Controller project.

Syntax

```
            viprcli vnasserver assign
            [-h]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|-n <vnas_server>
        [<vnas_server> ...]
            -project|-pr <project>
        [<project> ...]
            -tenant|-tn <tenant>
```

Description

The `viprcli vnasserver assign` command assigns one or more vNAS servers to a ViPR Controller project.

Options

name|n

List of vNAS servers. This is a mandatory parameter.

project|pr

Names of project. This is a mandatory parameter.

tenant|tn

Name of tenant. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

```
# viprcli vnasserver assign -name vdm_3b -project project1
```

viprcli vnasserver list

Return a list of vNAS servers.

Syntax

```
            viprcli vnasserver list
            [-hostname|hn <hostname>]
```

```
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-v|verbose]
            [-l|long]
            [-h]
```

Description

The `viprcli vnasserver list` command can list the details of vNAS servers.

Options

verbose|v

Lists the details of a vNAS server.

long||

Lists the details of a vNAS server in a table.

common arguments

This operation also takes the Common Parameters on page 24.

```
# viprcli vnasserver list
NAME
vdm_3a
vdm_3b
vdm_4a
vdm_4b
vdm_as2
vdm_as4
```

viprcli vnasserver show

Show the attributes of a vNAS server.

Syntax

```
            viprcli vnasserver show
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            [-xml <XML response>]
            [-name|-n <vNAS server>]
            [-h]
```

Description

The `viprcli vnasserver show` command shows the attributes of a vNAS server, such as its compatibility status and performance statics for dynamic workloads.

```
# viprcli vnasserver show -name vdm_3b
{
   "assigned_varrays": [],
   "avg_percentage_busy": "0.66",
   "cifs_servers": [,
     "LOSAT215_CIFS = PROVISIONING.BOURNE.LOCAL"
   ],

   "compatibility_status": "COMPATIBLE",
   "connected_varrays": [],
```

viprcli vnasserver unassign

Unassigns a vNAS server from a project

Syntax

```
            viprcli vnasserver assign
            [-h]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookie_file>]
            -name|-n
        <vnas_server>[<vnas_server> ...]
            -project|-pr <project>
        [<project> ...]
```

Description

The `viprcli vnasserver unassign` command unassigns one or more vNAS servers from a ViPR Controller project.

Options

name|n

List of vNAS servers. This is a mandatory parameter.

project|pr

Names of project. This is a mandatory parameter.

Common Arguments

This operation also takes the arguments listed in the Common Parameters on page 24.

```
# viprcli vnasserver unassign -name vdm_3b -project project1
```

CHAPTER 52

viprcli volume commands

This chapter contains the following topics:

- viprcli volume clone 632
- viprcli volume clone-activate 633
- viprcli volume clone-deactivate 634
- viprcli volume clone-checkprogress 635
- viprcli volume clone-detach 637
- viprcli volume clone-list 638
- viprcli volume clone-restore 639
- viprcli volume clone-resync 640
- viprcli volume clone-show 641
- viprcli volume continuous_copies copy 644
- viprcli volume continuous_copies delete 645
- viprcli volume continuous_copies establish 646
- viprcli volume continuous_copies failover 647
- viprcli volume continuous_copies failover-test 648
- viprcli volume continuous_copies failover-test-cancel 649
- viprcli volume continuous_copies list 650
- viprcli volume continuous_copies pause 651
- viprcli volume continuous_copies restore 653
- viprcli volume continuous_copies resume 654
- viprcli volume continuous_copies show 656
- viprcli volume continuous_copies start 658
- viprcli volume continuous_copies stop 660
- viprcli volume continuous_copies swap 662
- viprcli volume continuous_copies update-access-mode 663
- viprcli volume create 664
- viprcli volume delete 667
- viprcli volume expand 668
- viprcli volume list 669
- viprcli volume migration-cancel 671
- viprcli volume migration-deactivate 672
- viprcli volume migration-list 674
- viprcli volume migration-pause 675
- viprcli volume migration-resume 676
- viprcli volume migration-show 678
- viprcli volume protectionset show 679
- viprcli volume show 679
- viprcli volume tag 681
- viprcli volume tasks 682
- viprcli volume unmanaged ingest 684
- viprcli volume unmanaged show 685
- viprcli volume update 685

viprcli volume clone

Clone a volume with the given parameters.

Syntax

```
viprcli volume clone
    [-h]
    [-hostname|hn <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-consistencygroup|cg
<consistencygroup> | -volume|vol <volumename>]
    -project|pr
<projectname>
    [-tenant|tn <tenantname>]
    -name|n <fullcopyname>
    [-synchronous|sync]
    [-synctimeout|syncto]
    [-source_snapshot|src_snap
<source_snapshot>]
    [-count|cu <count>]
```

Description

`viprcli volume clone` command creates a volume clone.

Options

consistencygroup|cg

Name of a consistency group. When you specify a consistency group, you cannot specify a volume. This is an optional parameter.

volume|vol

Name of the source volume to be cloned. When you specify a volume, you cannot specify a consistency group. This is an optional parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

name|n

The name of the full copy. This is a mandatory parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

source_snapshot|src_snap

Name of the source snapshot from which to clone the volume. This is an optional parameter.

count|cu

Specifies the number of volumes to be cloned. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli volume clone-activate

Activates a fullcopy(clone) of a volume that was created inactive or all clones in a consistency group clone set.

Syntax

```
    viprcli volume clone-activate
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-consistencygroup|cg
<consistencygroup> | -volume|vol <volumename>]
    -project|pr
<projectname>
    [-tenant|tn <tenantname>]
    -name|n <fullcopyname>
    [-synchronous|sync]
    [-synctimeout|syncto]
```

Description

`viprcli volume clone-activate` command activates a volume clone or all volume clones in a consistency group. The command silently exits on success.

Options

consistencygroup|cg

Name of the consistency group. When you specify a consistency group, you cannot specify a volume. This is an optional parameter.

volume|vol

Name of the source volume to which the clone is resynchronized. When you specify a volume, you cannot specify a consistency group. This is an optional parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

name|n

The name of the full copy. This is a mandatory parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume clone-activate -project cg_pr -name VOLC1_Clone -
synchronous -consistencygroup cli_cg1 -sync
```

viprcli volume clone-deactivate

Deactivates a fullcopy(clone) of a volume or all clones in a consistency group clone set.

Syntax

```
            viprcli volume clone-deactivate
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-consistencygroup|cg
        <consistencygroup> | -volume|vol <volumename>]
            -project|pr
        <projectname>
            [-tenant|tn <tenantname>]
            -name|n <fullcopyname>
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

`viprcli volume clone-deactivate` command deactivates a volume clone or all volume clones in a consistency group. The command silently exits on success.

Options

consistencygroup|cg

Name of the consistency group. When you specify a consistency group, you cannot specify a volume. This is an optional parameter.

volume|vol

Name of the source volume to which the clone is resynchronized. When you specify a volume, you cannot specify a consistency group. This is an optional parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

name|n

The name of the full copy. This is a mandatory parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume clone-deactivate -project cg_pr -name VOLC1_Clone -
synchronous -consistencygroup cli_cg1 -sync
```

viprcli volume clone-checkprogress

Check progress of an operation on a single clone or a consistency group clone set, such as synchronization or restore.

Syntax

```
            viprcli volume clone-checkprogress
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-consistencygroup|cg
        <consistencygroup> | -volume|vol <volumename>]
            -project|pr
        <projectname>
            [-tenant|tn <tenantname>]
            -name|n <fullcopyname>
            [-synchronous|sync]
```

```
              [-synctimeout|syncto]
```

Description

`viprcli volume clone-checkprogress` command checks the progress of an operation on either a single clone or a consistency group clone set, such as synchronization or restore.

Options

consistencygroup|cg

Name of the consistency group. When you specify a consistency group, you cannot specify a volume. This is an optional parameter.

volume|vol

Name of the source volume for a clone. When you specify a volume, you cannot specify a consistency group. This is an optional parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

name|n

The name of the full copy. This is a mandatory parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume clone-checkprogress -pr myproj -volume post_driver  -
name clone-1
  IS_SYNC_ACTIVE PERCENT_SYNCED REPLICASTATE
  True           100            SYNCHRONIZED
```

viprcli volume clone-detach

Detach a clone from its source volume or detach all clones in a consistency group from their source volumes.

Syntax

```
viprcli volume clone-detach
    [-h]
    [-hostname <hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-consistencygroup|cg
<consistencygroup> | -volume|vol <volumename>]
    -project|pr
<projectname>
    [-tenant|tn <tenantname>]
    -name|n <fullcopyname>
    [-synchronous|sync]
    [-synctimeout|syncto]
```

Description

`viprcli volume clone-detach` command detaches a volume clone or all volume clones in a consistency group. The command silently exits on success.

Note

Once you detach a clone from its source volume, you can expand, export, unexport and delete it. However, you cannot use `viprcli volume clone-restore` to restore the full copy, or use `viprcli volume clone-resync` to synchronize the clone against the source volume.

Options

consistencygroup|cg

Name of the consistency group for which all clones are being detached from their source volumes. When you specify a consistency group, you cannot specify a volume name. This is an optional parameter.

volume|vol

Name of the source volume from which the clone is being detached. When you specify a volume, you cannot specify a consistency group. This is an optional parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

name|n

The name of the full copy. This is a mandatory parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume clone-detach -project cg_pr -name VOLC1_Clone -
synchronous -consistencygroup cli_cg1 -sync
```

viprcli volume clone-list

List the clones in a consistency group.

Syntax

```
            viprcli volume clone-list
            -project|pr
        <projectname>
            -consistencygroup|cg
        <consistencygroup>
            [-tenant|tn <tenantname>]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume clone-list` command lists the clones in a consistency group.

Options

consistencygroup|cg

Name of the consistency group. This is an mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume clone-list -pr myproj -cg df

NAME
  cd-1
  cd-2
```

viprcli volume clone-restore

Restore a single volume from its clone or all volumes in a consistency group from their clones.

Syntax

```
            viprcli volume clone-restore
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-consistencygroup|cg
        <consistencygroup> | -volume <volumename>]
            -project|pr
        <projectname>
            [-tenant|tn <tenantname>]
            -name|n <fullcopyname>
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

`viprcli volume clone` command creates a volume clone. The command silently exits on success.

Options

consistencygroup|cg

Name of the consistency group for which all volumes in the consistency group are being restored from the full copy. When you specify a consistency group, you cannot specify a volume. This is an optional parameter.

volume|vol

Name of the source volume being restored by the full copy. When you specify a volume, you cannot specify a consistency group. This is an optional parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

name|n

The name of the full copy. This is a mandatory parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume clone-restore -project cg_pr -name VOLC1_Clone -
consistencygroup cli_cg1 -sync
```

viprcli volume clone-resync

Resynchronizes a full copy(clone) of a single volume or all clones with their source volumes in a consistency group.

Syntax

```
            viprcli volume clone-resync
            [-h]
            [-hostname <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-consistencygroup|cg
        <consistencygroup> | -volume <volumename>]
            -project|pr
        <projectname>
            [-tenant|tn <tenantname>]
            -name|n <fullcopyname>
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

`viprcli volume clone-resync` command resynchronizes a volume clone or all volume clones in a consistency group. The command silently exits on success.

Options

consistencygroup|cg

Name of the consistency group. When you specify a consistency group, you cannot specify a volume. This is an optional parameter.

volume|vol

Name of the source volume to which the clone is resynchronized. When you specify a volume, you cannot specify a consistency group. This is an optional parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

name|n

The name of the full copy. This is a mandatory parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume clone-resync -project cg_pr -name VOLC1_Clone -
synchronous -consistencygroup cli_cg1 -sync
```

viprcli volume clone-show

Gets the details of a single clone or of all the clones in a consistency group.

Syntax

```
        viprcli volume clone-show
        -project|pr
    <projectname>
        -consistencygroup|cg
    <consistencygroup>
        -name|n <fullcopyname>
        [-tenant|tn <tenantname>]
        [-hostname <hostname>]
        [-port|po
    <port_number>]
        [-portui|pu
    <ui_port_number>]
        [-cookiefile|cf
    <cookiefile>]
        [-h]
```

Description

`viprcli volume clone-show` command gets the details of a single clone or all clones in a consistency group.

Options

consistencygroup|cg

Name of the consistency group. This is an mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

name|n

The name of the full copy. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume clone-show -pr myproj -cg df -name cd-1
volume URI is urn:storageos:Volume:262148ad-
d21c-49e2-9054-2bd0bfc45159:vdc1
resource URI is urn:storageos:BlockConsistencyGroup:
75502d0b-3104-48a9-b767-6a6cbfd2c05b:vdc1

{
   "allocated_capacity_gb": "0.00",
   "creation_time": 1436209890925,
   "device_label": "cd-1",
   "global": false,
   "high_availability_backing_volumes": [],
   "id": "urn:storageos:Volume:262148ad-
d21c-49e2-9054-2bd0bfc45159:vdc1",
   "inactive": false,
   "internal": false,
   "is_composite": false,
   "link": {
      "href": "/block/volumes/urn:storageos:Volume:262148ad-
d21c-49e2-9054-2bd0bfc45159:vdc1",
      "rel": "self"
   },
   "name": "cd-1",
   "native_id": "005FD",
   "pre_allocation_size_gb": "0.00",
   "project": {
      "id": "urn:storageos:Project:3dbb5faa-06dc-40b2-
a775-5af0e7c46dea:global",
      "link": {
         "href": "/projects/urn:storageos:Project:
3dbb5faa-06dc-40b2-a775-5af0e7c46dea:global",
         "rel": "self"
      }
   },
   "protection": {
      "full_copies": {
         "associated_source_volume": {
            "id": "urn:storageos:Volume:f01d74e2-d0cf-435b-
ba51-7f47c20d3ab1:vdc1",
            "link": {
```

```
                "href": "/block/volumes/
urn:storageos:Volume:f01d74e2-d0cf-435b-ba51-7f47c20d3ab1:vdc1",
                "rel": "self"
            }
         },
         "is_sync_active": true,
         "replicaState": "SYNCHRONIZED",
         "volumes": []
      }
   },
   "protocols": [
      "iSCSI"
   ],
   "provisioned_capacity_gb": "1.00",
   "remote": false,
   "requested_capacity_gb": "1.00",
   "storage_controller": "urn:storageos:StorageSystem:
8bd0af11-9a61-4bde-9df5-f583fa79b17b:vdc1",
   "storage_pool": {
      "id": "urn:storageos:StoragePool:71b7f979-866c-4835-
b0fa-5b8fee556238:vdc1",
      "link": {
         "href": "/vdc/storage-systems/urn:storageos:StorageSystem:
8bd0af11-9a61-4bde-9df5-f583fa79b17b:vdc1/storage-pools/
urn:storageos:StoragePool:71b7f979-866c-4835-
b0fa-5b8fee556238:vdc1",
         "rel": "self"
      }
   },
   "tags": [],
   "tenant": {
      "id": "urn:storageos:TenantOrg:989e7c38-836c-413a-
ac95-15b436c30e3d:global",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:
989e7c38-836c-413a-ac95-15b436c30e3d:global",
         "rel": "self"
      }
   },
   "thinly_provisioned": true,
   "varray": {
      "id": "urn:storageos:VirtualArray:e2661eb6-15f3-40f5-
b753-08eca26f68c3:vdc1",
      "link": {
         "href": "/vdc/varrays/
urn:storageos:VirtualArray:e2661eb6-15f3-40f5-
b753-08eca26f68c3:vdc1",
         "rel": "self"
      }
   },
   "vdc": {
      "id": "urn:storageos:VirtualDataCenter:d4127b9e-5b4d-4345-
aa35-8067c17bac2a:vdc1",
      "link": {
         "href": "/vdc/
urn:storageos:VirtualDataCenter:d4127b9e-5b4d-4345-
aa35-8067c17bac2a:vdc1",
         "rel": "self"
      }
   },
   "vpool": {
      "id": "urn:storageos:VirtualPool:389cee33-beba-4c33-
b818-82db0b83928a:vdc1",
      "link": {
         "href": "/block/vpools/urn:storageos:VirtualPool:389cee33-
beba-4c33-b818-82db0b83928a:vdc1",
         "rel": "self"
      }
```

```
    },
  "wwn": "60000970000198700420533030354644"
}
```

viprcli volume continuous_copies copy

Make a continuous-copy volume of the specified volume.

Syntax

```
viprcli volume continuous_copies copy
-name|n <volumename>
-project|pr <projectname>
-copyname|cp <volumecopyname>
[-count|cu <count>]
[tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli volume continuous_copies copy` command creates a continuous-copy volume of the specified volume. This command is silent on success.

Note

The volumes (continuous-copy) created using `viprcli volume continuous_copies copy` command can only be listed using `viprcli volume list` command and not using `viprcli volume continuous_copies list` command.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

copyname|cp

The name of the copy-volume. This is a mandatory parameter.

count|cu

The number of copies. This is an optional parameter.

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli volume continuous_copies delete

Delete the continuous-copy of a volume.

Syntax

```
viprcli volume continuous_copies delete
-name|n <volumename>
-project|pr <projectname>
-continuouscopyname|ccn <continuouscopyname>
[type|t <type>]
[tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli volume continuous_copies delete` command deletes the specified continuous-copy of a volume. The command is silent on success.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Name of the continuous-copy-volume. This is a mandatory parameter.

type|t

Protection type. Possible values are: native or rp. The default is native. This is an optional parameter.

Note

Protection type native must be understood same as mirror and rp must be understood same as recover point.

---

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 286

Example 286 (continued)

```
# viprcli volume continuous_copies delete -n vol01 -pr HR -ccn
contvol01 -t native
```

Use the list command to view result.

viprcli volume continuous_copies establish

Establishes the continuous-copying for a volume.

Syntax

```
viprcli volume continuous_copies establish
-name|n <volumename>
-project|pr <projectname>
-continuouscopyname|ccn <continuouscopyname>
[type|t <protectiontype>]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli volume continuous_copies establish` command establishes the continuous-copying for a specified volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Name of the continuous-copy-volume. This is a mandatory parameter.

type|t

Protection type. Possible values are: native or rp. The default is native. This is an optional parameter.

Note

Protection type native must be understood same as mirror and rp must be understood same as recover point.

---

tenant|tn

A valid tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli volume continuous_copies failover

Fail over continuous protection for a volume.

Syntax

```
    viprcli volume continuous_copies failover
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po
<port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -name|n <volumename>
    [-tenant|tn <tenantname>]
    -project|pr
<projectname>
    -continuouscopyname|ccn
<continuouscopyname>
    [-type|t <type>]
    [-pit|p <pit>]
```

Description

`viprcli volume continuous_copies failover` command fails over the continuous protection for a specified volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Name of the continuous-copy-volume. This is a mandatory parameter.

type|t

Protection type. Possible values are: native, rp, or srdf. The default is native. This is an optional parameter.

Note

Protection type native must be understood same as mirror and rp must be understood same as recover point.

---

pit|p

Any UTC point in time formatted as yyyy-MM-dd_HH:mm:ss or datetime in milliseconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 287

```
viprcli volume continuous_copies failover -name sg17924bvol1 -
project sanity -tenant standalone -continuouscopyname sg17924bvol1-
target-varray1 -type rp -pit 2016-01-19_12:03:00
```

viprcli volume continuous_copies failover-test

Fail over continuous protection test for a RecoverPoint protected volume.

Syntax

```
            viprcli volume continuous_copies failover-test
[-name|n <volume_name>]
[-project|pr <project_name>]
[-continuouscopyname|ccn <continuous_copy_name>]
[-tenant|tn <tenant_name>]
[-type|t <protection_type>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
            [-h]
```

Description

`viprcli volume continuous_copies failover-test` command performs a fail over continuous protection test on a volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

continuouscopyname|ccn

The name of the protected volume. This is a mandatory parameter.

tenant|tn

The name of the tenant.

type|t

Protection type is rp (RecoverPoint). The default is native. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli volume continuous_copies failover-test-cancel

Cancel a fail over continuous protection test for a RecoverPoint protected volume.

Syntax

```
            viprcli volume continuous_copies failover-test-cancel
[-name|n <volume_name>]
[-project|pr <project_name>]
[-continuouscopyname|ccn <continuous_copy_name>]
[-tenant|tn <tenant_name>]
[-type|t <protection_type>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli volume continuous_copies failover-test-cancel` command cancels a fail over continuous protection test on a volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

continuouscopyname|ccn

The name of the protected volume. This is a mandatory parameter.

tenant|tn

The name of the tenant.

type|t

Protection type is rp (RecoverPoint). The default is native. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli volume continuous_copies list

List the continuous-copies for a native or RecoverPoint protected volume.

Syntax

```
viprcli volume continuous_copies list
-name|n <volumename>
-project|pr <projectname>
[tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 288

```
# viprcli volume continuous_copies list  list -pr BLOCK_PROJECT -n
temp_vol_mirr1
  NAME     SOURCE_VOLUME VARRAY_NAME PROTOCOLS STORAGESYSTEM_NAME
  copy_mirr_vol1 temp_vol_mirr1 BANGALORE   FC        SYMMETRIX
+000195701573
```

Example 289

```
# viprcli volume continuous_copies list  list -pr BLOCK_PROJECT -n
temp_vol_mirr2
  NAME     SOURCE_VOLUME VARRAY_NAME PROTOCOLS STORAGESYSTEM_NAME
  copy_mirr_vol2 temp_vol_mirr2 BANGALORE   FC        SYMMETRIX
+000195701573
```

viprcli volume continuous_copies pause

Pause the continuous-copying for a volume.

Syntax

```
    viprcli volume continuous_copies pause
    -name|n <volumename>
    -project|pr
<projectname>
    -continuouscopyname|ccn
<continuouscopyname>
    [-split|sp <split> {true |
false}]
    [type|t <type>]
    [tenant|tn <tenant>]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-h]
```

Description

`viprcli volume continuous_copies pause` command pauses the continuous-copying for a specified volume.

Options

name|n

Name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Name of the continuous-copy-volume. This is a mandatory parameter.

split|sp

Synchronize the data between the mirrored volumes or between a source and target SRDF before splitting them into two separate entities. This is an optional parameter. Values are:

- true = synchronize
- false = do not synchronize (default)

type|t

Protection type. Possible values are: native, srdf, or rp. The default is native. This is a required parameter for SRDF types, but an optional parameter for other types.

Note

Protection type native must be understood same as mirror and rp must be understood same as recover point.

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 290

```
# viprcli volume continuous_copies pause -pr BLOCK_PROJECT -n
temp_vol_mirr -ccn test -t native
```

Use the show command to verify that the `syncState` value has changed.

```
# viprcli volume continuous_copies show -pr BLOCK_PROJECT -n
temp_vol_mirr -ccn test
{
   "creation_time": 1389337753032,
   "device_label": "test",
   "high_availability_backing_volumes": [],
   "id": "urn:storageos:BlockMirror:420bceae-c732-442e-a8b2-
b32db25b4773:",
   "inactive": false,
   "is_composite": false,
   "link": {
      "href": "/block/volumes/
urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:
      /protection/continuous-copies/urn:storageos:BlockMirror:
420bceae-c732-442e-a8b2-b32db25b4773:",
      "rel": "self"
   },
   "name": "test",
   "native_id": "0038C",
   "protection": null,
   "protocols": [
      "FC"
   ],
   "source": {
      "id":
"urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:",
      "link": {
         "href": "/block/volumes/
urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:",
         "rel": "self"
      },
      "name": "temp_vol_mirr"
   },
   "storage_controller": "urn:storageos:StorageSystem:
4d3451f2-1ec0-4f0c-9d8b-5bd61afeb9dd:",
   "syncState": "6",
   "syncType": "6",
   "tags": [],
   "varray": {
```

Example 290 (continued)

```
      "id": "urn:storageos:VirtualArray:5132e57a-f6c1-4152-8796-
bcf3b8004738:",
      "link": {
         "href": "/vdc/varrays/urn:storageos:VirtualArray:5132e57a-
f6c1-4152-8796-bcf3b8004738:",
         "rel": "self"
      }
   },
   "vpool": {
      "id": "urn:storageos:VirtualPool:b5eb9c9b-900d-470b-86c7-
e5eddb4cc157:",
      "link": {
         "href": "/block/vpools/
urn:storageos:VirtualPool:b5eb9c9b-900d-470b-86c7-e5eddb4cc157:",
         "rel": "self"
      }
   },
   "wwn": "60000970000195701573533030333843"
}
```

viprcli volume continuous_copies restore

Restores the continuous-copying for a volume.

Syntax

```
viprcli volume continuous_copies restore
-name|n <volumename>
-project|pr <projectname>
-continuouscopyname|ccn <continuouscopyname>
[type|t <protectiontype>]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli volume continuous_copies restore` command restores the continuous-copying for a specified volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Name of the continuous-copy-volume. This is a mandatory parameter.

type|t

Protection type. Possible values are: native or rp. The default is native. This is an optional parameter.

Note

Protection type native must be understood same as mirror and rp must be understood same as recover point.

tenant|tn

A valid tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli volume continuous_copies resume

Resume the continuous-copying for a volume.

Syntax

```
    viprcli volume continuous_copies resume
    -name|n <volumename>
    -project|pr
<projectname>
    -continuouscopyname|ccn
<continuouscopyname>
    [type|t <type>]
    [-tenant|tn
<tenantname>]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-h]
```

Description

`viprcli volume continuous_copies resume` command resumes the continuous-copying after it was paused for a specified volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Name of the continuous-copy-volume. This is a mandatory parameter.

type|t

Protection type. Possible values are: native or rp. The default is native. This is an optional parameter.

Note

Protection type native must be understood same as mirror and rp must be understood same as recover point.

tenant|tn

A valid tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 291

```
# viprcli volume continuous_copies resume -pr BLOCK_PROJECT -n
temp_vol_mirr -ccn test -t native
```

Use the show command to verify that the `syncState` value has changed.

```
# viprcli volume continuous_copies show -pr BLOCK_PROJECT -n
temp_vol_mirr -ccn test
{
   "creation_time": 1389337753032,
   "device_label": "test",
   "high_availability_backing_volumes": [],
   "id": "urn:storageos:BlockMirror:420bceae-c732-442e-a8b2-
b32db25b4773:",
   "inactive": false,
   "is_composite": false,
   "link": {
      "href": "/block/volumes/
urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:
      /protection/continuous-copies/urn:storageos:BlockMirror:
420bceae-c732-442e-a8b2-b32db25b4773:",
      "rel": "self"
   },
   "name": "test",
   "native_id": "0038C",
   "protection": null,
   "protocols": [
      "FC"
   ],
   "source": {
      "id":
"urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:",
      "link": {
         "href": "/block/volumes/
urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:",
         "rel": "self"
      },
      "name": "temp_vol_mirr"
   },
   "storage_controller": "urn:storageos:StorageSystem:
4d3451f2-1ec0-4f0c-9d8b-5bd61afeb9dd:",
   "syncState": "6",
   "syncType": "6",
   "tags": [],
   "varray": {
      "id": "urn:storageos:VirtualArray:5132e57a-f6c1-4152-8796-
```

Example 291 (continued)

```
bcf3b8004738:",
      "link": {
         "href": "/vdc/varrays/urn:storageos:VirtualArray:5132e57a-
f6c1-4152-8796-bcf3b8004738:",
         "rel": "self"
      }
   },
   "vpool": {
      "id": "urn:storageos:VirtualPool:b5eb9c9b-900d-470b-86c7-
e5eddb4cc157:",
      "link": {
         "href": "/block/vpools/
urn:storageos:VirtualPool:b5eb9c9b-900d-470b-86c7-e5eddb4cc157:",
         "rel": "self"
      }
   },
   "wwn": "60000970000195701573533030333843"
}
```

viprcli volume continuous_copies show

Return the details of continuous-copy for a volume.

Syntax

```
            viprcli volume continuous_copies show
            -name|n <volumename>
            -project|pr <projectname>
            -continuouscopyname|ccn
        <continuouscopyname>
            [tenant|tn <tenant>]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume continuous_copies show` command returns the details of continuous-copy for a specified volume.

Options

name|n

Name of the source volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Name of the continuous-copy-volume. This is a mandatory parameter.

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 292

```
# viprcli volume continuous_copies show -pr BLOCK_PROJECT -n
temp_vol_mirr -ccn test
{
   "creation_time": 1389337753032,
   "device_label": "test",
   "high_availability_backing_volumes": [],
   "id": "urn:storageos:BlockMirror:420bceae-c732-442e-a8b2-
b32db25b4773:",
   "inactive": false,
   "is_composite": false,
   "link": {
      "href": "/block/volumes/
urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:
      /protection/continuous-copies/urn:storageos:BlockMirror:
420bceae-c732-442e-a8b2-b32db25b4773:",
      "rel": "self"
   },
   "name": "test",
   "native_id": "0038C",
   "protection": null,
   "protocols": [
      "FC"
   ],
   "source": {
      "id":
"urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:",
      "link": {
         "href": "/block/volumes/
urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:",
         "rel": "self"
      },
      "name": "temp_vol_mirr"
   },
   "storage_controller": "urn:storageos:StorageSystem:
4d3451f2-1ec0-4f0c-9d8b-5bd61afeb9dd:",
   "syncState": "6",
   "syncType": "6",
   "tags": [],
   "varray": {
      "id": "urn:storageos:VirtualArray:5132e57a-f6c1-4152-8796-
bcf3b8004738:",
      "link": {
         "href": "/vdc/varrays/urn:storageos:VirtualArray:5132e57a-
f6c1-4152-8796-bcf3b8004738:",
         "rel": "self"
      }
   },
   "vpool": {
      "id": "urn:storageos:VirtualPool:b5eb9c9b-900d-470b-86c7-
e5eddb4cc157:",
      "link": {
         "href": "/block/vpools/
urn:storageos:VirtualPool:b5eb9c9b-900d-470b-86c7-e5eddb4cc157:",
         "rel": "self"
      }
```

Example 292 (continued)

```
    },
    "wwn": "60000970000195701573533030333843"
}
```

viprcli volume continuous_copies start

Start a continuous-copy for a specified volume.

Syntax

```
viprcli volume continuous_copies start
    -name|n <volumename>
    -project|pr
<projectname>
    -continuouscopyname|ccn
<continuouscopyname>
    [type|t <type> {native |
rp}]
    [-count|cu <count>]
    [tenant|tn <tenant>]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    [-h]
```

Description

`viprcli volume continuous_copies start` command starts a continuous-copy using the given name or label for the specified volume. The command is silent on success.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Continuous-copy-volume name to be created or a label for continuous-copy-volume. This is a mandatory parameter.

type|t

Protection type. Possible value are: native or rp. The default is native. This is an optional parameter.

Note

Protection type native must be understood same as mirror and rp must be understood same as recover point.

count|cu

The maximum number of continuous-copies. This is an optional parameter.

Note

Before setting up the count to a number using this command, you must have the following parameters of `viprcli vpools create` command set up appropriately based on the protection type:
For protection type native:

- [-maxcontinuouscopies|mcc] must be set to greater than '0' - zero. By default it is set to zero.
- [-expandable|ex] must be set to false. By default it is set to true.

For protection type rp:

- [-protectionvpool|vp] must be set.

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 293

```
# viprcli volume continuous_copies start -ccn copy_mirr_vol1 -pr
BLOCK_PROJECT -n temp_vol_mirr1 -t native
```

Use the list command to view result.

```
# viprcli volume continuous_copies list  list -pr BLOCK_PROJECT -n
temp_vol_mirr1
  NAME     SOURCE_VOLUME VARRAY_NAME PROTOCOLS STORAGESYSTEM_NAME
  copy_mirr_vol1 temp_vol_mirr1 BANGALORE   FC        SYMMETRIX
+000195701573
```

Example 294

```
# viprcli volume continuous_copies start -ccn copy_mirr_vol2 -pr
BLOCK_PROJECT -n temp_vol_mirr2 -t rp
```

Example 294 (continued)

Use the list command to view result.

```
# viprcli volume continuous_copies list  list -pr BLOCK_PROJECT -n
temp_vol_mirr2
  NAME     SOURCE_VOLUME VARRAY_NAME PROTOCOLS STORAGESYSTEM_NAME
  copy_mirr_vol2 temp_vol_mirr2 BANGALORE   FC        SYMMETRIX
+000195701573
```

viprcli volume continuous_copies stop

Stop the continuous-copy for a specified volume.

Syntax

```
            viprcli volume continuous_copies stop
            -name|n <volumename>
            -project|pr
        <projectname>
            -continuouscopyname|ccn
        <continuouscopyname>
            [type|t <type>]
            [-tenant|tn
        <tenantname>]
            [-hostname|hn
        <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume continuous_copies stop` command stops the continuous-copying for a specified volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Name of the continuous-copy-volume. This is a mandatory parameter.

type|t

Protection type. Possible values are: native or rp. The default is native. This is an optional parameter.

Note

Protection type native must be understood same as mirror and rp must be understood same as recover point.

tenant|tn

A valid tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 295

```
# viprcli volume continuous_copies stop -pr BLOCK_PROJECT -n
temp_vol_mirr -ccn test -t native
```

Use the show command to verify that the `syncState` value has changed.

```
# viprcli volume continuous_copies show -pr BLOCK_PROJECT -n
temp_vol_mirr -ccn test
{
   "creation_time": 1389337753032,
   "device_label": "test",
   "high_availability_backing_volumes": [],
   "id": "urn:storageos:BlockMirror:420bceae-c732-442e-a8b2-
b32db25b4773:",
   "inactive": false,
   "is_composite": false,
   "link": {
      "href": "/block/volumes/
urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:
      /protection/continuous-copies/urn:storageos:BlockMirror:
420bceae-c732-442e-a8b2-b32db25b4773:",
      "rel": "self"
   },
   "name": "test",
   "native_id": "0038C",
   "protection": null,
   "protocols": [
      "FC"
   ],
   "source": {
      "id":
"urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:",
      "link": {
         "href": "/block/volumes/
urn:storageos:Volume:ecee165a-931e-4084-8cf7-6687220c454c:",
         "rel": "self"
      },
      "name": "temp_vol_mirr"
   },
   "storage_controller": "urn:storageos:StorageSystem:
4d3451f2-1ec0-4f0c-9d8b-5bd61afeb9dd:",
   "syncState": "6",
   "syncType": "6",
   "tags": [],
   "varray": {
      "id": "urn:storageos:VirtualArray:5132e57a-f6c1-4152-8796-
```

Example 295 (continued)

```
bcf3b8004738:",
      "link": {
         "href": "/vdc/varrays/urn:storageos:VirtualArray:5132e57a-
f6c1-4152-8796-bcf3b8004738:",
         "rel": "self"
      }
   },
   "vpool": {
      "id": "urn:storageos:VirtualPool:b5eb9c9b-900d-470b-86c7-
e5eddb4cc157:",
      "link": {
         "href": "/block/vpools/
urn:storageos:VirtualPool:b5eb9c9b-900d-470b-86c7-e5eddb4cc157:",
         "rel": "self"
      }
   },
   "wwn": "60000970000195701573533030333843"
}
```

viprcli volume continuous_copies swap

Swap a continuous-copy for a specified volume.

Syntax

```
viprcli volume continuous_copies swap
[-name|n <volume_name>]
[-project|pr <project_name>]
[-continuouscopyname|ccn <continuous_copy_name>]
[-tenant|tn <tenant_name>]
[-type|t <protection_type>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli volume continuous_copies swap` command swaps a continuous-copy for a specified volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

continuouscopyname|ccn

The name of the protected volume. This is a mandatory parameter.

tenant|tn

The name of the tenant.

type|t

The type of protection.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli volume continuous_copies update-access-mode

Enable Direct Access for a specified volume.

Syntax

```
            viprcli volume continuous_copies update-access-mode
            -name|n <volumename>
            -project|pr
        <projectname>
[-accessmode|am <accessmode> {DIRECT_ACCESS}]
            -continuouscopyname|ccn
        <continuouscopyname>
            [type|t <type> {native |
        rp}]
             [tenant|tn <tenant>]
               [-h]
```

Description

`viprcli volume continuous_copies update-access-mode` command allows you to enable direct access. The command is silent on success.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

continuouscopyname|ccn

Continuous-copy-volume name to be created or a label for continuous-copy-volume. This is a mandatory parameter.

type|t

Protection type. Possible value are: native or rp. The default is native. This is an optional parameter.

Note

Protection type native must be understood same as mirror and rp must be understood same as recover point.

---

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

accessmode|am

Updates the access mode for RecoverPoint consistency groups only. Currently, the only supported value is `DIRECT_ACCESS`.

common arguments

This operation also takes the Common Parameters on page 24.

Example 296

```
# viprcli volume continuous_copies update-access-mode -name
sg17924bvol1 -project sanity -tenant standalone -continuouscopyname
sg17924bvol1-target-varray1 -type rp -accessmode DIRECT_ACCESS
```

viprcli volume create

Create a volume with the given parameters.

Syntax

```
            viprcli volume create
            [-h]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <volumename>
            -size|s <volumesize
        [kKmMgGtT]>
            -project|pr
        <projectname>
            [-tenant|tn
        <tenantname>]
            -vpool|vp <vpoolname>
            -varray|va <varray>
            [-count|cu <count>]
  <consistentgroupname>]
            [-consistencygroup|cg;]
   <portgroup>]
            [-portgroup|port_group_name;]
            [-serialnumber|sn] <serialnumber>
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

`viprcli volume create` command creates one or more volumes. This volume can be added to an export group so that users can access it.

Note

An AD user can belong to only one tenant at a time and therefore, avoid using the tenant (tn) parameter when creating a volume under a project, if you are an AD user with an ACL 'All' privilege.

Options

name|n

The name of the volume. This is a mandatory parameter.

size|s

Size is in bytes. Size can be followed by a size specifier (K or k, M or m, G or g, or T or t). K implies multiples of 1024 bytes, M implies multiples of (1024 * 1024) bytes, and so on. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

vpool|vp

A valid virtual pool name. This is a mandatory parameter.

varray|va

A valid virtual array name. This is a mandatory parameter.

count|cu

Count. This specifies the number of volumes to be created. This is an optional parameter.

consistencygroup|cg

The name of the consistency group. This is an optional parameter.

portgroup name|port_group name

The name of the port group. This is an optional parameter.

serial number|sn

The serial number of the port group's storage system. This is an optional parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters.

Example 297

```
viprcli volume create -n VNX_VOL-1 -s 1K
-project TEST_PROJECT11 -vp TEST_BLOCK_GOLD_VNX -va hopkinton
-protocol FC -sync

viprcli volume list -pr TEST_PROJECT11
```

Example 297 (continued)

```
Name                    Capacity       Protocols
B_VOL                   1024           FC
B_VOL-2-1               1024           FC
B_VOL-2-2               1024           FC
C_VOL                   10485          FC
TEST_VNX_FC_VOLUME1     1073741824     FC
VNX_VOL1                1024           FC
```

Example 298

```
viprcli volume create -n VNX_VOL-1 -s 1K
-project TEST_PROJECT11 -vp TEST_BLOCK_GOLD_VNX -va hopkinton
-protocol FC  -count 2

viprcli volume list -pr TEST_PROJECT11 -l

Name      Capacity  Maxsnapshots  MultivolumeConsistency
Protocols
B_VOL     1024      2             false                     FC
B_VOL2    1024      2             false                     FC
B_VOL     1024      2             false                     FC
C_VOL     1048576   2             false                     FC
VNX_VOL1  1024      2             false                     FC
```

Example 299

- Create volume with SRDF pool

```
viprcli volume create -n mysrdfvol -s 4G -va myvarray -pr sanity -
vp srdfpool1 -sync

viprcli volume list -l -pr sanity

NAME
                      PROVISIONED_CAPACITY_GB  PROTOCOLS
THINLY_PROVISIONED    TAGS
mysrdfvol

4.00
  FC         False
mysrdfvol-target-myvarray
4.00
  FC         False
```

Example 300

- Volume create with optional parameter portgroup

```
volume create -n New_HR_vol -pr HR -va v197test -vpool 197 -s 1g -
sn 197 -portgroup PG_HR
```

viprcli volume delete

Delete a volume.

Syntax

```
viprcli volume delete
    [-h]
    [-hostname|hn
<hostname>]
    [-port|po <port_number>]
    [-portui|pu
<ui_port_number>]
    [-cookiefile|cf
<cookiefile>]
    -name|n <volumename>
[<volumename> ...]
    [-tenant|tn <tenantname>]
    -project|pr
<projectname>
    [-synchronous|sync]
    [-synctimeout|syncto]
    [-forceDelete|fd]
    [-vipronly|vo]
```

Description

`viprcli volume delete` command succeeds only if the volume is not exported. Any snapshots associated with the volume will be deleted when the volume is deleted.

Note

Before you can delete the volume, you must detach any clones of the volume using viprcli volume clone-detach.

Options

name|n

A valid ViPR Controller volume name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

forceDelete|fd

Forcibly delete the volume. This is an optional parameter.

vipronly|vo

Delete the volume only from ViPR Controller. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 301

```
# viprcli volume delete -pr TEST_PROJECT11 -n B_VOL-2-2 VNX_VOL-1-1
```

viprcli volume expand

Expand the size of a volume.

Syntax

```
            viprcli volume expand
            [-h]
            [-hostname|hn <hostname>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            -name|n <volumename>
            [-tenant|tn
        <tenantname>]
            -project|pr
        <projectname>
            -size|s
        <volumesize[kKmMgGtT]>
            [-synchronous|sync]
            [-synctimeout|syncto]
```

Description

`viprcli volume expand` command expands the size of a volume.

Options

name|n

A valid ViPR Controller volume name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

size|s

The new size of the file system. This value must be higher than the current size of the file system, or the operation will fail. By default, this value is in bytes. You can

append k, m, g, or t to the numeric value to allocate kilobytes, megabytes, gigabytes or terabytes. This is a mandatory parameter.

synchronous|sync

Perform the file system creation synchronously. If this argument is not specified, the file system creation will be performed asynchronously. This is an optional parameter.

synctimeout|syncto

Sync timeout in seconds. This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 302

```
viprcli volume expand -s 123 -n VNX_VOL-1-2 -pr TEST_PROJECT11
error: Incorrect value of new size: 123 bytes
New size must be greater than current size: 1024 bytes
viprcli volume expand -s 2m -n VNX_VOL-1-2 -pr TEST_PROJECT11 -sync
```

viprcli volume list

Return a list of all volumes in the given project.

Syntax

```
viprcli volume list
-project|pr <projectname>
[-tenant|tn <tenantname>]
[-verbose|v]
[-long|l]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli volume list` command returns a list of all volumes in the given project.

Note

The `viprcli volume list` command also lists the volumes (continuous copy) created using `viprcli volume continuous_copies copy` command.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

tenant|tn

The name of a tenant. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

{verbose|v | long|l}

These options provide additional information. Use either verbose or long and not both. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 303

```
viprcli volume list -pr ProjectName -tn TenantName -v
[ {
      "capacity": "1073741824",
      "vpool": "urn:storageos:VirtualPool:
      180b6819-d5d4-4f38-a687-510af785f27c:",
      "vpoolParams": {
         "entry": [
            {
               "key": "resiliency",
               "value": "5"
            },
            {
               "key": "performance",
               "value": "Undefined"
            },
            {
               "key": "multiVolumeConsistency",
               "value": "false"
            },
            {
               "key": "maxSnapshots",
               "value": "2"
            },
            {
               "key": "numPaths",
               "value": "2"
            }
         ]
      },
      "id": "urn:storageos:Volume:
      70f06889-127a-40fe-a1dc-afe6c7e92595:",
      "inactive": "false",
      "name": "testvol",
      "varray": "urn:storageos:VirtualArray:
     87078d8b-504f-4613-b1c1-39554257c653:",
      "operationStatus": {
         "entry": {
            "key": "ebc9fe85-2b3a-410e-b312-d0077511d280",
            "value": {
               "message": "Created volume 7 named 'testvol'
                from pool Thin Pool 1",
               "status": "ready"
            }
         }
      },
      "project": "urn:storageos:Project:
      8d13e649-d85e-4842-a2d0-03924a949ceb:",
      "protocols": {
         "protocols": "FC"
      },
      "storageController": "urn:storageos:StorageDevice:
```

Example 303 (continued)

```
      4deda0ae-cbaf-4828-b361-b9637fe0b71f:"
   }
]
```

viprcli volume migration-cancel

Cancel the specified data migration, currently in progress.

Syntax

```
            viprcli volume migration-cancel
            -migration_id|id
        <migration_id>
            [-hostname|hn <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume migration-cancel` command cancels the specified data migration, currently in progress.

Options

migration_id|id

The migration ID. This parameter is mandatory.

common arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume  migration-cancel -id urn:storageos:Migration:
03416849-daf8-4ff1-bee2-5fbbc4a102b1:vdc1
{
   "associated_resources": [],
   "creation_time": 1449175021173,
   "description": "cancel migration",
   "global": false,
   "id": "urn:storageos:Task:380eaab8-109b-4bd0-81b5-
c7137766bc9b:vdc1",
   "inactive": false,
   "internal": false,
   "link": {
      "href": "/vdc/tasks/urn:storageos:Task:
380eaab8-109b-4bd0-81b5-c7137766bc9b:vdc1",
      "rel": "self"
   },
   "name": "CANCEL MIGRATION",
```

```
   "op_id": "62fe896b-829b-43ed-b14d-870325f42adb",
   "progress": 0,
   "remote": false,
   "resource": {
      "id": "urn:storageos:Volume:8768ec55-8fa9-45e8-a334-
a8a01f601026:vdc1",
      "link": {
         "href": "/block/volumes/urn:storageos:Volume:
8768ec55-8fa9-45e8-a334-a8a01f601026:vdc1",
         "rel": "self"
      },
      "name": "KRISFS"
   },
   "start_time": 1449175021172,
   "state": "pending",
   "tags": [],
   "tenant": {
      "id": "urn:storageos:TenantOrg:
6acd8ff4-09ed-40ba-9ad2-0cff4e6824f7:global",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:
6acd8ff4-09ed-40ba-9ad2-0cff4e6824f7:global",
         "rel": "self"
      }
   },
   "vdc": {
      "id": "urn:storageos:VirtualDataCenter:18f249eb-
c695-4780-807d-1afdf1820a6f:vdc1",
      "link": {
         "href": "/vdc/urn:storageos:VirtualDataCenter:18f249eb-
c695-4780-807d-1afdf1820a6f:vdc1",
         "rel": "self"
      }
   }
}
```

viprcli volume migration-deactivate

Remove the completed data migration from ViPR Controllerand VPLEX .

Syntax

```
            viprcli volume migration-deactivate
            -migration_id|id
        <migration_id>
            [-hostname|hn <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume migration-deactivate` command removes the completed data migration from ViPR Controllerand VPLEX .

Options

migration_id|id

The migration ID. This parameter is mandatory.

common arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume migration-deactivate -id urn:storageos:Migration:
7a9e794b-1eb2-497d-818a-c76a244a4288:vdc1
{
   "associated_resources": [],
   "creation_time": 1449152326802,
   "description": "delete migration",
   "global": false,
   "id": "urn:storageos:Task:f57f737d-
bdb9-4f71-924d-0d02541c2646:vdc1",
   "inactive": false,
   "internal": false,
   "link": {
      "href": "/vdc/tasks/urn:storageos:Task:f57f737d-
bdb9-4f71-924d-0d02541c2646:vdc1",
      "rel": "self"
   },
   "name": "DELETE MIGRATION",
   "op_id": "27ae6831-b3a6-4e0c-ad0a-b63b4d8ee6ef",
   "progress": 0,
   "remote": false,
   "resource": {
      "id": "urn:storageos:Volume:8768ec55-8fa9-45e8-a334-
a8a01f601026:vdc1",
      "link": {
         "href": "/block/volumes/urn:storageos:Volume:
8768ec55-8fa9-45e8-a334-a8a01f601026:vdc1",
         "rel": "self"
      },
      "name": "KRISFS"
   },
   "start_time": 1449152326800,
   "state": "pending",
   "tags": [],
   "tenant": {
      "id": "urn:storageos:TenantOrg:
6acd8ff4-09ed-40ba-9ad2-0cff4e6824f7:global",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:
6acd8ff4-09ed-40ba-9ad2-0cff4e6824f7:global",
         "rel": "self"
      }
   },
   "vdc": {
      "id": "urn:storageos:VirtualDataCenter:18f249eb-
c695-4780-807d-1afdf1820a6f:vdc1",
      "link": {
         "href": "/vdc/urn:storageos:VirtualDataCenter:18f249eb-
c695-4780-807d-1afdf1820a6f:vdc1",
         "rel": "self"
      }
   }
}
```

viprcli volume migration-list

Return a list of all data migrations.

Syntax

```
            viprcli volume migration-list
            [-hostname|hn <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume migration-list` command returns a list of all of the data migrations.

Options

common arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume migration-list
{
"migration": [
{
"id": "urn:storageos:Migration:73f64940-21bd-4e4d-9824-
fdfc2b8cebb1:vdc1",
"link":
{ "href": "/block/migrations/urn:storageos:Migration:
73f64940-21bd-4e4d-9824-fdfc2b8cebb1:vdc1", "rel": "self" }
,
"name": "M_151114-032854-120"
},
{
"id": "urn:storageos:Migration:9a7c4271-d59c-40be-
aab9-9af546f00348:vdc1",
"link":
{ "href": "/block/migrations/urn:storageos:Migration:9a7c4271-
d59c-40be-aab9-9af546f00348:vdc1", "rel": "self" }
,
"name": "M_151114-030339-859"
},
{
"id": "urn:storageos:Migration:f92eb130-abdf-4df4-86cc-
f21478ca7e75:vdc1",
"link":
{ "href": "/block/migrations/urn:storageos:Migration:f92eb130-
abdf-4df4-86cc-f21478ca7e75:vdc1", "rel": "self" }
,
"name": "M_151114-031418-805"
}
```

```
]
}
```

viprcli volume migration-pause

Pause the specified data migration, currently in progress.

Syntax

```
            viprcli volume migration-pause
            -migration_id|id
        <migration_id>
            [-hostname|hn <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume migration-pause` command pauses the specified data migration, currently in progress.

Options

migration_id|id

The migration ID. This parameter is mandatory.

common arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume migration-pause -id
"urn:storageos:Migration:df0e8ff2-92d4-4cfc-adb8-db229a337d6c:vdc1"
{
   "associated_resources": [],
   "creation_time": 1447472705345,
   "description": "puase migration",
   "global": false,
   "id": "urn:storageos:Task:2cb914c9-9ec1-4686-ae62-
ec8ac9a74015:vdc1",
   "inactive": false,
   "internal": false,
   "link": {
      "href": "/vdc/tasks/urn:storageos:Task:2cb914c9-9ec1-4686-
ae62-ec8ac9a74015:vdc1",
      "rel": "self"
   },
   "name": "PAUSE MIGRATION",
   "op_id": "d6f5d107-a427-4ea1-b9f0-c7928b01699d",
   "progress": 0,
   "remote": false,
```

```
   "resource": {
      "id": "urn:storageos:Volume:2b5cb82d-725c-4786-8c89-
a679df099c38:vdc1",
      "link": {
         "href": "/block/volumes/urn:storageos:Volume:
2b5cb82d-725c-4786-8c89-a679df099c38:vdc1",
         "rel": "self"
      },
      "name": "XIOBLK"
   },
   "start_time": 1447472705344,
   "state": "pending",
   "tags": [],
   "tenant": {
      "id": "urn:storageos:TenantOrg:
6acd8ff4-09ed-40ba-9ad2-0cff4e6824f7:global",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:
6acd8ff4-09ed-40ba-9ad2-0cff4e6824f7:global",
         "rel": "self"
      }
   },
   "vdc": {
      "id": "urn:storageos:VirtualDataCenter:18f249eb-
c695-4780-807d-1afdf1820a6f:vdc1",
      "link": {
         "href": "/vdc/urn:storageos:VirtualDataCenter:18f249eb-
c695-4780-807d-1afdf1820a6f:vdc1",
         "rel": "self"
      }
   }
```

viprcli volume migration-resume

Resume the specified data migration, that was previously paused using `viprcli volume migration-pause`.

Syntax

```
            viprcli volume migration-resume
            -migration_id|id
        <migration_id>
            [-hostname|hn <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume migration-resume` command resumes the specified data migration, which was previously paused.

Options

migration_id|id

The migration ID. This parameter is mandatory.

common arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume  migration-resume -id urn:storageos:Migration:
03416849-daf8-4ff1-bee2-5fbbc4a102b1:vdc1
{
   "associated_resources": [],
   "creation_time": 1449174913436,
   "description": "resume migration",
   "global": false,
   "id": "urn:storageos:Task:1b33c4ed-
a276-46d9-9662-3f6a3e91b6c6:vdc1",
   "inactive": false,
   "internal": false,
   "link": {
      "href": "/vdc/tasks/urn:storageos:Task:1b33c4ed-
a276-46d9-9662-3f6a3e91b6c6:vdc1",
      "rel": "self"
   },
   "name": "RESUME MIGRATION",
   "op_id": "d692afac-879d-4e45-a187-51bc8d36facc",
   "progress": 0,
   "remote": false,
   "resource": {
      "id": "urn:storageos:Volume:8768ec55-8fa9-45e8-a334-
a8a01f601026:vdc1",
      "link": {
         "href": "/block/volumes/urn:storageos:Volume:
8768ec55-8fa9-45e8-a334-a8a01f601026:vdc1",
         "rel": "self"
      },
      "name": "KRISFS"
   },
   "start_time": 1449174913435,
   "state": "pending",
   "tags": [],
   "tenant": {
      "id": "urn:storageos:TenantOrg:
6acd8ff4-09ed-40ba-9ad2-0cff4e6824f7:global",
      "link": {
         "href": "/tenants/urn:storageos:TenantOrg:
6acd8ff4-09ed-40ba-9ad2-0cff4e6824f7:global",
         "rel": "self"
      }
   },
   "vdc": {
      "id": "urn:storageos:VirtualDataCenter:18f249eb-
c695-4780-807d-1afdf1820a6f:vdc1",
      "link": {
         "href": "/vdc/urn:storageos:VirtualDataCenter:18f249eb-
c695-4780-807d-1afdf1820a6f:vdc1",
         "rel": "self"
      }
   }
}
```

viprcli volume migration-show

List the details of the specified data migration.

Syntax

```
            viprcli volume migration-show
            -migration_id|id
        <migration_id>
            [-hostname|hn <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume migration-show` command lists the details of the specified data migration.

Options

migration_id|id

The migration ID. This parameter is mandatory.

common arguments

This operation also takes the Common Parameters on page 24.

```
viprcli volume migration-show -id "urn:storageos:Migration:f92eb130-
abdf-4df4-86cc-f21478ca7e75:vdc1"
{
"global": null,
"percent_done": "100",
"remote": null,
"source": {
"id": "urn:storageos:Volume:32410086-
b791-4cd9-83be-0d5f5050ec07:vdc1",
"link":
{ "href": "/block/volumes/urn:storageos:Volume:32410086-
b791-4cd9-83be-0d5f5050ec07:vdc1", "rel": "self" }
},
"start_time": "Sat Nov 14 02:57:55 UTC 2015",
"status": "committed",
"tags": [],
"target": {
"id": "urn:storageos:Volume:925a5096-a010-4c4b-
ab84-08cac5d4e94e:vdc1",
"link":
{ "href": "/block/volumes/urn:storageos:Volume:925a5096-a010-4c4b-
ab84-08cac5d4e94e:vdc1", "rel": "self" }
},
"vdc": null,
"volume": {
"id": "urn:storageos:Volume:2b5cb82d-725c-4786-8c89-
```

```
a679df099c38:vdc1",
"link":
{ "href": "/block/volumes/urn:storageos:Volume:
2b5cb82d-725c-4786-8c89-a679df099c38:vdc1", "rel": "self" }
}
}
```

viprcli volume protectionset show

Return the protection set details for a volume.

Syntax

```
viprcli volume protectionset show
-name|n <volumename>
-project|pr <projectname>
[tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli volume protectionset show` command returns the protection set details for a volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli volume show

Return a detailed listing of the volume.

Syntax

```
viprcli volume show
-name|n <volumename>
```

```
            -project|pr
        <projectname>
            [-tenant|tn
        <tenantname>]
            [-xml]
            [-hostname|hn
        <hostname>]
            [-port|po
        <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-cookiefile|cf
        <cookiefile>]
            [-h]
```

Description

`viprcli volume show` command returns a detailed listing of the volume.

Note

If your volume name, project name or tenant name contains a forward-slash character (/), then use the `viprcli volume list` command instead of the `viprcli volume show` command.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 304

```
# viprcli volume show -tn TEST_TENANT11 -pr TEST_PROJECT11 -n
TEST_VNX_FC_VOLUME1
{
   "capacity": "1073741824",
   "vpool": "urn:storageos:VirtualPool:
180b6819-d5d4-4f38-a687-510af785f27c:",
   "vpoolParams": {
      "entry": [
         {
            "key": "resiliency",
            "value": "5"
         },
         {
```

Example 304 (continued)

```
            "key": "performance",
            "value": "Undefined"
         },
         {
            "key": "multiVolumeConsistency",
            "value": "false"
         },
         {
            "key": "maxSnapshots",
            "value": "2"
         },
         {
            "key": "numPaths",
            "value": "2"
         }
      ]
   },
   "id": "urn:storageos:Volume:
    70f06889-127a-40fe-a1dc-afe6c7e92595:",
   "inactive": "false",
   "name": "testvol",
   "varray": "urn:storageos:VirtualArray:
    87078d8b-504f-4613-b1c1-39554257c653:",
   "operationStatus": {
      "entry": {
         "key": "ebc9fe85-2b3a-410e-b312-d0077511d280",
         "value": {
            "message": "Created volume 7 named 'testvol' from
            pool Thin Pool 1",
            "status": "ready"
         }
      }
   },
   "project": "urn:storageos:Project:
   8d13e649-d85e-4842-a2d0-03924a949ceb:",
   "protocols": {
      "protocols": "FC"
   },
   "storageController": "urn:storageos:StorageDevice:
   4deda0ae-cbaf-4828-b361-b9637fe0b71f:"
}
```

viprcli volume tag

Add or remove tags to a volume.

Syntax

```
viprcli volume tag
-name|n <volumename>
-project|pr <projectname>
[-add <add_tags>]
[-remove <remove_tags>]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
```

```
            [-h]
```

Description

`viprcli volume tag` command add or removes tags to a volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

add

Add a tag to the volume. This is an optional parameter.

remove

Delete a tag from the volume. This is an optional parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli volume tasks

Check the status of an asynchronous task.

Syntax

```
        viprcli volume
tasks
        -project|pr <projectname>
        [-tenant|tn <tenantname>]
        [-name|n <filesystemname>]
        [-id <operationID>]
        [-verbose|v]
        [-hostname|hn <hostname>]
        [-port|po <port_number>]
        [-portui|pu <ui_port_number>]
        [-cookiefile|cf <cookiefile>]
        [-h]
```

Description

`viprcli volume tasks` command checks the status of an asynchronous task for a volume.

Options

project|pr

The name of a valid ViPR Controller project. This is a mandatory parameter.

tenant|tn

A valid tenant name. This is an optional parameter.

name|n

A valid file system name. This is an optional parameter.

id

An operation ID. Each asynchronous operation within ViPR Controller issues an operation id. This id must be provided through this CLI operation to retrieve information about an ongoing asynchronous task. This is an optional parameter.

verbose|v

Show additional details about asynchronous tasks. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 305

```
viprcli volume tasks
```

OP_ID	NAME	STATE
0bd5832d-5f3f-4cde-9f88-1b7579172b61	B_VOL-2-2	ready
0bd5832d-5f3f-4cde-9f88-1b7579172b61	VNX_VOL-1-1	ready
39b6265b-3273-4cff-a773-3f14f1b1c6fa	B_VOL	ready
54372cc0-21c0-4ecc-8847-0aa53fd0a641	B_VOL-2-1	ready
54372cc0-21c0-4ecc-8847-0aa53fd0a641	B_VOL-2-2	ready
60d7e9b5-4563-4b46-b65b-802095894fe9	TEST_VNX_FC_VOLUME1	error
74435737-af8d-4f85-bdea-107884a300ad	C_VOL	ready
809bdfe1-65e5-40a8-8033-d5d14b365275	TEST_VNX_FC_VOLUME1	error
84ea89df-2db4-4a8e-bbbe-e64b4efbda24	VNX_VOL-1-1	ready
84ea89df-2db4-4a8e-bbbe-e64b4efbda24	VNX_VOL-1-2	ready
a90954a2-8b45-4369-93bb-4a19eefdd000	VNX_VOL-1	ready
c285cc82-36ea-47da-8ed6-45a56ef5bc16	VNX_VOL-1-2	ready
fee0a360-32de-4fd6-908f-09a68455ece9	C_VOL	ready

viprcli volume unmanaged ingest

Ingest unmanaged volumes into ViPR Controller and provides the datacenter and vcenter options if the cluster is a part of Datacenter-Vcenter.

Syntax

```
            viprcli volume unmanaged ingest
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookiefile>]
[-vpool|vp <vpoolname>[
[-project|pr        <projectname>[
[-volumes|vol <volume>        [<volume>...]
[-varray|va <varray>]
[-tenant|tn <tenant>]
[-hostlabel|hl <host        label>]
[-cluster|cl <cluster        name>]
[-ingestmethod |inmd        <ingest_method>]
[-datacentername|dc <datacentername>]
[-vcentername|vc <vcentername>]
```

Description

`viprcli volume unmanaged ingest` command ingests unmanaged volumes into ViPR Controller.

Options

vpool|vp

Name of the virtual storage pool. This is a mandatory parameter.

project|pr

A valid project name. This is a mandatory parameter.

volumes|vol

Name or Id of the volumes. This is a mandatory parameter.

varray|va

The name of a virtual storage array. This is a mandatory parameter.

tenant|tn

A valid tenant name. If a tenant name is not specified, the default parent tenant is taken. This is an optional parameter.

hostlabel|hl

The name of client host to which volume is exported. This is an optional parameter.

cluster|cl

The name of the cluster. This is an optional parameter.

ingestmethod

The full value (default) ingests the virtual volumes and the backend volumes. The ingest only virtual volumes value ingests only the virtual volumes.

datacentername|dc

The datacenter name . This is an optional parameter.

vcentername|vc

The vCenter name . This is an optional parameter.

Common Arguments

This operation also takes the Common Parameters on page 24.

viprcli volume unmanaged show

Return the details of unmanaged volumes.

Syntax

```
viprcli volume unmanaged show
-volume|vol <volume> [<volume>...]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookiefile>]
[-h]
```

Description

`viprcli volume unmanaged show` command returns the details of unmanaged volumes.

Options

volume|vol

The name or Id of the volume. This is a mandatory parameter.

common arguments

This operation also takes the Common Parameters on page 24.

viprcli volume update

Update a volume.

Syntax

```
viprcli volume update
[-name|n <volume_name>]
[-project|pr <project_name>]
[-vpool|vp <virtual_pool_name>]
[-tenant|tn <tenantname>]
[-hostname|hn <hostname>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-cookiefile|cf <cookie_file>]
[-h]
```

Description

`viprcli volume update` command updates a volume.

Options

name|n

The name of the volume. This is a mandatory parameter.

project|pr

The name of the project. This is a mandatory parameter.

vpool|vp

The name of the new virtual pool. This is a mandatory parameter.

tenant|tn

The name of the tenant.

common arguments

This operation also takes the Common Parameters on page 24.

CHAPTER 53

viprcli vpool commands

This chapter contains the following topics:

- viprcli vpool create 688
- viprcli vpool list 694
- viprcli vpool show 696
- viprcli vpool update 698
- viprcli vpool delete 702
- viprcli vpool allow 703
- viprcli vpool disallow 704
- viprcli vpool add_pools 705
- viprcli vpool refresh_pools 706
- viprcli vpool get_pools 707
- viprcli vpool remove_pools 708

viprcli vpool create

Create a virtual storage pool in ViPR Controller. A virtual storage pool is a group of storage pools according to their attributes and properties.

Syntax

```
            viprcli vpool create
[-h]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu    <ui_port_number>]
[-cookiefile|cf    <cookie_file>]
[-name|n <vpoolname>
[-protocol|pl <protocol>    [<protocol. ...]
[-varrays|va <varray>    [<varrays> ...]
[-provisiontype|pt <provisiontype>    { Thick | Thin }]
[-maxsnapshots|msnp    <max_snapshots>]
[-maxretention|mxrtn        <max_retention>]
[-mindatacenters|mndcs        <min_datacenters>]
[-longtermretention|ltrtn    <long_term_retention>]
[-maxcontinuouscopies|mcc        <max_continuos_copies>]
[-continuouscopiesvpool|ccv    <continuoscopies_vpool>]
[-highavailability|ha    <highavailability>]
[-rp <rp> [<rp>    ...]]
[-rp_source_policy|rp_policy    <rp_source_policy>]
[-metropoint {true | false}]
[-activeProtectionAtHASite {true | false}]
[-file_replication_policy|frpol    <file_replication_policy>]
[-file_replication_copies    <file_replication_copies>
[<file_replication_copies> ...]]
[-systemtype|st <systemtype>    {isilon | vnxblock | vnxfile | vmax
| netapp | netappc |vplex | hds | openstack
        | scaleio | xtremio | ddmc | ibmxiv | vnxe | datadomain}]
[-raidlevel|rl <raidlevel>    [<raidlevel> ...]{RAID1 | RAID2 |
RAID3 | RAID4 | RAID5 | RAID6 |
        RAID10}]
[-fastpolicy|fp    <fastpolicy>]
[-drivetype|dt <drivetype> {SSD |        FC | SAS | NL_SAS | SATA |
NONE}]
[-type|t <vpooltype> { file |    block }]
[-description|desc]        <description>
[-usematchedpools|ump        <useMatchedPools>]
[-multivolconsistency|mvc        {true,false}]
[-expandable|ex <expandable>        ]
[-autoCrossConnectExport|acc        <autoCrossConnectExport> {true
| false}]
[-fastexpansion|fe        <fastexpansion>
[-autotierpolicynames|apn        <unique_auto_tier_policy_names>> ]
[-enablecompression <enable_compression>]
[-maxpaths|mxp        <max_paths>]
[-minpaths|mnp        <min_paths>]
[thinpreallocper|tpap        <thinpreallocper>]
[frontendbandwidth|febw        <frontendbandwidth>]
[iopersec|iops        <iopersec>]
[-pathsperinitiator|ppi        <PathsPerInitiator>]
[-snapshotschedule|snapsched        {true,false}]
[-srdf <SRDF> [<srdf>        ...]]
[-placementpolicy|pp <placementpolicy>]
```

Options

name|n

The name of the virtual storage pool. This is a mandatory parameter.

protocol|pl

For file storage, Protocol can be `NFS`, `NFSv4`,`CIFS` or `ScaleIO`. For block storage, Protocol is `FC`, `iSCSI`, or `ScaleIO`. This is a mandatory parameter.

varrays|va

The list of virtual storage arrays where the primary storage is available (empty set means available in all virtual storage arrays). This is a mandatory parameter.

provisiontype|pt

Provision type. Type of `Thin` or `Thick`. This is an optional parameter.

Note

This is a mandatory parameter for block storage type. If you are creating this virtual pool for VMAX journal volumes, `provisiontype` must be set to `Thick`, as ViPR Controller does not pre-allocate the volumes.

---

maxsnapshots|msnp

The maximum number of native snapshots. This must be greater than '0'. This is a mandatory parameter to create a snapshot. This is an optional parameter.

maxretention|mxrtn

Maximum retention period. This is an optional parameter.

mindatacenters|mndcs

Minimum number of data centers. This is an optional parameter.

longtermretention|ltrtn

Long term retention. This is an optional parameter.

maxcontinuouscopies|mcc

The maximum number of native continuous copies. You can set this only for a vpool of type Block. This is an optional parameter.

Note

This is set to zero '0' by default. You must set it to a value greater than zero and set `expandable` to **`false`**, if you want to create continuous copy of a volume using `viprcli volume continuous_copies create` command with the protection type as native.

---

continouscopiesvpool|ccv

The vpool name for continuous copies. You can set this only for a vpool of type Block. This is an optional parameter.

Note

Optionally, you can set this parameter, if you want to create continuous copy of a volume using `viprcli volume continuous_copies create` command with the protection type as native.

highavailability|ha

High-availability is available only for VPLEX arrays. The possible values are vplex_local or vplex_distributed. This is an optional parameter.

Note

For vplex_distributed you must also specify varray:vpool with high-availability type.

rp

RecoverPoint target. Specify in the form set of varray:vpool:journalsize. This is an optional parameter.

rp_source_policy|rp_policy

Sets parameters for RecoverPoint copies. The syntax is:

```
remotecopymode:rpovalue:rpotype:journalsize:journal_varray:journ
al_vpool:standby_journal_varray:standby_journal_vpool
```

- `remotecopymode` is the remote copy mode for the RecoverPoint CG. Values are: `ASYNCHRONOUS` (the default) or `SYNCHRONOUS`. When using `SYNCHRONOUS`, the next two parameters must be `None`, as follows:

  ```
  SYNCHRONOUS:None:None
  ```

- `rpovalue` is the value sent to RP. Use the value `None` if mode is `SYNCHRONOUS`.
- `rpotype` is the RPO unit. Default is SECONDS. Value values are: `SECONDS MINUTES HOURS WRITES BYTES KB MB GB TB`. Use the value `None` if mode is `SYNCHRONOUS`.
- `journalsize` is journal size.
- `journal_varray` is an optional separate virtual array for the RecoverPoint copy journal volume. The default is the same virtual array as the RecoverPoint copy.
- `journal_vpool` is an optional separate virtual pool for the RecoverPoint copy journal volume. The default is the same virtual pool as the RecoverPoint copy.
- `standby_journal_varray` is the target virtual array.
- `standby_journal_vpool` is the target virtual pool.

This is an optional parameter.

metropoint

Flag to indicate whether or not MetroPoint configuration will be used. Specify `true` or `false` as the parameter value. You must also specify `ha`. This is an optional parameter.

activeProtectionAtHASite

Indicates whether or not to use the HA side of the VPLEX as the RecoverPoint protected site in an RP+VPLEX setup. In a MetroPoint context, if true, this field indicates that the HA VPLEX site will be the active site. Specify `true` or `false` as the parameter value. You must also specify `ha`. This is an optional parameter.

file_replication_policy|frpol

File replication policy, for example, `replicationtype:copymode:rpovalue:rpotype`. This is an optional parameter.

file_replication_copies

File replication remote copies, for example, `varray1:vpool1 varray2:vpool2`. This is an optional parameter.

systemtype|st

Type of the storage system used to create the vpool. It could be isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, or datadomain. This is an optional parameter.

Note

Only when this parameter is set to `vmax`, can you define the `srdf` parameter.

---

raidlevel|rl

RAID level for the storage system used. Possible values are: `RAID1`, `RAID2`, `RAID3`, `RAID4`, `RAID5`, `RAID6`, or `RAID10`. This is an optional parameter.

fastpolicy|fp

Autotiering policy. Specify a fast policy only when you specify a system type. This is an optional parameter.

drivetype|dt

Supported drive type. This is an optional parameter. Supported types are SSD, FC, SAS, NL_SAS, SATA, or NONE.

type|t

Type of the virtual storage pool. Type of `file` or `block`. Default is `file` type. This is an optional parameter.

description|desc

A string describing the virtual storage pool. This is a mandatory parameter.

usematchedpools|ump

The virtual storage pool uses the matched pools. This is an optional parameter.

multivolconsistency|mvc

When included on the command line, resources provisioned from the pool support the use of consistency groups. If not included on the command line, a resource cannot be assigned to a consistency group when running ViPR Controller block provisioning services. This is an optional parameter.

expandable|ex

Indicates whether non-disruptive volume expansion should be supported. Native continuous copies are not supported. When not included, storage is selected based on performance over the ability to expand. This is an optional parameter.

autoCrossConnectExport|acc

Cross connect is enabled and export to both sides (true) or export to only one side (false). This is available only when there is High Availability VPLEX configuration, such as a shared VSAN between two VPLEX clusters in a VPLEX Metro configuraiton. This is an optional parameter.

fastexpansion|fe

Indicates that vpool volumes should use concatenated meta volumes, not striped. This is an optional parameter.

autotierpolicynames|apn

The Fully Automated Storage Tiering (FAST) policy for the virtual pool.

enablecompression

To only allow VMAX3 compression enabled storage pools to be included in the virtual pool.

maxpaths|mxp

The maximum number of paths that can be used between a host and storage volume. This is an optional parameter.

minpaths|mnp

The minimum number of paths that can be used between a host and storage volume. This is an optional parameter.

thinpreallocper|tpap

If you have `provisiontype` as `thin`, then this specifies the percentage of the physical storage to initially allocate to a volume. This is an optional percentage

frontendbandwidth|febw

Controls VMAX resource consumption at the storage group level by limiting the amount of front-end bandwidth that are consumed by the VMAX devices provisioned from this virtual pool. This value is measured in MB/s. To allow unlimited front-end bandwidth consumption, set this value to zero. This is an optional parameter.

Note

This parameter does not get set on a virtual pool that includes RecoverPoint protection.

---

iopersec|iops

Controls VMAX resource consumption at the storage group level by limiting the amount of I/Os per second (IOPS) that are consumed by the VMAX devices provisioned from this virtual pool. This value is measured in IOPS. To allow unlimited front-end I/O consumption, set this value to `0`. This is an optional parameter.

Note

This parameter does not get set on a virtual pool that includes RecoverPoint protection.

pathsperinitiator|ppi

The number of paths per host initiator. This is an optional parameter.

snapshotschedule|snapsched

Specifies whether to support a snapshot schedule. The allowable values are true and false. This is an optional parameter.

srdf

The VMAX SRDF protection parameters. There are two modes of protection: synchronous and asynchronous. This is available only between two VMAX arrays deployed in the datacenter. Each virtual storage array must have a protection vpool and policy. For example, myvarray:basepool:SYNCHRONOUS. This is an optional parameter.

Note

For using this parameter, you must have defined the `systemtype|st` parameter as **`VMAX`**.

-placementpolicy|pp

Resource placement policy (default_policy, or array_affinity) when set to array_affinity, provisioning will be performed based on the host/array affinity settings.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 306

This command creates a virtual storage pool with the NFS protocol and type file.

```
# viprcli vpool create -name SILVER -protocol NFS -t file -desc
"Protocol NFS"
```

Example 307 Using SRDF

This command creates a virtual storage pool with the SRDF options. This example shows a normal vpool creation first.

```
# viprcli vpool create -n basepool -t block -pl FC -pt Thick -va
myvarray -msnp 4

# viprcli vpool create -n srdfpool1 -t block -pl FC -pt Thick -va
myvarray -msnp 4 -st vmax -srdf myvarray:basepool:SYNCHRONOUS
```

Example 308 Using a High Availability VPLEX Configuration

This command creates a virtual storage pool with the `autoCrossConnectExport` option. This example shows auto cross connect enabled.

```
viprcli vpool create -name vplex_1_target1 -t block -pt Thin -pl FC
-desc "target pool" -ha vplex_local -autoCrossConnectExport true -
va Site1
```

Example 309 Creating a virtual pool for a journal volume

```
# viprcli vpool create -name myfull -protocol FC  -va  vmax573vplex154cl1varray -
provisiontype Thin -description test -type block  -rp vnx1303varray:my_base_new:
10GB:vmax573vplex154cl1varray:my_base_vplex -rp_source_policy
10GB:vmax573vplex154cl1varray:my_base_vplex:vmax573vplex154cl1varray:my_base_vplex:ASYNCHRO
NOUS:30:SECONDS -mvc true -ha vplex_distributed:vmax573vplex154cl1varray:my_base_vplex
```

Example 310 Creating a virtual pool for metropoint

```
# viprcli vpool create -n V1_rp_metropoint -t block -pl FC -pt Thin
-msnp 100 -mxp 1 -expandable true -mv true -acc true -desc
V1_rp_metropoint -va varray1 -ha
vplex_distributed:varray3:VP3_vplex -rp varray1:VP1_vplex:10GB
varray3:VP3_vplex:10GB -rp_source_policy ASYNCHRONOUS:25:SECONDS:
10GB:varray1:VP1_vplex:varray3:VP3_vplex -metropoint true -
activeProtectionAtHASite false
```

viprcli vpool list

Return a list of virtual storage pools accessible to the user.

Syntax

```
 viprcli vpool list
[-type|t { file | block }]
[-vdcname|vn <vdc_name>]
[{-verbose|v <verbose> | -long|l <long>}]
[-tenant|tn <tenant>]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli vpool list` command displays all virtual storage pools available to the user, or it can list virtual storage pools of a given type.

Options

type|t

One of file or block. If no type is provided, file is the default. This is an optional parameter.

vdcname|vn

The name of the virtual data center. This is an optional parameter.

tenant|tn

The name of the tenant. If user has permission on the input tenant, returns the vpools to which the tenant has access. If user does not have permission on the input tenant, an insufficient permission exception will be thrown.

{verbose|v | long|l}

These options provide additional information. Use either verbose or long and not both. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

```
viprcli vpool list

Name                    Type      Protocol
GOLD_VNX                block     FC
```

```
viprcli vpool list -type file

Name                  Type      Protocol
FILE_GOLD             file      NFS
```

```
viprcli vpool list -type block

Name                    Type     Protocol
BLOCK_GOLD_VNX                        block    FC
```

```
viprcli vpool list -l

Name                  Type     Protocol   Tenants_Allowed    Num
                                                            Paths
GOLD_VNX              block    FC         TEST_TENANT1       2
BLOCK_GOLD            file     NFS        VMWare             2
```

viprcli vpool show

Display a verbose listing of all parameters of a virtual pool that has a specified name.

Syntax

```
viprcli vpool show
       -name|n <vpoolname>
       [-type|t {file | block |
   object}]
       [-xml]
       [-hostname|hn
   <hostname>]
       [-port|po <port_number>]
       [-portui|pu
   <ui_port_number>]
       [-cookiefile|cf
   <cookiefile>]
       [-h]
```

Description

`viprcli vpool show` command displays details of a virtual pool. If no type is provided in the command line, only virtual pools of type file are displayed.

Options

name|n

A valid virtual storage pool. This is a mandatory parameter.

type|t

One of file, block, or object. This is an optional parameter.

xml

Information is returned in JSON format by default. This option returns the information in XML format.

common arguments

This operation also takes the Common Parameters on page 24.

Example 311

```
 viprcli vpool show -name ecs_vpool -type object
{
   "assigned_storage_pools": [],
   "creation_time": 1445529060904,
   "description": "cos_vpool for ECS",
   "global": false,
   "id": "urn:storageos:VirtualPool:048135b0-ca81-465b-
bb6f-6726b5d82e45:vdc1",
   "inactive": false,
   "internal": false,
   "invalid_matched_pools": [],
   "link": {
      "href": "/object/vpools/urn:storageos:VirtualPool:048135b0-
```

Example 311 (continued)


```
ca81-465b-bb6f-6726b5d82e45:vdc1",
      "rel": "self"
   },
   "matched_storage_pools": [
      {
         "id": "urn:storageos:StoragePool:c8f9ed0b-2b73-4919-9db8-
d3982c42a0bc:vdc1",
         "link": {
            "href": "/vdc/storage-systems/
urn:storageos:StorageSystem:eda5288d-53ee-4ae4-887b-
ab4f2fe162dd:vdc1/storage-pools/
urn:storageos:StoragePool:c8f9ed0b-2b73-4919-9db8-
d3982c42a0bc:vdc1",
            "rel": "self"
         }
      },
      {
         "id": "urn:storageos:StoragePool:
4dcd718d-94f0-4a1f-8022-08ad15ac2ad6:vdc1",
         "link": {
            "href": "/vdc/storage-systems/
urn:storageos:StorageSystem:eda5288d-53ee-4ae4-887b-
ab4f2fe162dd:vdc1/storage-pools/urn:storageos:StoragePool:
4dcd718d-94f0-4a1f-8022-08ad15ac2ad6:vdc1",
            "rel": "self"
         }
      }
   ],
   "max_retention": 0,
   "name": "ecs_vpool",
   "num_resources": 0,
   "protocols": [
      "S3"
   ],
   "provisioning_type": "Thick",
   "remote": false,
   "tags": [],
   "type": "object",
   "use_matched_pools": true,
   "varrays": [
      {
         "id": "urn:storageos:VirtualArray:
18b10752-7782-4aa4-8097-740a710e20fc:vdc1",
         "link": {
            "href": "/vdc/varrays/urn:storageos:VirtualArray:
18b10752-7782-4aa4-8097-740a710e20fc:vdc1",
            "rel": "self"
         }
      }
   ],
   "vdc": {
      "id": "urn:storageos:VirtualDataCenter:649d4065-f544-42da-
af89-f1a2a0d76b2f:vdc1",
      "link": {
         "href": "/vdc/urn:storageos:VirtualDataCenter:649d4065-
f544-42da-af89-f1a2a0d76b2f:vdc1",
         "rel": "self"
      }
   }
}
```

viprcli vpool update

Update the virtual pool.

Syntax

```
            viprcli vpool update
-name|n <vpoolname>
[-protocol_add|pa <protocol_add> [<protocol_add>...] {NFS | CIFS |
FC | iSCSI}]
[-protocol_remove|prm        <protocol_remove>
[<protocol_remove>...]]
[-varray_add|va_add <varray_add>        [<varray_add> ...] ]
[-varray_remove|va_rm        <varray_remove> [<varray_remove> ...] ]
[-usematchedpools|ump        <useMatchedPools>]
[-label|l <label>]
[-highavailability|ha        <highavailability>]
[-maxsnapshots|msnp        <max_snapshots>]
[-maxcontinuouscopies|mcc        <max_continuos_copies>]
[-type|t <vpooltype> { file |        block }]
[-description|desc        <description>]
[-mutlivolconsistency|mvc        <multivolconsistency>]
[-expandable|ex        <expandable>]
[-fastpolicy|fp        <fastpolicy>]
[-autotierpolicynames|apn        <unique_auto_tier_policy_names>]
[-enablecompression <enable_compression>]
[-maxpaths|mxp        <max_paths>]
[-minpaths|mnp        <min_paths>]
[-pathsperinitiator|ppi        <PathsPerInitiator>]
[-autoCrossConnectExport|acc        <autoCrossConnectExport> {true
| false}]
[-srdf_add <srdfadd> ]
[-srdf_remove <srdfremove>        ]
[-rp_add <rp_add> [<rp_add>        ...]]
[-rp_remove <rp_remove>        [<rp_remove> ...]]
[-rp_source_policy|rp_policy        <rp_source_policy>]
[-quota_enable|qe <quota_enable>        ]
[-quota_capacity|qc        <quota_capacity> ]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-h]
[-placementpolicy|pp <placementpolicy>]
```

Description

`viprcli vpool update` command updates a virtual pool in ViPR Controller. This command is silent on success.

Options

name|n

The name of the virtual pool. This is a mandatory parameter.

protocol_add|pa

The protocol to be added to the virtual pool. For file storage, Protocol can be NFS or CIFS. For block storage, Protocol is FC or iSCSI. This is an optional parameter.

protocol_remove|prm

The protocol to be removed from the virtual pool. For file storage, Protocol can be NFS or CIFS. For block storage, Protocol is FC or iSCSI. This is an optional parameter.

varray_add|va_add

The virtual storage arrays to be added to the virtual pool. This is an optional parameter.

varray_remove|va_rm

The virtual storage arrays to be removed from the virtual pool. This is an optional parameter.

usematchedpools|ump

The virtual pool uses the matched pools.

label|l

New name for the virtual pool. This is an optional parameter.

maxsnapshots|msnp

The maximum number of native snapshots. This must be greater than '0'. This is an optional parameter.

maxcontinuouscopies|mcc

The maximum number of native continuous copies. You can set this only for a vpool of type Block.

Note

This is set to zero '0' by default. You must set it to a value greater than zero and set `expandable` to **`false`**, if you want to create continuous copy of a volume using `viprcli volume continuous_copies create` command with the protection type as native.

---

expandable|ex

Expandable file system or volume. This is set to true by default. This is an optional parameter.

Note

You must set `maxcontinuouscopies` to a value greater than zero and set `expandable` to **`false`**, if you want to create continuous copy of a volume using `viprcli volume continuous_copies create` command with the protection type as native.

---

fastpolicy|fp

Name of the FAST policy. This is an optional parameter.

highavailability|ha

Highavailability is available only for VPLEX arrays. The possible values are vplex_local or vplex_distributed. This is an optional parameter.

type|t

Type of the virtual storage pool. Type of file or block. Default is file type. This is an optional parameter.

description|desc

A string describing the virtual storage pool. This is an optional parameter.

multivolconsistency|mvc

Mutliple volume consistency. This is an optional parameter.

autotierpolicynames|apn

Enable or disable unique auto-tiering policies for FAST policy. This is an optional parameter.

enablecompression

To only allow VMAX3 compression enabled storage pools to be included in the virtual pool.

maxpaths|mxp

The maximum number of paths that can be used between a host and storage volume. This is an optional parameter.

minpaths|mnp

The minimum number of paths that can be used between a host and storage volume. This is an optional parameter.

pathsperinitiator|ppi

The number of paths per host initiator. This is an optional parameter.

autoCrossConnectExport|acc

Cross connect is enabled and export to both sides (true) or export to only one side (false). This is available only when there is High Availability VPLEX configuration, such as a shared VSAN between two VPLEX clusters in a VPLEX Metro configuraiton. This is an optional parameter.

srdf_add

Add VMAX SRDF protection parameters. This is available only between two VMAX arrays deployed in the datacenter. For example, myvarray:basepool:SYNCHRONOUS. This is an optional parameter.

srdf_remove

Delete VMAX SRDF protection parameters. For example, myvarray:basepool:SYNCHRONOUS. This is an optional parameter.

rp_add

Add RP protection parameters.

rp_remove

Remove RP protection parameters.

rp_source_policy|rp_policy

Sets parameters for RecoverPoint copies. The syntax is:

```
remotecopymode:rpovalue:rpotype:journalsize:journal_varray:journ
al_vpool:standby_journal_varray:standby_journal_vpool
```

- `remotecopymode` is the remote copy mode for the RecoverPoint CG. Values are: `ASYNCHRONOUS` (the default) or `SYNCHRONOUS`. When using `SYNCHRONOUS`, the next two parameters must be `None`, as follows:

  ```
  SYNCHRONOUS:None:None
  ```

- `rpovalue` is the value sent to RP. Use the value `None` if mode is `SYNCHRONOUS`.
- `rpotype` is the RPO unit. Default is SECONDS. Value values are: `SECONDS MINUTES HOURS WRITES BYTES KB MB GB TB`. Use the value `None` if mode is `SYNCHRONOUS`.
- `journalsize` is journal size.
- `journal_varray` is an optional separate virtual array for the RecoverPoint copy journal volume. The default is the same virtual array as the RecoverPoint copy.
- `journal_vpool` is an optional separate virtual pool for the RecoverPoint copy journal volume. The default is the same virtual pool as the RecoverPoint copy.
- `standby_journal_varray` is the target virtual array.
- `standby_journal_vpool` is the target virtual pool.

This is an optional parameter.

quota_enable|qe

Enable or disable quota. Possible values are: true or false. This is an optional parameter.

quota_capacity|qc

Quota capacity in GB. Quota reserves the maximum limit for allocated storage space for the specified vpool user. This is an optional parameter and used only when you set the `quota_enable` to true.

-placementpolicy|pp

Resource placement policy (default_policy, or array_affinity) when set to array_affinity, provisioning will be performed based on the array_affinity settings.

Common Arguments

This operation also takes the Common Parameters on page 24.

Example 312

This command updates the virtual storage pool 'silver' with CIFS protocol.

```
# viprcli vpool update -n silver -pa CIFS
```

The command is silent on execution. Use the list command as shown here to view the details.

Example 312 (continued)

```
# viprcli vpool list
NAME   TYPE PROTOCOLS TENANTS_ALLOWED
silver file CIFS,NFS
```

Example 313 Using a High Availability VPLEX Configuration

This command updates the virtual storage pool with the autoCrossConnectExport option. This example shows auto cross connect disabled so that the export is only to one side.

```
viprcli vpool update -name vplex_1_check -t block -desc "target
pool" -ha vplex_local -autoCrossConnectExport false -va_add Site1
```

viprcli vpool delete

Delete a virtual pool.

Syntax

```
viprcli vpool delete
       -name|n <vpool>
       [-type|t { file | block }]
       [-hostname|hn
   <hostname>]
       [-cookiefile|cf
   <cookie_file>]
       [-port|po
   <port_number>]
       [-portui|pu
   <ui_port_number>]
       [-h]
```

Description

`viprcli vpool delete` command deletes the virtual pool with the given name. This operation succeeds only if no volume, fileshare or keypool is associated with the virtual pool. If no type is provided, 'file' is the default. ViPR Controller can support multiple virtual pools with the same name, but they cannot have the same name and the same type. For example you can build a virtual storage pool named Gold that is of type File, and a second virtual pool named Gold that is of type Block.

Options

name|n

The name of the virtual pool.

type|t

One of file or block. If no type is provided, file is the default.

common arguments

This operation also takes the Common Parameters on page 24.

Example 314

```
# viprcli vpool delete -name GOLD
  vpool GOLD (file) : Delete failed vpool GOLD (file) : not found

# viprcli vpool delete -name GOLD -t block

# viprcli vpool list

Name                      Type      Protocol
TEST_BLOCK_GOLD_VNX       block     FC
TEST_FILE_GOLD            file      NFS
TEST_VMAXBLOCK_FC_GOLD    block     FC
```

viprcli vpool allow

Allow a specified tenant to use a virtual pool.

Syntax

```
 viprcli vpool allow
-name|n <vpoolname>
[-tenant|tn <tenantname>]
[-type|t { file | block }]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli vpool allow` command allows the specified tenants to use the current virtual pool. This allows the ViPR Controller administrator to limit the number of tenants that are allowed to use storage assigned to a given virtual pool.

Options

name|n

The name of the virtual pool. This is a mandatory parameter.

tenant|tn

The name of a valid ViPR Controller tenant. This is an optional parameter.

type|t

One of file or block. If no type is provided, file is the default. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 315

```
# viprcli vpool create
-n TEST_BLOCK_GOLD_VNX -t block -pl FC  -mp 2

# viprcli vpool allow -name TEST_BLOCK_GOLD_VNX
-type block
```

viprcli vpool disallow

Disallow the tenant from using a virtual pool.

Syntax

```
 viprcli vpool disallow
-name|n <vpoolname>
[-tenant|tn <tenantname>]
[-type|t {file | block }]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli vpool disallow` command removes the named tenant from the list of tenants allowed to use a specified virtual pool. This allows the ViPR Controller administrator to limit the number of tenants that are allowed to use storage assigned to a given virtual pool.

Options

name|n

The name of the virtual pool. This is a mandatory parameter.

tenant|tn

The name of a valid ViPR Controller tenant. This is an optional parameter.

type|t

One of file or block. If no type is provided, file is the default. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 316

Example 316 (continued)

```
# viprcli vpool list
  NAME             TYPE PROTOCOLS TENANTS_ALLOWED
  TEST_FILE_GOLD11 file CIFS,NFS  TEST_TENANT11
  foo              file block
  foo2             file foo

# viprcli vpool disallow -n TEST_FILE_GOLD11 -tenant TEST_TENANT11

# viprcli vpool list
  NAME             TYPE PROTOCOLS TENANTS_ALLOWED
  TEST_FILE_GOLD11 file CIFS,NFS
  foo              file block
  foo2             file foo
```

viprcli vpool add_pools

Add the pools to a virtual pool.

Syntax

```
     viprcli vpool add_pools
            -name|n <vpoolname>
            -pools <pools> [<pools>
        ...]
            -serialnumber|sn
        <serialnumber>
            -devicetype|dt <devicetype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleio | xtremio | ddmc | ibmxiv | vnxe | datadomain}
            [-type|t {file | block }]
            [-hostname|hn
        <hostname>]
            [-cookiefile|cf
        <cookie_file>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-h]
```

Description

`viprcli vpool add_pools` command adds pools to a virtual pool. The pool you add must qualify to be added to the virtual pool. The storage pools in a virtual pool can be displayed by running the option `viprcli vpool get_pools`.

Options

name|n

A valid ViPR Controller virtual pool name. This is a mandatory parameter.

pools

The name of a ViPR Controller storage pool that is not in the target virtual pool. This is a mandatory parameter.

serialnumber|sn

The serial number of the array ion which the storage pool resides. This is a mandatory parameter.

devicetype|dt

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, or datadomain. This is a mandatory parameter.

type|t

One of file or block. If no type is provided, file is the default.

common arguments

This operation also takes the Common Parameters on page 24.

Example 317

```
# viprcli vpool add_pools -name myviprpool
       -pools 'Pool 0' -sn APM00112803075 -dt vnxblock -type block
```

viprcli vpool refresh_pools

Update the matched pools list of a virtual pool.

Syntax

```
 viprcli vpool refresh_pools
-name|n <vpoolname>
[-type|t <vpooltype> { file | block }]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli vpool refresh_pools` command recomputes the list of storage pools matched by the specified virtual pool. Use this operation after storage pool properties have been updated using `viprcli storagepool update`.

This operation differs from `viprcli vpool get-pools` because get-pools does not recalculate and reconstruct the matched pools list of the virtual pools.

Options

name|n

A valid ViPR Controller virtual pool name. This is a mandatory parameter.

type|t

One of file or block. If no type is provided, file is the default.

common arguments

This operation also takes the Common Parameters on page 24.

Example 318

```
# viprcli vpool refresh_pools
-name myviprpool -type block

  POOL_NAME     SUPPORTED_VOLUME_TYPES OPERATIONAL_STATUS
STORAGESYSTEM_NAME
  ViPR Pool 1   THIN_AND_THICK         READY              CLARIION
+APM00112803075
  Pool 0        THIN_AND_THICK         READY              CLARIION
+APM00112803075
```

viprcli vpool get_pools

Display the storage pools contained in a virtual pool.

Syntax

```
 viprcli vpool get_pools
-name|n <vpoolname>
[-type|t { file | block }]
[-hostname|hn <hostname>]
[-cookiefile|cf <cookie_file>]
[-port|po <port_number>]
[-portui|pu <ui_port_number>]
[-h]
```

Description

`viprcli vpool get_pools` command lists the storage pools currently assigned to the specified virtual pool. If pools are being added to the system (using `viprcli vpool add_pools` command) and this virtual pool has automatic matching enabled, the list displayed may not be complete because the pool assignment to the virtual storage pool may be pending. In this case, you may need to run `viprcli vpool refresh_pools` command then rerun `viprcli vpool get_pools`.

Options

name

A valid ViPR Controller virtual storage pool name. This is a mandatory parameter.

type|t

One of file or block. If no type is provided, file is the default. This is an optional parameter.

common arguments

This operation also takes the Common Parameters on page 24.

Example 319

```
viprcli vpool get_pools -n adasdasdasd -t block
  POOL_NAME     SUPPORTED_VOLUME_TYPES OPERATIONAL_STATUS
STORAGESYSTEM_NAME
  Bourne Pool 1 THIN_AND_THICK         READY              CLARIION
+APM0011280307
  Pool 0        THIN_AND_THICK         READY              CLARIION
+APM0011280307
```

viprcli vpool remove_pools

Remove the pool from a virtual pool.

Syntax

```
     viprcli vpool remove_pools
            -name|n <vpoolname>
            -pools <pools> [<pools>
        ...]
            -serialnumber|sn
        <serialnumber>
            -devicetype|dt <devicetype>
        {isilon | vnxblock | vnxfile | vmax | netapp | netappc |
vplex | hds | openstack
        | scaleio | xtremio | ddmc | ibmxiv | vnxe | datadomain}
            [-type|t { file | block }]
            [-hostname|hn
        <hostname>]
            [-cookiefile|cf
        <cookie_file>]
            [-port|po <port_number>]
            [-portui|pu
        <ui_port_number>]
            [-h]
```

Description

`viprcli vpool remove_pools` command removes the pools from the specified virtual pool. The pool you remove must be in the virtual pool. The storage pools in a virtual pool can be displayed by running the option `viprcli vpool get_pools`.

Options

name|n

A valid ViPR Controller virtual pool name. This is a mandatory parameter.

pools

The name of a pool in the virtual pool. Call `viprcli vpool get_pools` to see the list of pools in the virtual pool. This is a mandatory parameter.

serialnumber|sn

The serial number of the array ion which the storage pool resides. This is a mandatory parameter.

devicetype|dt

One of isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, or datadomain. This is a mandatory parameter.

type|t

One of file or block. If no type is provided, file is the default.

common arguments

This operation also takes the Common Parameters on page 24.

Example 320

```
viprcli vpool remove_pools -name adasdasdasd -pools 'Pool 0' -sn
APM00112803075 -dt vnxblock
-type block
```