Command Reference

Acropolis

Contents

1. Acropolis Command-Line Interface (aCLI)..... 4

CLI Reference Conventions..... 4
ads..... 4
core..... 5
ha..... 6
host..... 7
image..... 8
iscsi_client..... 11
microseg..... 12
net..... 13
nf..... 21
parcel..... 21
snapshot..... 22
vg..... 22
vm..... 30
vm_group..... 51

2. Nutanix Command-Line Interface (nCLI)..... 54

Installing the nCLI on Your Local System..... 54
Initiating an nCLI Session From Your Local System..... 55
Command Format..... 56
Embedded Help..... 56
nCLI Entities..... 57
Nutanix Command-Line Interface Reference..... 58
CLI Reference Conventions..... 58
alerts: Alert..... 58
authconfig: Authentication Configuration..... 60
cloud: Cloud..... 65
cluster: Cluster..... 69
container: Storage Container..... 82
data-at-rest-encryption: Data At Rest Encryption..... 86
data-at-rest-encryption-certificate: Data At Rest Encryption Certificate..... 87
datastore: Datastore..... 90
disk: Physical Disk..... 91
events: Event..... 92
failover-cluster: Failover Cluster..... 93
file-server: File Server..... 94
health-check: Health Check..... 127
host: Physical Host..... 128
http-proxy: HTTP Proxy..... 132
key-management-server: Key Management Server..... 134
license: License..... 135
managementserver: Management Server..... 136
multicluster: Multicluster..... 138
network: Network..... 138
nutanix-guest-tools: Nutanix Guest Tools..... 144
progress-monitor: Progress Monitor..... 145
protection-domain: Protection domain..... 146

pulse-config: Pulse Configuration....160
rackable-unit: Rackable unit....160
remote-site: Remote Site....161
rsyslog-config: RSyslog Configuration....166
smb-server: Nutanix SMB server....168
snapshot: Snapshot....169
snmp: SNMP....170
software: Software....174
ssl-certificate: SSL Certificate....176
storagepool: Storage Pool....177
storagetier: Storage Tier....178
task: Tasks....179
user: User....180
vdisk: Virtual Disk....183
virtual-disk: Virtual Disk....186
virtualmachine: Virtual Machine....187
volume-group: Volume Groups....189
vstore: VStore....194
vzone: vZone....195

3. Controller VM Commands....197

Specifying Credentials....197
cluster....197
diagnostics.py....250
genesis....256
ncc....263
setup_hyperv.py....279

1

Acropolis Command-Line Interface (aCLI)

Acropolis provides a command-line interface for managing hosts, networks, snapshots, and VMs.

Accessing the Acropolis CLI

To access the Acropolis CLI, log on to a Controller VM in the cluster with SSH and type `acli` at the shell prompt.

To exit the Acropolis CLI and return to the shell, type `exit` at the `<acropolis>` prompt.

CLI Reference Conventions

This command-line interface reference uses the following conventions.

- Parameters in italic are unique to your environment.

  ```
  value
  ```

- Parameters in square brackets are optional.

  ```
  [ value ]
  ```

- Parameters in curly brackets must be one of a limited set of values.

  ```
  { value1 | value2 }
  ```

  One example is boolean parameters: `{ true | false }`
- The `keyword` is a literal string required by the command, and the `value` is the unique value for your environment.

  ```
  keyword=value
  ```

ads

Operations

- Get current Acropolis Dynamic Scheduling (ADS) configuration : `ads.get`
- Enable or disable Acropolis Dynamic Scheduling (ADS) : `ads.update`

Get current Acropolis Dynamic Scheduling (ADS) configuration

`<acropolis> ads.get`

Required arguments

None

Enable or disable Acropolis Dynamic Scheduling (ADS)

If this behavior is not desirable, then the only option is to disable this feature completely.

`<acropolis> ads.update [ enable="`*{ true | false }*`" ]`

Required arguments

None

Optional arguments

`enable`

Enable ADS.

Type: boolean

Default: true

core

Operations

- Exits the CLI : `core.exit`
- Gets the current value of the given configuration options : `core.get`
- Provides help text for the named object : `core.help`
- Sets the value of the given configuration options : `core.set`

Exits the CLI

```
<acropolis> core.exit
```

Required arguments

None

Gets the current value of the given configuration options

```
<acropolis> core.get [options ]
```

Required arguments

None

Optional arguments

`options`

Comma-delimited list

Type: list of configuration options

Provides help text for the named object

```
<acropolis> core.help [name ]
```

Required arguments

None

Optional arguments

`name`

Command or namespace to describe

Type: command or namespace name

Sets the value of the given configuration options

```
<acropolis> core.set [ assume_default="{true | false }" ][ inline_help="{true | false }" ][
json="{true | false }" ][ page_size="page_size" ][ pretty="{true | false }" ][ use_dns="{true |
false }" ]
```

Required arguments

None

Optional arguments

`assume_default`

Reply to all prompts with default values.

Type: boolean

`inline_help`

Show inline help for tab completion.

Type: boolean

`json`

Output JSON.

Type: boolean

`page_size`

Number of lines before pagination.

Type: int

`pretty`

Show pretty output.

Type: boolean

`use_dns`

Perform DNS lookups.

Type: boolean

ha

Operations

- Get current HA configuration : `ha.get`
- Enable, disable or modify VM availability configuration : `ha.update`

Get current HA configuration

```
<acropolis> ha.get
```

Required arguments

None

Enable, disable or modify VM availability configuration

All VMs are evacuated from the host that is picked as reserved.

```
<acropolis> ha.update [ num_host_failures_to_tolerate="num_host_failures_to_tolerate"
][ reservation_type="reservation_type" ][ wait="{ true | false }" ]
```

Required arguments

None

Optional arguments

`num_host_failures_to_tolerate`

Number of host failures to tolerate.

Type: int

`reservation_type`

Reservation type

Type: HA reservation type

`wait`

If True, wait for the host evacuation attempt to finish

Type: boolean

Default: true

host

Operations

- Puts a host into maintenance mode : `host.enter_maintenance_mode`
- Check if AHV host can enter maintenance mode : `host.enter_maintenance_mode_check`
- Takes a host out of maintenance mode : `host.exit_maintenance_mode`
- Retrieves scheduler information about a Host : `host.get`
- Lists hosts in the cluster : `host.list`
- Lists VMs currently running on the host : `host.list_vms`

Puts a host into maintenance mode

The user should use the host.get command to determine the host's current maintenance mode state.

`<acropolis> host.enter_maintenance_mode` *host* `[ mode="`*mode*`" ][ non_migratable_vm_action="`*non_migratable_vm_action*`" ][ wait="{`*true* | *false*`}" ]`

Required arguments

host

Host identifier

Type: host

Optional arguments

`mode`

Evacuation mode ('live', 'cold', 'power_off')

Type: string

Default: live

`non_migratable_vm_action`

Action for non migratable VMs ('block', 'acpi_shutdown')

Type: string

Default: block

`wait`

If True, wait for the host evacuation attempt to finish

Type: boolean

Default: true

Check if AHV host can enter maintenance mode

User can proceed to 'host.enter_maintenance_mode' if check succeeds.

`<acropolis> host.enter_maintenance_mode_check` *host* `[ non_migratable_vm_action="`*non_migratable_vm_action*`" ]`

Required arguments

host

Host identifier

Type: host

Optional arguments

`non_migratable_vm_action`

Action for non migratable VMs ('block', 'acpi_shutdown')

Type: string

Default: block

Takes a host out of maintenance mode

Use the host.get command to check the host's current maintenance mode state.

`<acropolis> host.exit_maintenance_mode` *host*

Required arguments

host

Host identifier

Type: host

Retrieves scheduler information about a Host

`<acropolis> host.get` *host_list*

Required arguments

host_list

Host identifier

Type: list of hosts

Lists hosts in the cluster

`<acropolis> host.list`

Required arguments

None

Lists VMs currently running on the host

`<acropolis> host.list_vms` *host*

Required arguments

host

Host UUID

Type: host

image

Operations

- Create an image : `image.create`
- Delete an image(s) : `image.delete`
- Retrieves information about an image : `image.get`
- List all Images : `image.list`
- Update an image : `image.update`

Create an image

Optionally, a checksum may also be specified if we are creating an image from a source_url in order to verify the correctness of the image.

`<acropolis> image.create` *name* [`annotation="`*annotation*`"`][`architecture="`*architecture*`"`][`clone_from_vmdisk="`*clone_from_vmdisk*`"`][`compute_checksum="`*{ true | false }*`"`][`container="`*container*`"`][`image_type="`*{raw|vhd|vmdk|vdi|iso|qcow2|vhdx}*`"`][`product_name="`*product_name*`"`][`product_version="`*product_version*`"`

```
][ sha1_checksum="sha1_checksum" ][ sha256_checksum="sha256_checksum" ][
source_url="source_url" ][ wait="{ true | false }" ]
```

Required arguments

name

Comma-delimited list of image names

Type: list of strings with expansion wildcards

Optional arguments

`annotation`

Image description

Type: string

`architecture`

Disk image CPU architecture

Type: image architecture

`clone_from_vmdisk`

UUID of the source vmdisk

Type: VM disk

`compute_checksum`

If True, we will compute the checksum of the image

Type: boolean

Default: false

`container`

Destination Storage Container

Type: container

`image_type`

Image type

Type: image type

`product_name`

Name of the producer/distributor of the image

Type: string

`product_version`

Version string for the image

Type: string

`sha1_checksum`

SHA-1 checksum

Type: hex checksum

`sha256_checksum`

SHA-256 checksum

Type: hex checksum

`source_url`

URL location of the source image

Type: image URL

`wait`

If True, we will wait for the image creation to complete

Type: boolean

Default: true

Examples

1. Create an image named 'foo' from an image located at http://test.com/disk_image.

```
<acropolis> image.create foo source_url=http://test.com/image_iso
 container=default image_type=kIsoImage architecture=kX86_64
```

2. Create an image named 'bar' from a vmdisk 0b4fc60b-cc56-41c6-911e-67cc8406d096.

```
<acropolis> image.create bar clone_from_vmdisk=0b4fc60b-
cc56-41c6-911e-67cc8406d096 image_type=kDiskImage
```

Delete an image(s)

`<acropolis> image.delete` *image_list*

Required arguments

image_list

Image identifiers

Type: list of images

Retrieves information about an image

`<acropolis> image.get` *image_list* `[ include_vmdisk_paths="{`*true* | *false* `}" ][ include_vmdisk_sizes="{`*true* | *false* `}" ]`

Required arguments

image_list

Image identifiers

Type: list of images

Optional arguments

`include_vmdisk_paths`

Fetch vmdisk paths

Type: boolean

Default: false

`include_vmdisk_sizes`

Fetch vmdisk sizes (in bytes)

Type: boolean

Default: true

List all Images

`<acropolis> image.list`

Required arguments

None

Update an image

```
<acropolis> image.update [ annotation="annotation" ][ architecture="architecture"
][ image="image" ][ image_type="{raw|vhd|vmdk|vdi|iso|qcow2|vhdx}" ][ name="name" ][
product_name="product_name" ][ product_version="product_version" ]
```

Required arguments

None

Optional arguments

`annotation`

Image description

Type: string

`architecture`

Disk image CPU architecture

Type: image architecture

`image`

Image identifier

Type: image

`image_type`

Image type

Type: image type

`name`

Image name

Type: string

`product_name`

Name of the producer/distributor of the image

Type: string

`product_version`

Version string for the image

Type: string

Examples

1. Update the name of an image named 'foo'.

```
<acropolis> image.update foo name=bar
```

iscsi_client

Operations

- Retrieves information about an ISCSI client : `iscsi_client.get`
- Lists all ISCSI clients : `iscsi_client.list`
- Update an ISCSI client's parameters : `iscsi_client.update`

Retrieves information about an ISCSI client

```
<acropolis> iscsi_client.get [ client_uuid_list="client_uuid_list" ][
name_list="name_list" ][ network_id_list="network_id_list" ]
```

Required arguments

None

Optional arguments

`client_uuid_list`

Comma-delimited list of iscsi initiator client uuids

Type: List of iSCSI clients

`name_list`

Comma-delimited list of iscsi initiator names (IQNs)

Type: List of iSCSI clients

`network_id_list`

Comma-delimited list of iscsi initiator network identifiers

Type: List of iSCSI clients

Lists all ISCSI clients

```
<acropolis> iscsi_client.list
```

Required arguments

None

Update an ISCSI client's parameters

```
<acropolis> iscsi_client.update [ client_secret="client_secret" ][
client_uuid="client_uuid" ][ name="name" ][ network_id="network_id" ]
```

Required arguments

None

Optional arguments

`client_secret`

Client secret to be used in CHAP authentication

Type: string

`client_uuid`

iscsi client UUID

Type: iSCSI client name or network id

`name`

iscsi initiator name (IQNs)

Type: iSCSI client name or network id

`network_id`

iscsi initiator network identifier

Type: iSCSI client name or network id

microseg

Operations

- Disables microsegmentation feature : `microseg.disable`
- Enables microsegmentation feature : `microseg.enable`
- Retrieves microsegmentation feature configuration information : `microseg.get`

Disables microsegmentation feature

```
<acropolis> microseg.disable
```

Required arguments

None

Enables microsegmentation feature

```
<acropolis> microseg.enable
```

Required arguments

None

Retrieves microsegmentation feature configuration information

```
<acropolis> microseg.get
```

Required arguments

None

net

Operations

- Add a DHCP pool to a managed network : `net.add_dhcp_pool`
- Blacklists IP addresses for a managed network : `net.add_to_ip_blacklist`
- Clear the DHCP DNS configuration for a managed network : `net.clear_dhcp_dns`
- Clear the DHCP TFTP configuration for a managed network : `net.clear_dhcp_tftp`
- Clears the network function chain for this network : `net.clear_network_function_chain`
- Creates a new virtual network for VMs : `net.create`
- Creates a cluster wide vswitch configuration : `net.create_cluster_vswitch`
- Deletes a network : `net.delete`
- Delete a DHCP pool from a managed network : `net.delete_dhcp_pool`
- Removes IP addresses from a managed network's blacklist : `net.delete_from_ip_blacklist`
- Retrieves information about a network : `net.get`
- Retrieves information about a specific cluster_vswitch configuration : `net.get_cluster_vswitch`
- Lists all networks : `net.list`
- Lists all cluster_vswitch configurations : `net.list_cluster_vswitch`
- List blacklisted IPs for a managed network : `net.list_ip_blacklist`
- Lists VMs configured on the network : `net.list_vms`
- Updates network metadata : `net.update`
- Updates a cluster wide vswitch configuration : `net.update_cluster_vswitch`
- Configure the DHCP DNS configuration for a managed network : `net.update_dhcp_dns`
- Configure the DHCP TFTP configuration for a managed network : `net.update_dhcp_tftp`
- Sets the network function chain for this network : `net.update_network_function_chain`

Add a DHCP pool to a managed network

That is, a user may manually specify an address belonging to the pool when creating a virtual adapter.

`<acropolis> net.add_dhcp_pool` ***network*** `[ end="`***end***`" ][ start="`***start***`" ]`

Required arguments

network

Network identifier

Type: network

Optional arguments

`end`

Last IPv4 address

Type: IPv4 address

`start`

First IPv4 address

Type: IPv4 address

Examples

1. Auto-assign addresses from the inclusive range 192.168.1.16 - 192.168.1.32.

```
<acropolis> net.add_dhcp_pool vlan.16 start=192.168.1.16 end=192.168.1.32
```

Blacklists IP addresses for a managed network

This property may be useful for avoiding conflicts between VMs and other hosts on the physical network.

`<acropolis> net.add_to_ip_blacklist` *network* `[ ip_list="`*ip_list*`" ]`

Required arguments

network

Network identifier

Type: network

Optional arguments

`ip_list`

Comma-delimited list of IP addresses

Type: list of IPv4 addresses

Clear the DHCP DNS configuration for a managed network

`<acropolis> net.clear_dhcp_dns` *network*

Required arguments

network

Network identifier

Type: network

Examples

1. Clear DNS servers and search domains.

```
<acropolis> net.clear_dhcp_dns vlan.123
```

Clear the DHCP TFTP configuration for a managed network

`<acropolis> net.clear_dhcp_tftp` *network*

Required arguments

network

Network identifier

Type: network

Examples

1. Clear TFTP server name and boot filename.

```
<acropolis> net.clear_dhcp_tftp vlan.123
```

Clears the network function chain for this network

`<acropolis> net.clear_network_function_chain` *network*

Required arguments

network

Network identifier

Type: network

Creates a new virtual network for VMs

For more about managed networks, see the following commands: net.add_dhcp_pool net.add_to_ip_blacklist net.clear_dhcp_dns net.clear_dhcp_tftp net.delete_dhcp_pool net.delete_from_ip_blacklist net.list_ip_blacklist net.update_dhcp_dns net.update_dhcp_tftp

`<acropolis> net.create` *name* `[ annotation="`*annotation*`" ][ dhcp_address="`*dhcp_address*`" ][ ip_config="`*ip_config*`" ][ mtu="`*mtu*`" ][ vlan="`*vlan*`" ][ vswitch_name="`*vswitch_name*`" ]`

Required arguments

name

Network name

Type: string

Optional arguments

`annotation`

Annotation string

Type: string

`dhcp_address`

DHCP server address (for managed networks)

Type: IPv4 address

`ip_config`

IP configuration in CIDR notation ("default_gateway/prefix")

Type: string

`mtu`

MTU setting

Type: int

`vlan`

VLAN ID

Type: int

`vswitch_name`

Vswitch name

Type: string

Examples

1. Create an unmanaged network on VLAN 66.

```
<acropolis> net.create mynet vlan=66
```

2. Create an unmanaged network on VLAN 66 with MTU 9000.

```
<acropolis> net.create mynet vlan=66 mtu=9000
```

3. Create a managed network on VLAN 99, bound to vswitch br1. The managed IPv4 range is 10.1.1.0 - 10.1.1.255, the default gateway is 10.1.1.1, and the DHCP server is 10.1.1.254

```
<acropolis> net.create mynet vlan=99 vswitch_name=br1 ip_config=10.1.1.1/24
```

4. Create an untagged managed network. The managed IPv4 range is 192.168.0.0 - 192.168.3.255, and the default gateway is 192.168.5.254. In this example, the DHCP server will be automatically configured as 192.168.5.253 to avoid collision with the default gateway.

```
<acropolis> net.create mynet vlan=0 ip_config=192.168.5.254/22
```

Creates a cluster wide vswitch configuration

`<acropolis> net.create_cluster_vswitch` *bridge_name* `[ bond="`*bond*`" ][ host_override="`*host_override*`" ][ lacp="`*{ true | false }*`" ][ lacp_fallback="`*{ true | false }*`" ][ lacp_timeout="`*lacp_timeout*`" ][ nic_team_policy="`*nic_team_policy*`" ][ uplink_grouping="`*uplink_grouping*`" ]`

Required arguments

bridge_name

Bridge name

Type: string

Optional arguments

`bond`

Bond name

Type: string

`host_override`

Per-host override grouping of uplink interfaces for the bond

Type: list of strings

`lacp`

Whether to enable LACP on the bond

Type: boolean

`lacp_fallback`

Whether to enable LACP fallback to active-backup on LACP negotiation failure

Type: boolean

`lacp_timeout`

LACP timeout value

Type: LACP timeout value

`nic_team_policy`

NIC teaming policy to be set on the bond

Type: NIC Team policy

`uplink_grouping`

Default uplink grouping based on NIC speed

Type: Uplink Grouping type

Examples

1. Create cluster_vswitch configuration for bridge br0 and bond 10G interfaces

```
<acropolis> net.create_cluster_vswitch br0 uplink_grouping=kAll10G
 nic_team_policy=kActiveBackup
```

2. Create cluster_vswitch configuration for bridge br0 and bond 100G withlacp

```
<acropolis> net.create_cluster_vswitch br0 uplink_grouping=kAll100G
 nic_team_policy=kBalanceTcp lacp=true
```

3. Create cluster_vswitch configuration for bridge br0 and bond 100G with lacp and hostoverride for host1:<uuid1> with uplinks=eth0,eth1 and for host2:<uuid2> with uplinks=eth3,eth4

```
<acropolis> net.create_cluster_vswitch br0 uplink_grouping=All100G
 nic_team_policy=kBalanceSlb lacp=true host_override=[uuid1:eth0:eth1],
[uuid2:eth2:eth3]
```

Deletes a network

To determine which VMs are on a network, use net.list_vms.

`<acropolis> net.delete` *network*

Required arguments

network

Network identifier

Type: network

Delete a DHCP pool from a managed network

See network.add_dhcp_pool for more information.

`<acropolis> net.delete_dhcp_pool` *network* `[ start="`*start*`" ]`

Required arguments

network

Network identifier

Type: network

Optional arguments

`start`

First IPv4 address

Type: IPv4 address

Removes IP addresses from a managed network's blacklist

`<acropolis> net.delete_from_ip_blacklist` *network* `[ ip_list="`*ip_list*`" ]`

Required arguments

network

Network identifier

Type: network

Optional arguments

`ip_list`

Comma-delimited list of IP addresses

Type: list of IPv4 addresses

Retrieves information about a network

```
<acropolis> net.get network_list
```

Required arguments

network_list

Network identifier

Type: list of networks

Retrieves information about a specific cluster_vswitch configuration

```
<acropolis> net.get_cluster_vswitch bridge_name_list
```

Required arguments

bridge_name_list

List of bridge names.

Type: list of Bridge Names

Lists all networks

```
<acropolis> net.list
```

Required arguments

None

Lists all cluster_vswitch configurations

```
<acropolis> net.list_cluster_vswitch
```

Required arguments

None

List blacklisted IPs for a managed network

```
<acropolis> net.list_ip_blacklist network
```

Required arguments

network

Network identifier

Type: network

Lists VMs configured on the network

```
<acropolis> net.list_vms network
```

Required arguments

network

Network identifier

Type: network

Updates network metadata

```
<acropolis> net.update network [ annotation="annotation" ][ name="name" ]
```

Required arguments

network

Network identifier

Type: network

Optional arguments

`annotation`

Annotation string

Type: string

`name`

Network name

Type: string

Updates a cluster wide vswitch configuration

```
<acropolis> net.update_cluster_vswitch bridge_name [ bond="bond" ][
host_override="host_override" ][ lacp="{ true | false }" ][ lacp_fallback="{ true |
false }" ][ lacp_timeout="lacp_timeout" ][ nic_team_policy="nic_team_policy" ][
uplink_grouping="uplink_grouping" ]
```

Required arguments

bridge_name

Bridge Name

Type: Bridge Name

Optional arguments

`bond`

Bond name

Type: string

`host_override`

Per-host override grouping of uplink interfaces for the bond

Type: list of strings

`lacp`

Whether to enable LACP on the bond

Type: boolean

`lacp_fallback`

Whether to enable LACP fallback to active-backup on LACP negotiation failure

Type: boolean

`lacp_timeout`

LACP timeout value

Type: LACP timeout value

`nic_team_policy`

NIC teaming policy to be set on the bond

Type: NIC Team policy

`uplink_grouping`

Default uplink grouping based on NIC speed

Type: Uplink Grouping type

Examples

1. Update cluster_vswitch configuration for cluster_vswitch_uuid uuid1 with uplink_grouping to include all 10G interfaces

```
<acropolis> net.update_cluster_vswitch uuid1 uplink_grouping=All10G
```

2. Update cluster_vswitch configuration for cluster_vswitch_uuid uuid1with lacp enabled

```
<acropolis> net.update_cluster_vswitch br0 uplink_grouping=All100G lacp=true
```

3. Update cluster_vswitch configuration for cluster_vswitch_uuid uuid with lacp and hostoverride for host1:<uuid1> with uplinks=eth0,eth1 and for host2:<uuid2> with uplinks=eth3,eth4

```
<acropolis> net.update_cluster_vswitch uuid1 uplink_grouping=All100G lacp=true
 host_override=[uuid1:eth0:eth1],[uuid2:eth2:eth3]
```

4. Update to clear host_overrides for cluster_vswitch configuration for cluster_vswitch_uuid uuid

```
<acropolis> net.update_cluster_vswitch uuid1 host_override=[]
```

Configure the DHCP DNS configuration for a managed network

However, the DHCP server hands out infinite leases, so clients will need to manually renew to pick up the new settings.

`<acropolis> net.update_dhcp_dns` *network* `[ domains="`*domains*`" ][ servers="`*servers*`" ]`

Required arguments

network

Network identifier

Type: network

Optional arguments

`domains`

Comma-delimited list of search domains

Type: list of DNS domains

`servers`

Comma-delimited list of DNS server IP addresses

Type: list of IPv4 addresses

Examples

1. Configure DNS servers and search domains.

```
<acropolis> net.update_dhcp_dns vlan.123 servers=10.1.1.1,10.1.1.2
 domains=eng.nutanix.com,corp.nutanix.com
```

Configure the DHCP TFTP configuration for a managed network

However, the TFTP server hands out infinite leases, so clients will need to manually renew to pick up the new settings.

`<acropolis> net.update_dhcp_tftp` *network* `[ bootfile_name="`*bootfile_name*`" ][ server_name="`*server_name*`" ]`

Required arguments

network

Network identifier

Type: network

Optional arguments

`bootfile_name`

Boot file name

Type: string

`server_name`

TFTP server name

Type: string

Examples

1. Configure TFTP server and bootfile.

```
<acropolis> net.update_dhcp_tftp vlan.123 server_name=10.1.1.1
 bootfile_name=ARDBP32.BIN
```

Sets the network function chain for this network

`<acropolis> net.update_network_function_chain` *network chain*

Required arguments

network

Network identifier

Type: network

chain

Network function chain identifier

Type: network function chain

nf

Operations

- Gets specific network function chains : `nf.chain_get`
- Lists all network function chains : `nf.chain_list`

Gets specific network function chains

`<acropolis> nf.chain_get` *chains*

Required arguments

chains

Network function chain

Type: list of network function chains

Lists all network function chains

`<acropolis> nf.chain_list`

Required arguments

None

parcel

Operations

- Retrieves information about a parcel : `parcel.get`
- List all Parcels : `parcel.list`

Retrieves information about a parcel

`<acropolis> parcel.get` *parcel_list*

Required arguments

parcel_list

Parcel identifiers

Type: list of parcels

List all Parcels

```
<acropolis> parcel.list
```

Required arguments

None

snapshot

Operations

- Deletes one or more snapshots : `snapshot.delete`
- Retrieves information about a snapshot : `snapshot.get`
- Lists all snapshots : `snapshot.list`

Deletes one or more snapshots

```
<acropolis> snapshot.delete snapshot_list
```

Required arguments

snapshot_list

Comma-delimited list of snapshot identifiers

Type: list of snapshots

Retrieves information about a snapshot

```
<acropolis> snapshot.get snapshot_list
```

Required arguments

snapshot_list

Snapshot identifier

Type: list of snapshots

Lists all snapshots

```
<acropolis> snapshot.list
```

Required arguments

None

vg

Operations

- Allow volume group to be accessed from an external intiator : `vg.attach_external`
- Attach a VG to the specified VM : `vg.attach_to_vm`
- Clones a Volume Group : `vg.clone`
- Creates one or more VGs : `vg.create`
- Deletes one or more VGs and its backing disks : `vg.delete`
- Stop allowing volume group to be accessed from an external intiator : `vg.detach_external`
- Detach a VG from the specified VM : `vg.detach_from_vm`
- Add a new disk to a VG : `vg.disk_create`
- Remove a disk from a VG : `vg.disk_delete`
- Updates the backing for the specified volume group disk : `vg.disk_update`
- Retrieves information about a VG : `vg.get`

- Lists all VGs : `vg.list`
- Updates the specified VGs : `vg.update`
- Update preferred_host for a given external attachment for a VG : `vg.update_external`

Allow volume group to be accessed from an external intiator

`<acropolis> vg.attach_external` *vg* `[ initiator_name="`*initiator_name*`" ][ initiator_network_id="`*initiator_network_id*`" ][ num_virtual_targets="`*num_virtual_targets*`" ][ use_redirection="{` *true* | *false* `}" ]`

Required arguments

vg

VG identifier

Type: volume group

Optional arguments

`initiator_name`

Name of external initiator as a valid IQN

Type: iSCSI IQN

`initiator_network_id`

Network identifier of external initiator as a valid IPv4 address

Type: string

`num_virtual_targets`

Number of virtual targets

Type: int

`use_redirection`

Use iSCSI redirection for iSCSI login to this target?

Type: boolean

Examples

1. Attach an external client identified by name, iqn.1994-05.com.redhat:71eef92fe6c, to the VG, vg1, :

```
<acropolis> vg.attach_external vg1.attach_external
 initiator_name=iqn.1994-05.com.redhat:71eef92fe6c
```

2. Attach an external client identified by network id, 10.1.1.1, to the VG, vg2, :

```
<acropolis> vg.attach_external vg2.attach_external
 initiator_network_id=10.1.1.1
```

Attach a VG to the specified VM

`<acropolis> vg.attach_to_vm` *vg vm* `[ index="`*index*`" ]`

Required arguments

vg

VG identifier

Type: volume group

vm

VM identifier

Type: VM

Optional arguments

`index`

Device index on the scsi bus

Type: int

Clones a Volume Group

If the ISCSI target names for the clones are not specified through the 'iscsi_target_prefix_list' argument, then default values will be used.

`<acropolis> vg.clone` *name_list* `[ clone_from_vg="`*clone_from_vg*`" ][ iscsi_target_prefix_list="`*iscsi_target_prefix_list*`" ][ load_balance_vm_attachments="{` *true* | *false* `}" ][ target_secret_list="`*target_secret_list*`" ]`

Required arguments

name_list

Comma-delimited list of VG names

Type: list of strings with expansion wildcards

Optional arguments

`clone_from_vg`

VG from which to clone

Type: volume group

`iscsi_target_prefix_list`

Comma-delimited list of iscsi target prefixes for each of the VGs

Type: list of strings

`load_balance_vm_attachments`

Whether to enable/disable VG load balance VM attachments

Type: boolean

`target_secret_list`

Comma delimited CHAP secrets associated with each of the VGs. To delete the secret, set target_secret="" using vg.update

Type: list of strings

Examples

1. Clone two VGs vg1 and vg2 with iscsi targets vgt1 and vgt2 from source-vg

```
<acropolis> vg.clone vg1,vg2 clone_from_vg=source-vg
 iscsi_target_prefix_list=vgt1,vgt2
```

2. Clone two VGs vg1 and vg2 with target_secrets vg1_target_secret and vg2_target_secret

```
<acropolis> vg.clone vg1,vg2 clone_from_vg=source-vg
 target_secret_list=vg1_target_secret,vg2_target_secret
```

3. Clone VG vg2 from vg1 with load_balance_vm_attachments set

```
<acropolis> vg.clone vg.clone vg2 clone_from_vg=vg1
 load_balance_vm_attachments=True
```

Creates one or more VGs

`<acropolis> vg.create` *name_list* `[ annotation="`*annotation*`" ][ flash_mode="{` *true* | *false* `}" ][ iscsi_target_prefix_list="`*iscsi_target_prefix_list*`" ][`

`load_balance_vm_attachments="{ true | false }" ][ shared="{ true | false }" ][ target_secret_list="target_secret_list" ]`

Required arguments

name_list

Comma-delimited list of VG names

Type: list of strings with expansion wildcards

Optional arguments

`annotation`

Annotation string

Type: string

`flash_mode`

Whether to enable/disable flash mode

Type: boolean

`iscsi_target_prefix_list`

Comma-delimited list of iscsi target prefixes for each of the VGs

Type: list of strings

`load_balance_vm_attachments`

Whether to enable/disable VG load balance VM attachments

Type: boolean

`shared`

Allow VG to be attached to multiple VMs simultaneously?

Type: boolean

`target_secret_list`

Comma delimited CHAP secrets associated with each of the VGs. To delete the secret, set target_secret="" using vg.update

Type: list of strings

Deletes one or more VGs and its backing disks

`<acropolis> vg.delete vg_list`

Required arguments

vg_list

Comma-delimited VG identifiers

Type: list of volume groups

Stop allowing volume group to be accessed from an external intiator

`<acropolis> vg.detach_external vg [ initiator_name="initiator_name" ][ initiator_network_id="initiator_network_id" ]`

Required arguments

vg

VG identifier

Type: volume group

Optional arguments

`initiator_name`

Name of external initiator as a valid IQN

Type: VG external initiator name

`initiator_network_id`

Network identifier of external initiator as a valid IPv4 address

Type: VG external initiator name

Detach a VG from the specified VM

```
<acropolis> vg.detach_from_vm vg vm
```

Required arguments

vg

VG identifier

Type: volume group

vm

VM identifier

Type: VM attached to VG

Add a new disk to a VG

This can be used to expand a disk image at clone time.

```
<acropolis> vg.disk_create vg [ clone_from_adsf_file="clone_from_adsf_file"
][ clone_from_vmdisk="clone_from_vmdisk" ][ clone_min_size="clone_min_size" ][
container="container" ][ create_size="create_size" ][ index="index" ]
```

Required arguments

vg

VG identifier

Type: volume group

Optional arguments

`clone_from_adsf_file`

Path to an ADSF file

Type: ADSF path

`clone_from_vmdisk`

A vmdisk UUID

Type: VM disk

`clone_min_size`

Minimum size of the resulting clone (only applies to cloned disks)

Type: size with cskKmMgGtT suffix

`container`

Storage Container (only applies to newly-created disks)

Type: container

`create_size`

Size of new disk

Type: size with cskKmMgGtT suffix

`index`

Device index on bus

Type: int

Examples

1. Create a blank 5GiB disk on ctr, and add it to my_vg at index 3

```
<acropolis> vg.disk_create my_vg create_size=5G container=ctr index=3
```

2. Clone a disk from the ADSF file /ctr/plan9.iso, and add it to first open slot

```
<acropolis> vg.disk_create my_vg clone_from_adsf_file=/ctr/plan9.iso
```

3. Clone a disk from the existing vmdisk, and add it to the first open slot

```
<acropolis> vg.disk_create my_vg clone_from_vmdisk=0b4fc60b-
cc56-41c6-911e-67cc8406d096
```

Remove a disk from a VG

`<acropolis> vg.disk_delete` *vg index*

Required arguments

vg

VG identifier

Type: volume group

index

Disk index

Type: VG disk index

Updates the backing for the specified volume group disk

But for the other options, the existing disk image will be deleted and replaced by the new image (which may be a clone of the existing image).

`<acropolis> vg.disk_update` *vg index* `[ clone_from_adsf_file="`*clone_from_adsf_file*`" ][ clone_from_image="`*clone_from_image*`" ][ clone_from_vmdisk="`*clone_from_vmdisk*`" ][ clone_min_size="`*clone_min_size*`" ][ container="`*container*`" ][ create_size="`*create_size*`" ][ flash_mode="{`*true* | *false*`}" ][ new_size="`*new_size*`" ]`

Required arguments

vg

VG identifier

Type: volume group

index

Disk index

Type: VG disk index

Optional arguments

`clone_from_adsf_file`

Path to an ADSF file

Type: ADSF path

`clone_from_image`

An image name/UUID

Type: image

`clone_from_vmdisk`

A vmdisk UUID

Type: VM disk

`clone_min_size`

Minimum size of the resulting clone(only applies to cloned disks)

Type: size with cskKmMgGtT suffix

`container`

Storage Container (only applies to newly-created disks)

Type: container

`create_size`

Size of new disk

Type: size with cskKmMgGtT suffix

`flash_mode`

Enable/Disable flash mode on this disk

Type: boolean

`new_size`

New size for the existing disk

Type: size with cskKmMgGtT suffix

Examples

1. Replace the disk at index 0 with blank 5GiB disk on ctr.

```
<acropolis> vg.disk_update my_vg 0 create_size=5G container=ctr
```

2. Replace the disk at index 0 with a clone of /ctr/plan9.iso.

```
<acropolis> vg.disk_update my_vg 0 clone_from_adsf_file=/ctr/plan9.iso
```

3. Replace the disk at index 0 with a clone of the existing vmdisk.

```
<acropolis> vg.disk_update my_vg 0 clone_from_vmdisk=0b4fc60b-
cc56-41c6-911e-67cc8406d096
```

4. Update the size of disk at index 0 to 5GiB.

```
<acropolis> vg.disk_update my_vg 0 new_size=5G
```

5. Disable flash mode for disk at index 0.

```
<acropolis> vg.disk_update my_vg 0 flash_mode=false
```

Retrieves information about a VG

`<acropolis> vg.get` *vg_list* `[ include_vmdisk_paths="{` *true* | *false* `}" ][ include_vmdisk_sizes="{` *true* | *false* `}" ]`

Required arguments

vg_list

VG identifier

Type: list of volume groups

Optional arguments

`include_vmdisk_paths`

Fetch disk paths

Type: boolean

Default: false

`include_vmdisk_sizes`

Fetch disk sizes (in bytes)

Type: boolean

Default: true

Lists all VGs

```
<acropolis> vg.list
```

Required arguments

None

Updates the specified VGs

```
<acropolis> vg.update vg_list [ annotation="annotation" ][
cbr_not_capable_reason="cbr_not_capable_reason" ][ flash_mode="{ true
| false }" ][ iscsi_target_prefix_list="iscsi_target_prefix_list" ][
load_balance_vm_attachments="{ true | false }" ][ name="name" ][ shared="{ true | false }" ][
target_secret_list="target_secret_list" ]
```

Required arguments

vg_list

Comma-delimited list of VG identifiers

Type: list of volume groups

Optional arguments

`annotation`

Annotation string

Type: string

`cbr_not_capable_reason`

If set, marks the VG incapable of CBR workflows

Type: string

`flash_mode`

Whether to enable/disable flash mode

Type: boolean

`iscsi_target_prefix_list`

Comma-delimited list of iscsi target prefixes for each of the VGs

Type: list of strings

`load_balance_vm_attachments`

Whether to enable/disable VG load balance VM attachments

Type: boolean

`name`

VG name

Type: string

`shared`

Allow VG to be attached to multiple VMs simultaneously?

Type: boolean

`target_secret_list`

Comma delimited CHAP secrets associated with each of the VGs. Enter "" to delete the secret.

Type: list of strings

Examples

1. Update target_secret for VG vg1 and delete target_secret for VG vg2

```
<acropolis> vg.update vg1,vg2 target_secret_list=new_vg1_secret,""
```

Update preferred_host for a given external attachment for a VG

`<acropolis> vg.update_external` *vg* `[ initiator_name="`*initiator_name*`" ][ initiator_network_id="`*initiator_network_id*`" ][ use_redirection="{`*true* | *false*`}" ]`

Required arguments

vg

VG identifier

Type: volume group

Optional arguments

`initiator_name`

Name of external initiator as a valid IQN

Type: VG external initiator name

`initiator_network_id`

Network identifier of external initiator as a valid IPv4 address

Type: VG external initiator name

`use_redirection`

Use iSCSI redirection for iSCSI login to this target?

Type: boolean

vm

Operations

- Retrieves affinity nodes for a given VM : `vm.affinity_nodeget`
- Enable VM-host affinity : `vm.affinity_set`
- Unsets affinity setting of specified VMs : `vm.affinity_unset`
- Clones a VM : `vm.clone`
- Creates one or more VMs : `vm.create`
- Deletes one or more VMs : `vm.delete`
- Attaches a new disk drive to a VM : `vm.disk_create`
- Detaches a disk drive from a VM and deletes the underlying disk : `vm.disk_delete`
- Gets details about the disks attached to a VM : `vm.disk_get`
- Lists the disks attached to a VM : `vm.disk_list`
- Updates the backing for the specified disk drive : `vm.disk_update`
- Force VM into the powered off state : `vm.force_off`
- Retrieves information about a VM : `vm.get`
- Attaches a new GPU to a VM : `vm.gpu_assign`
- Detaches a GPU from a VM : `vm.gpu_deassign`

- Initiates a Guest level Reboot of the VMs : `vm.guest_reboot`
- Initiates a Guest level Shutdown of the VMs : `vm.guest_shutdown`
- Lists all VMs : `vm.list`
- Live migrates a VM to another host : `vm.migrate`
- Clear the network function chain for this NIC : `vm.nic_clear_network_function_chain`
- Attaches a network adapter to a VM : `vm.nic_create`
- Deletes a NIC from a VM : `vm.nic_delete`
- Gets details about the NICs attached to a VM : `vm.nic_get`
- Lists the NICs attached to a VM : `vm.nic_list`
- Updates a network adapter, specified by the MAC address, on a VM : `vm.nic_update`
- Updates the network function chain for this NIC : `vm.nic_update_network_function_chain`
- Powers off the specified VMs : `vm.off`
- Powers on the specified VMs : `vm.on`
- Pauses the specified VMs : `vm.pause`
- Power cycles the specified VMs : `vm.power_cycle`
- Initiates a reboot by issuing an ACPI event : `vm.reboot`
- Resets the specified VMs : `vm.reset`
- Restores a VM to a snapshotted state : `vm.restore`
- Resumes the specified VMs : `vm.resume`
- Resumes all paused VMs : `vm.resume_all`
- Attaches a new serial port to a VM : `vm.serial_port_create`
- Detaches a serial port from a VM : `vm.serial_port_delete`
- Initiates a shutdown by issuing an ACPI event : `vm.shutdown`
- Creates one or more snapshots in a single consistency group : `vm.snapshot_create`
- Prints the graph representation of the snapshot history for a VM : `vm.snapshot_get_tree`
- Gets a list of all snapshots associated with a VM : `vm.snapshot_list`
- Updates the specified VMs : `vm.update`
- Updates a VM's boot device : `vm.update_boot_device`

Retrieves affinity nodes for a given VM

```
<acropolis> vm.affinity_nodeget vm_list
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Enable VM-host affinity

Host_list is the list of hosts that VM can affine to.

```
<acropolis> vm.affinity_set vm_list [ host_list="host_list" ]
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Optional arguments

`host_list`

Host list

Type: list of hosts

Unsets affinity setting of specified VMs

This will unset a VM affinity configuration, including policy, constraint, and binding entities.

```
<acropolis> vm.affinity_unset vm_list
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Clones a VM

The following suffixes are valid: M=2^20, G=2^30.

```
<acropolis> vm.clone name_list [ clone_affinity="{ true
| false }" ][ clone_from_snapshot="clone_from_snapshot" ][
clone_from_vm="clone_from_vm" ][ clone_ip_address="clone_ip_address"
][ memory="memory" ][ num_cores_per_vcpu="num_cores_per_vcpu" ][
num_threads_per_core="num_threads_per_core" ][ num_vcpus="num_vcpus" ]
```

Required arguments

name_list

Comma-delimited list of VM names

Type: list of strings with expansion wildcards

Optional arguments

`clone_affinity`

Clone source VM's affinity rules.

Type: boolean

`clone_from_snapshot`

Snapshot from which to clone

Type: snapshot

`clone_from_vm`

VM from which to clone

Type: VM

`clone_ip_address`

IP addresses to assign to clones

Type: list of IPv4 addresses

`memory`

Memory size

Type: size with MG suffix

`num_cores_per_vcpu`

Number of cores per vCPU

Type: int

`num_threads_per_core`

Number of threads per core

Type: int

`num_vcpus`

Number of vCPUs

Type: int

Creates one or more VMs

Enabling metrics allows host-specific metrics to be percolated to this VM.

```
<acropolis> vm.create name_list [ agent_vm="{ true | false }"
][ container="container" ][ disable_branding="{ true | false }" ][
enable_metrics="{ true | false }" ][ extra_flags="extra_flags" ][
memory="memory" ][ num_cores_per_vcpu="num_cores_per_vcpu" ][
num_threads_per_core="num_threads_per_core" ][ num_vcpus="num_vcpus" ][
num_vnuma_nodes="num_vnuma_nodes" ][ uefi_boot="{ true | false }" ][ vcpu_hard_pin="{ true
| false }" ]
```

Required arguments

name_list

Comma-delimited list of VM names

Type: list of strings with expansion wildcards

Optional arguments

`agent_vm`

Agent VM

Type: boolean

`container`

Container to store NVRAM disk if uefi_boot=True

Type: container

`disable_branding`

Disable Nutanix branding

Type: boolean

`enable_metrics`

Enable host metrics for this VM

Type: boolean

`extra_flags`

Additional VM flags as key=value pairs, separated by semicolon

Type: string

`memory`

Memory size

Type: size with MG suffix

Default: 2G

`num_cores_per_vcpu`

Number of cores per vCPU

Type: int

`num_threads_per_core`

Number of threads per core

Type: int

`num_vcpus`

Number of vCPUs

Type: int

Default: 1

`num_vnuma_nodes`

Number of vNUMA nodes

Type: int

`uefi_boot`

UEFI boot

Type: boolean

`vcpu_hard_pin`

Enable hard pinning vcpu to pcpus

Type: boolean

Deletes one or more VMs

If the VM is powered on, it will be powered off and then deleted.

```
<acropolis> vm.delete vm_list [ delete_snapshots="{ true | false }" ]
```

Required arguments

vm_list

Comma-delimited VM identifiers

Type: list of VMs

Optional arguments

`delete_snapshots`

Delete snapshots?

Type: boolean

Default: false

Attaches a new disk drive to a VM

Note that certain buses, like IDE, are not hot-pluggable.

```
<acropolis> vm.disk_create vm [ bus="bus" ][ cdrom="{ true | false }" ][
clone_from_adsf_file="clone_from_adsf_file" ][ clone_from_image="clone_from_image"
][ clone_from_vmdisk="clone_from_vmdisk" ][ clone_min_size="clone_min_size" ][
container="container" ][ create_size="create_size" ][ device_uuid="device_uuid" ][
empty="{ true | false }" ][ index="index" ][ scsi_passthru="{ true | false }" ]
```

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`bus`

Device bus

Type: Bus Type

`cdrom`

Indicates if the disk is a CDROM drive

Type: boolean

`clone_from_adsf_file`

Path to an ADSF file

Type: ADSF path

`clone_from_image`

An image name/UUID

Type: image

`clone_from_vmdisk`

A vmdisk UUID

Type: VM disk

`clone_min_size`

Minimum size of the resulting clone (only applies to cloned disks)

Type: size with cskKmMgGtT suffix

`container`

Container (only applies to newly-created disks)

Type: container

`create_size`

Size of new disk

Type: size with cskKmMgGtT suffix

`device_uuid`

Device UUID

Type: UUID

`empty`

Whether the disk is empty (only applies to CDROMs)

Type: boolean

`index`

Device index on bus

Type: int

`scsi_passthru`

Passthrough disk?

Type: boolean

Examples

1. Create a blank 5GiB disk on ctr, and attach it as SCSI:3.

```
<acropolis> vm.disk_create my_vm create_size=5G container=ctr bus=scsi index=3
```

2. Clone a disk from the ADSF file /ctr/plan9.iso, and use it as the backing image for a newly-created CD-ROM drive on the first available IDE slot.

```
<acropolis> vm.disk_create my_vm clone_from_adsf_file=/ctr/plan9.iso cdrom=1
```

3. Clone a disk from the existing vmdisk, and attach it to the first available SCSI slot.

```
<acropolis> vm.disk_create my_vm clone_from_vmdisk=0b4fc60b-
cc56-41c6-911e-67cc8406d096
```

4. Create a disk from an Acropolis image and attach it to the first SCSI slot.

```
<acropolis> vm.disk_create my_vm clone_from_image=my_image
```

5. Create a new empty CD-ROM drive, and attach it to the first available IDE slot.

```
<acropolis> vm.disk_create my_vm empty=1 cdrom=1
```

Detaches a disk drive from a VM and deletes the underlying disk

Note that certain buses, like IDE, are not hot-pluggable.

`<acropolis> vm.disk_delete` *vm* `[ device_uuid="`*device_uuid*`" ][ disk_addr="`*disk_addr*`" ]`

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`device_uuid`

Device UUID

Type: UUID

`disk_addr`

Disk address ("bus.index")

Type: VM disk

Gets details about the disks attached to a VM

`<acropolis> vm.disk_get` *vm* `[ device_uuid="`*device_uuid*`" ][ disk_addr="`*disk_addr*`" ][ include_vmdisk_paths="{`*true* | *false*`}" ][ include_vmdisk_sizes="{`*true* | *false*`}" ]`

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`device_uuid`

Device UUID

Type: UUID

`disk_addr`

Disk address ("bus.index")

Type: VM disk

`include_vmdisk_paths`

Fetch vmdisk paths

Type: boolean

Default: false

`include_vmdisk_sizes`

Fetch vmdisk sizes (in bytes)

Type: boolean

Default: true

Lists the disks attached to a VM

```
<acropolis> vm.disk_list vm
```

Required arguments

vm

VM identifier

Type: VM

Updates the backing for the specified disk drive

The existing disk image will be deleted and replaced by the new image (which may be a clone of the existing image).

```
<acropolis> vm.disk_update vm [ clone_from_adsf_file="clone_from_adsf_file" ][
clone_from_image="clone_from_image" ][ clone_from_vmdisk="clone_from_vmdisk" ][
clone_min_size="clone_min_size" ][ container="container" ][ create_size="create_size"
][ device_uuid="device_uuid" ][ disk_addr="disk_addr" ][ empty="{ true | false }" ][
new_size="new_size" ]
```

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`clone_from_adsf_file`

Path to an ADSF file

Type: ADSF path

`clone_from_image`

An image name/UUID

Type: image

`clone_from_vmdisk`

A vmdisk UUID

Type: VM disk

`clone_min_size`

Minimum size of the resulting clone (only applies to cloned disks)

Type: size with cskKmMgGtT suffix

`container`

Container (only applies to newly-created disks)

Type: container

`create_size`

Size of new disk

Type: size with cskKmMgGtT suffix

`device_uuid`

Device UUID

Type: UUID

`disk_addr`

Disk address ("bus.index")

Type: VM disk

`empty`

Whether the disk is empty (only applies to CDROMs)

Type: boolean

`new_size`

New size for the existing disk

Type: size with cskKmMgGtT suffix

Examples

1. Replace the disk at SCSI:0 with blank 5GiB disk on ctr.

```
<acropolis> vm.disk_update my_vm disk_addr=scsi.0 create_size=5G container=ctr
```

2. Replace the disk at IDE:0 with a clone of /ctr/plan9.iso. Note that if IDE:0 is a CD-ROM drive, it remains such.

```
<acropolis> vm.disk_update my_vm disk_addr=ide.0 clone_from_adsf_file=/ctr/
plan9.iso
```

3. Replace the disk at SCSI:0 with a clone of the existing vmdisk.

```
<acropolis> vm.disk_update my_vm disk_addr=scsi.0 clone_from_vmdisk=0b4fc60b-
cc56-41c6-911e-67cc8406d096
```

4. Eject the image from the CD-ROM drive at IDE:0.

```
<acropolis> vm.disk_update my_vm disk_addr=ide.0 empty=1
```

5. Update the size of disk at SCSSI:1 to 5GiB.

```
<acropolis> vm.disk_update my_vm disk_addr=scsi.1 new_size=5G
```

Force VM into the powered off state

Therefore, the user should take adequate precautions to ensure that the old instance is really gone.

`<acropolis> vm.force_off` *vm*

Required arguments

vm

VM identifier

Type: VM

Retrieves information about a VM

`<acropolis> vm.get` *vm_list* `[ include_address_assignments="{` *true* | *false* `}" ][ include_vmdisk_paths="{` *true* | *false* `}" ][ include_vmdisk_sizes="{` *true* | *false* `}" ]`

Required arguments

vm_list

VM identifier

Type: list of VMs

Optional arguments

`include_address_assignments`

Fetch configured IP addresses

Type: boolean

Default: true

`include_vmdisk_paths`

Fetch vmdisk paths

Type: boolean

Default: false

`include_vmdisk_sizes`

Fetch vmdisk sizes (in bytes)

Type: boolean

Default: true

Attaches a new GPU to a VM

Changes to the GPU configuration can only be made when VM is powered off.

`<acropolis> vm.gpu_assign` *vm* `[ extra_param="`*extra_param*`" ][ gpu="`*gpu*`" ]`

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`extra_param`

extra_parameters

Type: string

`gpu`

GPU

Type: gpu

Examples

1. Add a new GPU in passthrough mode

```
<acropolis> vm.gpu_assign my_vm gpu=Nvidia_Tesla_M60
```

Detaches a GPU from a VM

Changes to the GPU configuration can only be made when VM is powered off.

`<acropolis> vm.gpu_deassign` *vm* `[ gpu="`*gpu*`" ]`

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`gpu`

GPU

Type: gpu

Examples

1. Remove a new GPU in passthrough mode

```
<acropolis> vm.gpu_deassign my_vm gpu=Nvidia_Tesla_M60
```

Initiates a Guest level Reboot of the VMs

Requires NGT.

`<acropolis> vm.guest_reboot` *vm_list* `[ enable_script_exec="{` *true* | *false* `}" ][ fail_on_script_failure="{` *true* | *false* `}" ]`

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Optional arguments

`enable_script_exec`

Whether to execute pre-reboot script.

Type: boolean

Default: false

`fail_on_script_failure`

Whether to abort reboot if script fails.

Type: boolean

Default: false

Initiates a Guest level Shutdown of the VMs

Requires NGT.

`<acropolis> vm.guest_shutdown` *vm_list* `[ enable_script_exec="{` *true* | *false* `}" ][ fail_on_script_failure="{` *true* | *false* `}" ]`

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Optional arguments

`enable_script_exec`

Whether to execute pre-shutdown script.

Type: boolean

Default: false

`fail_on_script_failure`

Whether to abort shutdown if script fails.

Type: boolean

Default: false

Lists all VMs

```
<acropolis> vm.list [ memory="memory" ][ num_vcpus="num_vcpus" ][
power_state="power_state" ]
```

Required arguments

None

Optional arguments

`memory`

List VMs with the given amount of memory

Type: size with MG suffix

`num_vcpus`

List VMs with the given number of VCPUs

Type: int

`power_state`

List VMs with the given power state

Type: string

Live migrates a VM to another host

This limit is applied to each of the migrations individually.

```
<acropolis> vm.migrate vm_list [ bandwidth_mbps="bandwidth_mbps" ][ host="host" ][
live="{ true | false }" ][ max_downtime_ms="max_downtime_ms" ]
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Optional arguments

`bandwidth_mbps`

Maximum bandwidth in MiB/s

Type: int

Default: 0

`host`

Destination host

Type: host

`live`

Live migration or suspended migration?

Type: boolean

Default: true

`max_downtime_ms`

Maximum time in ms for which this Vm can be stunned while migrating

Type: int

Default: 0

Clear the network function chain for this NIC

```
<acropolis> vm.nic_clear_network_function_chain vm mac_addr
```

Required arguments

vm

VM identifier

Type: VM

mac_addr

NIC MAC address

Type: NIC address

Attaches a network adapter to a VM

If the VM is running, the NIC is hot-added to the VM.

```
<acropolis> vm.nic_create vm [ connected="{ true | false }" ][
ip="ip" ][ mac="mac" ][ model="model" ][ network="network" ][
network_function_nic_type="network_function_nic_type" ][ queues="queues" ][
request_ip="{ true | false }" ][ trunked_networks="trunked_networks" ][ type="type" ][
vlan_mode="vlan_mode" ]
```

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`connected`

Whether or not the NIC is connected.

Type: boolean

`ip`

IPv4 address

Type: IPv4 address

`mac`

MAC address

Type: MAC address

`model`

Virtual hardware model. Defaults to 'virtio', can also specify 'e1000'.

Type: string

`network`

Network identifier

Type: network

`network_function_nic_type`

Network function nic type

Type: Network function NIC Type

`queues`

Maximum number of Tx/Rx queue pairs (default: 1).

Type: int

`request_ip`

If true, then try to request an IP from Acropolis (static if the 'ip=' field is set, otherwise dynamic if the network is managed.) If false, then don't request an IP from Acropolis at all (even if 'ip=' is set.

Type: boolean

`trunked_networks`

List of trunked networks.

Type: list of ints

`type`

NIC Type

Type: NIC Type

Default: 1

`vlan_mode`

VLan Mode. Access by default.

Type: Vlan Type

Deletes a NIC from a VM

If the NIC to be removed is specified as the boot device in the boot configuration, the boot device configuration will be cleared as a side effect of removing the NIC.

```
<acropolis> vm.nic_delete vm mac_addr
```

Required arguments

vm

VM identifier

Type: VM

mac_addr

NIC MAC address

Type: NIC address

Gets details about the NICs attached to a VM

```
<acropolis> vm.nic_get vm [ include_address_assignments="{ true | false }" ][
mac_addr="mac_addr" ]
```

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`include_address_assignments`

Fetch configured IP addresses

Type: boolean

Default: true

`mac_addr`

NIC MAC address

Type: NIC address

Lists the NICs attached to a VM

```
<acropolis> vm.nic_list vm [ include_address_assignments="{ true | false }" ]
```

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`include_address_assignments`

Fetch configured IP addresses

Type: boolean

Default: true

Updates a network adapter, specified by the MAC address, on a VM

3) List of trunked networks.

```
<acropolis> vm.nic_update vm mac_addr [ connected="{ true | false }" ][
ip="ip" ][ network="network" ][ queues="queues" ][ request_ip="{ true | false }" ][
trunked_networks="trunked_networks" ][ type="type" ][ update_vlan_trunk_info="{ true |
false }" ][ vlan_mode="vlan_mode" ]
```

Required arguments

vm

VM identifier

Type: VM

mac_addr

NIC MAC address

Type: NIC address

Optional arguments

`connected`

Whether or not the NIC is connected.

Type: boolean

`ip`

IPv4 address

Type: IPv4 address

`network`

Network identifier

Type: network

`queues`

Maximum number of Tx/Rx queue pairs (default: 1).

Type: int

`request_ip`

If true, request a new IP address.

Type: boolean

`trunked_networks`

List of trunked networks.

Type: list of ints

`type`

NIC Type

Type: NIC Type

`update_vlan_trunk_info`

If true, then update vlan type and trunked network list.

Type: boolean

`vlan_mode`

VLan Mode. Access by default.

Type: Vlan Type

Updates the network function chain for this NIC

`<acropolis> vm.nic_update_network_function_chain` *vm mac_addr chain*

Required arguments

vm

VM identifier

Type: VM

mac_addr

NIC MAC address

Type: NIC address

chain

Network function chain

Type: network function chain

Powers off the specified VMs

`<acropolis> vm.off` *vm_list*

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Powers on the specified VMs

Note that no such host may be available.

`<acropolis> vm.on` *vm_list* `[ host="`*host*`" ]`

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Optional arguments

`host`

Host on which to power on the VM

Type: host

Pauses the specified VMs

```
<acropolis> vm.pause vm_list
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Power cycles the specified VMs

```
<acropolis> vm.power_cycle vm_list [ change_host="{ true | false }" ][ host="host" ]
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Optional arguments

`change_host`

Whether to power on on a different host

Type: boolean

Default: false

`host`

Host on which to power on the VM

Type: host

Initiates a reboot by issuing an ACPI event

```
<acropolis> vm.reboot vm_list
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Resets the specified VMs

```
<acropolis> vm.reset vm_list
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Restores a VM to a snapshotted state

In this case, the user may choose not to restore the VM's network adapters using the "restore_network_config" keyword argument.

```
<acropolis> vm.restore vm snapshot [ restore_network_config="{ true | false }" ]
```

Required arguments

vm

VM identifier

Type: VM

snapshot

Snapshot identifier

Type: snapshot

Optional arguments

`restore_network_config`

Whether to restore the VM's networking configuration

Type: boolean

Default: true

Resumes the specified VMs

```
<acropolis> vm.resume vm_list
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Resumes all paused VMs

```
<acropolis> vm.resume_all
```

Required arguments

None

Attaches a new serial port to a VM

Changes to the serial port configuration only take effect after a full power cycle.

```
<acropolis> vm.serial_port_create vm [ index="index" ][ type="type" ]
```

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`index`

Serial port index

Type: int

`type`

Serial port type

Type: serial port type

Examples

1. Add a new serial port at COM1 in server mode.

```
<acropolis> vm.serial_port_create my_vm index=0 type=kServer
```

Detaches a serial port from a VM

Changes to the serial port configuration only take effect after a full power cycle.

```
<acropolis> vm.serial_port_delete vm index
```

Required arguments

vm

VM identifier

Type: VM

index

Serial port index

Type: int

Examples

1. Remove the serial port at COM2.

```
<acropolis> vm.serial_port_delete my_vm 1
```

Initiates a shutdown by issuing an ACPI event

`<acropolis> vm.shutdown` *vm_list*

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Creates one or more snapshots in a single consistency group

If no snapshot name is provided, the snapshot will be referred to as "<vm_name>-<timestamp>", where the timestamp is in ISO 8601 format (YYYY-MM-DDTHH:MM:SS.mmmmmm).

`<acropolis> vm.snapshot_create` *vm_list* `[ snapshot_name_list="`*snapshot_name_list*`" ]`

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Optional arguments

`snapshot_name_list`

Comma-delimited list of names for each snapshot

Type: list of strings

Examples

1. Create a snapshot named 'dev-vm-gold' from a VM named 'dev-vm'.

```
<acropolis> vm.snapshot_create dev-vm snapshot_name_list=dev-vm-gold
```

2. Create a consistent snapshot across several VMs, using the default naming scheme.

```
<acropolis> vm.snapshot_create vm1,vm2,vm3
```

Prints the graph representation of the snapshot history for a VM

`<acropolis> vm.snapshot_get_tree` *vm*

Required arguments

vm

VM identifier

Type: VM

Gets a list of all snapshots associated with a VM

```
<acropolis> vm.snapshot_list vm
```

Required arguments

vm

VM identifier

Type: VM

Updates the specified VMs

Enabling metrics allows host-specific metrics to be percolated to this VM.

```
<acropolis> vm.update vm_list [ agent_vm="{ true | false }" ][
annotation="annotation" ][ cbr_not_capable_reason="cbr_not_capable_reason"
][ cpu_passthrough="{ true | false }" ][ disable_branding="{ true | false }"
][ enable_metrics="{ true | false }" ][ extra_flags="extra_flags" ][
ha_priority="ha_priority" ][ hwclock_timezone="hwclock_timezone" ][
memory="memory" ][ name="name" ][ num_cores_per_vcpu="num_cores_per_vcpu"
][ num_threads_per_core="num_threads_per_core" ][ num_vcpus="num_vcpus"
][ num_vnuma_nodes="num_vnuma_nodes" ][ vcpu_hard_pin="{ true | false }" ][
vga_console="{ true | false }" ]
```

Required arguments

vm_list

Comma-delimited list of VM identifiers

Type: list of VMs

Optional arguments

`agent_vm`

Agent vm

Type: boolean

`annotation`

Annotation string

Type: string

`cbr_not_capable_reason`

If set, marks the VM incapable of CBR workflows

Type: string

`cpu_passthrough`

Pass through all host CPU features.

Type: boolean

`disable_branding`

Disable Nutanix branding

Type: boolean

`enable_metrics`

Enable host metrics

Type: boolean

`extra_flags`

Additional VM flags as key=value pairs, separated by semicolon

Type: string

`ha_priority`

Numeric priority for HA restart. Negative value indicates no restart.

Type: int

`hwclock_timezone`

Hardware clock timezone

Type: timezone

`memory`

Memory size

Type: size with MG suffix

`name`

VM name

Type: string

`num_cores_per_vcpu`

Number of cores per vCPU

Type: int

`num_threads_per_core`

Number of threads per core

Type: int

`num_vcpus`

Number of vCPUs

Type: int

`num_vnuma_nodes`

Number of vNUMA nodes

Type: int

`vcpu_hard_pin`

Enable hard pinning vcpu to pcpus

Type: boolean

`vga_console`

VGA console device

Type: boolean

Updates a VM's boot device

Changes to the boot order only take effect after a full VM power cycle.

`<acropolis> vm.update_boot_device` *vm* `[ boot_device_order="`*boot_device_order*`" ][ disk_addr="`*disk_addr*`" ][ mac_addr="`*mac_addr*`" ]`

Required arguments

vm

VM identifier

Type: VM

Optional arguments

`boot_device_order`

Device boot order

Type: list of boot device types

`disk_addr`

Disk bus address

Type: VM disk

`mac_addr`

NIC MAC address

Type: NIC address

vm_group

Operations

- Add VMs to a VM group : `vm_group.add_vms`
- Enables vm-vm preferential anti-affinity : `vm_group.antiaffinity_set`
- Clears vm-vm anti-affinity : `vm_group.antiaffinity_unset`
- Creates one or more VM groups : `vm_group.create`
- Deletes one or more VM groups : `vm_group.delete`
- Retrieves information about a VM group : `vm_group.get`
- Lists all VM groups : `vm_group.list`
- Lists VMs configured to the VmGroup : `vm_group.list_vms`
- Remove VMs from a VM group : `vm_group.remove_vms`
- Updates the specified VM groups : `vm_group.update`

Add VMs to a VM group

`<acropolis> vm_group.add_vms` *vm_group* `[ vm_list="`*vm_list*`" ]`

Required arguments

vm_group

VmGroup identifier

Type: VM group type

Optional arguments

`vm_list`

Comma-delimited VM identifiers

Type: list of VMs

Enables vm-vm preferential anti-affinity

It is OK to violate the rule during VM placement if scheduler cannot honor the policy.

`<acropolis> vm_group.antiaffinity_set` *vm_group_list*

Required arguments

vm_group_list

Comma-delimited list of VM group identifiers

Type: list of VM group types

Clears vm-vm anti-affinity

`<acropolis> vm_group.antiaffinity_unset` *vm_group_list*

Required arguments

vm_group_list

Comma-delimited list of VM group identifiers

Type: list of VM group types

Creates one or more VM groups

```
<acropolis> vm_group.create name_list [ uuid="uuid" ]
```

Required arguments

name_list

Comma-delimited list of VM group names

Type: list of strings

Optional arguments

`uuid`

uuid

Type: string

Deletes one or more VM groups

```
<acropolis> vm_group.delete vm_group_list
```

Required arguments

vm_group_list

Comma-delimited VM group identifiers

Type: list of VM group types

Retrieves information about a VM group

```
<acropolis> vm_group.get vm_group_list
```

Required arguments

vm_group_list

VM group identifier

Type: list of VM group types

Lists all VM groups

```
<acropolis> vm_group.list
```

Required arguments

None

Lists VMs configured to the VmGroup

```
<acropolis> vm_group.list_vms vm_group
```

Required arguments

vm_group

VmGroup identifier

Type: VM group type

Remove VMs from a VM group

```
<acropolis> vm_group.remove_vms vm_group [ vm_list="vm_list" ]
```

Required arguments

vm_group

VmGroup identifier

Type: VM group type

Optional arguments

`vm_list`

Comma-delimited VM identifiers

Type: List of VM configured to VmGroup

Updates the specified VM groups

```
<acropolis> vm_group.update vm_group_list [ name="name" ]
```

Required arguments

vm_group_list

Comma-delimited list of VM group identifiers

Type: list of VM group types

Optional arguments

`name`

VM group name

Type: string

2

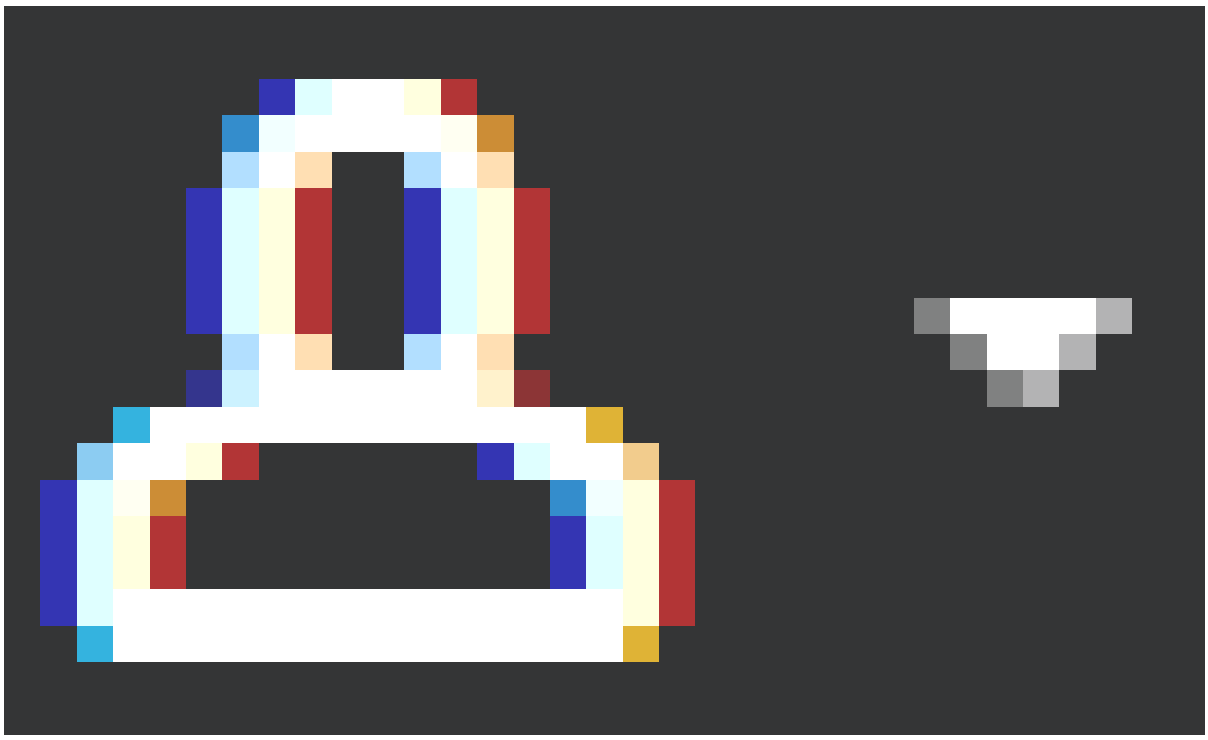
Nutanix Command-Line Interface (nCLI)

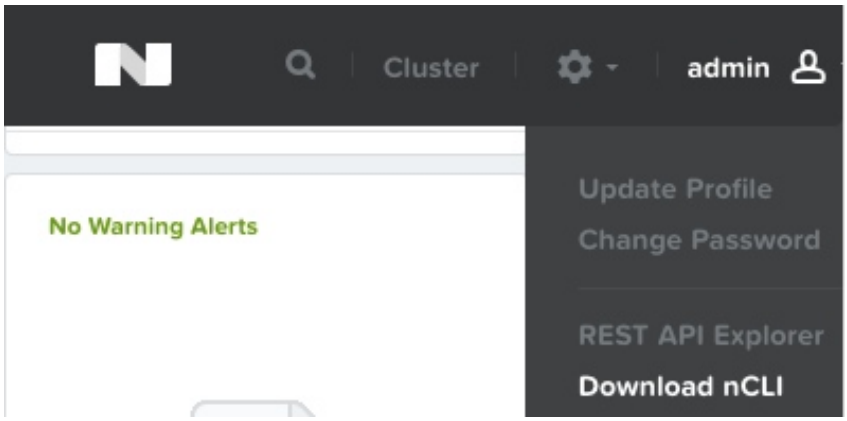
The Nutanix command-line interface (nCLI) allows you to run system administration commands against the Nutanix cluster from any of the following machines:

- Your local machine (preferred)
- Any Controller VM in the cluster

Installing the nCLI on Your Local System

Tip: Refer to Default Cluster Credentials for the default credentials of all cluster components.

1. Verify that your system has Java Runtime Environment (JRE) version 5.0 or higher.
   To check which version of Java is installed on your system or to download the latest version, go to http://www.java.com/en/download/installed.jsp.
2. Download the nCLI.
   a. Connect to the Nutanix web console.
   b. Click the user icon
   
   at the top of the console.



c. Select **Download nCLI**.

d. Download and unzip the file on your local system.

3. Configure your Windows %PATH% or Linux $PATH environment variables.

   - `ncli` directory (for example, `C:\ncli`)
   - JRE `bin` directory (for example, `C:\Program Files\Java\jre1.6.0_21\bin`)

   The procedure to complete this step depends on your operating system. For more information, go to http://java.com/en/download/help/path.xml.

Initiating an nCLI Session From Your Local System

Before you begin

Install the nCLI on your local system by following Installing the nCLI on Your Local System on page 54.

1. On your local system, open a command prompt (such as `bash` for Linux or `CMD` for Windows).
2. At the command prompt, start the nCLI by using one of the following commands.

   a. `ncli` -s *management_ip_addr* -u '*username*' -p '*user_password*'
   In this case, the console displays the password.

   b. `ncli` -s *management_ip_addr* -u '*username*' -p
   In this case, you are prompted to specify the password.

   - Replace *management_ip_addr* with the IP address of any Nutanix Controller VM in the cluster.
   - Replace *username* with the name of the user (if not specified, the default is admin).
   - (Optional) Replace *user_password* with the password of the user.

   Note: When specifying the `-p 'user_password'` command-line argument to establish an nCLI session interactively, the credentials are stored in the user's history file. This flag is intended for use in automated scripts or third-party application integration to nCLI.

   To prevent retention of the password for interactive login, specify the `-p` command flag alone without the password as described in step 2(b). An interactive prompt is provided to supply the password, which will not be stored in the user's history file.

Table 1: Troubleshooting

Error	Explanation/Resolution
`ncli` not found or not recognized as a command	The Windows %PATH% or Linux $PATH environment variable is not set.
`Error: Bad credentials`	The admin user password has been changed from the default and you did not specify the correct password. Type `exit` and start the nCLI again with the correct password.
`Error: Could not connect to Nutanix Gateway`	The cluster is not started. Log on to a Controller VM as the nutanix user and run the following command: `cluster start` When the cluster is up, exit the nCLI and start it again.

Results

The Nutanix CLI is now in interactive mode. To exit this mode, type `exit` at the `ncli>` prompt.

Command Format

Nutanix Command-Line Interface commands must match the following format:

```
ncli> entity action parameter1=value parameter2=value ...
```

entity can be replaced by any Nutanix entity, such as *cluster* or *disk*.

action can be replaced by any valid action for the preceding entity. Each entity has a unique set of actions, but a common action across all entities is list. For example, you can type the following command to request a list of all storage pools in the cluster.

```
ncli> storagepool list
```

Some actions require parameters at the end of the command. For example, when creating an NFS datastore, you need to provide both the name of the datastore as it will appear to the hypervisor and the name of the source storage container.

```
ncli> datastore create name="NTNX-NFS" ctr-name="nfs-ctr"
```

Parameter-value pairs can be listed in any order, as long as they are preceded by a valid entity and action.

Tip: To avoid syntax errors, surround all string values with double-quotes, as demonstrated in the preceding example. This is particularly important when specifying parameters that accept a list of values.

Embedded Help

The nCLI provides assistance on all entities and actions. By typing `help` at the command line, you can request additional information at one of three levels of detail.

`help`
: Provides a list of entities and their corresponding actions

`entity help`
: Provides a list of all actions and parameters associated with the entity, as well as which parameters are required, and which are optional

`entity action help`
: Provides a list of all parameters associated with the action, as well as a description of each parameter

The nCLI provides additional details at each level. To control the scope of the nCLI help output, add the detailed parameter, which can be set to either *true* or *false*.

For example, type the following command to request a detailed list of all actions and parameters for the `cluster` entity.

```
ncli> cluster help detailed=true
```

You can also type the following command if you prefer to see a list of parameters for the `cluster edit-params` action without descriptions.

```
ncli> cluster edit-params help detailed=false
```

nCLI Entities

alerts: An Alert
authconfig: Configuration information used to authenticate user
cloud: Manage AWS or AZURE Cloud
cluster: A Nutanix Complete Cluster
container: A Storage Container is a container for virtual disks
data-at-rest-encryption: Manage data-at-rest-encryption related operations
data-at-rest-encryption-certificate: Manage data-at-rest-encryption related digital certificates
datastore: An NFS Datastore
disk: A Physical Disk
events: An Event
failover-cluster: Hyper-V failover cluster
file-server: Minerva file server
health-check: A health check
host: A Physical Host hosts Virtual Machines
http-proxy: An HTTP Proxy
key-management-server: Manage key management servers
license: License for a Nutanix cluster
managementserver: An infrastructure management server such as VCenter
multicluster: A Nutanix Management Console to manage multiple clusters
network: Network specific commands
nutanix-guest-tools: Admin commands for Nutanix Guest Tools
progress-monitor: Monitor progress of long running tasks
protection-domain: A protection domain to be used for Data Protection
pulse-config: Configuration information used for Pulse setup
rackable-unit: A rackable unit
remote-site: A remote cluster to be used for replicating data
rsyslog-config: Configuration information to send logs to remote servers
smb-server: The Nutanix SMB file server
snapshot: Snapshot of a Virtual Disk
snmp: An SNMP agent
software: NOS Software Release
ssl-certificate: Manage SSL certificates
storagepool: A Pool of Physical Disks
storagetier: A Tier of physical storage
task: A Task
user: A User
vdisk: A Virtual Disk
virtual-disk: Commands for performing different actions on Virtual Disks.

virtualmachine: A Virtual Machine
volume-group: A Volume Group
vstore: A file namespace in a Storage Container
vzone: A vZone

Nutanix Command-Line Interface Reference

CLI Reference Conventions

This command-line interface reference uses the following conventions.

- Parameters in italic are unique to your environment.

  ```
  value
  ```

- Parameters in square brackets are optional.

  ```
  [ value ]
  ```

- Parameters in curly brackets must be one of a limited set of values.

  ```
  { value1 | value2 }
  ```

  One example is boolean parameters: `{ true | false }`
- The `keyword` is a literal string required by the command, and the `value` is the unique value for your environment.

  ```
  keyword=value
  ```

`alerts`: Alert

Description An Alert

Alias `alert`

Operations

- Acknowledge Alerts : `acknowledge` | `ack`
- Update Alert Configuration : `edit-alert-config` | `update-alert-config`
- List Alert Configuration : `get-alert-config`
- List history of Alerts : `history`
- List of unresolved Alerts : `list` | `ls`
- Resolve Alerts : `resolve`

Acknowledge Alerts

```
ncli> alerts { acknowledge  | ack } ids="ids"
```

Required arguments

`ids`

A comma-separated list of ids of the Alerts

Update Alert Configuration

```
ncli> alerts { edit-alert-config  | update-alert-config }[ enable="{ true | false }" ][
enable-default-nutanix-email="{ true | false }" ][ enable-email-digest="{ true | false }" ][
email-contacts="email_contacts" ]
```

Required arguments

None

Optional arguments

`enable`

Enable Alert emails

`enable-default-nutanix-email`

Enable default Nutanix email

`enable-email-digest`

Enable alert email digest

`email-contacts`

Comma-separated list of emails to be used while sending alerts. Set to '-' to clear all the existing emails.

List Alert Configuration

```
ncli> alerts { get-alert-config }
```

Required arguments

None

List history of Alerts

```
ncli> alerts { history } duration="duration" [ acknowledged="acknowledged" ][
resolved="resolved" ][ auto-resolved="auto_resolved" ][ max-alerts="max_alerts" ]
```

Required arguments

`duration`

Duration (in days) for getting the history of Alerts

Optional arguments

`acknowledged`

Acknowledged Alerts?

`resolved`

Resolved Alerts?

`auto-resolved`

Auto resolved Alerts?

`max-alerts`

Maximum number of Alerts to fetch (permitted max value = 1000)

Default: 100

List of unresolved Alerts

```
ncli> alerts { list  | ls }[ acknowledged="acknowledged" ][ max-alerts="max_alerts" ]
```

Required arguments

None

Optional arguments

`acknowledged`

Acknowledged Alerts?

`max-alerts`

Maximum number of Alerts to fetch (permitted max value = 1000)

Default: 100

Resolve Alerts

```
ncli> alerts { resolve } ids="ids"
```

Required arguments

`ids`

A comma-separated list of ids of the Alerts

`authconfig`: Authentication Configuration

Description Configuration information used to authenticate user

Alias

Operations

- Add directory role mapping : `add-role-mapping`
- Comma-separated list of values to be added to the existing directory role mapping : `add-to-role-mapping-values`
- Create Directory Service Configuration : `create-directory` | `add-directory`
- delete ca chain certificate : `delete-ca-chain-certificate` | `delete-ca-chain-cert`
- Delete directory role mappings : `delete-role-mapping`
- Update Authentication Configuration : `edit` | `update`
- Update Directory Service Configuration : `edit-directory` | `update-directory`
- Get Client Auth Configuration : `get-client-authentication-config`
- Get the list of entity values for the specified entity type and the directory name : `get-directory-values-by-type`
- List Authentication Configuration : `list` | `ls`
- List Authentication Configuration : `list-directory` | `ls-directory`
- Get the list of directory role mappings : `list-role-mappings` | `ls-role-mappings`
- Remove Authentication Configuration : `remove` | `rm`
- Remove Directory Service Configuration : `remove-directory` | `rm-directory`
- Comma-separated list of values to be removed from the existing directory role mapping : `remove-from-role-mapping-values`
- Test LDAP Connection and authentication : `test-ldap-connection`
- Enable/Disable client authentication : `update-client-authentication`
- Update service account (CAC) : `update-service-account`

Add directory role mapping

```
ncli> authconfig { add-role-mapping }role="role" entity-type="entity_type" entity-values="entity_values" name="name"
```

Required arguments

`role`

Role of the directory.

`entity-type`

Type of directory entity.

`entity-values`

List of entity values.

`name`

Directory Name

Comma-separated list of values to be added to the existing directory role mapping

```
ncli> authconfig { add-to-role-mapping-values }name="name" role="role" entity-type="entity_type" entity-values="entity_values"
```

Required arguments

`name`

Name

`role`

Role

`entity-type`

Entity Type

`entity-values`

Values

Create Directory Service Configuration

```
ncli> authconfig { create-directory  | add-directory}directory-
type="directory_type" connection-type="connection_type" directory-url="directory_url"
domain="domain" name="name" service-account-username="service_account_username"
[ group-search-type="group_search_type" ][ user-object-class="user_object_class" ][
user-search-base="user_search_base" ][ username-attribute="username_attribute" ][
group-object-class="group_object_class" ][ group-search-base="group_search_base"
][ group-member-attribute="group_member_attribute" ][ group-member-
attribute-value="group_member_attribute_value" ][ service-account-
password="service_account_password" ]
```

Required arguments

`directory-type`

Type of the Directory Service

`connection-type`

Connection type for the Directory Service

`directory-url`

Url to connect to the Directory Service

`domain`

Domain of the Directory Service

`name`

Name of the Directory Service

`service-account-username`

User name of the directory administrator

Optional arguments

`group-search-type`

Type of the search whether RECURSIVE or NON_RECURSIVE

`user-object-class`

Object class for users

`user-search-base`

Search base for users

`username-attribute`

Unique identifier for a user that can be used for authentication

`group-object-class`

Object class for groups

`group-search-base`

Search base for groups

`group-member-attribute`

Attribute in group that associates users to group

`group-member-attribute-value`

User attributevalue that will be used in group entity to associate user to the group

`service-account-password`

Password of the directory administrator

delete ca chain certificate

```
ncli> authconfig { delete-ca-chain-certificate  | delete-ca-chain-cert }ca-chain-certificate-name="ca_chain_certificate_name"
```

Required arguments

`ca-chain-certificate-name`

Name of the Certificate

Delete directory role mappings

```
ncli> authconfig { delete-role-mapping }name="name"[ role="role" ][ entity-type="entity_type" ]
```

Required arguments

`name`

Directory Name

Optional arguments

`role`

Role

`entity-type`

Entity Type

Update Authentication Configuration

```
ncli> authconfig { edit  | update }auth-type-list="auth_type_list"[ directory-list="directory_list" ]
```

Required arguments

`auth-type-list`

Authentication Types

Optional arguments

`directory-list`

Id of an alert type.

Update Directory Service Configuration

```
ncli> authconfig { edit-directory  | update-directory }directory-type="directory_type" connection-type="connection_type" directory-url="directory_url" domain="domain" name="name"[ group-search-type="group_search_type" ][ user-object-class="user_object_class" ][ user-search-base="user_search_base" ][ username-attribute="username_attribute" ][ group-object-class="group_object_class" ][ group-search-base="group_search_base" ][ group-member-attribute="group_member_attribute" ][ group-member-attribute-value="group_member_attribute_value" ][ service-account-username="service_account_username" ][ service-account-password="service_account_password" ]
```

Required arguments

`directory-type`

Type of the Directory Service

`connection-type`
: Connection type for the Directory Service

`directory-url`
: Url to connect to the Directory Service

`domain`
: Domain of the Directory Service

`name`
: Name of the Directory Service

Optional arguments

`group-search-type`
: Type of the search whether RECURSIVE or NON_RECURSIVE

`user-object-class`
: Object class for users

`user-search-base`
: Search base for users

`username-attribute`
: Unique identifier for a user that can be used for authentication

`group-object-class`
: Object class for groups

`group-search-base`
: Search base for groups

`group-member-attribute`
: Attribute in group that associates users to group

`group-member-attribute-value`
: User attributevalue that will be used in group entity to associate user to the group

`service-account-username`
: User name of the directory administrator

`service-account-password`
: Password of the directory administrator

Get Client Auth Configuration

```
ncli> authconfig { get-client-authentication-config }
```

Required arguments

None

Get the list of entity values for the specified entity type and the directory name.

```
ncli> authconfig { get-directory-values-by-type }name="name" directory-
name="directory_name" username="username" password="password" entity-
type="entity_type"
```

Required arguments

`name`
: Name of the Directory

`directory-name`
: Name of the directory.

`username`

User name.

`password`

Password for corresponding username.

`entity-type`

Type of the directory entity.

List Authentication Configuration

```
ncli> authconfig {  list   | ls }
```

Required arguments

None

List Authentication Configuration

```
ncli> authconfig {  list-directory   | ls-directory }[ name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Directory Service

Get the list of directory role mappings

```
ncli> authconfig {  list-role-mappings   | ls-role-mappings } name="name" [
role="role" ][ entity-type="entity_type" ]
```

Required arguments

`name`

Name

Optional arguments

`role`

Role

`entity-type`

Entity Type

Remove Authentication Configuration

```
ncli> authconfig {  remove   | rm }
```

Required arguments

None

Remove Directory Service Configuration

```
ncli> authconfig {  remove-directory   | rm-directory } name="name"
```

Required arguments

`name`

Name of the Directory

Comma-separated list of values to be removed from the existing directory role mapping

```
ncli> authconfig {  remove-from-role-mapping-values  } name="name" role="role"
entity-type="entity_type" entity-values="entity_values"
```

Required arguments

`name`

Name

`role`

Role

`entity-type`

Entity Type

`entity-values`

Values

Test LDAP Connection and authentication

```
ncli> authconfig { test-ldap-connection }directory-name="directory_name"
password="password" username="username"
```

Required arguments

`directory-name`

Directory name to test LDAP configuration.

`password`

Password to test LDAP configuration.

`username`

Username to test LDAP configuration.

Enable/Disable client authentication

```
ncli> authconfig { update-client-authentication }enable-client-auth="{ true |
false }"
```

Required arguments

`enable-client-auth`

Enable client auth

Update service account (CAC)

```
ncli> authconfig { update-service-account }[ enable="{ true | false }" ][ directory-
name="directory_name" ][ username="username" ][ password="password" ]
```

Required arguments

None

Optional arguments

`enable`

Enable Service Account (CAC)

`directory-name`

Service Account (CAC) Directory Name

`username`

Service Account (CAC) Username

`password`

Service Account (CAC) Password

`cloud`: Cloud

Description Manage AWS or AZURE Cloud

Alias

Operations

- Add AWS or AZURE credentials : `add-credentials`
- Clear all cloud credentials : `clear-all-credentials`
- Deploy and configure a Nutanix CVM on cloud, and create a Remote Site on the local cluster which points to the new CVM : `deploy-remote-site`
- Destroy a cloud remote site : `destroy-remote-site`
- List AWS credentials : `ls-credentials`
- List AWS CVM images : `ls-cvm-images`
- List AWS CVMs : `ls-cvms`
- List AWS VPC subnets : `ls-subnets`
- Remove AWS credentials : `remove-credentials`
- Set default AWS credentials : `set-default-credentials`

Add AWS or AZURE credentials

```
ncli> cloud { add-credentials }cloud-type="cloud_type"[ name="name" ][ aws-access-key="aws_access_key" ][ aws-secret-key="aws_secret_key" ][ azure-certificate-file-path="azure_certificate_file_path" ][ azure-subscription-id="azure_subscription_id" ]
```

Required arguments

`cloud-type`

Type of the cloud service

Optional arguments

`name`

Given name of the credentials

`aws-access-key`

AWS access key

`aws-secret-key`

AWS secret key

`azure-certificate-file-path`

Path to the AZURE certificate file

`azure-subscription-id`

AZURE subscription id

Clear all cloud credentials

```
ncli> cloud { clear-all-credentials }cloud-type="cloud_type"
```

Required arguments

`cloud-type`

Cloud type

Deploy and configure a Nutanix CVM on cloud, and create a Remote Site on the local cluster which points to the new CVM

```
ncli> cloud { deploy-remote-site }cloud-type="cloud_type" region="region" remote-site-name="remote_site_name" local-ctr-name="local_ctr_name" connectivity-type="connectivity_type"[ instance-name="instance_name" ][ credential-name="credential_name" ][ image-id="image_id" ][ image-name="image_name" ][ admin-password="admin_password" ][ remote-sp-name="remote_sp_name" ][ remote-ctr-name="remote_ctr_name" ][ subnet-id="subnet_id" ][ ssh-tunnel-port="ssh_tunnel_port" ][ azure-virtual-network="azure_virtual_network" ][ enable-proxy="{ true | false }" ][ enable-on-wire-compression="{ true | false }" ][ max-bandwidth="max_bandwidth" ][ instance-type="instance_type" ]
```

Required arguments

`cloud-type`

Type of the cloud service

`region`

Name of the region, eg. us-east-1 | eu-west-1 | East Asia | Brazil South

`remote-site-name`

Name of the Remote Site on the local cluster

`local-ctr-name`

Name of a local Storage Container to be backed-up to the deployed CVM

`connectivity-type`

The platform to use for the cloud instance. Choose between 'vpn'(recommended) and 'ssh-tunnel'

Optional arguments

`instance-name`

Prefix for the name of the instance deployed in the cloud

`credential-name`

Given name of the credentials

`image-id`

ID of the CVM image to use for deployment

`image-name`

Name of the CVM image to use for deployment

`admin-password`

Password for the nutanix user on the CVM deployed in the cloud

`remote-sp-name`

Name of the storage pool to be created on the deployed CVM

`remote-ctr-name`

Name of Storage Container to be created on the deployed CVM

`subnet-id`

Cloud subnet ID

`ssh-tunnel-port`

Port to use for SSH tunnel, in the range of 3000-3099

`azure-virtual-network`

Azure VPN Name

`enable-proxy`

Boolean parameter to indicate whether the addresses specified in address-list can be used as a proxy to communicate with other Nutanix components on the remote site

`enable-on-wire-compression`

Enable or disable on-wire compression of data during replication

Default: true

`max-bandwidth`

Maximum bandwidth (in kilobytes per sec) to be used while replicating to the remote site. If not specified, restriction is not placed on maximum bandwidth used by replication

`instance-type`

Cloud instance type

Destroy a cloud remote site

```
ncli> cloud { destroy-remote-site }remote-site-name="remote_site_name"[
credential-id="credential_id" ]
```

Required arguments

`remote-site-name`

Name of the Remote Site

Optional arguments

`credential-id`

Given id of the credentials

List AWS credentials

```
ncli> cloud { ls-credentials }cloud-type="cloud_type"[ name="name" ][ is-
default="is_default" ]
```

Required arguments

`cloud-type`

Type of the cloud service

Optional arguments

`name`

Given name of the credentials

`is-default`

Whether the AWS Credentials are being used as the default

List AWS CVM images

```
ncli> cloud { ls-cvm-images }cloud-type="cloud_type"[ credential-
name="credential_name" ][ region="region" ][ list-local-only="list_local_only" ]
```

Required arguments

`cloud-type`

Type of the cloud service

Optional arguments

`credential-name`

Given name of the credentials

`region`

Name of the region, eg. us-east-1 | eu-west-1 | East Asia | Brazil South

`list-local-only`

Whether the list is local to the system (only for Azure cloud type)

List AWS CVMs

```
ncli> cloud { ls-cvms }cloud-type="cloud_type" region="region"[ credential-
name="credential_name" ]
```

Required arguments

`cloud-type`

Type of the cloud service

`region`

Name of the region, eg. us-east-1 | eu-west-1 | East Asia | Brazil South

Optional arguments

`credential-name`

Given name of the credentials

List AWS VPC subnets

```
ncli> cloud { ls-subnets }cloud-type="cloud_type" region="region"[ credential-name="credential_name" ]
```

Required arguments

`cloud-type`

Type of the cloud service

`region`

Name of the region, eg. us-east-1 | eu-west-1 | East Asia | Brazil South

Optional arguments

`credential-name`

Given name of the credentials

Remove AWS credentials

```
ncli> cloud { remove-credentials }cloud-type="cloud_type" name="name"
```

Required arguments

`cloud-type`

Type of the cloud service

`name`

Given name of the credentials

Set default AWS credentials

```
ncli> cloud { set-default-credentials }name="name" cloud-type="cloud_type"
```

Required arguments

`name`

Given name of the credentials

`cloud-type`

Type of the cloud service

`cluster`: Cluster

Description A Nutanix Complete Cluster

Alias

Operations

- Add the configured node to the cluster : `add-node`
- Add public key to the cluster : `add-public-key`
- Add addresses to the list of name servers : `add-to-name-servers`
- Add addresses to the list of NFS subnet whitelist : `add-to-nfs-whitelist`
- Add addresses to the list of NTP servers : `add-to-ntp-servers`
- Add a Metro Availability Witness : `add-witness`
- Clear the external IP address of the Cluster : `clear-external-ip-address`
- Clear configuration of SMTP Server used for transmitting alerts and report emails to Nutanix support : `clear-smtp-server`
- Configure discovered node with IP addresses (Hypervisor, CVM and IPMI addresses) : `configure-node`
- Discover new nodes available to add to the cluster : `discover-nodes`
- Edit the security params of a Cluster : `edit-cvm-security-params`
- Edit the hypervisor LLDP config of the Cluster : `edit-hypervisor-lldp-params`
- Edit the hypervisor security compliance config of a Cluster : `edit-hypervisor-security-params`
- Edit params of a Cluster : `edit-params` | `edit-info`
- Generates and downloads the csr from discovered node based on certification information from the cluster : `generate-csr-for-discovered-node`
- Get the security config for the Cluster : `get-cvm-security-config`
- Get the domain fault tolerance state of the cluster : `get-domain-fault-tolerance-status` | `get-dm-ft-stat`
- Get the hypervisor LLDP config of the Cluster : `get-hypervisor-lldp-config`
- Get the hypervisor security config for the Cluster : `get-hypervisor-security-config`
- Get Ipmi monitoring status : `get-ipmi-monitoring-status`
- Gets the Microsoft Azure Account details : `get-la-jolla-azure-info`
- Get the list of name servers : `get-name-servers`
- Get the list of NFS subnet whitelist : `get-nfs-whitelist`
- Get the list of NTP servers : `get-ntp-servers`
- Get params of a Cluster : `get-params` | `info`
- Get the redundancy state of the cluster : `get-redundancy-state`
- Get remote support status on a Cluster : `get-remote-support-status`
- Check the removal status for Physical Hosts : `get-remove-status` | `get-rm-status`
- Get configuration of SMTP Server used for transmitting alerts and report emails to Nutanix support : `get-smtp-server`
- Join the Nutanix storage cluster to the Windows AD domain specified in the cluster name : `join-domain`
- Get the list of public keys configured in the cluster : `list-public-keys` | `ls-public-keys`
- List All registered Metro Availability Witness : `list-witness`
- Remove all the public keys previously configured in the cluster : `remove-all-public-keys` | `rm-all-public-keys`
- Remove addresses from the list of name servers : `remove-from-name-servers`
- Remove addresses from the list of NFS subnet whitelist : `remove-from-nfs-whitelist`
- Remove addresses from the list of NTP servers : `remove-from-ntp-servers`
- Delete public key with the specified name from the cluster : `remove-public-key` | `rm-public-key`
- Begin the process of removing a Physical Host : `remove-start` | `rm-start` | `delete`
- Delete a Metro Availability Witness : `remove-witness`
- Send an email to test the SMTP Server configuration : `send-test-email`
- Sets the DVM configuration parameters for LaJolla cluster : `set-dvm-configuration`
- Set the external IP address (IPv4) of the Cluster : `set-external-ip-address`
- Enable or disable Ipmi monitoring : `set-ipmi-monitoring-status`

Add the configured node to the cluster. In case of compute only node, cvm ip corresponds to host ip

```
ncli> cluster { add-node } node-uuid="node_uuid" [ server-certificate-
list="server_certificate_list" ]
```

Required arguments

`node-uuid`

UUID of the new node

Optional arguments

`server-certificate-list`

Comma-separated list of the key management server uuid and corresponding certificate file path. List should be of format <server_uuid:path_to_certificate>

Add public key to the cluster

```
ncli> cluster { add-public-key } name="name" file-path="file_path"
```

Required arguments

`name`

Name of the cluster public key

`file-path`

Full path of the public key file

Add addresses to the list of name servers

```
ncli> cluster { add-to-name-servers } servers="servers"
```

Required arguments

`servers`

Comma-separated list of IP addresses to be included in the name servers list

Add addresses to the list of NFS subnet whitelist

```
ncli> cluster { add-to-nfs-whitelist } ip-subnet-masks="ip_subnet_masks"
```

Required arguments

`ip-subnet-masks`

Comma-separated list with entries of the form 'IP/subnet mask'(a.b.c.d/l.m.n.o) to be included in the NFS subnet whitelist

Add addresses to the list of NTP servers

```
ncli> cluster { add-to-ntp-servers } servers="servers"
```

Required arguments

`servers`

Comma-separated list of IP addresses/Host names to be included in the NTP servers list

Add a Metro Availability Witness

```
ncli> cluster { add-witness } external-ip-address="external_ip_address"
username="username" password="password" [ test-only="test_only" ]
```

Required arguments

`external-ip-address`

external IP address of the Witness

`username`

username

`password`

password

Optional arguments

`test-only`

Only test the config without storing it.

Default: false

Clear the external IP address of the Cluster

```
ncli> cluster { clear-external-ip-address }
```

Required arguments

None

Clear configuration of SMTP Server used for transmitting alerts and report emails to Nutanix support

```
ncli> cluster { clear-smtp-server }
```

Required arguments

None

Configure discovered node with IP addresses (Hypervisor, CVM and IPMI addresses)

```
ncli> cluster { configure-node }node-uuid="node_uuid"[ cvm-ip="cvm_ip" ][
hypervisor-ip="hypervisor_ip" ][ ipmi-ip="ipmi_ip" ][ ipmi-netmask="ipmi_netmask" ][
ipmi-gateway="ipmi_gateway" ]
```

Required arguments

`node-uuid`

UUID of the new node

Optional arguments

`cvm-ip`

IP address of the controller VM

`hypervisor-ip`

IP address of the Hypervisor Host

`ipmi-ip`

IPMI address of the node

`ipmi-netmask`

IPMI netmask of the node

`ipmi-gateway`

IPMI gateway of the node

Discover new nodes available to add to the cluster

```
ncli> cluster { discover-nodes }
```

Required arguments

None

Edit the security params of a Cluster

```
ncli> cluster { edit-cvm-security-params }[ enable-aide="{ true | false }" ][ enable-
core="{ true | false }" ][ enable-high-strength-password="{ true | false }" ][ enable-
banner="{ true | false }" ][ enable-snmpv3-only="{ true | false }" ][ schedule="schedule" ]
```

Required arguments

None

Optional arguments

`enable-aide`

Enable intrusion detection service.

`enable-core`

Enable kernel core dumps.

`enable-high-strength-password`

Enable to set the minimum length for password to 14 and remember to 24 in PAM.

`enable-banner`

Enable DoD knowledge of consent banner for SSH and console logins.

`enable-snmpv3-only`

Enable to set the CVM host to either ignore or explicitly restrict v1 and v2 snmp based connections.

`schedule`

Set cron schedule to run Salt periodically.

Edit the hypervisor LLDP config of the Cluster

```
ncli> cluster { edit-hypervisor-lldp-params }[ enable-lldp-tx="{ true | false }" ]
```

Required arguments

None

Optional arguments

`enable-lldp-tx`

Enable LLDP TX mode.

Edit the hypervisor security compliance config of a Cluster

```
ncli> cluster { edit-hypervisor-security-params }[ enable-aide="{ true | false }" ][
enable-core="{ true | false }" ][ enable-high-strength-password="{ true | false }" ][ enable-
banner="{ true | false }" ][ schedule="schedule" ]
```

Required arguments

None

Optional arguments

`enable-aide`

Enable intrusion detection service.

`enable-core`

Enable kernel core dumps.

`enable-high-strength-password`

Enable to set the minimum length for password to 14 and remember to 24 in PAM.

`enable-banner`

Enable DoD knowledge of consent banner for SSH and console logins.

`schedule`

Set cron schedule to run Salt periodically.

Edit params of a Cluster

```
ncli> cluster { edit-params  | edit-info }[ new-name="new_name"
][ support-verbosity-level="support_verbosity_level" ][ external-ip-
address="external_ip_address" ][ masquerading-ip-address="masquerading_ip_address"
][ masquerading-port="masquerading_port" ][ external-data-services-ip-
```

```
address="external_data_services_ip_address" ][ enable-shadow-clones="{ true | false }"
][ enable-password-remote-login="{ true | false }" ][ logon-name="logon_name" ][
password="password" ][ disable-degraded-node-monitoring="{ true | false }" ][ enable-on-
disk-dedup="{ true | false }" ][ enable-common-criteria-mode="{ true | false }" ]
```

Required arguments

None

Optional arguments

`new-name`

Name of the cluster

`support-verbosity-level`

Verbosity level settings for populating support information

`external-ip-address`

External IP address to access the cluster. Set to '-' to clear the existing value

`masquerading-ip-address`

Masquerading IP address to access the cluster. Set to '-' to clear the existing value

`masquerading-port`

Masquerading port to access the cluster. Set to '-' to clear the existing value

`external-data-services-ip-address`

External Data Services IP address of the cluster. Set to '-' to clear the existing value

`enable-shadow-clones`

Enable/Disable shadow clones

`enable-password-remote-login`

Enable/Disable password remote login to cluster

`logon-name`

Logon name (domain\username) of a domain user/administrator account that has privileges to perform the operation

`password`

Password for the account specified by the logon account name

`disable-degraded-node-monitoring`

Whether to disable detection of partially degraded nodes.

`enable-on-disk-dedup`

Whether to enable on-disk deduplication for any new writes at cluster level. It might be disabled again when storage issue detected.

`enable-common-criteria-mode`

Enables Common Criteria mode on the cluster when set to 'true'. Disables when set to 'false'

Generates and downloads the csr from discovered node based on certification information from the cluster

```
ncli> cluster { generate-csr-for-discovered-node }cvm-ip="cvm_ip" file-
path="file_path"
```

Required arguments

`cvm-ip`

IPv6 address of the controller VM of discovered node

`file-path`

Path where csr from the discovered node needs to be downloaded

Get the security config for the Cluster

```
ncli> cluster { get-cvm-security-config }
```

Required arguments

None

Get the domain fault tolerance state of the cluster

```
ncli> cluster { get-domain-fault-tolerance-status  | get-dm-ft-stat } type="type"
```

Required arguments

`type`

Fault tolerance domain type { rack, rackable_unit, node, disk }

Get the hypervisor LLDP config of the Cluster

```
ncli> cluster { get-hypervisor-lldp-config }
```

Required arguments

None

Get the hypervisor security config for the Cluster

```
ncli> cluster { get-hypervisor-security-config }
```

Required arguments

None

Get Ipmi monitoring status

```
ncli> cluster { get-ipmi-monitoring-status }
```

Required arguments

None

Gets the Microsoft Azure Account details

```
ncli> cluster { get-la-jolla-azure-info }
```

Required arguments

None

Get the list of name servers

```
ncli> cluster { get-name-servers }
```

Required arguments

None

Get the list of NFS subnet whitelist

```
ncli> cluster { get-nfs-whitelist }
```

Required arguments

None

Get the list of NTP servers

```
ncli> cluster { get-ntp-servers }
```

Required arguments

None

Get params of a Cluster

```
ncli> cluster { get-params  | info}
```

Required arguments

None

Get the redundancy state of the cluster

```
ncli> cluster { get-redundancy-state }
```

Required arguments

None

Get remote support status on a Cluster

```
ncli> cluster { get-remote-support-status }
```

Required arguments

None

Check the removal status for Physical Hosts

```
ncli> cluster { get-remove-status  | get-rm-status}[ id="id" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Physical Host

Get configuration of SMTP Server used for transmitting alerts and report emails to Nutanix support

```
ncli> cluster { get-smtp-server }
```

Required arguments

None

Join the Nutanix storage cluster to the Windows AD domain specified in the cluster name. This operation is only valid for clusters having hosts running Hyper-V.

```
ncli> cluster { join-domain }domain="domain" logon-name="logon_name"
name-server-ip="name_server_ip"[ cluster-name="cluster_name" ][ external-ip-
address="external_ip_address" ][ password="password" ][ ou-path="ou_path" ][ cps-
prefix="cps_prefix" ]
```

Required arguments

`domain`

Full name of the domain

`logon-name`

Logon name (domain\username) of a domain user/administrator account that has privileges to perform the operation

`name-server-ip`

IP address of the name server that can resolve the domain name

Optional arguments

`cluster-name`

Name of the cluster

`external-ip-address`

External IP address to access the cluster. Set to '-' to clear the existing value

`password`

Password for the account specified by the logon account name

`ou-path`

Organizational Unit path of the domain

`cps-prefix`

CPS prefix path of the domain

Get the list of public keys configured in the cluster

```
ncli> cluster { list-public-keys  | ls-public-keys }[ name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the cluster public key

List All registered Metro Availability Witness

```
ncli> cluster { list-witness }
```

Required arguments

None

Remove all the public keys previously configured in the cluster

```
ncli> cluster { remove-all-public-keys  | rm-all-public-keys }
```

Required arguments

None

Remove addresses from the list of name servers

```
ncli> cluster { remove-from-name-servers } servers="servers"
```

Required arguments

`servers`

Comma-separated list of IP addresses to be removed from the name servers list

Remove addresses from the list of NFS subnet whitelist

```
ncli> cluster { remove-from-nfs-whitelist } ip-subnet-masks="ip_subnet_masks"
```

Required arguments

`ip-subnet-masks`

Comma-separated list with entries of the form 'IP/subnet mask' (a.b.c.d/l.m.n.o) to be removed from the NFS subnet whitelist

Remove addresses from the list of NTP servers

```
ncli> cluster { remove-from-ntp-servers } servers="servers"
```

Required arguments

`servers`

Comma-separated list of IP addresses/Host names to be removed from the NTP servers list

Delete public key with the specified name from the cluster

```
ncli> cluster { remove-public-key  | rm-public-key } name="name"
```

Required arguments

`name`

Name of the cluster public key

Begin the process of removing a Physical Host

```
ncli> cluster { remove-start  | rm-start | delete } id="id" [ skip-space-
check="{ true | false }" ][ force="force" ]
```

Required arguments

`id`

ID of the Physical Host

Optional arguments

`skip-space-check`

Skip checking storage space-related constraints when initiating removal of a host from the cluster

Default: false

`force`

Forcefully perform the requested operation skipping any constraint validation

Default: false

Delete a Metro Availability Witness

```
ncli> cluster { remove-witness } id="id" [ recover="recover" ]
```

Required arguments

`id`

Id of the Metro Availability Witness

Optional arguments

`recover`

Recover from a Metro Availability Witness an unsuccessful remove operation

Default: false

Send an email to test the SMTP Server configuration

```
ncli> cluster { send-test-email } recipient="recipient" subject="subject"
```

Required arguments

`recipient`

Recipient of the test email

`subject`

Subject of the test email

Sets the DVM configuration parameters for LaJolla cluster

```
ncli> cluster { set-dvm-configuration } ip-start="ip_start" ip-end="ip_end"
ip-mask="ip_mask" gateway="gateway" company-name="company_name" [ license-
key="license_key" ][ data-telemetry="data_telemetry" ]
```

Required arguments

`ip-start`

IP Address range start.

`ip-end`

IP Address range end.

`ip-mask`

Subnet mask.

`gateway`

Default Gateway IP.

`company-name`

Company name setting dvm configuraiton

Optional arguments

`license-key`

License key of Microsoft

`data-telemetry`

True/False indicating whether you want your data to be sent to Microsoft.

Set the external IP address (IPv4) of the Cluster

```
ncli> cluster { set-external-ip-address }[ external-ip-
address="external_ip_address" ][ logon-name="logon_name" ][ password="password" ]
```

Required arguments

None

Optional arguments

`external-ip-address`

External IP address to access the cluster. Set to '-' to clear the existing value

`logon-name`

Logon name (domain\username) of a domain user/administrator account that has privileges to perform the operation

`password`

Password for the account specified by the logon account name

Enable or disable Ipmi monitoring

```
ncli> cluster { set-ipmi-monitoring-status } enable="{ true | false }"
```

Required arguments

`enable`

Enable or disable Ipmi monitoring

Set the redundancy state of the cluster

```
ncli> cluster { set-redundancy-state } desired-redundancy-
factor="desired_redundancy_factor"
```

Required arguments

`desired-redundancy-factor`

Desired redundancy factor of the cluster

Set configuration of SMTP Server used for transmitting alert and report emails to Nutanix support

```
ncli> cluster { set-smtp-server } address="address" [ port="port" ][
username="username" ][ password="password" ][ security-mode="security_mode" ][ from-
email-address="from_email_address" ]
```

Required arguments

`address`

Fully Qualified Domain Name(FQDN) or IPv4 address of the SMTP Server

Optional arguments

`port`

Port number of the SMTP Server. By default, port 25 is used

`username`

Username to access the SMTP Server

`password`

Password to access the SMTP Server

`security-mode`

Security mode used by SMTP Server for data encryption and authentication. SMTP Server in Nutanix cluster can be configured with one of the following mode: 'none', 'ssl' or 'starttls'

Default: none

`from-email-address`

From email address to be used while sending emails (Set to '-' to clear the existing value)

Set the timezone of the Cluster

```
ncli> cluster { set-timezone }timezone="timezone"
```

Required arguments

`timezone`

Timezone of the cluster

Starts the CPS deployment in Lajolla cluster

```
ncli> cluster { start-la-jolla-deployment }ip-start="ip_start" ip-end="ip_end" ip-
mask="ip_mask" gateway="gateway" company-name="company_name" username="username"
password="password" storage_admin_password="storage_admin_password"
azure_onboarding_enabled="azure_onboarding_enabled"[ license-
key="license_key" ][ azure_subscription_name="azure_subscription_name" ][
azure_subscription_id="azure_subscription_id" ][
azure_site_recovery_region="azure_site_recovery_region" ][
azure_ops_insights_region="azure_ops_insights_region" ]
```

Required arguments

`ip-start`

IP Address range start.

`ip-end`

IP Address range end.

`ip-mask`

Subnet mask.

`gateway`

Default Gateway IP.

`company-name`

Company name setting dvm configuraiton

`username`

Active Directory User name

`password`

Active directory password

`storage_admin_password`

Admin password for the cluster

`azure_onboarding_enabled`

Set to true or false based on whether Azure info details are provided or not

Optional arguments

`license-key`

License key of Microsoft

`azure_subscription_name`

Subscription name of Microsoft Account

`azure_subscription_id`

Subscription Id of Microsoft Account

`azure_site_recovery_region`

Site recovery region of Microsoft

`azure_ops_insights_region`

Ops insight region of Microsoft

Start remote support on a Cluster

```
ncli> cluster { start-remote-support }[ duration="duration" ]
```

Required arguments

None

Optional arguments

`duration`

Time (in minutes) to keep remote support enabled

Get the status of all services in the cluster

```
ncli> cluster { status }
```

Required arguments

None

Stop remote support on a Cluster

```
ncli> cluster { stop-remote-support }[ duration="duration" ]
```

Required arguments

None

Optional arguments

`duration`

Time (in minutes) to keep remote support disabled

Update a Metro Availability Witness

```
ncli> cluster { update-witness } id="id" password="password"
```

Required arguments

`id`

Id of the Metro Availability Witness

`password`

password

Get version of software running on a Cluster

```
ncli> cluster { version }
```

Required arguments

None

`container`: Storage Container

Description A Storage Container is a container for virtual disks

Alias `ctrstorage-container`

Operations

- Add addresses to Storage Container's NFS subnet whitelist : `add-to-nfs-whitelist`
- Create a new Storage Container : `create` | `add`
- Edit a Storage Container : `edit` | `update`
- Get the down-migrate times (in minutes) for Storage Tiers in a Storage Container : `get-down-migrate-times` | `get-dm-times`
- List Storage Containers : `list` | `ls`
- Get stats data for Storage Containers : `list-stats` | `ls-stats`
- Delete a Storage Container : `remove` | `rm`
- Remove addresses from Storage Container's NFS subnet whitelist : `remove-from-nfs-whitelist`
- Set the down-migrate times (in minutes) for a Storage Tier in a Storage Container : `set-down-migrate-times` | `set-dm-times`

Add addresses to Storage Container's NFS subnet whitelist

```
ncli> container { add-to-nfs-whitelist } ip-subnet-masks="ip_subnet_masks" [
id="id" ][ name="name" ]
```

Required arguments

`ip-subnet-masks`

Comma-separated list with entries of the form 'IP/subnet mask'(a.b.c.d/l.m.n.o) to be included in the Storage Container's NFS subnet whitelist

Optional arguments

`id`

ID of the Storage Container

`name`

Name of the Storage Container

Create a new Storage Container

```
ncli> container { create  | add } name="name" [ res-capacity="res_capacity"
][ adv-capacity="adv_capacity" ][ sp-id="sp_id" ][ sp-name="sp_name" ][ rf="rf" ][
random-io-priority-order="random_io_priority_order" ][ sequential-io-priority-
order="sequential_io_priority_order" ][ enable-compression="{ true | false }" ][ fingerprint-
on-write="fingerprint_on_write" ][ on-disk-dedup="on_disk_dedup" ][ compression-
delay="compression_delay" ][ erasure-code="erasure_code" ][ prefer-higher-ec-fault-
domain="prefer_higher_ec_fault_domain" ][ erasure-code-delay="erasure_code_delay" ][ ip-
subnet-masks="ip_subnet_masks" ][ enable-software-encryption="{ true | false }" ]
```

Required arguments

`name`

Name of the Storage Container

Optional arguments

`res-capacity`

Explicit reserved Capacity (GiB) of the Storage Container

`adv-capacity`

Advertised Capacity (GiB) of the Storage Container

`sp-id`

ID of the Storage Pool for the Storage Container

`sp-name`

Name of the Storage Pool for the Storage Container

`rf`

Replication Factor for all data in the Storage Container

`random-io-priority-order`

Comma-separated random I/O priority order (high to low) of Storage Tiers in a Storage Container

`sequential-io-priority-order`

Comma-separated sequential I/O priority order (high to low) of Storage Tiers in a Storage Container

`enable-compression`

Enable or disable compression on a Storage Container

Default: false

`fingerprint-on-write`

Fingerprint on writes to the Storage Container {on, off, none}

`on-disk-dedup`

On-disk dedup of the Storage Container {none, off, post-process}

`compression-delay`

Time delay in minutes for compressing/uncompressing the data on Storage Container

`erasure-code`

Erasure code should be of the form: 'on', 'off' or <N>/<K> where N, K are valid positive integers

`prefer-higher-ec-fault-domain`

If set to true, EC strips will be created such that they are higher fault domain aware even if they are shorter. Existing longer EC strips that yield higher space savings but are lower fault domain aware will be made shorter so that they become higher fault domain aware.

`erasure-code-delay`

Erasure code delay (mins) of the Storage Container. 'clear' for clearing the Erasure code delay setting

`ip-subnet-masks`

Comma-separated list with entries of the form 'IP/subnet mask'(a.b.c.d/l.m.n.o) to be included in the Storage Container's NFS subnet whitelist

`enable-software-encryption`

Enable or disable software encryption on the Storage Container

Edit a Storage Container

```
ncli> container { edit  | update }[ id="id" ][ name="name" ][ new-name="new_name"
][ res-capacity="res_capacity" ][ adv-capacity="adv_capacity" ][ rf="rf" ][
random-io-priority-order="random_io_priority_order" ][ sequential-io-priority-
order="sequential_io_priority_order" ][ enable-compression="{ true | false }" ][ compression-
delay="compression_delay" ][ fingerprint-on-write="fingerprint_on_write" ][ on-disk-
dedup="on_disk_dedup" ][ erasure-code="erasure_code" ][ prefer-higher-ec-fault-
domain="prefer_higher_ec_fault_domain" ][ erasure-code-delay="erasure_code_delay" ][ ip-
subnet-masks="ip_subnet_masks" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Storage Container

`name`

Name of the Storage Container

`new-name`

Name of the Storage Container

`res-capacity`

Explicit reserved Capacity (GiB) of the Storage Container

`adv-capacity`

Advertised Capacity (GiB) of the Storage Container

`rf`

Replication Factor for all data in the Storage Container

`random-io-priority-order`

Comma-separated random I/O priority order (high to low) of Storage Tiers in a Storage Container

`sequential-io-priority-order`

Comma-separated sequential I/O priority order (high to low) of Storage Tiers in a Storage Container

`enable-compression`

Enable or disable compression on a Storage Container

`compression-delay`

Time delay in minutes for compressing/uncompressing the data on Storage Container

`fingerprint-on-write`

Fingerprint on writes to the Storage Container {on, off, none}

`on-disk-dedup`

On-disk dedup of the Storage Container {none, off, post-process}

`erasure-code`

Erasure code should be of the form: 'on', 'off' or <N>/<K> where N, K are valid positive integers

`prefer-higher-ec-fault-domain`

If set to true, EC strips will be created such that they are higher fault domain aware even if they are shorter. Existing longer EC strips that yield higher space savings but are lower fault domain aware will be made shorter so that they become higher fault domain aware.

`erasure-code-delay`

Erasure code delay (mins) of the Storage Container. 'clear' for clearing the Erasure code delay setting

`ip-subnet-masks`

Comma-separated list with entries of the form 'IP/subnet mask'(a.b.c.d/l.m.n.o) to be included in the Storage Container's NFS subnet whitelist

Get the down-migrate times (in minutes) for Storage Tiers in a Storage Container

```
ncli> container { get-down-migrate-times  | get-dm-times }[ id="id" ][ name="name" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Storage Container

`name`

Name of the Storage Container

List Storage Containers

```
ncli> container { list  | ls }[ id="id" ][ name="name" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Storage Container

`name`

Name of the Storage Container

Get stats data for Storage Containers

```
ncli> container { list-stats  | ls-stats }[ id="id" ][ name="name" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Storage Container

`name`

Name of the Storage Container

Delete a Storage Container

```
ncli> container { remove  | rm }[ id="id" ][ name="name" ][ ignore-small-files="ignore_small_files" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Storage Container

`name`

Name of the Storage Container

`ignore-small-files`

Ignore all the small files on the Storage Container {true, false} while marking for removal

Default: false

Remove addresses from Storage Container's NFS subnet whitelist

```
ncli> container { remove-from-nfs-whitelist }ip-subnet-masks="ip_subnet_masks"[
id="id" ][ name="name" ]
```

Required arguments

`ip-subnet-masks`

Comma-separated list with entries of the form 'IP/subnet mask'(a.b.c.d/l.m.n.o) to be included in the Storage Container's NFS subnet whitelist

Optional arguments

`id`

ID of the Storage Container

`name`

Name of the Storage Container

Set the down-migrate times (in minutes) for a Storage Tier in a Storage Container

```
ncli> container { set-down-migrate-times  | set-dm-times }tier-names="tier_names"
[ id="id" ][ name="name" ][ time-in-min="time_in_min" ]
```

Required arguments

`tier-names`

A comma-separated list of Storage Tiers

Optional arguments

`id`

ID of the Storage Container

`name`

Name of the Storage Container

`time-in-min`

Time in minutes after which to down-migrate data in a given Storage Tier in a Storage Container

Default: 30

`data-at-rest-encryption`: Data At Rest Encryption

Description Manage data-at-rest-encryption related operations

Alias

Operations

- List of results of the certificate tests that were performed against key management servers : `get-recent-certificate-test-results`
- Get current encryption status of the cluster : `get-status`
- Enable or disable disk passwords for encryption on the cluster : `password`
- Assigns new passwords to encryption capable disks when cluster is password protected : `rekey-disks`
- Test encryption configuration on given hosts and key management servers : `test-configuration`

List of results of the certificate tests that were performed against key management servers

```
ncli> data-at-rest-encryption { get-recent-certificate-test-results }[ host-
ids="host_ids" ][ key-management-server-names="key_management_server_names" ]
```

Required arguments

None

Optional arguments

`host-ids`

List of Host ids

`key-management-server-names`

List of Key Management Server names

Get current encryption status of the cluster

```
ncli> data-at-rest-encryption { get-status }
```

Required arguments

None

Enable or disable disk passwords for encryption on the cluster

```
ncli> data-at-rest-encryption { password } enable="{ true | false }"
```

Required arguments

`enable`

Enable or disable disk passwords for encryption

Assigns new passwords to encryption capable disks when cluster is password protected. If disk ids are not given, rekey will be performed on all disks of the cluster

```
ncli> data-at-rest-encryption { rekey-disks }[ disk-ids="disk_ids" ]
```

Required arguments

None

Optional arguments

`disk-ids`

IDs of the Physical Disks

Test encryption configuration on given hosts and key management servers. If no parameters are specified, test will be conducted on all nodes and key management servers configured in the cluster

```
ncli> data-at-rest-encryption { test-configuration }[ host-ids="host_ids" ][ key-management-server-names="key_management_server_names" ]
```

Required arguments

None

Optional arguments

`host-ids`

A comma-separated list of the ids of the Physical Hosts

`key-management-server-names`

Comma-separated list of key management server names

`data-at-rest-encryption-certificate`: Data At Rest Encryption Certificate

Description Manage data-at-rest-encryption related digital certificates

Alias

Operations

- Download one or more CSRs from the cluster as zip file : `download-csrs`
- Get certification information : `get-csr-information`
- Get the list of ca certificates stored in the cluster : `list-ca-certificates` | `ls-ca-certificates`
- Delete ca certificate with the specified name from the cluster : `remove-ca-certificate` | `rm-ca-certificate`
- Delete certificate installed on a node for a key management server : `remove-cvm-certificate` | `rm-cvm-certificate`
- Replace expired digital certificate : `replace-cvm-certificate`
- Update certification information : `update-csr-information`
- Upload ca certificate to the cluster : `upload-ca-certificate`
- Upload digital certificates as a single file or a zip file : `upload-cvm-certificates`

Download one or more CSRs from the cluster as zip file

```
ncli> data-at-rest-encryption-certificate { download-csrs } file-path="file_path" [
host-ids="host_ids" ]
```

Required arguments

`file-path`

Path where zip file needs to be downloaded

Optional arguments

`host-ids`

A comma-separated list of the ids of the Physical Hosts

Get certification information

```
ncli> data-at-rest-encryption-certificate { get-csr-information }
```

Required arguments

None

Get the list of ca certificates stored in the cluster

```
ncli> data-at-rest-encryption-certificate { list-ca-certificates  | ls-ca-
certificates }
```

Required arguments

None

Delete ca certificate with the specified name from the cluster

```
ncli> data-at-rest-encryption-certificate { remove-ca-certificate  | rm-ca-
certificate } ca-name="ca_name"
```

Required arguments

`ca-name`

Certificate Authority name

Delete certificate installed on a node for a key management server

```
ncli> data-at-rest-encryption-certificate { remove-cvm-certificate
| rm-cvm-certificate } host-id="host_id" key-management-server-
name="key_management_server_name"
```

Required arguments

`host-id`

Id of the node on which certificate is installed

`key-management-server-name`

Key management Server name for which certificate is installed

Replace expired digital certificate.

```
ncli> data-at-rest-encryption-certificate { replace-cvm-certificate }host-
id="host_id" key-management-server-name="key_management_server_name" file-
path="file_path"
```

Required arguments

`host-id`

ID of the Physical Host

`key-management-server-name`

Key management server name

`file-path`

Certificate path

Update certification information

```
ncli> data-at-rest-encryption-certificate { update-csr-information }[
country-code="country_code" ][ state="state" ][ city="city" ][ organization-
name="organization_name" ][ email-address="email_address" ][ organizational-
units="organizational_units" ][ domain-name="domain_name" ]
```

Required arguments

None

Optional arguments

`country-code`

Country Code

`state`

State

`city`

City

`organization-name`

Organization Name

`email-address`

Email Address

`organizational-units`

List of comma-separated Organizational Units. Set to '-' to clear existing value(s)

`domain-name`

Domain name that is to be included in Common Name. Set to '-' to clear the existing value

Upload ca certificate to the cluster

```
ncli> data-at-rest-encryption-certificate { upload-ca-certificate }ca-
name="ca_name" file-path="file_path"
```

Required arguments

`ca-name`

Certificate Authority Name

`file-path`

Certificate path

Upload digital certificates as a single file or a zip file.

```
ncli> data-at-rest-encryption-certificate { upload-cvm-certificates }key-
management-server-name="key_management_server_name" file-path="file_path"
```

Required arguments

`key-management-server-name`

Key management server name

`file-path`

Path of a zipped or regular file

`datastore`: Datastore

Description An NFS Datastore

Alias

Operations

- Create a new NFS datastore on the Physical Hosts using the Storage Container (ESX only) : `create` | `add`
- Delete the NFS datastore on the Physical Hosts : `delete` | `remove` | `rm`
- List NFS Datastores : `list` | `ls`

Create a new NFS datastore on the Physical Hosts using the Storage Container (ESX only). Storage Container name will be used as datastore name, if datastore name is not specified

```
ncli> datastore { create  | add }ctr-name="ctr_name"[ name="name" ][ host-
ids="host_ids" ][ read-only="read_only" ]
```

Required arguments

`ctr-name`

Name of the Storage Container

Optional arguments

`name`

Name of the Datastore (default = Storage Container name)

`host-ids`

A comma-separated list of the ids of the Physical Hosts (default = Includes all hosts)

`read-only`

Whether a Physical Host must have only read-only access to the Datastore

Default: false

Delete the NFS datastore on the Physical Hosts

```
ncli> datastore { delete  | remove | rm }name="name"[ host-ids="host_ids" ]
```

Required arguments

`name`

Name of the Datastore

Optional arguments

`host-ids`

A comma-separated list of the ids of the Physical Hosts (default = Includes all hosts)

List NFS Datastores

```
ncli> datastore { list  | ls }
```

Required arguments

None

`disk`: Physical Disk

Description A Physical Disk

Alias

Operations

- Check the removal status for Physical Disks : `get-remove-status` | `get-rm-status`
- List Physical Disks : `list` | `ls`
- List Physical Disks that are not assigned to any Storage Pool : `list-free` | `ls-free`
- Get stats data for Physical Disks : `list-stats` | `ls-stats`
- Begin the process of removing a Physical Disk : `remove-start` | `rm-start` | `delete`

Check the removal status for Physical Disks

```
ncli> disk { get-remove-status  | get-rm-status }[ id="id" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Physical Disk

List Physical Disks

```
ncli> disk { list  | ls }[ id="id" ][ sp-id="sp_id" ][ sp-name="sp_name" ][ tier-
name="tier_name" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Physical Disk

`sp-id`

ID of the Storage Pool for the Physical Disk

`sp-name`

Name of the Storage Pool for the Physical Disk

`tier-name`

Corresponding Storage Tier

List Physical Disks that are not assigned to any Storage Pool

```
ncli> disk { list-free  | ls-free }
```

Required arguments

None

Get stats data for Physical Disks

```
ncli> disk { list-stats  | ls-stats }[ id="id" ][ sp-id="sp_id" ][ tier-
name="tier_name" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Physical Disk

`sp-id`

ID of the Storage Pool for the Physical Disk

`tier-name`

Corresponding Storage Tier

Begin the process of removing a Physical Disk

```
ncli> disk { remove-start  | rm-start | delete } id="id" [ force="force" ]
```

Required arguments

`id`

ID of the Physical Disk

Optional arguments

`force`

Forcefully perform the requested operation skipping any constraint validation

Default: false

`events`: Event

Description An Event

Alias `event`

Operations

- Acknowledge Events : `acknowledge` | `ack`
- List history of Events : `history`
- List of unacknowledged Events : `list` | `ls`

Acknowledge Events

```
ncli> events { acknowledge  | ack } ids="ids"
```

Required arguments

`ids`

A comma-separated list of ids of the Events

List history of Events

```
ncli> events { history } duration="duration" [ acknowledged="acknowledged" ][ max-events="max_events" ]
```

Required arguments

`duration`

Duration (in days) for getting the history of Events

Optional arguments

`acknowledged`

Acknowledged Events?

`max-events`

Maximum number of Events to fetch

Default: 100

List of unacknowledged Events

```
ncli> events { list  | ls }[ max-events="max_events" ]
```

Required arguments

None

Optional arguments

`max-events`

Maximum number of Events to fetch

Default: 100

`failover-cluster`: Failover Cluster

Description Hyper-V failover cluster

Alias

Operations

- Create a failover cluster : `create`
- Join nodes to a failover cluster : `join-nodes`
- List all failover clusters : `list`
- Remove node from a failover cluster : `remove-node`

Create a failover cluster

```
ncli> failover-cluster { create } name="name" ip-address="ip_address" logon-name="logon_name" [ password="password" ][ host-ids="host_ids" ]
```

Required arguments

`name`

Simple name of the failover cluster. Must be at most 15 characters long.

`ip-address`

IP address of the failover cluster

`logon-name`

Logon name (domain\username) of a domain user/administrator account that has privileges to perform the operation

Optional arguments

`password`

Password for the account specified by the logon account name

`host-ids`

A comma-separated list of the ids of the Physical Hosts

Join nodes to a failover cluster

```
ncli> failover-cluster { join-nodes } name="name" logon-name="logon_name" [ password="password" ][ host-ids="host_ids" ]
```

Required arguments

`name`

Simple name of the failover cluster. Must be at most 15 characters long.

`logon-name`

Logon name (domain\username) of a domain user/administrator account that has privileges to perform the operation

Optional arguments

`password`

Password for the account specified by the logon account name

`host-ids`

A comma-separated list of the ids of the Physical Hosts

List all failover clusters

```
ncli> failover-cluster { list }
```

Required arguments

None

Remove node from a failover cluster

```
ncli> failover-cluster { remove-node } name="name" logon-name="logon_name" host-id="host_id" [ password="password" ]
```

Required arguments

`name`

Simple name of the failover cluster. Must be at most 15 characters long.

`logon-name`

Logon name (domain\username) of a domain user/administrator account that has privileges to perform the operation

`host-id`

ID of the Physical Host

Optional arguments

`password`

Password for the account specified by the logon account name

`file-server`: File Server

Description Minerva file server

Alias `fs`

Operations

- Activate a file server : `activate` | `activate`
- Add a admin user : `add-admin-user`
- Add DNS entries : `add-dns`
- Add a icap server : `add-icap-server`
- Add a Quota Policy : `add-quota-policy`
- Add a Share : `add-share`
- Add a Snapshot Policy : `add-snapshot-policy`
- Add a user : `add-user` | `add-user`
- Add a User Mapping : `add-user-mapping` | `add-user-mapping`
- Add a virus scan policy : `add-virus-scan-policy`
- Clone a file server : `clone` | `clone`
- Join and unjoin the File Server to the Windows AD domain or bind and unbind from LDAP : `configure-name-services` | `configure-name-services`
- Add a File Server : `create` | `add`
- Delete a File Server : `delete` | `remove` | `rm`
- Delete a admin user : `delete-admin-user`
- Delete a icap server : `delete-icap-server`
- Delete infected files : `delete-infected-files`
- Delete a Quota Policy : `delete-quota-policy`
- Delete a Share : `delete-share`
- Delete a Snapshot Policy : `delete-snapshot-policy`
- Delete a user : `delete-user`
- Delete a virus scan policy : `delete-virus-scan-policy`
- Update a File Server : `edit` | `update`
- Show an individual FileServer's details : `get`
- Get a admin user : `get-admin-user`
- Show built in groupsper File Server : `get-builtin-groups`
- Get icap server : `get-icap-server`
- Get infected file : `get-infected-file`
- Get principal type from principal name : `get-principal-type` | `get-principal-type`
- Get a Quota Policy : `get-quota-policy`
- Show an individual share's details : `get-share` | `show-share`
- Get a file server share snapshot : `get-snapshot`
- Get a User Mapping : `get-user-mapping` | `get-user-mapping`
- Get a virus scan policy : `get-virus-scan-policy`
- Join the File Server to the Windows AD domain specified : `join-domain` | `join-domain`
- Leave File Server from domain : `leave-domain` | `leave-domain`
- List all File Servers with filtering : `list` | `ls`
- List all admin users with filtering : `list-admin-users`
- List all shares with filtering : `list-all-fs-shares`
- List all DNS entries : `list-dns`
- List all icap servers with filtering : `list-icap-servers`
- List all infected files : `list-infected-files`
- List all quota policies with filtering : `list-quota-policies`
- List all shares with filtering : `list-shares` | `list-shares`
- List all Snapshot policies with filtering : `list-snapshot-policies`
- List all Snapshots with filtering : `list-snapshots`
- List users : `list-user`
- Load balance a File Server : `load-balance`
- Add a protection domain for File Server : `protect`
- Quarantine infected files : `quarantine-infected-files`
- Remove DNS entries : `remove-dns`
- Rescan infected files : `rescan-infected-files`

Activate a file server.

```
ncli> file-server { activate  | activate}uuid="uuid" dns-domain-
name="dns_domain_name" dns-server-ip-address-list="dns_server_ip_address_list"
ntp-servers="ntp_servers" internal-virtual-network="internal_virtual_network"
external-virtual-network="external_virtual_network"[ internal-virtual-
network-gateway="internal_virtual_network_gateway" ][ internal-virtual-
network-mask="internal_virtual_network_mask" ][ internal-virtual-
network-ips="internal_virtual_network_ips" ][ external-virtual-network-
gateway="external_virtual_network_gateway" ][ external-virtual-network-
mask="external_virtual_network_mask" ][ external-virtual-network-
ips="external_virtual_network_ips" ][ windows-ad-domain-name="windows_ad_domain_name"
][ windows-ad-username="windows_ad_username" ][ windows-ad-
password="windows_ad_password" ][ overwrite="overwrite" ][ add-user-as-afs-
admin="add_user_as_afs_admin" ][ organizational-unit="organizational_unit" ][ rfc-
enabled="rfc_enabled" ][ preferred-domain-controller="preferred_domain_controller" ][ ad-
protocol-type="ad_protocol_type" ][ ldap-protocol-type="ldap_protocol_type" ][ local-
protocol-type="local_protocol_type" ][ nfsv4-domain="nfsv4_domain" ][ ldap-server-
uri="ldap_server_uri" ][ ldap-base-dn="ldap_base_dn" ][ ldap-username="ldap_username"
][ ldap-password="ldap_password" ][ ldap-ssh-key="ldap_ssh_key" ][ dns-
operation-type="dns_operation_type" ][ dns-server="dns_server" ][ dns-server-
username="dns_server_username" ][ dns-server-password="dns_server_password" ][
nfsversion="nfsversion" ]
```

Required arguments

`uuid`

Uuid of the file server

`dns-domain-name`

Fully qualified domain name (fileserver namespace). This, along with the fileserver name, constitutes the namespace of the fileserver. Example: fileserver_name.corp.companyname.com. This is also used to create fileserver DNS entries on the nameservers so that clients can access the fileserver using its name.

`dns-server-ip-address-list`

List of comma-separated dns server ip addresses for file server configuration.

`ntp-servers`

List of comma-separated ntp servers for file server configuration.

`internal-virtual-network`

The identifier of the internal virtual network. Please use network UUID on AHV and ESX.

`external-virtual-network`

The identifier of the external virtual network. Please use network UUID on AHV and ESX.

Optional arguments

`internal-virtual-network-gateway`

The gateway ip address of the internal virtual network associated with file server VMs.

`internal-virtual-network-mask`

The network mask of the internal virtual network associated with file server VMs.

`internal-virtual-network-ips`

IPs used by new file server VMs in internal virtual network. Comma-separated IP ranges or IPs, e.g. "10.4.1.3-10.4.1.5,10.4.1.11".

`external-virtual-network-gateway`

The gateway ip address of the external virtual network associated with file server VMs.

`external-virtual-network-mask`

The network mask of the internal virtual network associated with file server VMs.

`external-virtual-network-ips`

IPs used by new file server VMs in external virtual network.

`windows-ad-domain-name`

The windows AD domain the file server is associated with.

`windows-ad-username`

The name of a user account with administrative privileges in the AD domain the file server is associated with.

`windows-ad-password`

The password for the above Windows AD account

`overwrite`

If a machine account with the same name as file-server name is present on the AD, then overwrite it during join-domain operation.

Default: false

`add-user-as-afs-admin`

AD user or group name as 'name' or 'NETBIOS\name' format.

`organizational-unit`

An Organizational unit container is where the AFS machine account will be created as part of domain join operation. Use a / to specify hierarchical Organization Units. Default container OU is "Computers". Examples: Engineering, Department/Engineering.

`rfc-enabled`

RFC 2307 ENABLED (true, false)

Default: false

`preferred-domain-controller`

Use a specific domain controller for join-domain operation in a multi DC Active Directory setups. By default, AFS discovers a site local domain controller for join-domain operation.

`ad-protocol-type`

Protocol Type(NONE,SMB,NFS,"NFS,SMB")

`ldap-protocol-type`

Protocol Type(NONE,NFS)

`local-protocol-type`

Protocol Type(NONE,NFS)

`nfsv4-domain`

NFSV4 Domain

`ldap-server-uri`

LDAP URI.

`ldap-base-dn`

LDAP Base DN.

`ldap-username`

LDAP Username.

`ldap-password`

LDAP Password.

`ldap-ssh-key`

LDAP SSH Certificate Key.

`dns-operation-type`

DNS Nameserver operation type: AFS specific DNS entries are needed on the nameservers so that clients can connect to the fileserver using its FQDN (fileserver_name.domain_name). AFS can add these entries automatically for MS-DNS servers or they can be added manually. Select one of "MS_DNS" or "MANUAL".

`dns-server`

Preferred Nameserver FQDN (if not given, AFS finds one using an NS query)

`dns-server-username`

Username to use for adding fileserver DNS entries on the nameservers.

`dns-server-password`

Password to use for adding fileserver DNS entries on the nameservers.

`nfsversion`

Nfs version (NFSV3, NFSV4, NFSV3V4)

Add a admin user

```
ncli> file-server { add-admin-user }uuid="uuid" user="user" role="role"
```

Required arguments

`uuid`

Uuid of the file server that admin user is associated with

`user`

AD user or group name as 'name' or 'NETBIOS\name' format.

`role`

Role of the given user or group (one of AdMIN or BACKUP_OPERATOR).

Add DNS entries

```
ncli> file-server { add-dns }uuid="uuid" dns-operation-type="dns_operation_type"
dns-server-username="dns_server_username" dns-server-password="dns_server_password" [
dns-server="dns_server" ]
```

Required arguments

`uuid`

UUID of the FileServer

`dns-operation-type`

DNS Nameserver operation type: AFS specific DNS entries are needed on the nameservers so that clients can connect to the fileserver using its FQDN (fileserver_name.domain_name). AFS can add these entries automatically for MS-DNS servers or they can be added manually. Select one of "MS_DNS" or "MANUAL".

`dns-server-username`

Username to use for adding fileserver DNS entries on the nameservers.

`dns-server-password`

Password to use for adding fileserver DNS entries on the nameservers.

Optional arguments

`dns-server`

Preferred Nameserver FQDN (if not given, AFS finds one using an NS query)

Add a icap server

```
ncli> file-server { add-icap-server } uuid="uuid" ip-address-or-host-
name="ip_address_or_host_name" port="port" [ description="description" ]
```

Required arguments

`uuid`

Uuid of the file server that icap server is associated with

`ip-address-or-host-name`

Antivirus server ip address.

`port`

Antivirus server port number.

Optional arguments

`description`

Antivirus server description.

Add a Quota Policy

```
ncli> file-server { add-quota-policy } uuid="uuid" share-uuid="share_uuid"
principal-value="principal_value" quota-size-gib="quota_size_gib" quota-
enforcement-type="quota_enforcement_type" [ principal-type="principal_type" ][ send-
quota-notifications-to-user="send_quota_notifications_to_user" ][ notification-
recipients="notification_recipients" ]
```

Required arguments

`uuid`

Uuid of the file server that quota policy is associated with

`share-uuid`

Uuid of the share that quota policy is associated with

`principal-value`

Quota policy principal value (user or group name)

`quota-size-gib`

Quota size in Gibs

`quota-enforcement-type`

Quota enforcement type (Hard or Soft)

Optional arguments

`principal-type`

Quota policy principal type (enter 'user' or 'group')

`send-quota-notifications-to-user`

Send quota notifications to user

`notification-recipients`

Additional notification recipients (comma-separated)

Add a Share

```
ncli> file-server { add-share } uuid="uuid" name="name" [ description="description"
][ enable-windows-previous-version="{ true | false }" ][ share-type="share_type" ][
share-size-gib="share_size_gib" ][ default-quota-limit-gib="default_quota_limit_gib"
][ quota-enforcement-type="quota_enforcement_type" ][ send-quota-notifications-to-
user="send_quota_notifications_to_user" ][ enable-access-based-enumeration="{ true | false }"
][ protocol-type="protocol_type" ][ secondary-protocol-type="secondary_protocol_type" ][
enable-concurrent-reads="{ true | false }" ][ enable-case-sensitive-namespace="{ true |
```

false }"][enable-symlink-creation="*{ true* | *false }*"][enable-simultaneous-access="*{ true* | *false }*"][share-path="*share_path*"][parent-share-uuid="*parent_share_uuid*"][share-auth-type="*share_auth_type*"][default-share-access-type="*default_share_access_type*"][client-with-read-write-access="*client_with_read_write_access*"][client-with-read-only-access="*client_with_read_only_access*"][client-with-no-access="*client_with_no_access*"][anonymous-uid="*anonymous_uid*"][anonymous-gid="*anonymous_gid*"][squash-type="*squash_type*"]

Required arguments

`uuid`

uuid of the File Server

`name`

Name of the Share

Optional arguments

`description`

Description of the Share

`enable-windows-previous-version`

Enable self service restore flag

`share-type`

Type of Share. Homes or General (General Purpose)

`share-size-gib`

Share size in Gibs

`default-quota-limit-gib`

Default quota limit in Gibs (Quota applies to all users of the share)

`quota-enforcement-type`

Quota enforcement type (Hard or Soft)

`send-quota-notifications-to-user`

Send quota notifications to user

`enable-access-based-enumeration`

Enable access based enumeration flag

`protocol-type`

Primary protocol type (SMB or NFS)

`secondary-protocol-type`

Secondary protocol type (SMB or NFS)

`enable-concurrent-reads`

Concurrent Reads Enabled

`enable-case-sensitive-namespace`

Case sensitive Creation Enabled

`enable-symlink-creation`

Symlink Creation Enabled

`enable-simultaneous-access`

Simultaneous Access Enabled

`share-path`

Share path for nested share

`parent-share-uuid`

UUID of parent share

`share-auth-type`

Type of Share Authentication. (System-none, Kerberos)

`default-share-access-type`

Type of Default Share Access. (Read-write, Read-only, No-Access)

`client-with-read-write-access`

Comma-separated list of clients

`client-with-read-only-access`

Comma-separated list of clients

`client-with-no-access`

Comma-separated list of clients

`anonymous-uid`

Anonymous UID.

`anonymous-gid`

Anonymous GID.

`squash-type`

Squash Type.(root-squash,all-squash,none)

Add a Snapshot Policy

```
ncli> file-server { add-snapshot-policy }file-server-uuid="file_server_uuid"
snapshot-policy-type="snapshot_policy_type" frequency="frequency" local-
retention="local_retention"[ days-of-week="days_of_week" ][ days-of-
month="days_of_month" ]
```

Required arguments

`file-server-uuid`

Uuid of the file server that snapshot policy is associated with

`snapshot-policy-type`

Snapshot policy type - hourly, daily, weekly, monthly

`frequency`

Snapshot policy frequency. Repeat snapshot every nth hour/day. For weekly and monthly, frequency is 1.

`local-retention`

Maximum number of snapshots to retain locally

Optional arguments

`days-of-week`

Comma-separated day of week values for the schedule: 1-7 (starts with Sunday(1))

`days-of-month`

Comma-separated day of month values for the schedule: 1-31

Add a user

```
ncli> file-server { add-user  | add-user }uuid="uuid" user="user"[
password="password" ]
```

Required arguments

`uuid`

Uuid of the file server

`user`

File server user name.

Optional arguments

`password`

The password for the above file server user.

Add a User Mapping

```
ncli> file-server { add-user-mapping  | add-user-mapping } uuid="uuid" [
template-mapping-rule="template_mapping_rule" ][ smb-user-and-group-with-no-
nfs-mapping-action="smb_user_and_group_with_no_nfs_mapping_action" ][ nfs-user-
no-smb-mapping-action="nfs_user_no_smb_mapping_action" ][ smb-user-map-to-nfs-
uid="smb_user_map_to_nfs_uid" ][ smb-group-map-to-nfs-gid="smb_group_map_to_nfs_gid"
][ nfs-user-map-to-smb-name="nfs_user_map_to_smb_name" ][ nfs-group-map-to-smb-
name="nfs_group_map_to_smb_name" ][ one-to-one-mapping="one_to_one_mapping" ][ wild-
card-mapping="wild_card_mapping" ][ deny-access-smb-users="deny_access_smb_users"
][ deny-access-smb-groups="deny_access_smb_groups" ][ deny-access-nfs-
users="deny_access_nfs_users" ][ deny-access-nfs-groups="deny_access_nfs_groups" ][
file-path="file_path" ]
```

Required arguments

`uuid`

Uuid of the file server

Optional arguments

`template-mapping-rule`

Pre-defined templates to map SMB and NFS identities. Allowed values(NO_TEMPLATE_MAPPING, SMB_NAME_NFS_NAME)

`smb-user-and-group-with-no-nfs-mapping-action`

SMB user and group with no NFS mapping action. Allowed values (DENY_ACCESS, MAP_IDENTITIES)

`nfs-user-no-smb-mapping-action`

NFS user and group with no SMB mapping action. Allowed values (DENY_ACCESS, MAP_IDENTITIES).

`smb-user-map-to-nfs-uid`

SMB user mapping with NFS uid

`smb-group-map-to-nfs-gid`

SMB group mapping with NFS uid

`nfs-user-map-to-smb-name`

NFS user mapping with SMB name, eg. domain\username.

`nfs-group-map-to-smb-name`

NFS group mapping with SMB name, eg. domain\username.

`one-to-one-mapping`

One to one explicit mapping associated with file server

`wild-card-mapping`

Wild card explicit mapping associated with file server

`deny-access-smb-users`

Deny access SMB users, eg. domain\username.

`deny-access-smb-groups`

Deny access SMB groups, eg. domain\username.

`deny-access-nfs-users`

Deny access NFS users, eg. domain\username.

`deny-access-nfs-groups`

Deny access NFS groups, eg. domain\username.

`file-path`

File path

Add a virus scan policy

```
ncli> file-server { add-virus-scan-policy }uuid="uuid"[ scan-time-interval-
secs="scan_time_interval_secs" ][ share-uuid="share_uuid" ][ scan-on-write="scan_on_write"
][ scan-on-read="scan_on_read" ][ file-size-exclusion-bytes="file_size_exclusion_bytes" ][
block-access-file="block_access_file" ][ enable-anti-virus="{ true | false }" ][ file-type-
exclusions="file_type_exclusions" ]
```

Required arguments

`uuid`

Uuid of the file server

Optional arguments

`scan-time-interval-secs`

Scan time interval in uses

`share-uuid`

Uuid of the Share

`scan-on-write`

Scan on write policy

`scan-on-read`

Scan on read policy

`file-size-exclusion-bytes`

File size exclusion

`block-access-file`

Block access file

`enable-anti-virus`

Enable antivirus

`file-type-exclusions`

File type exclusion list

Clone a file server.

```
ncli> file-server { clone  | clone }uuid="uuid" name="name" dns-domain-
name="dns_domain_name" dns-server-ip-address-list="dns_server_ip_address_list"
ntp-servers="ntp_servers" internal-virtual-network="internal_virtual_network"
external-virtual-network="external_virtual_network"[ snapshot-uuid="snapshot_uuid"
][ internal-virtual-network-gateway="internal_virtual_network_gateway"
][ internal-virtual-network-mask="internal_virtual_network_mask" ][
internal-virtual-network-ips="internal_virtual_network_ips" ][ external-
virtual-network-gateway="external_virtual_network_gateway" ][ external-
virtual-network-mask="external_virtual_network_mask" ][ external-
virtual-network-ips="external_virtual_network_ips" ][ windows-ad-domain-
name="windows_ad_domain_name" ][ windows-ad-username="windows_ad_username" ][
windows-ad-password="windows_ad_password" ][ overwrite="overwrite" ][ add-user-
```

```
as-afs-admin="add_user_as_afs_admin" ][ organizational-unit="organizational_unit" ][
preferred-domain-controller="preferred_domain_controller" ][ rfc-enabled="rfc_enabled" ][
ad-protocol-type="ad_protocol_type" ][ ldap-protocol-type="ldap_protocol_type" ][ local-
protocol-type="local_protocol_type" ][ nfsv4-domain="nfsv4_domain" ][ ldap-server-
uri="ldap_server_uri" ][ ldap-base-dn="ldap_base_dn" ][ ldap-username="ldap_username"
][ ldap-password="ldap_password" ][ ldap-ssh-key="ldap_ssh_key" ][ dns-
operation-type="dns_operation_type" ][ dns-server="dns_server" ][ dns-server-
username="dns_server_username" ][ dns-server-password="dns_server_password" ][
nfsversion="nfsversion" ]
```

Required arguments

`uuid`

Uuid of the file server

`name`

Name of the file server clone

`dns-domain-name`

Fully qualified domain name (fileserver namespace). This, along with the fileserver name, constitutes the namespace of the fileserver. Example: fileserver_name.corp.companyname.com. This is also used to create fileserver DNS entries on the nameservers so that clients can access the fileserver using its name.

`dns-server-ip-address-list`

List of comma-separated dns server ip addresses for file server configuration.

`ntp-servers`

List of comma-separated ntp servers for file server configuration.

`internal-virtual-network`

The identifier of the internal virtual network. Please use network UUID on AHV and ESX.

`external-virtual-network`

The identifier of the external virtual network. Please use network UUID on AHV and ESX.

Optional arguments

`snapshot-uuid`

File Server snapshot uuid

`internal-virtual-network-gateway`

The gateway ip address of the internal virtual network associated with file server VMs.

`internal-virtual-network-mask`

The network mask of the internal virtual network associated with file server VMs.

`internal-virtual-network-ips`

IPs used by new file server VMs in internal virtual network. Comma-separated IP ranges or IPs, e.g. "10.4.1.3-10.4.1.5,10.4.1.11".

`external-virtual-network-gateway`

The gateway ip address of the external virtual network associated with file server VMs.

`external-virtual-network-mask`

The network mask of the internal virtual network associated with file server VMs.

`external-virtual-network-ips`

IPs used by new file server VMs in external virtual network.

`windows-ad-domain-name`

The windows AD domain the file server is associated with.

`windows-ad-username`

The name of a user account with administrative privileges in the AD domain the file server is associated with.

`windows-ad-password`

The password for the above Windows AD account

`overwrite`

If a machine account with the same name as file-server name is present on the AD, then overwrite it during join-domain operation.

Default: false

`add-user-as-afs-admin`

AD user or group name as 'name' or 'NETBIOS\name' format.

`organizational-unit`

An Organizational unit container is where the AFS machine account will be created as part of domain join operation. Use a / to specify hierarchical Organization Units. Default container OU is "Computers". Examples: Engineering, Department/Engineering.

`preferred-domain-controller`

Use a specific domain controller for join-domain operation in a multi DC Active Directory setups. By default, AFS discovers a site local domain controller for join-domain operation.

`rfc-enabled`

RFC 2307 ENABLED (true, false)

Default: false

`ad-protocol-type`

Protocol Type(NONE,SMB,NFS,"NFS,SMB")

`ldap-protocol-type`

Protocol Type(NONE,NFS)

`local-protocol-type`

Protocol Type(NONE,NFS)

`nfsv4-domain`

NFSV4 Domain

`ldap-server-uri`

LDAP URI.

`ldap-base-dn`

LDAP Base DN.

`ldap-username`

LDAP Username.

`ldap-password`

LDAP Password.

`ldap-ssh-key`

LDAP SSH Certificate Key.

`dns-operation-type`

DNS Nameserver operation type: AFS specific DNS entries are needed on the nameservers so that clients can connect to the fileserver using its FQDN (fileserver_name.domain_name). AFS can add these entries automatically for MS-DNS servers or they can be added manually. Select one of "MS_DNS" or "MANUAL".

`dns-server`

Preferred Nameserver FQDN (if not given, AFS finds one using an NS query)

`dns-server-username`

Username to use for adding fileserver DNS entries on the nameservers.

`dns-server-password`

Password to use for adding fileserver DNS entries on the nameservers.

`nfsversion`

Nfs version (NFSV3, NFSV4, NFSV3V4)

Join and unjoin the File Server to the Windows AD domain or bind and unbind from LDAP.

```
ncli> file-server { configure-name-services  | configure-name-services
} uuid="uuid" [ windows-ad-username="windows_ad_username" ][ organizational-
unit="organizational_unit" ][ windows-ad-password="windows_ad_password" ][
overwrite="overwrite" ][ add-user-as-afs-admin="add_user_as_afs_admin" ][ rfc-
enabled="rfc_enabled" ][ use-ad-credential-for-dns="use_ad_credential_for_dns"
][ preferred-domain-controller="preferred_domain_controller" ][ ad-protocol-
type="ad_protocol_type" ][ ldap-protocol-type="ldap_protocol_type" ][ local-
protocol-type="local_protocol_type" ][ nfsv4-domain="nfsv4_domain" ][ ldap-server-
uri="ldap_server_uri" ][ ldap-base-dn="ldap_base_dn" ][ ldap-username="ldap_username" ][
ldap-password="ldap_password" ][ ldap-ssh-key="ldap_ssh_key" ][ nfsversion="nfsversion"
]
```

Required arguments

`uuid`

Uuid of the file server

Optional arguments

`windows-ad-username`

The name of a user account with administrative privileges in the AD domain the file server is associated with.

`organizational-unit`

An Organizational unit container is where the AFS machine account will be created as part of domain join operation. Use a / to specify hierarchical Organization Units. Default container OU is "Computers". Examples: Engineering, Department/Engineering.

`windows-ad-password`

The password for the above Windows AD account

`overwrite`

If a machine account with the same name as file-server name is present on the AD, then overwrite it during join-domain operation.

Default: false

`add-user-as-afs-admin`

AD user or group name as 'name' or 'NETBIOS\name' format.

`rfc-enabled`

RFC 2307 ENABLED (true, false)

Default: false

`use-ad-credential-for-dns`

Use the same AD credential for dns

Default: false

`preferred-domain-controller`

Use a specific domain controller for join-domain operation in a multi DC Active Directory setups. By default, AFS discovers a site local domain controller for join-domain operation.

`ad-protocol-type`

Protocol Type(NONE,SMB,NFS,"NFS,SMB")

`ldap-protocol-type`

Protocol Type(NONE,NFS)

`local-protocol-type`

Protocol Type(NONE,NFS)

`nfsv4-domain`

NFSV4 Domain

`ldap-server-uri`

LDAP URI.

`ldap-base-dn`

LDAP Base DN.

`ldap-username`

LDAP Username.

`ldap-password`

LDAP Password.

`ldap-ssh-key`

LDAP SSH Certificate Key.

`nfsversion`

Nfs version (NFSV3, NFSV4, NFSV3V4)

Add a File Server

```
ncli> file-server { create  | add } name="name" dns-domain-
name="dns_domain_name" dns-server-ip-address-list="dns_server_ip_address_list"
ntp-servers="ntp_servers" size-gib="size_gib" internal-virtual-
network="internal_virtual_network" external-virtual-network="external_virtual_network"
[ internal-virtual-network-gateway="internal_virtual_network_gateway" ][
internal-virtual-network-mask="internal_virtual_network_mask" ][ internal-
virtual-network-ips="internal_virtual_network_ips" ][ external-virtual-
network-gateway="external_virtual_network_gateway" ][ external-virtual-
network-mask="external_virtual_network_mask" ][ external-virtual-network-
ips="external_virtual_network_ips" ][ nvm-count="nvm_count" ][ memory="memory" ][
cpu-count="cpu_count" ][ ad-protocol-type="ad_protocol_type" ][ ldap-protocol-
type="ldap_protocol_type" ][ local-protocol-type="local_protocol_type" ][ windows-ad-
domain-name="windows_ad_domain_name" ][ windows-ad-username="windows_ad_username"
][ windows-ad-password="windows_ad_password" ][ overwrite="overwrite" ][
organizational-unit="organizational_unit" ][ rfc-enabled="rfc_enabled" ][ pd-
name="pd_name" ][ add-user-as-afs-admin="add_user_as_afs_admin" ][ preferred-domain-
controller="preferred_domain_controller" ][ nfsv4-domain="nfsv4_domain" ][ ldap-server-
uri="ldap_server_uri" ][ ldap-base-dn="ldap_base_dn" ][ ldap-username="ldap_username"
][ ldap-password="ldap_password" ][ ldap-ssh-key="ldap_ssh_key" ][ dns-
operation-type="dns_operation_type" ][ dns-server="dns_server" ][ dns-server-
username="dns_server_username" ][ dns-server-password="dns_server_password" ][
nfsversion="nfsversion" ]
```

Required arguments

`name`

Name of the file server

`dns-domain-name`

Fully qualified domain name (fileserver namespace). This, along with the fileserver name, constitutes the namespace of the fileserver. Example: fileserver_name.corp.companyname.com. This is also used to create fileserver DNS entries on the nameservers so that clients can access the fileserver using its name.

`dns-server-ip-address-list`

List of comma-separated dns server ip addresses for file server configuration.

`ntp-servers`

List of comma-separated ntp servers for file server configuration.

`size-gib`

File server size in gibs

`internal-virtual-network`

The identifier of the internal virtual network. Please use network UUID on AHV and ESX.

`external-virtual-network`

The identifier of the external virtual network. Please use network UUID on AHV and ESX.

Optional arguments

`internal-virtual-network-gateway`

The gateway ip address of the internal virtual network associated with file server VMs.

`internal-virtual-network-mask`

The network mask of the internal virtual network associated with file server VMs.

`internal-virtual-network-ips`

IPs used by new file server VMs in internal virtual network. Comma-separated IP ranges or IPs, e.g. "10.4.1.3-10.4.1.5,10.4.1.11".

`external-virtual-network-gateway`

The gateway ip address of the external virtual network associated with file server VMs.

`external-virtual-network-mask`

The network mask of the internal virtual network associated with file server VMs.

`external-virtual-network-ips`

IPs used by new file server VMs in the external virtual network. Comma-separated IP ranges or IPs, e.g. "10.4.1.3-10.4.1.5,10.4.1.11"

`nvm-count`

Total number of file server VMs associated with the file server.

Default: 3

`memory`

Memory associated with each file server VM.

Default: 12

`cpu-count`

Number of vCPUs per file server VM.

Default: 4

`ad-protocol-type`

Protocol Type(NONE,SMB,NFS,"NFS,SMB")

`ldap-protocol-type`

Protocol Type(NONE,NFS)

`local-protocol-type`

Protocol Type(NONE,NFS)

`windows-ad-domain-name`

The windows AD domain the file server is associated with.

`windows-ad-username`

The name of a user account with administrative privileges in the AD domain the file server is associated with.

`windows-ad-password`

The password for the above Windows AD account

`overwrite`

If a machine account with the same name as file-server name is present on the AD, then overwrite it during join-domain operation.

Default: false

`organizational-unit`

An Organizational unit container is where the AFS machine account will be created as part of domain join operation. Use a / to specify hierarchical Organization Units. Default container OU is "Computers". Examples: Engineering, Department/Engineering.

`rfc-enabled`

RFC 2307 ENABLED (true, false)

Default: false

`pd-name`

Name of the pd going to be associated with the file server.

`add-user-as-afs-admin`

AD user or group name as 'name' or 'NETBIOS\name' format.

`preferred-domain-controller`

Use a specific domain controller for join-domain operation in a multi DC Active Directory setups. By default, AFS discovers a site local domain controller for join-domain operation.

`nfsv4-domain`

NFSV4 Domain

`ldap-server-uri`

LDAP URI.

`ldap-base-dn`

LDAP Base DN.

`ldap-username`

LDAP Username.

`ldap-password`

LDAP Password.

`ldap-ssh-key`

LDAP SSH Certificate Key.

`dns-operation-type`

DNS Nameserver operation type: AFS specific DNS entries are needed on the nameservers so that clients can connect to the fileserver using its FQDN (fileserver_name.domain_name). AFS can add these entries automatically for MS-DNS servers or they can be added manually. Select one of "MS_DNS" or "MANUAL".

`dns-server`

Preferred Nameserver FQDN (if not given, AFS finds one using an NS query)

`dns-server-username`

Username to use for adding fileserver DNS entries on the nameservers.

`dns-server-password`

Password to use for adding fileserver DNS entries on the nameservers.

`nfsversion`

Nfs version (NFSV3, NFSV4, NFSV3V4)

Delete a File Server

```
ncli> file-server { delete  | remove | rm } uuid="uuid" [ force="force" ][ delete-pd-and-snapshots="delete_pd_and_snapshots" ]
```

Required arguments

`uuid`

UUID of the FileServer

Optional arguments

`force`

force delete file server

`delete-pd-and-snapshots`

delete pd and all snapshots

Delete a admin user

```
ncli> file-server { delete-admin-user } uuid="uuid" admin-user-uuid="admin_user_uuid"
```

Required arguments

`uuid`

Uuid of the file server that admin user is associated with

`admin-user-uuid`

uuid of the admin user

Delete a icap server

```
ncli> file-server { delete-icap-server } uuid="uuid" icap-server-uuid="icap_server_uuid"
```

Required arguments

`uuid`

Uuid of the file server that icap is associated with

`icap-server-uuid`

uuid of the icap server

Delete infected files

```
ncli> file-server { delete-infected-files } uuid="uuid" infected-file-uuids="infected_file_uuids"
```

Required arguments

`uuid`

uuid of the file server

`infected-file-uuids`

Comma-separated list of infected file uuids

Delete a Quota Policy

```
ncli> file-server { delete-quota-policy }uuid="uuid" share-uuid="share_uuid"
quota-policy-uuid="quota_policy_uuid"
```

Required arguments

`uuid`

Uuid of the file server that quota policy is associated with

`share-uuid`

Uuid of the share that quota policy is associated with

`quota-policy-uuid`

uuid of the quota policy

Delete a Share

```
ncli> file-server { delete-share }uuid="uuid" share-uuid="share_uuid"[
force="force" ]
```

Required arguments

`uuid`

uuid of the File Server

`share-uuid`

uuid of the FileServer share

Optional arguments

`force`

force delete Share

Delete a Snapshot Policy

```
ncli> file-server { delete-snapshot-policy }file-server-uuid="file_server_uuid"
uuid="uuid"
```

Required arguments

`file-server-uuid`

Uuid of the file server that snapshot policy is associated with

`uuid`

uuid of the snapshot policy

Delete a user

```
ncli> file-server { delete-user }uuid="uuid" user="user"
```

Required arguments

`uuid`

Uuid of the file server that user is associated with

`user`

Name of the user

Delete a virus scan policy

```
ncli> file-server { delete-virus-scan-policy }uuid="uuid" share-uuid="share_uuid"
```

Required arguments

`uuid`

Uuid of the file server that virus scan policy is associated with

`share-uuid`

Uuid of the share that virus scan policy is associated with

Update a File Server

```
ncli> file-server { edit  | update } uuid="uuid" [ name="name" ][ dns-
domain-name="dns_domain_name" ][ total-nvm-count="total_nvm_count" ][
internal-virtual-network-ips="internal_virtual_network_ips" ][ external-
virtual-network-ips="external_virtual_network_ips" ][ dns-server-ip-address-
list="dns_server_ip_address_list" ][ ntp-servers="ntp_servers" ][ size-gib="size_gib" ][
memory="memory" ][ cpu-count="cpu_count" ]
```

Required arguments

`uuid`

Uuid of the file server

Optional arguments

`name`

Name of the file server

`dns-domain-name`

Fully qualified domain name (fileserver namespace). This, along with the fileserver name, constitutes the namespace of the fileserver. Example: fileserver_name.corp.companyname.com. This is also used to create fileserver DNS entries on the nameservers so that clients can access the fileserver using its name.

`total-nvm-count`

Total number of file server VMs associated with the file server.

`internal-virtual-network-ips`

IPs used by new file server VMs in the internal virtual network.

`external-virtual-network-ips`

IPs used by new file server VMs in the external virtual network. Comma-separated IP ranges or IPs, e.g. "10.4.1.3-10.4.1.5,10.4.1.11"

`dns-server-ip-address-list`

List of comma-separated dns server ip addresses for file server configuration.

`ntp-servers`

List of comma-separated ntp servers for file server configuration.

`size-gib`

File server size in gibs

`memory`

Memory associated with each file server VM.

`cpu-count`

Number of vCPUs per file server VM.

Show an individual FileServer's details

```
ncli> file-server { get } uuid="uuid" [ projection="projection" ]
```

Required arguments

`uuid`

uuid of the FileServer

Optional arguments

`projection`

Projections on the attributes

Get a admin user

```
ncli> file-server { get-admin-user }uuid="uuid" admin-user-uuid="admin_user_uuid"
```

Required arguments

`uuid`

Uuid of the file server that admin user is associated with

`admin-user-uuid`

uuid of the admin user

Show built in groupsper File Server

```
ncli> file-server { get-builtin-groups }uuid="uuid"[ projection="projection" ]
```

Required arguments

`uuid`

uuid of the FileServer

Optional arguments

`projection`

Projections on the attributes

Get icap server

```
ncli> file-server { get-icap-server }uuid="uuid" icap-server-
uuid="icap_server_uuid"
```

Required arguments

`uuid`

uuid of the file server

`icap-server-uuid`

uuid of the icap server

Get infected file

```
ncli> file-server { get-infected-file }uuid="uuid" infected-file-
uuid="infected_file_uuid"[ quarantined="quarantined" ]
```

Required arguments

`uuid`

uuid of the file server

`infected-file-uuid`

uuid of the infected file

Optional arguments

`quarantined`

Quarantined

Get principal type from principal name

```
ncli> file-server { get-principal-type  | get-principal-type}uuid="uuid" share-
uuid="share_uuid" principal-value="principal_value"[ principal-type="principal_type" ][
protocol-type="protocol_type" ]
```

Required arguments

`uuid`

Uuid of the file server

`share-uuid`

Uuid of the Share

`principal-value`

Quota policy principal value (user or group name)

Optional arguments

`principal-type`

Quota policy principal type (enter 'user' or 'group')

`protocol-type`

Primary protocol type (SMB or NFS)

Get a Quota Policy

```
ncli> file-server { get-quota-policy }uuid="uuid" share-uuid="share_uuid" quota-policy-uuid="quota_policy_uuid"
```

Required arguments

`uuid`

Uuid of the file server that quota policy is associated with

`share-uuid`

uuid of the share

`quota-policy-uuid`

uuid of the quota policy

Show an individual share's details

```
ncli> file-server { get-share  | show-share }uuid="uuid" share-uuid="share_uuid"[ projection="projection" ]
```

Required arguments

`uuid`

uuid of the file server

`share-uuid`

uuid of the FileServer share

Optional arguments

`projection`

Projections on the attributes

Get a file server share snapshot

```
ncli> file-server { get-snapshot }uuid="uuid" share-uuid="share_uuid" snapshot-uuid="snapshot_uuid"
```

Required arguments

`uuid`

uuid of the file server

`share-uuid`

uuid of the file server share

`snapshot-uuid`

uuid of the file server share snapshot

Get a User Mapping

```
ncli> file-server { get-user-mapping  | get-user-mapping } uuid="uuid"
```

Required arguments

`uuid`

Uuid of the file server

Get a virus scan policy

```
ncli> file-server { get-virus-scan-policy } uuid="uuid" [ share-uuid="share_uuid" ]
```

Required arguments

`uuid`

uuid of the file server

Optional arguments

`share-uuid`

uuid of the file server share

Join the File Server to the Windows AD domain specified.

```
ncli> file-server { join-domain  | join-domain } uuid="uuid" windows-ad-
domain-name="windows_ad_domain_name" windows-ad-username="windows_ad_username" [
organizational-unit="organizational_unit" ][ windows-ad-password="windows_ad_password"
][ overwrite="overwrite" ][ add-user-as-afs-admin="add_user_as_afs_admin" ][ preferred-
domain-controller="preferred_domain_controller" ]
```

Required arguments

`uuid`

Uuid of the file server

`windows-ad-domain-name`

The windows AD domain the file server is associated with.

`windows-ad-username`

The name of a user account with administrative privileges in the AD domain the file server is associated with.

Optional arguments

`organizational-unit`

An Organizational unit container is where the AFS machine account will be created as part of domain join operation. Use a / to specify hierarchical Organization Units. Default container OU is "Computers". Examples: Engineering, Department/Engineering.

`windows-ad-password`

The password for the above Windows AD account

`overwrite`

If a machine account with the same name as file-server name is present on the AD, then overwrite it during join-domain operation.

Default: false

`add-user-as-afs-admin`

AD user or group name as 'name' or 'NETBIOS\name' format.

`preferred-domain-controller`

Use a specific domain controller for join-domain operation in a multi DC Active Directory setups. By default, AFS discovers a site local domain controller for join-domain operation.

Leave File Server from domain

```
ncli> file-server { leave-domain | leave-domain } uuid="uuid" [ windows-ad-
password="windows_ad_password" ][ windows-ad-username="windows_ad_username" ]
```

Required arguments

`uuid`

Uuid of the file server

Optional arguments

`windows-ad-password`

The password for the above Windows AD account

`windows-ad-username`

The name of a user account with administrative privileges in the AD domain the file server is associated with.

List all File Servers with filtering

```
ncli> file-server { list | ls }[ count="count" ][ filter-criteria="filter_criteria" ][
search-string="search_string" ][ projection="projection" ]
```

Required arguments

None

Optional arguments

`count`

Number of Containers to retrieve

`filter-criteria`

Filter criteria

`search-string`

Search string

`projection`

Projections on the attributes

List all admin users with filtering

```
ncli> file-server { list-admin-users } uuid="uuid"
```

Required arguments

`uuid`

uuid of the file server

List all shares with filtering

```
ncli> file-server { list-all-fs-shares }[ count="count" ][ filter-
criteria="filter_criteria" ][ search-string="search_string" ][ search-attribute-
list="search_attribute_list" ][ projection="projection" ]
```

Required arguments

None

Optional arguments

`count`

Number of shares to retrieve

`filter-criteria`

Filter criteria

`search-string`

Search string

`search-attribute-list`

Search attribute list

`projection`

Projections on the attributes

List all DNS entries

```
ncli> file-server { list-dns }uuid="uuid"
```

Required arguments

`uuid`

uuid of the file server

List all icap servers with filtering

```
ncli> file-server { list-icap-servers }uuid="uuid"
```

Required arguments

`uuid`

uuid of the file server

List all infected files

```
ncli> file-server { list-infected-files }uuid="uuid" quarantined="quarantined"
[ share-uuid="share_uuid" ][ count="count" ][ page="page" ][ filter-
criteria="filter_criteria" ][ search-string="search_string" ][ projection="projection" ]
```

Required arguments

`uuid`

uuid of the file server

`quarantined`

Quarantined

Optional arguments

`share-uuid`

Uuid of the share

`count`

Number of infected files to retrieve

`page`

Page number

`filter-criteria`

Filter criteria

`search-string`

Search string

`projection`

Projections on the attributes

List all quota policies with filtering

```
ncli> file-server { list-quota-policies }uuid="uuid" share-uuid="share_uuid"
```

Required arguments

`uuid`

Uuid of the file server that quota policy is associated with

`share-uuid`

uuid of the share

List all shares with filtering

```
ncli> file-server { list-shares  | list-shares } uuid="uuid" [ count="count" ][
filter-criteria="filter_criteria" ][ search-string="search_string" ][ search-attribute-
list="search_attribute_list" ][ projection="projection" ]
```

Required arguments

`uuid`

uuid of the file server

Optional arguments

`count`

Number of Containers to retrieve

`filter-criteria`

Filter criteria

`search-string`

Search string

`search-attribute-list`

Search attribute list

`projection`

Projections on the attributes

List all Snapshot policies with filtering

```
ncli> file-server { list-snapshot-policies } file-server-uuid="file_server_uuid"
```

Required arguments

`file-server-uuid`

uuid of the file server

List all Snapshots with filtering

```
ncli> file-server { list-snapshots } uuid="uuid" share-uuid="share_uuid"
```

Required arguments

`uuid`

uuid of the file server

`share-uuid`

uuid of the file server share

List users

```
ncli> file-server { list-user } uuid="uuid"
```

Required arguments

`uuid`

uuid of the file server

Load balance a File Server

```
ncli> file-server { load-balance } uuid="uuid" action="action" [ total-nvm-
count="total_nvm_count" ][ internal-virtual-network-ips="internal_virtual_network_ips" ][
external-virtual-network-ips="external_virtual_network_ips" ][ memory="memory" ][ cpu-
count="cpu_count" ]
```

Required arguments

`uuid`

Uuid of the file server

`action`

Action based on the recommendation associated with file server.i.e {scaleup, scaleout or rebalance}

Optional arguments

`total-nvm-count`

Total number of file server VMs associated with the file server.

`internal-virtual-network-ips`

IPs used by new file server VMs in the internal virtual network.

`external-virtual-network-ips`

IPs used by new file server VMs in the external virtual network. Comma-separated IP ranges or IPs, e.g. "10.4.1.3-10.4.1.5,10.4.1.11"

`memory`

Memory associated with each file server VM.

`cpu-count`

Number of vCPUs per file server VM.

Add a protection domain for File Server.

```
ncli> file-server { protect }uuid="uuid"[ pd-name="pd_name" ]
```

Required arguments

`uuid`

UUID of the FileServer

Optional arguments

`pd-name`

pd name

Quarantine infected files

```
ncli> file-server { quarantine-infected-files }uuid="uuid" infected-file-
uuids="infected_file_uuids"
```

Required arguments

`uuid`

uuid of the file server

`infected-file-uuids`

Comma-separated list of infected file uuids

Remove DNS entries

```
ncli> file-server { remove-dns }uuid="uuid" dns-operation-
type="dns_operation_type" dns-server-username="dns_server_username" dns-server-
password="dns_server_password"[ dns-server="dns_server" ]
```

Required arguments

`uuid`

UUID of the FileServer

`dns-operation-type`

DNS Nameserver operation type: AFS specific DNS entries are needed on the nameservers so that clients can connect to the fileserver using its FQDN

(fileserver_name.domain_name). AFS can add these entries automatically for MS-DNS servers or they can be added manually. Select one of "MS_DNS" or "MANUAL".

`dns-server-username`

Username to use for adding fileserver DNS entries on the nameservers.

`dns-server-password`

Password to use for adding fileserver DNS entries on the nameservers.

Optional arguments

`dns-server`

Preferred Nameserver FQDN (if not given, AFS finds one using an NS query)

Rescan infected files

```
ncli> file-server { rescan-infected-files }uuid="uuid" infected-file-uuids="infected_file_uuids"
```

Required arguments

`uuid`

uuid of the file server

`infected-file-uuids`

Comma-separated list of infected file uuids

Reset infected files

```
ncli> file-server { reset-infected-files }uuid="uuid" infected-file-uuids="infected_file_uuids"
```

Required arguments

`uuid`

uuid of the file server

`infected-file-uuids`

Comma-separated list of infected file uuids

Test icap server connection

```
ncli> file-server { test-icap-connection }uuid="uuid" icap-server-uuid="icap_server_uuid"
```

Required arguments

`uuid`

Uuid of the file server that icap server is associated with

`icap-server-uuid`

uuid of the icap server

Unquarantine infected files

```
ncli> file-server { unquarantine-infected-files }uuid="uuid" infected-file-uuids="infected_file_uuids"
```

Required arguments

`uuid`

uuid of the file server

`infected-file-uuids`

Comma-separated list of infected file uuids

Update a admin user

```
ncli> file-server { update-admin-user }uuid="uuid" admin-user-
uuid="admin_user_uuid"[ user="user" ][ role="role" ]
```

Required arguments

`uuid`

Uuid of the file server that admin user is associated with

`admin-user-uuid`

Admin user uuid.

Optional arguments

`user`

AD user or group name as 'name' or 'NETBIOS\name' format.

`role`

Role of the given user or group (one of AdMIN or BACKUP_OPERATOR).

Update a icap server

```
ncli> file-server { update-icap-server }uuid="uuid" icap-server-
uuid="icap_server_uuid"[ description="description" ][ enabled="{ true | false }" ]
```

Required arguments

`uuid`

Uuid of the file server that Icap is associated with

`icap-server-uuid`

Uuid of the antivirus server

Optional arguments

`description`

Antivirus server description.

`enabled`

Enable or diable antivirus server.

Update network of a File Server

```
ncli> file-server { update-network }uuid="uuid"[ dns-server-ip-address-
list="dns_server_ip_address_list" ][ ntp-servers="ntp_servers" ][ internal-virtual-
network="internal_virtual_network" ][ external-virtual-network="external_virtual_network"
][ internal-virtual-network-gateway="internal_virtual_network_gateway" ][
internal-virtual-network-mask="internal_virtual_network_mask" ][ internal-
virtual-network-ips="internal_virtual_network_ips" ][ external-virtual-
network-gateway="external_virtual_network_gateway" ][ external-virtual-
network-mask="external_virtual_network_mask" ][ external-virtual-network-
ips="external_virtual_network_ips" ]
```

Required arguments

`uuid`

Uuid of the file server

Optional arguments

`dns-server-ip-address-list`

List of comma-separated dns server ip addresses for file server configuration.

`ntp-servers`

List of comma-separated ntp servers for file server configuration.

`internal-virtual-network`

The identifier of the internal virtual network. Please use network UUID on AHV and ESX.

`external-virtual-network`

The identifier of the external virtual network. Please use network UUID on AHV and ESX.

`internal-virtual-network-gateway`

The gateway ip address of the internal virtual network associated with file server VMs.

`internal-virtual-network-mask`

The network mask of the internal virtual network associated with file server VMs.

`internal-virtual-network-ips`

IPs used by new file server VMs in internal virtual network. Comma-separated IP ranges or IPs, e.g. "10.4.1.3-10.4.1.5,10.4.1.11".

`external-virtual-network-gateway`

The gateway ip address of the external virtual network associated with file server VMs.

`external-virtual-network-mask`

The network mask of the internal virtual network associated with file server VMs.

`external-virtual-network-ips`

IPs used by new file server VMs in the external virtual network. Comma-separated IP ranges or IPs, e.g. "10.4.1.3-10.4.1.5,10.4.1.11"

Update a Quota Policy

```
ncli> file-server { update-quota-policy }uuid="uuid" share-uuid="share_uuid"
quota-policy-uuid="quota_policy_uuid"[ quota-size-gib="quota_size_gib" ][ quota-
enforcement-type="quota_enforcement_type" ][ send-quota-notifications-to-
user="send_quota_notifications_to_user" ][ notification-recipients="notification_recipients" ]
```

Required arguments

`uuid`

Uuid of the file server that quota policy is associated with

`share-uuid`

Uuid of the share that quota policy is associated with

`quota-policy-uuid`

Uuid of the Quota Policy

Optional arguments

`quota-size-gib`

Quota size in Gibs

`quota-enforcement-type`

Quota enforcement type (Hard or Soft)

`send-quota-notifications-to-user`

Send quota notifications to user

`notification-recipients`

Additional notification recipients (comma-separated)

Update a Share

```
ncli> file-server { update-share  | edit-share }uuid="uuid" share-
uuid="share_uuid"[ name="name" ][ enable-windows-previous-version="{ true | false }" ][
description="description" ][ share-size-gib="share_size_gib" ][ default-quota-limit-
gib="default_quota_limit_gib" ][ quota-enforcement-type="quota_enforcement_type" ][ send-
quota-notifications-to-user="send_quota_notifications_to_user" ][ enable-access-based-
```

```
enumeration="{ true | false }" ][ protocol-type="protocol_type" ][ secondary-protocol-
type="secondary_protocol_type" ][ enable-concurrent-reads="{ true | false }" ][ enable-
case-sensitive-namespace="{ true | false }" ][ enable-symlink-creation="{ true | false }"
][ enable-simultaneous-access="{ true | false }" ][ share-auth-type="share_auth_type"
][ default-share-access-type="default_share_access_type" ][ client-with-
read-write-access="client_with_read_write_access" ][ client-with-read-only-
access="client_with_read_only_access" ][ client-with-no-access="client_with_no_access"
][ anonymous-uid="anonymous_uid" ][ anonymous-gid="anonymous_gid" ][ squash-
type="squash_type" ]
```

Required arguments

`uuid`

uuid of the File Server

`share-uuid`

Uuid of the Share

Optional arguments

`name`

Name of the Share

`enable-windows-previous-version`

Enable self service restore flag

`description`

Description of the Share

`share-size-gib`

Share size in Gibs

`default-quota-limit-gib`

Default quota limit in Gibs (Quota applies to all users of the share)

`quota-enforcement-type`

Quota enforcement type (Hard or Soft)

`send-quota-notifications-to-user`

Send quota notifications to user

`enable-access-based-enumeration`

Enable access based enumeration flag

`protocol-type`

Primary protocol type (SMB or NFS)

`secondary-protocol-type`

Secondary protocol type (SMB or NFS)

`enable-concurrent-reads`

Concurrent Reads Enabled

`enable-case-sensitive-namespace`

Case sensitive Creation Enabled

`enable-symlink-creation`

Symlink Creation Enabled

`enable-simultaneous-access`

Simultaneous Access Enabled

`share-auth-type`

Type of Share Authentication. (System-none, Kerberos)

`default-share-access-type`

Type of Default Share Access. (Read-write, Read-only, No-Access)

`client-with-read-write-access`

Comma-separated list of clients

`client-with-read-only-access`

Comma-separated list of clients

`client-with-no-access`

Comma-separated list of clients

`anonymous-uid`

Anonymous UID.

`anonymous-gid`

Anonymous GID.

`squash-type`

Squash Type.(root-squash,all-squash,none)

Update a Snapshot Policy

```
ncli> file-server { update-snapshot-policy } file-server-uuid="file_server_uuid"
uuid="uuid" [ snapshot-policy-type="snapshot_policy_type" ][ frequency="frequency"
][ days-of-week="days_of_week" ][ days-of-month="days_of_month" ][ local-
retention="local_retention" ]
```

Required arguments

`file-server-uuid`

Uuid of the file server that snapshot policy is associated with

`uuid`

Uuid of the Snapshot Policy

Optional arguments

`snapshot-policy-type`

Snapshot policy type - hourly, daily, weekly, monthly

`frequency`

Snapshot policy frequency. Repeat snapshot every nth hour/day. For weekly and monthly, frequency is 1.

`days-of-week`

Comma-separated day of week values for the schedule: 1-7 (starts with Sunday(1))

`days-of-month`

Comma-separated day of month values for the schedule: 1-31

`local-retention`

Maximum number of snapshots to retain locally

Update a user

```
ncli> file-server { update-user } uuid="uuid" [ user="user" ][ password="password" ]
```

Required arguments

`uuid`

Uuid of the file server that user is associated with

Optional arguments

`user`

File server user name.

`password`

The password for the above file server user.

Update a User Mapping

```
ncli> file-server { update-user-mapping  | update-user-mapping } uuid="uuid"
[ template-mapping-rule="template_mapping_rule" ][ smb-user-and-group-with-no-
nfs-mapping-action="smb_user_and_group_with_no_nfs_mapping_action" ][ nfs-user-
no-smb-mapping-action="nfs_user_no_smb_mapping_action" ][ smb-user-map-to-nfs-
uid="smb_user_map_to_nfs_uid" ][ smb-group-map-to-nfs-gid="smb_group_map_to_nfs_gid"
][ nfs-user-map-to-smb-name="nfs_user_map_to_smb_name" ][ nfs-group-map-to-smb-
name="nfs_group_map_to_smb_name" ][ one-to-one-mapping="one_to_one_mapping" ][ wild-
card-mapping="wild_card_mapping" ][ deny-access-smb-users="deny_access_smb_users"
][ deny-access-smb-groups="deny_access_smb_groups" ][ deny-access-nfs-
users="deny_access_nfs_users" ][ deny-access-nfs-groups="deny_access_nfs_groups" ][
file-path="file_path" ]
```

Required arguments

`uuid`

Uuid of the file server

Optional arguments

`template-mapping-rule`

Pre-defined templates to map SMB and NFS identities. Allowed values(NO_TEMPLATE_MAPPING, SMB_NAME_NFS_NAME)

`smb-user-and-group-with-no-nfs-mapping-action`

SMB user and group with no NFS mapping action. Allowed values (DENY_ACCESS, MAP_IDENTITIES)

`nfs-user-no-smb-mapping-action`

NFS user and group with no SMB mapping action. Allowed values (DENY_ACCESS, MAP_IDENTITIES).

`smb-user-map-to-nfs-uid`

SMB user mapping with NFS uid

`smb-group-map-to-nfs-gid`

SMB group mapping with NFS uid

`nfs-user-map-to-smb-name`

NFS user mapping with SMB name, eg. domain\username.

`nfs-group-map-to-smb-name`

NFS group mapping with SMB name, eg. domain\username.

`one-to-one-mapping`

One to one explicit mapping associated with file server

`wild-card-mapping`

Wild card explicit mapping associated with file server

`deny-access-smb-users`

Deny access SMB users, eg. domain\username.

`deny-access-smb-groups`

Deny access SMB groups, eg. domain\username.

`deny-access-nfs-users`

Deny access NFS users, eg. domain\username.

`deny-access-nfs-groups`

Deny access NFS groups, eg. domain\username.

`file-path`

File path

Update a virus scan policy

```
ncli> file-server { update-virus-scan-policy } uuid="uuid" [ scan-time-interval-secs="scan_time_interval_secs" ][ share-uuid="share_uuid" ][ scan-on-write="scan_on_write" ][ scan-on-read="scan_on_read" ][ file-size-exclusion-bytes="file_size_exclusion_bytes" ][ block-access-file="block_access_file" ][ enable-anti-virus="{ true | false }" ][ file-type-exclusions="file_type_exclusions" ]
```

Required arguments

`uuid`

Uuid of the file server that antivirus scan policy is associated with

Optional arguments

`scan-time-interval-secs`

Scan time interval in uses

`share-uuid`

Uuid of the Share

`scan-on-write`

Scan on write policy

`scan-on-read`

Scan on read policy

`file-size-exclusion-bytes`

File size exclusion

`block-access-file`

Block access file

`enable-anti-virus`

Enable antivirus

`file-type-exclusions`

File type exclusion list

Upgrade given list of File Servers

```
ncli> file-server { upgrade  | upgrade } version="version" [ uuid="uuid" ][ upgrade-all-file-servers="upgrade_all_file_servers" ]
```

Required arguments

`version`

Version of the file server

Optional arguments

`uuid`

Uuid of the file server

`upgrade-all-file-servers`

Upgrade all file servers flag.

Verify DNS entries

```
ncli> file-server { verify-dns } uuid="uuid" [ dns-server="dns_server" ]
```

Required arguments

`uuid`

UUID of the FileServer

Optional arguments

`dns-server`

Dns server IP address

`health-check`: Health Check

Description A health check

Alias `check`

Operations

- Edit a Health Check : `edit` | `update`
- List Health Checks : `list` | `ls`

Edit a Health Check

```
ncli> health-check { edit | update } id="id" [ enable="{ true | false }" ][
interval="interval" ][ parameter-thresholds="parameter_thresholds" ][ severity="severity"
][ enable-severity-threshold="{ true | false }" ][ parameter-name="parameter_name" ][
parameter-value="parameter_value" ][ auto-resolve="auto_resolve" ]
```

Required arguments

`id`

Id of the Health Check

Optional arguments

`enable`

Enable/Disable Health Check

`interval`

Scheduled interval at which Health Check should be run (in seconds)

`parameter-thresholds`

List of comma-separated parameter and it's threshold. All the values should be of format <parameter-name:value>.

`severity`

Severity Threshold to update

`enable-severity-threshold`

Enable or Disable the selected Severity Threshold

`parameter-name`

Name of the parameter to update.

`parameter-value`

Updated value of the parameter.

`auto-resolve`

Enable/Disable Auto Resolve feature for this check. Value should be set to either true or false

List Health Checks

```
ncli> health-check { list  | ls }[ id="id" ]
```

Required arguments

None

Optional arguments

`id`

Id of the Health Check

host: Physical Host

Description A Physical Host hosts Virtual Machines

Alias

Operations

- Add the configured node to the cluster : `add-node`
- Configure discovered node with IP addresses (Hypervisor, CVM and IPMI addresses) : `configure-node`
- Discover new nodes available to add to the cluster : `discover-nodes`
- Edit Physical Host : `edit` | `update`
- Enable metadata store on a Physical Host : `enable-metadata-store`
- Generates and downloads the csr from discovered node based on certification information from the cluster : `generate-csr-for-discovered-node`
- Get certificate related information : `get-certificate-information`
- Check the removal status for Physical Hosts : `get-remove-status` | `get-rm-status`
- Join one or more host(s) to a domain : `join-domain`
- List Physical Hosts : `list` | `ls`
- Get stats data for Physical Host : `list-stats` | `ls-stats`
- Begin the process of removing a Physical Host : `remove-start` | `rm-start` | `delete`
- Reset to factory setting, the default location to be used for storing the virtual machine configuration files and the virtual hard disk files : `reset-default-vm-vhd-location`
- Set the default location to be used for storing the virtual machine configuration files and the virtual hard disk files : `set-default-vm-vhd-location`
- Set the monitoring status of Physical Hosts : `set-monitoring`

Add the configured node to the cluster. In case of compute only node, cvm ip corresponds to host ip

```
ncli> host { add-node } node-uuid="node_uuid" [ server-certificate-
list="server_certificate_list" ]
```

Required arguments

`node-uuid`

UUID of the new node

Optional arguments

`server-certificate-list`

Comma-separated list of the key management server uuid and corresponding certificate file path. List should be of format <server_uuid:path_to_certificate>

Configure discovered node with IP addresses (Hypervisor, CVM and IPMI addresses)

```
ncli> host { configure-node }node-uuid="node_uuid"[ cvm-ip="cvm_ip" ][
hypervisor-ip="hypervisor_ip" ][ ipmi-ip="ipmi_ip" ][ ipmi-netmask="ipmi_netmask" ][
ipmi-gateway="ipmi_gateway" ]
```

Required arguments

`node-uuid`

UUID of the new node

Optional arguments

`cvm-ip`

IP address of the controller VM

`hypervisor-ip`

IP address of the Hypervisor Host

`ipmi-ip`

IPMI address of the node

`ipmi-netmask`

IPMI netmask of the node

`ipmi-gateway`

IPMI gateway of the node

Discover new nodes available to add to the cluster

```
ncli> host { discover-nodes }
```

Required arguments

None

Edit Physical Host

```
ncli> host { edit  | update }id="id"[ hypervisor-username="hypervisor_username"
][ hypervisor-password="hypervisor_password" ][ new-ms-name="new_ms_name"
][ oplog-disk-pct="oplog_disk_pct" ][ ipmi-username="ipmi_username" ][ ipmi-
password="ipmi_password" ][ cvm-nat-ip="cvm_nat_ip" ][ cvm-nat-port="cvm_nat_port" ]
```

Required arguments

`id`

ID of the Physical Host

Optional arguments

`hypervisor-username`

Username to access the Hypervisor Host

`hypervisor-password`

Password to access the Hypervisor Host

`new-ms-name`

Name of the Management Server

`oplog-disk-pct`

The percentage of the oplog-disk to use for oplog data

`ipmi-username`

Username for IPMI access to a Physical Host

`ipmi-password`

Password for IPMI access to a Physical Host

`cvm-nat-ip`

NAT IP address of the controller VM. Set to '-' to clear the existing value

`cvm-nat-port`

NAT port of the controller VM. Set to '-' to clear the existing value

Enable metadata store on a Physical Host

```
ncli> host { enable-metadata-store }id="id"
```

Required arguments

`id`

ID of the Physical Host

Generates and downloads the csr from discovered node based on certification information from the cluster

```
ncli> host { generate-csr-for-discovered-node }cvm-ip="cvm_ip" file-
path="file_path"
```

Required arguments

`cvm-ip`

IPv6 address of the controller VM of discovered node

`file-path`

Path where csr from the discovered node needs to be downloaded

Get certificate related information.

```
ncli> host { get-certificate-information }[ id="id" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Physical Host

Check the removal status for Physical Hosts

```
ncli> host { get-remove-status  | get-rm-status}[ id="id" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Physical Host

Join one or more host(s) to a domain. This operation is only valid for hosts running Hyper-V.

```
ncli> host { join-domain }domain="domain" logon-name="logon_name"
restart="restart"[ name-server-ip="name_server_ip" ][ host-name-
prefix="host_name_prefix" ][ password="password" ][ host-ids="host_ids" ][ host-
names="host_names" ][ ou-path="ou_path" ][ cps-prefix="cps_prefix" ]
```

Required arguments

`domain`

Full name of the domain

`logon-name`

Logon name (domain\username) of a domain user/administrator account that has privileges to perform the operation

`restart`

Whether to restart the host(s) after the successful execution of a domain join/unjoin operation

Optional arguments

`name-server-ip`

IP address of the name server that can resolve the domain name

`host-name-prefix`

The prefix to be used in naming the hosts running Hyper-V. Must not exceed 11 characters in length

`password`

Password for the account specified by the logon account name

`host-ids`

A comma-separated list of the ids of the Physical Hosts

`host-names`

A comma-separated list of the names of the Physical Hosts

`ou-path`

Organizational Unit path of the domain

`cps-prefix`

CPS prefix path of the domain

List Physical Hosts

```
ncli> host { list  | ls }[ id="id" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Physical Host

Get stats data for Physical Host

```
ncli> host { list-stats  | ls-stats }[ id="id" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Physical Host

Begin the process of removing a Physical Host

```
ncli> host { remove-start  | rm-start | delete }id="id"[ skip-space-check="{true | false}" ][ force="force" ]
```

Required arguments

`id`

ID of the Physical Host

Optional arguments

`skip-space-check`

Skip checking storage space-related constraints when initiating removal of a host from the cluster

Default: false

`force`

Forcefully perform the requested operation skipping any constraint validation

Default: false

Reset to factory setting, the default location to be used for storing the virtual machine configuration files and the virtual hard disk files. This operation is only valid for hosts running Hyper-V.

```
ncli> host { reset-default-vm-vhd-location } host-ids="host_ids"
```

Required arguments

`host-ids`

A comma-separated list of the ids of the Physical Hosts

Set the default location to be used for storing the virtual machine configuration files and the virtual hard disk files. This operation is only valid for hosts running Hyper-V.

```
ncli> host { set-default-vm-vhd-location } ctr-for-vm-config="ctr_for_vm_config"
ctr-for-vhd-files="ctr_for_vhd_files" [ host-ids="host_ids" ]
```

Required arguments

`ctr-for-vm-config`

Name of the Storage Container to be used for storing VM configuration files.

`ctr-for-vhd-files`

Name of the Storage Container to be used for storing virtual hard disk files.

Optional arguments

`host-ids`

A comma-separated list of the ids of the Physical Hosts

Set the monitoring status of Physical Hosts

```
ncli> host { set-monitoring } enabled="{ true | false }" ids="ids"
```

Required arguments

`enabled`

Enable monitoring of Physical Hosts?

`ids`

A comma-separated list of the ids of the Physical Hosts

`http-proxy`: HTTP Proxy

Description An HTTP Proxy

Alias `proxy`

Operations

- Create a new HTTP Proxy : `add`
- Add HTTP Proxy whitelist entry : `add-to-whitelist`
- Delete HTTP Proxy whitelist entry : `delete-from-whitelist`
- Edit an HTTP Proxy : `edit` | `update`
- Get HTTP Proxy whitelist : `get-whitelist`
- List HTTP Proxies : `list` | `ls`
- Remove an HTTP Proxy : `remove` | `rm`

Create a new HTTP Proxy

```
ncli> http-proxy { add }name="name" address="address" port="port"[
username="username" ][ password="password" ][ proxy-types="proxy_types" ]
```

Required arguments

`name`

Proxy name.

`address`

Address of the proxy

`port`

Port on which proxy is binding

Optional arguments

`username`

Username for proxy authentication

`password`

Password for proxy authentication

`proxy-types`

Proxy types to send applicable traffic

Add HTTP Proxy whitelist entry

```
ncli> http-proxy { add-to-whitelist }target-type="target_type" target="target"
```

Required arguments

`target-type`

Type of the target. Values can be {IPV4_ADDRESS, IPV4_NETWORK_MASK, DOMAIN_NAME_SUFFIX, HOST_NAME}

`target`

Http proxy whitelist target. If the target-type is IPV4_NETWORK_MASK, specify the netmask using traditional notation (for example, 172.16.0.0/255.240.0.0), not CIDR notation.

Delete HTTP Proxy whitelist entry

```
ncli> http-proxy { delete-from-whitelist }target="target"
```

Required arguments

`target`

Http proxy whitelist target. If the target-type is IPV4_NETWORK_MASK, specify the netmask using traditional notation (for example, 172.16.0.0/255.240.0.0), not CIDR notation.

Edit an HTTP Proxy

```
ncli> http-proxy { edit  | update }name="name"[ address="address" ][
username="username" ][ password="password" ][ port="port" ]
```

Required arguments

`name`

Name of the HTTP Proxy

Optional arguments

`address`

Address of the HTTP Proxy

`username`

Username to access the HTTP Proxy

`password`

Password to access the HTTP Proxy

`port`

Port number of the HTTP Proxy

Get HTTP Proxy whitelist

```
ncli> http-proxy { get-whitelist }
```

Required arguments

None

List HTTP Proxies

```
ncli> http-proxy { list  | ls }[ name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the HTTP Proxy

Remove an HTTP Proxy

```
ncli> http-proxy { remove  | rm } name="name"
```

Required arguments

`name`

Name of an HTTP Proxy

`key-management-server`: Key Management Server

Description Manage key management servers

Alias

Operations

- Add key management server : `add`
- Get specified key management server : `get`
- List all key management servers : `list` | `ls`
- Remove key management server : `remove` | `rm`
- Update key management server : `update`

Add key management server

```
ncli> key-management-server { add } name="name" address-list="address_list"
```

Required arguments

`name`

Key management server name

`address-list`

List of comma-separated addresses of the key management server. All addresses should be of format <ipaddress> or of format <ipaddress:port>. Port 5696 is used as default if port number is not specified

Get specified key management server

```
ncli> key-management-server { get } name="name"
```

Required arguments

`name`

Key management server name

List all key management servers

```
ncli> key-management-server { list  | ls }
```

Required arguments

None

Remove key management server

```
ncli> key-management-server { remove  | rm } name="name"
```

Required arguments

`name`

Key management server name

Update key management server

```
ncli> key-management-server { update } name="name" [ address-list="address_list" ][
force="force" ]
```

Required arguments

`name`

Key management server name

Optional arguments

`address-list`

List of comma-separated addresses of the key management server. All addresses should be of format <ipaddress> or of format <ipaddress:port>. Port 5696 is used as default if port number is not specified

`force`

Forcefully perform the requested operation skipping any constraint validation

`license`: License

Description License for a Nutanix cluster

Alias

Operations

- Apply a license file to the cluster : `apply-license`
- Download cluster info as a file : `download-cluster-info`
- Get cluster info from the cluster : `generate-cluster-info`
- Read allowances for features as listed in the license : `get-allowances`
- Read license file from the cluster : `get-license`

Apply a license file to the cluster

```
ncli> license { apply-license } license-file="license_file"
```

Required arguments

`license-file`

License file

Download cluster info as a file

```
ncli> license { download-cluster-info } file-path="file_path"
```

Required arguments

`file-path`

File path

Get cluster info from the cluster

```
ncli> license { generate-cluster-info }
```

Required arguments

None

Read allowances for features as listed in the license

```
ncli> license { get-allowances }[ feature-name="feature_name" ]
```

Required arguments

None

Optional arguments

`feature-name`

Feature name

Read license file from the cluster

```
ncli> license { get-license }
```

Required arguments

None

`managementserver`: Management Server

Description An infrastructure management server such as VCenter

Alias `ms`

Operations

- Add a new Management Server : `add`
- Add a new Management Server : `edit` | `update`
- List Management Servers : `list` | `ls`
- Returns a list of information for management servers which are used for managing the cluster : `list-management-server-info`
- Create and register a management server extension for Nutanix : `register`
- Delete a Management Server : `remove` | `rm`
- Unregister the management server extension for Nutanix : `unregister`

Add a new Management Server

```
ncli> managementserver { add } name="name" url="url" username="username"
password="password" [ type="type" ]
```

Required arguments

`name`

Name of the Management Server

`url`

Access URL of the Management Server

`username`

Username for access to the Management Server

`password`

Password for access to the Management Server

Optional arguments

`type`

Hypervisor type of the Management Server

Default: vmware

Add a new Management Server

```
ncli> managementserver { edit  | update } name="name" [ url="url" ][
username="username" ][ password="password" ]
```

Required arguments

`name`

Name of the Management Server

Optional arguments

`url`

Access URL of the Management Server

`username`

Username for access to the Management Server

`password`

Password for access to the Management Server

List Management Servers

```
ncli> managementserver { list  | ls }[ name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Management Server

Returns a list of information for management servers which are used for managing the cluster.

```
ncli> managementserver { list-management-server-info }
```

Required arguments

None

Create and register a management server extension for Nutanix.

```
ncli> managementserver { register } ip-address="ip_address" port="port" admin-
username="admin_username" admin-password="admin_password"
```

Required arguments

`ip-address`

Address of the management server.

`port`

Port of the management server.

`admin-username`

Administrator username of the management server.

`admin-password`

Administrator password of the management server.

Delete a Management Server

```
ncli> managementserver { remove  | rm } name="name"
```

Required arguments

`name`

Name of the Management Server

Unregister the management server extension for Nutanix.

```
ncli> managementserver { unregister }ip-address="ip_address" port="port" admin-
username="admin_username" admin-password="admin_password"
```

Required arguments

`ip-address`

Address of the management server.

`port`

Port of the management server.

`admin-username`

Administrator username of the management server.

`admin-password`

Administrator password of the management server.

`multicluster`: Multicluster

Description A Nutanix Management Console to manage multiple clusters

Alias

Operations

- Add to multicluster : `add-to-multicluster`
- Get cluster state : `get-cluster-state`

Add to multicluster

```
ncli> multicluster { add-to-multicluster }external-ip-address-or-svm-
ips="external_ip_address_or_svm_ips" username="username" password="password"
```

Required arguments

`external-ip-address-or-svm-ips`

External IP address or list of SVM IP addresses

`username`

username

`password`

password

Get cluster state

```
ncli> multicluster { get-cluster-state }[ cluster-id="cluster_id" ]
```

Required arguments

None

Optional arguments

`cluster-id`

Id of the cluster

`network`: Network

Description Network specific commands

Alias `net`

Operations

- Add an SNMP Profile : `add-snmp-profile`
- Add a switch configuration : `add-switch-config`
- Delete an SNMP Profile : `delete-snmp-profile`
- Delete a switch configuration : `delete-switch-config`
- Update an SNMP Profile : `edit-snmp-profile`
- Update a switch collector configuration : `edit-switch-collector-config`
- Update a switch configuration : `edit-switch-config`
- Get switch collector configuration : `get-switch-collector-config`
- List all host interfaces : `list-host-nics`
- List all host virtual interfaces : `list-host-vnics`
- List all SNMP Profiles : `list-snmp-profile`
- List all switches information : `list-switch`
- List all switch interfaces information : `list-switch-ports`
- List all VM virtual Nics : `list-vm-nics`

Add an SNMP Profile

```
ncli> network { add-snmp-profile } name="name" [ uuid="uuid" ][ default="default" ][ version="version" ][ community="community" ][ level="level" ][ username="username" ][ auth-type="auth_type" ][ auth-key="auth_key" ][ priv-type="priv_type" ][ priv-key="priv_key" ]
```

Required arguments

`name`

SNMP profile name

Optional arguments

`uuid`

SNMP profile UUID

`default`

Used as Default SNMP profile

`version`

SNMP version [snmpv2c, snmpv3]

`community`

SNMP community string. Used for snmpv2c only. If not set, default to "public"

`level`

SNMP security level [NoAuthNoPriv, AuthnoPriv, AuthPriv].

`username`

SNMP username

`auth-type`

SNMP authentication type [SHA]

`auth-key`

SNMP authentication key

`priv-type`

SNMP encryption type [AES or DES]

`priv-key`

SNMP encryption key

Add a switch configuration

```
ncli> network { add-switch-config } switch-address="switch_address" [
switch-id="switch_id" ][ snmp-profile-name="snmp_profile_name" ][ host-
addresses="host_addresses" ][ version="version" ][ community="community" ][ level="level"
][ username="username" ][ auth-type="auth_type" ][ auth-key="auth_key" ][ priv-
type="priv_type" ][ priv-key="priv_key" ]
```

Required arguments

`switch-address`
: Switch address

Optional arguments

`switch-id`
: Switch ID

`snmp-profile-name`
: SNMP profile name to apply on switch config

`host-addresses`
: List of comma-separated Host addresses which is connected to this switch

`version`
: SNMP version [snmpv2c, snmpv3]

`community`
: SNMP community string. Used for snmpv2c only. If not set, default to "public"

`level`
: SNMP security level [NoAuthNoPriv, AuthnoPriv, AuthPriv].

`username`
: SNMP username

`auth-type`
: SNMP authentication type [SHA]

`auth-key`
: SNMP authentication key

`priv-type`
: SNMP encryption type [AES or DES]

`priv-key`
: SNMP encryption key

Delete an SNMP Profile

```
ncli> network { delete-snmp-profile } uuid="uuid"
```

Required arguments

`uuid`
: UUID of the SNMP Profile

Delete a switch configuration

```
ncli> network { delete-switch-config } switch-id="switch_id"
```

Required arguments

`switch-id`
: ID of the switch

Update an SNMP Profile

```
ncli> network { edit-snmp-profile }uuid="uuid"[ name="name" ][ default="default"
][ version="version" ][ community="community" ][ level="level" ][ username="username"
][ auth-type="auth_type" ][ auth-key="auth_key" ][ priv-type="priv_type" ][ priv-
key="priv_key" ]
```

Required arguments

`uuid`

SNMP profile UUID

Optional arguments

`name`

SNMP profile name

`default`

Used as Default SNMP profile

`version`

SNMP version [snmpv2c, snmpv3]

`community`

SNMP community string. Used for snmpv2c only. If not set, default to "public"

`level`

SNMP security level [NoAuthNoPriv, AuthnoPriv, AuthPriv].

`username`

SNMP username

`auth-type`

SNMP authentication type [SHA]

`auth-key`

SNMP authentication key

`priv-type`

SNMP encryption type [AES or DES]

`priv-key`

SNMP encryption key

Update a switch collector configuration

```
ncli> network { edit-switch-collector-config }[ enabled="{ true | false }" ][
schedule-interval-in-secs="schedule_interval_in_secs" ][ schedule-discovery-in-
secs="schedule_discovery_in_secs" ]
```

Required arguments

None

Optional arguments

`enabled`

Enable/Disable switch collector

`schedule-interval-in-secs`

Scheduled interval to collect the switch stats (in seconds)

`schedule-discovery-in-secs`

Switch discovery interval (in seconds)

Update a switch configuration

```
ncli> network { edit-switch-config } switch-id="switch_id" [ switch-
address="switch_address" ][ snmp-profile-name="snmp_profile_name" ][ host-
addresses="host_addresses" ][ version="version" ][ community="community" ][ level="level"
][ username="username" ][ auth-type="auth_type" ][ auth-key="auth_key" ][ priv-
type="priv_type" ][ priv-key="priv_key" ]
```

Required arguments

`switch-id`

Switch ID

Optional arguments

`switch-address`

Switch address

`snmp-profile-name`

SNMP profile name to apply on switch config

`host-addresses`

List of comma-separated Host addresses which is connected to this switch

`version`

SNMP version [snmpv2c, snmpv3]

`community`

SNMP community string. Used for snmpv2c only. If not set, default to "public"

`level`

SNMP security level [NoAuthNoPriv, AuthnoPriv, AuthPriv].

`username`

SNMP username

`auth-type`

SNMP authentication type [SHA]

`auth-key`

SNMP authentication key

`priv-type`

SNMP encryption type [AES or DES]

`priv-key`

SNMP encryption key

Get switch collector configuration

```
ncli> network { get-switch-collector-config }
```

Required arguments

None

List all host interfaces

```
ncli> network { list-host-nics } host-id="host_id" [ pnic-id="pnic_id" ]
```

Required arguments

`host-id`

ID of the Host

Optional arguments

`pnic-id`

ID of Host Nic

List all host virtual interfaces

```
ncli> network { list-host-vnics } host-id="host_id" [ host-vnic-id="host_vnic_id" ]
```

Required arguments

`host-id`

ID of the Host

Optional arguments

`host-vnic-id`

ID of Host Virtual Nic

List all SNMP Profiles

```
ncli> network { list-snmp-profile } [ uuid="uuid" ]
```

Required arguments

None

Optional arguments

`uuid`

UUID of the SNMP Profile

List all switches information

```
ncli> network { list-switch } [ switch-id="switch_id" ] [ config-only="config_only" ]
```

Required arguments

None

Optional arguments

`switch-id`

ID of the switch

`config-only`

Config-only flag

List all switch interfaces information

```
ncli> network { list-switch-ports } switch-id="switch_id" [ port-id="port_id" ]
```

Required arguments

`switch-id`

ID of the switch

Optional arguments

`port-id`

port ID of the switch interface

List all VM virtual Nics

```
ncli> network { list-vm-nics } vm-id="vm_id" [ vnic-id="vnic_id" ]
```

Required arguments

`vm-id`

ID of the VM

Optional arguments

`vnic-id`

ID of VM Nic

`nutanix-guest-tools`: Nutanix Guest Tools

Description Admin commands for Nutanix Guest Tools

Alias `ngt`

Operations

- Delete Nutanix Guest Tools : `delete`
- Disable Nutanix Guest Tools : `disable`
- Disable Applications in Nutanix Guest Tools : `disable-applications`
- Enable Nutanix Guest Tools : `enable`
- Enable Applications in Nutanix Guest Tools : `enable-applications`
- Get Nutanix Guest Tools : `get`
- List Nutanix Guest Tools : `list`
- List applications supported by Nutanix Guest Tools : `list-applications`
- Mount Nutanix Guest Tools : `mount`
- Unmount Nutanix Guest Tools : `unmount`

Delete Nutanix Guest Tools

```
ncli> nutanix-guest-tools { delete } vm-id="vm_id"
```

Required arguments

`vm-id`

ID of the Virtual Machine

Disable Nutanix Guest Tools

```
ncli> nutanix-guest-tools { disable } vm-id="vm_id"
```

Required arguments

`vm-id`

ID of the Virtual Machine

Disable Applications in Nutanix Guest Tools

```
ncli> nutanix-guest-tools { disable-applications } vm-id="vm_id" application-
names="application_names"
```

Required arguments

`vm-id`

ID of the Virtual Machine

`application-names`

Comma-separated list of application names in Nutanix Guest Tools

Enable Nutanix Guest Tools

```
ncli> nutanix-guest-tools { enable } vm-id="vm_id" [ application-
names="application_names" ]
```

Required arguments

`vm-id`

ID of the Virtual Machine

Optional arguments

`application-names`

Comma-separated list of application names in Nutanix Guest Tools

Enable Applications in Nutanix Guest Tools

```
ncli> nutanix-guest-tools { enable-applications } vm-id="vm_id" application-names="application_names"
```

Required arguments

`vm-id`

ID of the Virtual Machine

`application-names`

Comma-separated list of application names in Nutanix Guest Tools

Get Nutanix Guest Tools

```
ncli> nutanix-guest-tools { get } vm-id="vm_id"
```

Required arguments

`vm-id`

Id of Virtual machine

List Nutanix Guest Tools

```
ncli> nutanix-guest-tools { list }[ application-names="application_names" ][ vm-names="vm_names" ]
```

Required arguments

None

Optional arguments

`application-names`

Comma separated list of application names. If no application name given all entities will be returned.

`vm-names`

Comma separated list of vm names.

List applications supported by Nutanix Guest Tools

```
ncli> nutanix-guest-tools { list-applications }
```

Required arguments

None

Mount Nutanix Guest Tools

```
ncli> nutanix-guest-tools { mount } vm-id="vm_id"
```

Required arguments

`vm-id`

ID of the Virtual Machine

Unmount Nutanix Guest Tools

```
ncli> nutanix-guest-tools { unmount } vm-id="vm_id"
```

Required arguments

`vm-id`

ID of the Virtual Machine

`progress-monitor`: Progress Monitor

Description Monitor progress of long running tasks

Alias

Operations

- Remove a specific Progress Monitor : `delete` | `remove`
- List all or specific Progress Monitors : `list` | `ls`

Remove a specific Progress Monitor

```
ncli> progress-monitor { delete  | remove } id="id"
```

Required arguments

`id`

Id of the progress monitor to be deleted

List all or specific Progress Monitors

```
ncli> progress-monitor { list  | ls }[ operation="operation" ][ entity="entity" ][
entity-id="entity_id" ][ verbose="verbose" ][ has-fanout-details="has_fanout_details" ]
```

Required arguments

None

Optional arguments

`operation`

Operation to be monitored

`entity`

Entity (for example: VM, VDisk etc.)

`entity-id`

ID of Entity to be monitored

`verbose`

Set verbose to get detailed information

Default: false

`has-fanout-details`

Returns Fanout tasks details

Default: false

`protection-domain`: Protection domain

Description A protection domain to be used for Data Protection

Alias `pd`

Operations

- Abort replications of a Protection domain : `abort-replication` | `abort-repl`
- Mark Protection domain as active : `activate`
- Add a daily snapshot schedule to a Protection domain : `add-daily-schedule`
- Add an hourly snapshot schedule to a Protection domain : `add-hourly-schedule`
- Add an minutely snapshot schedule to a Protection domain : `add-minutely-schedule`
- Add a monthly snapshot schedule to a Protection domain : `add-monthly-schedule`
- Create a new out of band snapshot schedule in a Protection domain to take a snapshot at a specified time : `add-one-time-snapshot` | `create-one-time-snapshot`
- Add a weekly snapshot schedule to a Protection domain : `add-weekly-schedule`
- Clear retention policies for snapshot schedules of a Protection domain : `clear-retention-policy`
- Remove all snapshot schedules from a Protection domain : `clear-schedules`
- Create a new active Protection domain : `create` | `add`
- List all Protection domains : `list` | `ls`
- Get the status of replication in a Protection domain : `list-replication-status` | `ls-repl-status`
- List Snapshots of a Protection domain : `list-snapshots` | `ls-snaps`
- List all pending actions for Protection domains : `ls-pending-actions`
- List out of band snapshot schedules of Protection domains : `ls-pending-one-time-snapshots`
- List the snapshot schedules of a Protection domain : `ls-schedules`
- Disable metro availability for a specific Protection domain : `metro-avail-disable`
- Enable metro availability for a specific Protection domain : `metro-avail-enable`
- Mark Protection domain as inactive and failover to the specified Remote Site : `migrate`
- Pause replications of a Protection domain : `pause-replication` | `pause-repl`
- Promote to active metro availability role for a Protection domain : `promote-to-active`
- Add Virtual Machines and NFS files to a Protection domain : `protect`
- Mark a Protection domain for removal : `remove` | `rm`
- Remove a snapshot schedule from a Protection domain : `remove-from-schedules`
- Restore Virtual Machines and/or NFS files in a Snapshot : `restore-snapshot`
- Resume previously paused replications of a Protection domain : `resume-replication` | `resume-repl`
- Resume all schedules in the Protection domain : `resume-schedules`
- Retain snapshots forever of a Protection domain : `retain-snapshot` | `retain-snap`
- Remove out of band snapshot schedules from a Protection domain : `rm-one-time-schedules`
- Remove snapshots of a Protection domain : `rm-snapshot` | `rm-snap`
- Rollback an active Protection domain to a snapshot : `rollback-pd`
- Set retention policies for snapshot schedules of a Protection domain : `set-retention-policy`
- The data protection status of the Protection domain : `status`
- Suspend all schedules in the Protection domain : `suspend-schedules`
- Mark Virtual Machines and NFS files for removal from a given Protection domain : `unprotect`
- Change failure handling mode for a protection domain : `update-failure-handling`

Abort replications of a Protection domain

```
ncli> protection-domain { abort-replication  | abort-repl } name="name"
replication-ids="replication_ids"
```

Required arguments

`name`

Name of the Protection domain

`replication-ids`

List of comma-separated identifier of replications

Mark Protection domain as active

```
ncli> protection-domain { activate } name="name" [ snap-id="snap_id" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`snap-id`

Id of the Snapshot to be restored while making Protection domain as active. By default, latest snapshot is restored.

Default: -1

Add a daily snapshot schedule to a Protection domain

```
ncli> protection-domain { add-daily-schedule } name="name" [ start-
time="start_time" ][ end-time="end_time" ][ every-nth-day="every_nth_day" ][ local-
retention="local_retention" ][ remote-retention="remote_retention" ][ app-consistent-
snapshots="app_consistent_snapshots" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`start-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become active. If not specified, schedule will become active immediately

`end-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become inactive. If not specified, schedule will be active indefinitely

`every-nth-day`

Repeat the daily schedule every nth day

Default: 1

`local-retention`

Maximum number of snapshots to retain locally

Default: 1

`remote-retention`

Comma-separated entries in the form of <remote_site_name>:<quantity> to specify the remote sites' retention policies for the schedule.

`app-consistent-snapshots`

Whether Consistency group created for Virtual Machine performs application consistent snapshots. Such special Consistency group can contain one and only one Virtual Machine

Default: false

Add an hourly snapshot schedule to a Protection domain

```
ncli> protection-domain { add-hourly-schedule }name="name"[ start-
time="start_time" ][ end-time="end_time" ][ every-nth-hour="every_nth_hour" ][ local-
retention="local_retention" ][ remote-retention="remote_retention" ][ app-consistent-
snapshots="app_consistent_snapshots" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`start-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become active. If not specified, schedule will become active immediately

`end-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become inactive. If not specified, schedule will be active indefinitely

`every-nth-hour`

Repeat the hourly schedule every nth hour

Default: 1

`local-retention`

Maximum number of snapshots to retain locally

Default: 1

`remote-retention`

Comma-separated entries in the form of <remote_site_name>:<quantity> to specify the remote sites' retention policies for the schedule.

`app-consistent-snapshots`

Whether Consistency group created for Virtual Machine performs application consistent snapshots. Such special Consistency group can contain one and only one Virtual Machine

Default: false

Add an minutely snapshot schedule to a Protection domain

```
ncli> protection-domain { add-minutely-schedule }name="name"[ start-
time="start_time" ][ end-time="end_time" ][ every-nth-minute="every_nth_minute" ][ local-
retention="local_retention" ][ remote-retention="remote_retention" ][ local-retention-
type="local_retention_type" ][ remote-retention-type="remote_retention_type" ][ app-
consistent-snapshots="app_consistent_snapshots" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`start-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become active. If not specified, schedule will become active immediately

`end-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become inactive. If not specified, schedule will be active indefinitely

`every-nth-minute`

Repeat the minutely schedule every nth minute

Default: 1

`local-retention`

Maximum number of snapshots to retain or days/weeks/months a snapshot is to be retained locally.

Default: 1

`remote-retention`

Comma-separated entries in the form of <remote_site_name>:<quantity/period> to specify the remote sites' retention policies for the schedule. Quantity corresponds to maximum number of snapshots to retain and period is number of days/weeks/months for which a snapshot is to be retained remotely. Default value is 1 for a remote site

`local-retention-type`

Type of Local Retention (DAYS | WEEKS | MONTHS | NUM_SNAPSHOTS)

Default: NUM_SNAPSHOTS

`remote-retention-type`

Comma-separated entries in the form of <remote_site_name>:<retention_type> to specify the remote sites' retention types for the schedule.Type of Remote Retention can be anything from (DAYS | WEEKS | MONTHS | NUM_SNAPSHOTS)

`app-consistent-snapshots`

Whether Consistency group created for Virtual Machine performs application consistent snapshots. Such special Consistency group can contain one and only one Virtual Machine

Default: false

Add a monthly snapshot schedule to a Protection domain

```
ncli> protection-domain { add-monthly-schedule } name="name" [ days-of-
month="days_of_month" ][ start-time="start_time" ][ end-time="end_time" ][ local-
retention="local_retention" ][ remote-retention="remote_retention" ][ app-consistent-
snapshots="app_consistent_snapshots" ][ timezone="timezone" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`days-of-month`

Comma-separated day of month values for the schedule: 1-31

`start-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become active. If not specified, schedule will become active immediately

`end-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become inactive. If not specified, schedule will be active indefinitely

`local-retention`

Maximum number of snapshots to retain locally

Default: 1

`remote-retention`

Comma-separated entries in the form of <remote_site_name>:<quantity> to specify the remote sites' retention policies for the schedule.

`app-consistent-snapshots`

Whether Consistency group created for Virtual Machine performs application consistent snapshots. Such special Consistency group can contain one and only one Virtual Machine

Default: false

`timezone`

Timezone of the time values specified. (e.g: PST, America/Los_Angeles, GMT-8:00)

Default: GMT

Create a new out of band snapshot schedule in a Protection domain to take a snapshot at a specified time

```
ncli> protection-domain { add-one-time-snapshot  | create-one-time-snapshot
} name="name" [ snap-time="snap_time" ][ remote-sites="remote_sites" ][ retention-
time="retention_time" ][ app-consistent-snapshots="app_consistent_snapshots" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`snap-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which snapshot is to be taken. If not specified, snapshot will be taken immediately

`remote-sites`

Comma-separated list of Remote Site to which snapshots are replicated. If not specified, remote replication is not performed

`retention-time`

Number of seconds to retain the snapshot. Aged snapshots will be garbage collected. By default, snapshot is retained forever

`app-consistent-snapshots`

Whether Consistency group created for Virtual Machine performs application consistent snapshots. Such special Consistency group can contain one and only one Virtual Machine

Default: false

Add a weekly snapshot schedule to a Protection domain

```
ncli> protection-domain { add-weekly-schedule } name="name" [ days-of-
week="days_of_week" ][ start-time="start_time" ][ end-time="end_time" ][ local-
retention="local_retention" ][ remote-retention="remote_retention" ][ app-consistent-
snapshots="app_consistent_snapshots" ][ timezone="timezone" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`days-of-week`

Comma-separated day of week values for the schedule: 1-7 (starts with Sunday(1)), or (sun | mon | tue | wed | thu | fri | sat)

`start-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become active. If not specified, schedule will become active immediately

`end-time`

Specify time in format MM/dd/yyyy [HH:mm:ss [z]] at which the schedule become inactive. If not specified, schedule will be active indefinitely

`local-retention`

Maximum number of snapshots to retain locally

Default: 1

`remote-retention`

Comma-separated entries in the form of <remote_site_name>:<quantity> to specify the remote sites' retention policies for the schedule.

`app-consistent-snapshots`

Whether Consistency group created for Virtual Machine performs application consistent snapshots. Such special Consistency group can contain one and only one Virtual Machine

Default: false

`timezone`

Timezone of the time values specified. (e.g: PST, America/Los_Angeles, GMT-8:00)

Default: GMT

Clear retention policies for snapshot schedules of a Protection domain

```
ncli> protection-domain { clear-retention-policy } name="name" id="id"
```

Required arguments

`name`

Name of the Protection domain

`id`

ID of a cron schedule of a Protection domain

Remove all snapshot schedules from a Protection domain

```
ncli> protection-domain { clear-schedules } name="name"
```

Required arguments

`name`

Name of the Protection domain

Create a new active Protection domain

```
ncli> protection-domain { create  | add } name="name"
```

Required arguments

`name`

Name of the Protection domain

List all Protection domains

```
ncli> protection-domain { list  | ls }[ name="name" ][ metro-avail="metro_avail"
][ vstore-name="vstore_name" ][ remote-site="remote_site" ][ forward-to-remote-
site="forward_to_remote_site" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Protection domain

`metro-avail`

Whether the Protection domain is stretched or not

`vstore-name`

Name of a VStore

`remote-site`

Name of the Remote Site

`forward-to-remote-site`

Name of the Remote Site to get the PD details

Get the status of replication in a Protection domain

```
ncli> protection-domain {  list-replication-status  | ls-repl-status }[
name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Protection domain

List Snapshots of a Protection domain

```
ncli> protection-domain {  list-snapshots  | ls-snaps }[ name="name" ][ snap-
id="snap_id" ][ state="state" ][ oob-schedule-id="oob_schedule_id" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Protection domain

`snap-id`

Id of the Snapshot

`state`

State of the Snapshot

`oob-schedule-id`

Id of the out of band schedule that created the Snapshot

List all pending actions for Protection domains

```
ncli> protection-domain {  ls-pending-actions  }[ name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Protection domain

List out of band snapshot schedules of Protection domains

```
ncli> protection-domain {  ls-pending-one-time-snapshots  }[ name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Protection domain

List the snapshot schedules of a Protection domain

```
ncli> protection-domain { ls-schedules } name="name" [ timezone-offset="timezone_offset" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`timezone-offset`

Timezone offset in seconds w.r.t GMT. (e.g: GMT+1:00 => 3600)

Disable metro availability for a specific Protection domain

```
ncli> protection-domain { metro-avail-disable } name="name" [ skip-remote-check="{ true | false }" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`skip-remote-check`

Skip checking remoteProtection domain

Default: false

Enable metro availability for a specific Protection domain

```
ncli> protection-domain { metro-avail-enable } name="name" [ ctr-name="ctr_name" ][ remote-site="remote_site" ][ re-enable="{ true | false }" ][ skip-remote-check="{ true | false }" ][ failure-handling="failure_handling" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`ctr-name`

Name of the Storage Container

`remote-site`

Name of the Remote Site

`re-enable`

Resynchronize operation.

Default: false

`skip-remote-check`

Skip checking remoteProtection domain

Default: false

`failure-handling`

Failure handling mode (Modes : [Automatic, Manual, Witness])

Mark Protection domain as inactive and failover to the specified Remote Site

```
ncli> protection-domain { migrate } name="name" remote-site="remote_site" [ skip-vm-mobility-check="{ true | false }" ]
```

Required arguments

`name`

Name of the Protection domain

`remote-site`

Remote Site to be used for planned failover

Optional arguments

`skip-vm-mobility-check`

Skip the vm mobility check while migrating a Protection domain

Pause replications of a Protection domain

```
ncli> protection-domain { pause-replication  | pause-repl } name="name" replication-ids="replication_ids"
```

Required arguments

`name`

Name of the Protection domain

`replication-ids`

List of comma-separated identifier of replications

Promote to active metro availability role for a Protection domain

```
ncli> protection-domain { promote-to-active } name="name" [ skip-remote-check="{ true | false }" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`skip-remote-check`

Skip checking remoteProtection domain

Default: false

Add Virtual Machines and NFS files to a Protection domain

```
ncli> protection-domain { protect } name="name" [ vm-names="vm_names" ][ vm-ids="vm_ids" ][ volume-group-uuids="volume_group_uuids" ][ host-id="host_id" ][ files="files" ][ cg-name="cg_name" ][ ignore-duplicates="ignore_duplicates" ][ app-consistent-snapshots="app_consistent_snapshots" ][ auto-protect-related-entities="auto_protect_related_entities" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`vm-names`

Comma-separated list of Virtual Machine names to be added in Protection domain

`vm-ids`

Comma-separated list of Virtual Machine ids to be added in Protection domain

`volume-group-uuids`

UUIDs of the Volume Groups

`host-id`

Add all Virtual Machines in a Physical Host to a Protection domain

`files`

Comma-separated list of NFS files to be added in Protection domain

`cg-name`

Name of the Consistency group to which Virtual Machines are added. If not specified, each Virtual Machines is added to a Consistency group which has same name as the Virtual Machine name

`ignore-duplicates`

Whether to ignore if any of the specified Virtual Machines already exist in the specified Protection domain

Default: true

`app-consistent-snapshots`

Whether Consistency group created for Virtual Machine performs application consistent snapshots. Such special Consistency group can contain one and only one Virtual Machine

Default: false

`auto-protect-related-entities`

Whether the related entities need to be auto-protected

Default: false

Mark a Protection domain for removal. Protection domain will be removed from the appliance when all outstanding operations on it are cancelled

```
ncli> protection-domain { remove | rm } name="name" [ skip-remote-check="{ true | false }" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`skip-remote-check`

Skip checking remoteProtection domain

Default: false

Remove a snapshot schedule from a Protection domain

```
ncli> protection-domain { remove-from-schedules } name="name" id="id"
```

Required arguments

`name`

Name of the Protection domain

`id`

ID of a cron schedule of a Protection domain

Restore Virtual Machines and/or NFS files in a Snapshot

```
ncli> protection-domain { restore-snapshot } name="name" [ vm-names="vm_names"
][ vm-uuids="vm_uuids" ][ files="files" ][ volume-group-uuids="volume_group_uuids"
][ path-prefix="path_prefix" ][ vm-name-prefix="vm_name_prefix" ][ vg-name-
prefix="vg_name_prefix" ][ snap-id="snap_id" ]
```

Required arguments

`name`

Name of the Protection Domain

Optional arguments

`vm-names`

Names of VM to be restored.

`vm-uuids`

Uuids of VM to be restored.

`files`

Names of NFS files to be restored

`volume-group-uuids`

UUIDs of Volume Groups to be restored

`path-prefix`

Path prefix to be applied for cloning VMs. It is recommended to not use path_prefix while restoring a virtual machine or volume group.

`vm-name-prefix`

Name prefix to be applied for cloning VMs.

`vg-name-prefix`

Name prefix to be applied for cloning VGs.

`snap-id`

Id of the snapshot to restore entities from.

Resume previously paused replications of a Protection domain

```
ncli> protection-domain { resume-replication | resume-repl } name="name"
replication-ids="replication_ids"
```

Required arguments

`name`

Name of the Protection domain

`replication-ids`

List of comma-separated identifier of replications

Resume all schedules in the Protection domain

```
ncli> protection-domain { resume-schedules } name="name"
```

Required arguments

`name`

Name of the Protection Domain

Retain snapshots forever of a Protection domain

```
ncli> protection-domain { retain-snapshot | retain-snap } name="name" snap-
ids="snap_ids"
```

Required arguments

`name`

Name of the Protection domain

`snap-ids`

List of comma-separated identifier of Snapshots

Remove out of band snapshot schedules from a Protection domain

```
ncli> protection-domain { rm-one-time-schedules } name="name" schedule-
ids="schedule_ids"
```

Required arguments

`name`

Name of the Protection domain

`schedule-ids`

List of comma-separated identifier of Out of band schedules

Remove snapshots of a Protection domain

```
ncli> protection-domain { rm-snapshot  | rm-snap } name="name" [ snap-
ids="snap_ids" ][ clear-all="clear_all" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`snap-ids`

List of comma-separated identifier of Snapshots

`clear-all`

Remove all entities of a particular type

Default: false

Rollback an active Protection domain to a snapshot

```
ncli> protection-domain { rollback-pd } name="name" snap-id="snap_id"
```

Required arguments

`name`

Name of the Protection domain

`snap-id`

Snapshot to which Protection domain is to be rollbacked

Set retention policies for snapshot schedules of a Protection domain

```
ncli> protection-domain { set-retention-policy } name="name" id="id" [ local-
retention="local_retention" ][ remote-retention="remote_retention" ][ local-retention-
type="local_retention_type" ][ remote-retention-type="remote_retention_type" ]
```

Required arguments

`name`

Name of the Protection domain

`id`

ID of a cron schedule of a Protection domain

Optional arguments

`local-retention`

Maximum number of snapshots to retain or days/weeks/months a snapshot is to be retained locally.

`remote-retention`

Comma-separated entries in the form of <remote_site_name>:<quantity/period> to specify the remote sites' retention policies for the schedule. Quantity corresponds to maximum number of snapshots to retain and period is number of days/weeks/months for which a snapshot is to be retained remotely. Default value is 1 for a remote site

`local-retention-type`

Type of Local Retention (DAYS | WEEKS | MONTHS | NUM_SNAPSHOTS)

Default: NUM_SNAPSHOTS

`remote-retention-type`

Comma-separated entries in the form of <remote_site_name>:<retention_type> to specify the remote sites' retention types for the schedule.Type of Remote Retention can be anything from (DAYS | WEEKS | MONTHS | NUM_SNAPSHOTS)

The data protection status of the Protection domain

```
ncli> protection-domain { status }
```

Required arguments

None

Suspend all schedules in the Protection domain

```
ncli> protection-domain { suspend-schedules } name="name"
```

Required arguments

`name`

Name of the Protection Domain

Mark Virtual Machines and NFS files for removal from a given Protection domain. They will be removed when all outstanding operations on them are completed/cancelled

```
ncli> protection-domain { unprotect } name="name" [ files="files" ][ vm-
names="vm_names" ][ vm-ids="vm_ids" ][ volume-group-uuids="volume_group_uuids" ]
```

Required arguments

`name`

Name of the Protection domain

Optional arguments

`files`

Comma-separated list of NFS files to be removed from Protection domain

`vm-names`

Comma-separated list of Virtual Machine name to be removed from Protection domain

`vm-ids`

Comma-separated list of Virtual Machine name to be removed from Protection domain

`volume-group-uuids`

UUIDs of the Volume Groups

Change failure handling mode for a protection domain

```
ncli> protection-domain { update-failure-handling } name="name" failure-
handling="failure_handling"
```

Required arguments

`name`

Name of the Protection domain

`failure-handling`

Failure handling mode (Modes : [Automatic, Manual, Witness])

`pulse-config`: Pulse Configuration

Description Configuration information used for Pulse setup

Alias

Operations
- Update Pulse Configuration : `edit` | `update`
- List Pulse Configuration : `list` | `ls`

Update Pulse Configuration

```
ncli> pulse-config { edit  | update }[ enable="{ true | false }" ][ enable-default-nutanix-email="{ true | false }" ][ support-verbosity-level="support_verbosity_level" ][ email-contacts="email_contacts" ]
```

Required arguments

None

Optional arguments

`enable`

Enable Pulse emails

`enable-default-nutanix-email`

Enable Pulse default Nutanix email

`support-verbosity-level`

List Pulse Configuration

`email-contacts`

Comma-separated list of emails to be used while sending Pulse info. Set to '-' to clear all the existing emails.

List Pulse Configuration

```
ncli> pulse-config { list  | ls }
```

Required arguments

None

`rackable-unit`: Rackable unit

Description A rackable unit

Alias `ru`

Operations
- Edit a Rackable unit : `edit` | `update`
- List Rackable unit : `list` | `ls`
- Remove a Rackable unit : `remove` | `rm`

Edit a Rackable unit

```
ncli> rackable-unit { edit  | update } id="id" location="location"
```

Required arguments

`id`

Id of the Rackable unit

`location`

Location of the Rackable unit

List Rackable unit

```
ncli> rackable-unit { list  | ls}
```

Required arguments

None

Remove a Rackable unit

```
ncli> rackable-unit { remove  | rm}id="id"
```

Required arguments

`id`

Id of the Rackable unit

`remote-site`: Remote Site

Description A remote cluster to be used for replicating data

Alias `rs`

Operations

- Add bandwidth policy : `add-bandwidth-schedule`
- Add a network mapping : `add-network-mapping`
- Create a new Remote Site : `create` | `add`
- Edit a Remote Site : `edit` | `update`
- List Remote Sites : `list` | `ls`
- List schedules for bandwidth throttling : `list-bandwidth-schedules`
- List network mapping(s) corresponding to a remote site : `list-network-mapping`
- List networks corresponding to the local cluster or a remote site : `list-networks`
- List Snapshots of a Remote Site : `list-snapshots` | `ls-snaps`
- Mark a Remote Site for removal : `remove` | `rm`
- Remove a bandwidth schedule : `remove-bandwidth-schedule`
- Remove a network mapping : `remove-network-mapping`
- Download a snapshot from a Remote Site : `retrieve-snapshot`
- Remove snapshots of a Protection domain : `rm-snapshot` | `rm-snap`

Add bandwidth policy

```
ncli> remote-site { add-bandwidth-schedule }remote-site-name="remote_site_name"
[ days-of-week="days_of_week" ][ start-time="start_time" ][ end-time="end_time" ][ max-
bandwidth="max_bandwidth" ][ default-bandwidth="default_bandwidth" ]
```

Required arguments

`remote-site-name`

Name of the Remote Site

Optional arguments

`days-of-week`

Comma-separated day of week values for the policy: 1-7 (starts with Sunday(1)), or (sun | mon | tue | wed | thu | fri | sat)

`start-time`

Specify time in format [hh:mm:ss aa] at which this bandwidth policy will start on a particular day.

`end-time`

Specify time in format [hh:mm:ss aa] at which this bandwidth policy will end on a particular day.

`max-bandwidth`

Maximum bandwidth for policy in kilobytes per second.

`default-bandwidth`

Maximum bandwidth (in kilobytes per sec) to be used while replicating to the remote site. If not specified, restriction is not placed on maximum bandwidth used by replication

Add a network mapping

```
ncli> remote-site { add-network-mapping } remote-site-name="remote_site_name"
src-network="src_network" dest-network="dest_network"
```

Required arguments

`remote-site-name`

Name of the Remote Site

`src-network`

Name of the source network

`dest-network`

Name of the destination network

Create a new Remote Site

```
ncli> remote-site { create  | add } name="name" address-list="address_list" [
capabilities="capabilities" ][ enable-proxy="{ true | false }" ][ enable-ssh-tunnel="{ true |
false }" ][ enable-compression="{ true | false }" ][ vstore-map="vstore_map" ]
```

Required arguments

`name`

Name of the Remote Site

`address-list`

List of comma-separated addresses of the remote site. All addresses should be of format <ip> or of format <ip:port>. Default port is used if the port number is not specified

Optional arguments

`capabilities`

Capabilities of the Remote Site; comma-separated values of (backup | disaster_recovery). Backup sites only allow data backup, whereas disaster recovery allows the user to run VMs in the event of a disaster.

`enable-proxy`

Boolean parameter to indicate whether the addresses specified in address-list can be used as a proxy to communicate with other Nutanix components on the remote site

`enable-ssh-tunnel`

Boolean parameter to indicate whether the addresses specified in address-list can be used as a SSH tunnel to communicate with other Nutanix components on the remote site. Enabling SSH tunnel mode will also enable proxy mode

`enable-compression`

Enable or disable compression of data during replication

`vstore-map`

By default, data from a local vStore is replicated to a identically named vStore in the remote site. To setup data replication from local vStore to remote vStore having different

names, provide comma-separated list of <local vStore>:<target vStore> mapping. Mapping is not required if the names of local and target vStore are same

Edit a Remote Site

```
ncli> remote-site { edit  | update } name="name" [ capabilities="capabilities"
][ enable-proxy="{ true | false }" ][ enable-ssh-tunnel="{ true | false }" ][ enable-
compression="{ true | false }" ][ address-add="address_add" ][ address-del="address_del"
][ address-list="address_list" ][ vstore-map-add="vstore_map_add" ][ vstore-map-
del="vstore_map_del" ][ enable-bandwidth-policy="{ true | false }" ]
```

Required arguments

`name`

Name of the Remote Site

Optional arguments

`capabilities`

Capabilities of the Remote Site; comma-separated values of (backup | disaster_recovery). Backup sites only allow data backup, whereas disaster recovery allows the user to run VMs in the event of a disaster.

`enable-proxy`

Boolean parameter to indicate whether the addresses specified in address-list can be used as a proxy to communicate with other Nutanix components on the remote site

`enable-ssh-tunnel`

Boolean parameter to indicate whether the addresses specified in address-list can be used as a SSH tunnel to communicate with other Nutanix components on the remote site. Enabling SSH tunnel mode will also enable proxy mode

`enable-compression`

Enable or disable compression of data during replication

`address-add`

Address to be included in the remote site address list. Address should be of the format <ip> or of the format <ip:port> and should conform to the format of the current remote site address list

`address-del`

Address to be removed from the remote site address list. Port number, if provided, is ignored

`address-list`

List of comma-separated addresses of the remote site. All addresses should be of format <ip> or of format <ip:port>. Default port is used if the port number is not specified

`vstore-map-add`

Entry of the form <local vStore>:<target vStore> to be included in the vStore replication map. If mapping for a local vStore already exists, mapping is updated with the the new target vStore

`vstore-map-del`

Entry of the form <local vStore>:<target vStore> to be removed from the vStore replication map

`enable-bandwidth-policy`

Enable or Disable bandwidth policy

List Remote Sites

```
ncli> remote-site {  list  | ls }[ name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Remote Site

List schedules for bandwidth throttling.

```
ncli> remote-site { list-bandwidth-schedules } name="name"
```

Required arguments

`name`

Name of the Remote Site

List network mapping(s) corresponding to a remote site

```
ncli> remote-site { list-network-mapping } remote-site-name="remote_site_name"
```

Required arguments

`remote-site-name`

Name of the Remote Site

List networks corresponding to the local cluster or a remote site. If remote-site-name is provided then networks corresponding to that remote site are returned else local cluster's networks are returned

```
ncli> remote-site { list-networks }[ remote-site-name="remote_site_name" ]
```

Required arguments

None

Optional arguments

`remote-site-name`

Name of the Remote Site

List Snapshots of a Remote Site

```
ncli> remote-site { list-snapshots  | ls-snaps }[ name="name" ][ pd-
name="pd_name" ][ snap-id="snap_id" ][ state="state" ][ oob-schedule-
id="oob_schedule_id" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Remote Site

`pd-name`

Name of the Protection domain

`snap-id`

Id of the Snapshot

`state`

State of the Snapshot

`oob-schedule-id`

Id of the out of band schedule that created the Snapshot

Mark a Remote Site for removal. Site will be removed from the appliance when all outstanding operations that are using the remote site are cancelled

```
ncli> remote-site { remove  | rm } name="name"
```

Required arguments

`name`

Name of the Remote Site

Remove a bandwidth schedule

```
ncli> remote-site { remove-bandwidth-schedule } name="name" [ schedule-
id="schedule_id" ][ remove-all-schedules="remove_all_schedules" ][ remove-default-
bandwidth-policy="remove_default_bandwidth_policy" ]
```

Required arguments

`name`

Name of the Remote Site

Optional arguments

`schedule-id`

Id of bandwidth schedule

`remove-all-schedules`

Remove all bandwidth schedules

Default: false

`remove-default-bandwidth-policy`

Remove default bandwidth policy

Default: false

Remove a network mapping

```
ncli> remote-site { remove-network-mapping } remote-site-name="remote_site_name"
src-network="src_network" dest-network="dest_network"
```

Required arguments

`remote-site-name`

Name of the Remote Site

`src-network`

Name of the source network

`dest-network`

Name of the destination network

Download a snapshot from a Remote Site

```
ncli> remote-site { retrieve-snapshot } name="name" pd-name="pd_name" snap-
id="snap_id"
```

Required arguments

`name`

Name of the Remote Site

`pd-name`

Name of the Protection domain

`snap-id`

Id of the Snapshot

Remove snapshots of a Protection domain

```
ncli> remote-site { rm-snapshot  | rm-snap } name="name" pd-name="pd_name" snap-ids="snap_ids"
```

Required arguments

`name`

Name of the Remote Site

`pd-name`

Name of the Protection domain

`snap-ids`

List of comma-separated identifier of Snapshots

`rsyslog-config`: RSyslog Configuration

Description Configuration information to send logs to remote servers

Alias

Operations

- Create RSyslog Server Module : `create-module` | `add-module`
- Create RSyslog Configuration : `create-server` | `add-server`
- Remove RSyslog Server Module : `delete-module` | `remove-module`
- Remove RSyslog Server : `delete-server` | `remove-server`
- Update RSyslog Server : `edit-server` | `update-server`
- Returns the status of RSyslog service : `get-status`
- List RSyslog Servers : `list` | `ls`
- List RSyslog Server Modules : `list-modules` | `ls-modules`
- List RSyslog Servers : `list-servers` | `ls-servers`
- Sets the status of RSyslog service : `set-status`

Create RSyslog Server Module

```
ncli> rsyslog-config { create-module  | add-module } server-name="server_name" module-name="module_name" level="level" [ include-monitor-logs="{ true | false }" ]
```

Required arguments

`server-name`

Log Server Name

`module-name`

Name of the RSyslog Server Module

`level`

Log level for RSyslog Server Module

Optional arguments

`include-monitor-logs`

Include monitor logs for the given RSyslog Server Module

Create RSyslog Configuration

```
ncli> rsyslog-config { create-server  | add-server } name="name" ip-address="ip_address" port="port" [ network-protocol="network_protocol" ][ relp-enabled="relp_enabled" ]
```

Required arguments

`name`

Name of the RSyslog Server

`ip-address`

Ip address of the RSyslog Server

`port`

port number

Optional arguments

`network-protocol`

Protocol for RSyslog server configuration

`relp-enabled`

Reliable Event Logging Protocol option

Remove RSyslog Server Module

```
ncli> rsyslog-config { delete-module  | remove-module } server-name="server_name"
module-name="module_name"
```

Required arguments

`server-name`

Name of the RSyslog Server

`module-name`

Name of the RSyslog Server Module

Remove RSyslog Server

```
ncli> rsyslog-config { delete-server  | remove-server } name="name"
```

Required arguments

`name`

Name of the log server

Update RSyslog Server

```
ncli> rsyslog-config { edit-server  | update-server } name="name" [ ip-
address="ip_address" ][ port="port" ][ network-protocol="network_protocol" ][ relp-
enabled="relp_enabled" ]
```

Required arguments

`name`

Name of the RSyslog Server

Optional arguments

`ip-address`

Ip address of the RSyslog Server

`port`

port number

`network-protocol`

Protocol for RSyslog server configuration

`relp-enabled`

Reliable Event Logging Protocol option

Returns the status of RSyslog service

```
ncli> rsyslog-config { get-status }
```

Required arguments

None

List RSyslog Servers

```
ncli> rsyslog-config { list  | ls }
```

Required arguments

None

List RSyslog Server Modules

```
ncli> rsyslog-config { list-modules  | ls-modules } server-name="server_name"
```

Required arguments

`server-name`

Name of the log server

List RSyslog Servers

```
ncli> rsyslog-config { list-servers  | ls-servers }
```

Required arguments

None

Sets the status of RSyslog service

```
ncli> rsyslog-config { set-status } enable="{ true | false }"
```

Required arguments

`enable`

Enable RSyslog Status

`smb-server`: Nutanix SMB server

Description The Nutanix SMB file server

Alias

Operations

- Disable Kerberos security services in the SMB server : `disable-kerberos`
- Enable Kerberos security services in the SMB server : `enable-kerberos`
- Get the status of Kerberos for the SMB server : `get-kerberos-status`

Disable Kerberos security services in the SMB server. This operation is only valid for clusters having hosts running Hyper-V.

```
ncli> smb-server { disable-kerberos } logon-name="logon_name" [
password="password" ]
```

Required arguments

`logon-name`

Logon name (domain\username) of a domain user/administrator account that has privileges to perform the operation

Optional arguments

`password`

Password for the account specified by the logon account name

Enable Kerberos security services in the SMB server. This operation is only valid for clusters having hosts running Hyper-V.

```
ncli> smb-server { enable-kerberos } logon-name="logon_name" [ password="password"
]
```

Required arguments

`logon-name`

Logon name (domain\username) of a domain user/administrator account that has privileges to perform the operation

Optional arguments

`password`

Password for the account specified by the logon account name

Get the status of Kerberos for the SMB server. This operation is only valid for clusters having hosts running Hyper-V.

```
ncli> smb-server { get-kerberos-status }
```

Required arguments

None

snapshot: Snapshot

Description Snapshot of a Virtual Disk

Alias `snap`

Operations

- Create a (fast) clone based on a Snapshot : `clone`
- Create a new Snapshot of a Virtual Disk or a NFS file : `create` | `add`
- List Snapshots : `list` | `ls`
- Get stats data for Snapshots : `list-stats` | `ls-stats`
- Delete a Snapshot : `remove` | `rm`

Create a (fast) clone based on a Snapshot

```
ncli> snapshot { clone }[ name="name" ][ clone-names="clone_names" ][ src-
file="src_file" ][ dest-files="dest_files" ][ allow-overwrite="allow_overwrite" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Snapshot

`clone-names`

A comma-separated list of names for the newly created Snapshots

`src-file`

Absolute path of the NFS snapshot file

`dest-files`

Absolute path(s) of the clone(s) to be created

`allow-overwrite`

Enable overwriting if a NFS file already exists in the destination path

Default: false

Create a new Snapshot of a Virtual Disk or a NFS file

```
ncli> snapshot { create  | add }[ name="name" ][ vdisk-name="vdisk_name" ][ src-
file="src_file" ][ dest-file="dest_file" ][ allow-overwrite="allow_overwrite" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Snapshot. If name is not specified, format snap_<vDisk-name>_YYYY_MM_DD_HH_MM_SS is used to generate the name

`vdisk-name`

Name of the Virtual Disk

`src-file`

Absolute path of the NFS snapshot file

`dest-file`

Absolute path of the snapshot file to be created

`allow-overwrite`

Enable overwriting if a NFS file already exists in the destination path

Default: false

List Snapshots

```
ncli> snapshot { list  | ls }[ name="name" ][ vdisk-name="vdisk_name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Snapshot

`vdisk-name`

Name of the corresponding Virtual Disk

Get stats data for Snapshots

```
ncli> snapshot { list-stats  | ls-stats }[ name="name" ][ vdisk-name="vdisk_name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Snapshot

`vdisk-name`

Name of the corresponding Virtual Disk

Delete a Snapshot

```
ncli> snapshot { remove  | rm } name="name"
```

Required arguments

`name`

Name of the Snapshot

snmp: SNMP

Description An SNMP agent

Alias

Operations

- Add a transport to the list of snmp transports : `add-transport`
- Add a trap sink to the list of trap sinks : `add-trap`
- Add an snmp user along with its authentication and privacy keys : `add-user`
- Edit one of the trap sinks from the list of trap sinks : `edit-trap` | `update-trap`
- Modify the authentication and encrytption information of an existing snmp user : `edit-user` | `update-user`
- Returns the status of the snmp service : `get-status`
- List all the transports specified for the snmp agent : `list-transports` | `ls-transports`
- List all the configured trap sinks along with their user information : `list-traps` | `ls-traps`
- Lists all the snmp users along with their properties like authentication and privacy information : `list-users` | `ls-users`
- Remove a transport from the list of snmp transports : `remove-transport` | `delete-transport`
- Remove a trap from the list of snmp traps : `remove-trap` | `delete-trap`
- Remove a user from the list of snmp users : `remove-user` | `delete-user`
- Sets the status of the snmp service : `set-status`

Add a transport to the list of snmp transports. Each transport is a protocol:port pair

```
ncli> snmp { add-transport } protocol="protocol" port="port"
```

Required arguments

`protocol`

Protocol for the snmp agent or trap sink. Currently supported protocols are UDP, TCP and UDP_6

`port`

Port number on which an snmp agent listens for requests or on which a trap sink is waiting traps

Add a trap sink to the list of trap sinks. Each trap sink is a combination of trap sink address, username and authentication information

```
ncli> snmp { add-trap } address="address" [ username="username" ][ port="port"
][ protocol="protocol" ][ version="version" ][ community="community" ][ engine-
id="engine_id" ][ inform="inform" ]
```

Required arguments

`address`

Address of an snmp trap sink. This should be an IP address or FQDN

Optional arguments

`username`

Identity of an snmp user. It is required for version snmpv3. It is not used for version snmpv2c.

`port`

Port number on which an snmp agent listens for requests or on which a trap sink is waiting traps

`protocol`

Protocol for the snmp agent or trap sink. Currently supported protocols are UDP, TCP and UDP_6

`version`

SNMP version [snmpv2c, snmpv3]

`community`

SNMP community string. Used for snmpv2c only. If not set, default to "public"

`engine-id`

Engine id of the snmp trap sink. This must be a hex string starting with 0x. It is set for snmpv3, not used for snmpv2c.

`inform`

Flag that specifies whether a trap sink is actually an inform sink

Default: false

Add an snmp user along with its authentication and privacy keys

```
ncli> snmp { add-user }username="username" auth-key="auth_key" auth-
type="auth_type"[ priv-key="priv_key" ][ priv-type="priv_type" ]
```

Required arguments

`username`

Identity of an snmp user. It is required for version snmpv3. It is not used for version snmpv2c.

`auth-key`

Authentication key for an snmp user

`auth-type`

Authentication type for snmp user. Can be SHA

Optional arguments

`priv-key`

Encryption key for an snmp user

`priv-type`

Encryption type for an snmp user. Can be AES

Edit one of the trap sinks from the list of trap sinks. Editable properties are username, authentication and privacy settings and protocol

```
ncli> snmp { edit-trap  | update-trap }address="address"[ port="port" ][
protocol="protocol" ][ version="version" ][ community="community" ][ engine-
id="engine_id" ][ inform="inform" ][ username="username" ]
```

Required arguments

`address`

Address of an snmp trap sink. This should be an IP address or FQDN

Optional arguments

`port`

Port number on which an snmp agent listens for requests or on which a trap sink is waiting traps

`protocol`

Protocol for the snmp agent or trap sink. Currently supported protocols are UDP, TCP and UDP_6

`version`

SNMP version [snmpv2c, snmpv3]

`community`

SNMP community string. Used for snmpv2c only. If not set, default to "public"

`engine-id`

Engine id of the snmp trap sink. This must be a hex string starting with 0x. It is set for snmpv3, not used for snmpv2c.

`inform`

Flag that specifies whether a trap sink is actually an inform sink

`username`

Identity of an snmp user. It is required for version snmpv3. It is not used for version snmpv2c.

Modify the authentication and encrytption information of an existing snmp user

```
ncli> snmp { edit-user  | update-user } username="username" [ auth-key="auth_key" ][ auth-type="auth_type" ][ priv-key="priv_key" ][ priv-type="priv_type" ]
```

Required arguments

`username`

Identity of an snmp user. It is required for version snmpv3. It is not used for version snmpv2c.

Optional arguments

`auth-key`

Authentication key for an snmp user

`auth-type`

Authentication type for snmp user. Can be SHA

`priv-key`

Encryption key for an snmp user

`priv-type`

Encryption type for an snmp user. Can be AES

Returns the status of the snmp service

```
ncli> snmp { get-status }
```

Required arguments

None

List all the transports specified for the snmp agent. Each transport is a protocol:port pair

```
ncli> snmp { list-transports  | ls-transports }
```

Required arguments

None

List all the configured trap sinks along with their user information.

```
ncli> snmp { list-traps  | ls-traps }[ address="address" ]
```

Required arguments

None

Optional arguments

`address`

Address of an snmp trap sink. This should be an IP address or FQDN

Lists all the snmp users along with their properties like authentication and privacy information

```
ncli> snmp { list-users  | ls-users }[ username="username" ]
```

Required arguments

None

Optional arguments

`username`

Identity of an snmp user. It is required for version snmpv3. It is not used for version snmpv2c.

Remove a transport from the list of snmp transports

```
ncli> snmp { remove-transport  | delete-transport } protocol="protocol" port="port"
```

Required arguments

`protocol`

Protocol for the snmp agent or trap sink. Currently supported protocols are UDP, TCP and UDP_6

`port`

Port number on which an snmp agent listens for requests or on which a trap sink is waiting traps

Remove a trap from the list of snmp traps

```
ncli> snmp { remove-trap  | delete-trap } address="address"
```

Required arguments

`address`

Address of an snmp trap sink. This should be an IP address or FQDN

Remove a user from the list of snmp users

```
ncli> snmp { remove-user  | delete-user } username="username"
```

Required arguments

`username`

Identity of an snmp user. It is required for version snmpv3. It is not used for version snmpv2c.

Sets the status of the snmp service

```
ncli> snmp { set-status } enable="{ true | false }"
```

Required arguments

`enable`

Enable or disable snmp agent on a cluster

`software`: Software

Description NOS Software Release

Alias

Operations

- Toggle automatic download of a Software : `automatic-download`
- Download a Software : `download`
- List Software : `list` | `ls`
- Pause Downloading / Uploading a Software : `pause`
- Delete a Software : `remove` | `rm` | `delete`
- Upload a Software : `upload`

Toggle automatic download of a Software

```
ncli> software { automatic-download }[ enable="{ true | false }" ]
```

Required arguments

None

Optional arguments

`enable`

Enable/Disable automatic downloads

Download a Software

```
ncli> software { download } name="name" software-type="software_type"
```

Required arguments

`name`

Name of the software

`software-type`

Type of the software (NOS | HYPERVISOR | FIRMWARE_DISK | NCC | FILE_SERVER | PRISM_CENTRAL_DEPLOY)

List Software

```
ncli> software { list  | ls }[ name="name" ][ software-type="software_type" ]
```

Required arguments

None

Optional arguments

`name`

Name of the software

`software-type`

Type of the software (NOS | HYPERVISOR | FIRMWARE_DISK | NCC | FILE_SERVER | PRISM_CENTRAL_DEPLOY)

Pause Downloading / Uploading a Software

```
ncli> software { pause } name="name" software-type="software_type"
```

Required arguments

`name`

Name of the software

`software-type`

Type of the software (NOS | HYPERVISOR | FIRMWARE_DISK | NCC | FILE_SERVER | PRISM_CENTRAL_DEPLOY)

Delete a Software

```
ncli> software { remove  | rm | delete } name="name" software-type="software_type"
```

Required arguments

`name`

Name of the software

`software-type`

Type of the software (NOS | HYPERVISOR | FIRMWARE_DISK | NCC | FILE_SERVER | PRISM_CENTRAL_DEPLOY)

Upload a Software

```
ncli> software { upload } file-path="file_path" software-type="software_type" [
hypervisor-type="hypervisor_type" ][ meta-file-path="meta_file_path" ]
```

Required arguments

`file-path`

Path to the software to be uploaded

`software-type`

Type of the software (NOS | HYPERVISOR | FIRMWARE_DISK | NCC | FILE_SERVER | PRISM_CENTRAL_DEPLOY)

Optional arguments

`hypervisor-type`

Type of the Hypervisor

`meta-file-path`

Path to the metadata file of the software to be uploaded

`ssl-certificate`: SSL Certificate

Description Manage SSL certificates

Alias

Operations

- Change password for pfx file : `change-pfx-file-password`
- Generates SSL Certificate with cipher Strength 2048 bits and replaces the existing certificate : `generate`
- Import SSL Certificate, key and CA certificate or chain file : `import`

Change password for pfx file

```
ncli> ssl-certificate { change-pfx-file-password }
```

Required arguments

None

Generates SSL Certificate with cipher Strength 2048 bits and replaces the existing certificate

```
ncli> ssl-certificate { generate }
```

Required arguments

None

Import SSL Certificate, key and CA certificate or chain file. This import replaces the existing certificate

```
ncli> ssl-certificate { import }certificate-path="certificate_path" cacertificate-
path="cacertificate_path" key-path="key_path" key-type="key_type"
```

Required arguments

`certificate-path`

Path of the SSL certificate

`cacertificate-path`

Path of the CA certificate or chain file

`key-path`

Path of the private key

`key-type`

Type of Private key. Must be either RSA_2048 or ECDSA_256 or ECDSA_384 or ECDSA_521

`storagepool`: Storage Pool

Description A Pool of Physical Disks

Alias `sp`

Operations

- Create a new Storage Pool : `create` | `add`
- Edit a Storage Pool : `edit` | `update`
- List Storage Pools : `list` | `ls`
- Get stats data for Storage Pools : `list-stats` | `ls-stats`

Create a new Storage Pool

```
ncli> storagepool { create  | add } name="name" [ disk-ids="disk_ids" ][ add-all-free-disks="add_all_free_disks" ][ force="force" ]
```

Required arguments

`name`

Name of the Storage Pool

Optional arguments

`disk-ids`

IDs of Physical Disk in the Storage Pool

`add-all-free-disks`

Add all free disks that are not part of any storage pool?

`force`

Forcefully perform the requested operation skipping any constraint validation

Default: false

Edit a Storage Pool

```
ncli> storagepool { edit  | update }[ id="id" ][ name="name" ][ new-name="new_name" ][ add-all-free-disks="add_all_free_disks" ][ add-disk-ids="add_disk_ids" ][ rm-disk-ids="rm_disk_ids" ][ force="force" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Storage Pool

`name`

Name of the Storage Pool

`new-name`

Name of the Storage Pool

`add-all-free-disks`

Add all free disks that are not part of any storage pool?

`add-disk-ids`

A comma-separated list of Physical Disk IDs to be added to the Storage Pool

`rm-disk-ids`

A comma-separated list of Physical Disk IDs to be removed from the Storage Pool

`force`

Forcefully perform the requested operation skipping any constraint validation

Default: false

List Storage Pools

```
ncli> storagepool { list  | ls }[ id="id" ][ name="name" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Storage Pool

`name`

Name of the Storage Pool

Get stats data for Storage Pools

```
ncli> storagepool { list-stats  | ls-stats }[ id="id" ][ name="name" ]
```

Required arguments

None

Optional arguments

`id`

ID of the Storage Pool

`name`

Name of the Storage Pool

`storagetier`: Storage Tier

Description A Tier of physical storage

Alias `tier`

Operations

- List the (global) default I/O priority order of Storage Tiers : `get-default-io-priority-order` | `get-def-io-pri`
- List Storage Tiers : `list` | `ls`
- List types of Storage Tiers : `list-supported-types` | `ls-supported-types`
- Remove a Storage Tier : `remove` | `rm`
- Set the (global) default I/O priority order of Storage Tiers : `set-default-io-priority-order` | `set-def-io-pri`

List the (global) default I/O priority order of Storage Tiers

```
ncli> storagetier { get-default-io-priority-order  | get-def-io-pri }
```

Required arguments

None

List Storage Tiers

```
ncli> storagetier { list  | ls }[ name="name" ][ type="type" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Storage Tier

`type`

Type of Storage Tier (as provided by the 'list-supported-types' operation)

List types of Storage Tiers

```
ncli> storagetier { list-supported-types  | ls-supported-types }
```

Required arguments

None

Remove a Storage Tier

```
ncli> storagetier { remove  | rm } name="name"
```

Required arguments

`name`

Name of the Storage Tier

Set the (global) default I/O priority order of Storage Tiers

```
ncli> storagetier { set-default-io-priority-order  | set-def-io-pri }
random-io-priority-order="random_io_priority_order" sequential-io-priority-
order="sequential_io_priority_order"
```

Required arguments

`random-io-priority-order`

Random I/O priority order (high to low) of Storage Tiers

`sequential-io-priority-order`

Sequential I/O priority order (high to low) of Storage Tiers

`task`: Tasks

Description A Task

Alias

Operations

- Inspect Task : `get`
- List all Tasks : `list` | `ls`
- Poll Task to completion : `wait-for-task`

Inspect Task

```
ncli> task { get } taskid="taskid" [ include-entity-names="{ true | false }" ]
```

Required arguments

`taskid`

Id of the task

Optional arguments

`include-entity-names`

Include entity names

List all Tasks

```
ncli> task { list  | ls }[ entity-types="entity_types" ][ entity-uuids="entity_uuids" ][
operation-type-list="operation_type_list" ][ include-completed="{ true | false }" ][ epoch-
cut-off-time="epoch_cut_off_time" ][ count="count" ][ include-entity-names="{ true |
false }" ]
```

Required arguments

None

Optional arguments

`entity-types`

Comma separated Entity types

`entity-uuids`

Comma separated Entity types

`operation-type-list`

Comma separated Operation types

`include-completed`

Include Completed Tasks

`epoch-cut-off-time`

Tasks greater than cut off epoch time in microseconds will be returned. This is applicable only when include completed is set to True.

`count`

Maximum number of tasks

`include-entity-names`

Include entity names

Poll Task to completion

`ncli> task { wait-for-task } taskid="`*taskid*`" [ timeoutseconds="`*timeoutseconds*`" ][ include-entity-names="{`*true* | *false*`}" ]`

Required arguments

`taskid`

Id of the task

Optional arguments

`timeoutseconds`

Timeout seconds

`include-entity-names`

Include entity names

`user`: User

Description A User

Alias

Operations

- Change the password of a User : `change-password`
- Add a new User : `create` | `add`
- Delete a User : `delete` | `remove` | `rm`
- Disable a User : `disable`
- Edit a User : `edit` | `update`
- Enable a User : `enable`
- Get the IP Addresses and browser details of a user who is currently logged in : `get-logged-in-user` | `get-logged-in-user`
- Get a list of all users who are currently logged in to the system along with their IP Addresses and browser details : `get-logged-in-users` | `get-logged-in-users`
- Grant cluster administration role to a User : `grant-cluster-admin-role`
- Grant user administration role to a User : `grant-user-admin-role`
- List Users : `list` | `ls`
- Reset the password of a User : `reset-password`
- Revoke cluster administration role from a User : `revoke-cluster-admin-role`
- Revoke user administration role from a User : `revoke-user-admin-role`
- Show profile of current User : `show-profile`

Change the password of a User

```
ncli> user { change-password }current-password="current_password" new-
password="new_password"
```

Required arguments

`current-password`

Current password of the user

`new-password`

New password of the user

Add a new User

```
ncli> user { create  | add}user-name="user_name" user-password="user_password"
first-name="first_name" last-name="last_name" email-id="email_id"[ middle-
initial="middle_initial" ]
```

Required arguments

`user-name`

User name of the user

`user-password`

Password of the user

`first-name`

First name of the user

`last-name`

Last name of the user

`email-id`

Email address of the user

Optional arguments

`middle-initial`

Middle Initial of the user

Delete a User

```
ncli> user { delete  | remove | rm }user-name="user_name"
```

Required arguments

`user-name`

User name of the user

Disable a User

```
ncli> user { disable }user-name="user_name"
```

Required arguments

`user-name`

User name of the user

Edit a User

```
ncli> user { edit  | update }user-name="user_name"[ first-name="first_name" ][
last-name="last_name" ][ middle-initial="middle_initial" ][ email-id="email_id" ]
```

Required arguments

`user-name`

User name of the user

Optional arguments

`first-name`

First name of the user

`last-name`

Last name of the user

`middle-initial`

Middle Initial of the user

`email-id`

Email address of the user

Enable a User

```
ncli> user { enable }user-name="user_name"
```

Required arguments

`user-name`

User name of the user

Get the IP Addresses and browser details of a user who is currently logged in

```
ncli> user { get-logged-in-user  | get-logged-in-user }username="username"
```

Required arguments

`username`

UserName of the Logged in User

Get a list of all users who are currently logged in to the system along with their IP Addresses and browser details

```
ncli> user { get-logged-in-users  | get-logged-in-users }
```

Required arguments

None

Grant cluster administration role to a User

```
ncli> user { grant-cluster-admin-role }user-name="user_name"
```

Required arguments

`user-name`

User name of the user

Grant user administration role to a User

```
ncli> user { grant-user-admin-role }user-name="user_name"
```

Required arguments

`user-name`

User name of the user

List Users

```
ncli> user { list  | ls }[ user-name="user_name" ]
```

Required arguments

None

Optional arguments

`user-name`

User name of the user

Reset the password of a User

```
ncli> user { reset-password }user-name="user_name" password="password"
```

Required arguments

`user-name`

User name of the user

`password`

password

Revoke cluster administration role from a User

```
ncli> user { revoke-cluster-admin-role }user-name="user_name"
```

Required arguments

`user-name`

User name of the user

Revoke user administration role from a User

```
ncli> user { revoke-user-admin-role }user-name="user_name"
```

Required arguments

`user-name`

User name of the user

Show profile of current User

```
ncli> user { show-profile }
```

Required arguments

None

`vdisk`: Virtual Disk

Description A Virtual Disk

Alias

Operations

- Create a (fast) clone of a Virtual Disk : `clone`
- Create a new Virtual Disk : `create` | `add`
- Edit a Virtual Disk : `edit` | `update`
- List Virtual Disks : `list` | `ls`
- List Snapshots : `list-snapshots` | `ls-snaps`
- Get stats data for Virtual Disks : `list-stats` | `ls-stats`
- Delete a Virtual Disk : `remove` | `rm`
- Remove reserved capacity of vdisks in a given container : `remove-reservation`
- Create a new Snapshot of a Virtual Disk : `snapshot` | `snap`

Create a (fast) clone of a Virtual Disk

```
ncli> vdisk { clone } name="name" clone-names="clone_names" [ snap-
name="snap_name" ]
```

Required arguments

`name`

Name of the Virtual Disk

`clone-names`

A comma-separated list of names for the newly created Snapshots

Optional arguments

`snap-name`

Name of the Snapshot on which the clone is based

Create a new Virtual Disk

```
ncli> vdisk { create  | add } name="name" max-capacity="max_capacity" [ ctr-
id="ctr_id" ][ ctr-name="ctr_name" ][ res-capacity="res_capacity" ][ fingerprint-
on-write="fingerprint_on_write" ][ on-disk-dedup="on_disk_dedup" ][ erasure-code-
delay="erasure_code_delay" ][ shared="shared" ]
```

Required arguments

`name`

Name of the Virtual Disk

`max-capacity`

Max Capacity (GiB) of the Virtual Disk

Optional arguments

`ctr-id`

ID of the Storage Container for the Virtual Disk

`ctr-name`

Name of the Storage Container for the Virtual Disk

`res-capacity`

Reserved Capacity (GiB) of the Virtual Disk

`fingerprint-on-write`

Fingerprint on writes to the Virtual Disk {on, off, none}. This VDisk level setting overrides the Storage Container level setting

`on-disk-dedup`

On-disk dedup of the Virtual Disk {off, post-process, none}. This VDisk level setting overrides the Storage Container level setting.

`erasure-code-delay`

Erasure code delay (secs) of the Virtual Disk

`shared`

Is this a shared Virtual Disk?

Default: false

Edit a Virtual Disk

```
ncli> vdisk { edit  | update } name="name" [ max-capacity="max_capacity" ][ res-capacity="res_capacity" ][ fingerprint-on-write="fingerprint_on_write" ][ on-disk-dedup="on_disk_dedup" ][ erasure-code-delay="erasure_code_delay" ]
```

Required arguments

`name`

Name of the Virtual Disk

Optional arguments

`max-capacity`

Max Capacity (GiB) of the Virtual Disk

`res-capacity`

Reserved Capacity (GiB) of the Virtual Disk

`fingerprint-on-write`

Fingerprint on writes to the Virtual Disk {on, off, none}. This VDisk level setting overrides the Storage Container level setting

`on-disk-dedup`

On-disk dedup of the Virtual Disk {off, post-process, none}. This VDisk level setting overrides the Storage Container level setting.

`erasure-code-delay`

Erasure code delay (secs) of the Virtual Disk

List Virtual Disks

```
ncli> vdisk { list  | ls }[ names="names" ][ vm-id="vm_id" ][ ctr-id="ctr_id" ]
```

Required arguments

None

Optional arguments

`names`

A comma-separated list of the names of the Virtual Disks

`vm-id`

ID of a Virtual Machine that the Virtual Disk is mapped to

`ctr-id`

Get Virtual Disks in the specified Storage Container

List Snapshots

```
ncli> vdisk { list-snapshots  | ls-snaps }[ name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Virtual Disk or Snapshot

Get stats data for Virtual Disks

```
ncli> vdisk { list-stats  | ls-stats }[ names="names" ][ vm-id="vm_id" ][ ctr-id="ctr_id" ]
```

Required arguments

None

Optional arguments

`names`

A comma-separated list of the names of the Virtual Disks

`vm-id`

ID of a Virtual Machine that the Virtual Disk is mapped to

`ctr-id`

Get Virtual Disks in the specified Storage Container

Delete a Virtual Disk

```
ncli> vdisk { remove  | rm } name="name"
```

Required arguments

`name`

Name of the Virtual Disk

Remove reserved capacity of vdisks in a given container. If container id not specified, reserved capacity will be removed for all vdisks

```
ncli> vdisk { remove-reservation }[ ctr-id="ctr_id" ]
```

Required arguments

None

Optional arguments

`ctr-id`

ID of the Storage Container

Create a new Snapshot of a Virtual Disk

```
ncli> vdisk { snapshot  | snap } name="name" [ snap-name="snap_name" ]
```

Required arguments

`name`

Name of the Virtual Disk

Optional arguments

`snap-name`

Name of the Snapshot

`virtual-disk`: Virtual Disk

Description Commands for performing different actions on Virtual Disks.

Alias

Operations

- List Virtual Disk : `list` | `ls`

List Virtual Disk

```
ncli> virtual-disk { list  | ls }[ id="id" ]
```

Required arguments

None

Optional arguments

`id`

Id of the Virtual Disk

`virtualmachine`: Virtual Machine

Description A Virtual Machine

Alias `vm`

Operations

- Attach a disk from file level restore capable snapshot to a VM : `attach-flr-disk`
- Detach a file level restore disk from a VM : `detach-flr-disk`
- List Virtual Machine : `list` | `ls`
- Get all file level restore capable snapshots attached to a VM : `list-attached-flr-snapshots`
- Get file level restore capable snapshots of a VM : `list-flr-snapshots` | `ls-flr-snaps`
- Get snapshots of a VM : `list-snapshots` | `ls-snaps`
- Get stats data for Virtual Machine : `list-stats` | `ls-stats`
- Update FingerPrintOnWrite on all vdisks of a VM : `update-fingerprint-on-write`
- Update OnDiskDedup on all vdisks of a VM : `update-on-disk-dedup`

Attach a disk from file level restore capable snapshot to a VM

```
ncli> virtualmachine { attach-flr-disk } vm-id="vm_id" snap-id="snap_id" disk-label="disk_label"
```

Required arguments

`vm-id`

ID of the Virtual Machine

`snap-id`

Id of the Snapshot

`disk-label`

Label for disk to be attached

Detach a file level restore disk from a VM

```
ncli> virtualmachine { detach-flr-disk } vm-id="vm_id" attached-disk-label="attached_disk_label"
```

Required arguments

`vm-id`

ID of the Virtual Machine

`attached-disk-label`

Attached disk label

List Virtual Machine

```
ncli> virtualmachine { list  | ls }[ name="name" ][ id="id" ][ host-id="host_id" ][ get-unprotected-cbr-vms-only="get_unprotected_cbr_vms_only" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Virtual Machine

`id`

ID of the Virtual Machine

`host-id`

ID of the Physical Host running Virtual Machines

`get-unprotected-cbr-vms-only`

Get only unprotected Virtual Machines that can participate in Nutanix Converged Backup and Recovery?

Default: false

Get all file level restore capable snapshots attached to a VM.

```
ncli> virtualmachine { list-attached-flr-snapshots } vm-id="vm_id" [ snap-id="snap_id" ]
```

Required arguments

`vm-id`

ID of the Virtual Machine

Optional arguments

`snap-id`

Id of the Snapshot

Get file level restore capable snapshots of a VM.

```
ncli> virtualmachine { list-flr-snapshots  | ls-flr-snaps } vm-id="vm_id" [ snapshot-count="snapshot_count" ]
```

Required arguments

`vm-id`

ID of the Virtual Machine

Optional arguments

`snapshot-count`

Number of snapshots to be fetched. A value of 0 indicates all snapshots will be returned.

Get snapshots of a VM.

```
ncli> virtualmachine { list-snapshots  | ls-snaps } vm-id="vm_id" [ snap-id="snap_id" ]
```

Required arguments

`vm-id`

ID of the Virtual Machine

Optional arguments

`snap-id`

Id of the Snapshot

Get stats data for Virtual Machine

```
ncli> virtualmachine { list-stats  | ls-stats }[ name="name" ][ id="id" ][ host-id="host_id" ]
```

Required arguments

None

Optional arguments

`name`

Name of the Virtual Machine

`id`

ID of the Virtual Machine

`host-id`

ID of the Physical Host running Virtual Machines

Update FingerPrintOnWrite on all vdisks of a VM

```
ncli> virtualmachine { update-fingerprint-on-write } vm-id="vm_id" fingerprint-on-write="fingerprint_on_write"
```

Required arguments

`vm-id`

Uuid of Virtual machine

`fingerprint-on-write`

Finger Print on Write

Update OnDiskDedup on all vdisks of a VM

```
ncli> virtualmachine { update-on-disk-dedup } vm-id="vm_id" on-disk-dedup="on_disk_dedup"
```

Required arguments

`vm-id`

Uuid of Virtual machine

`on-disk-dedup`

On Disk Dedup

`volume-group`: Volume Groups

Description A Volume Group

Alias

Operations

- Attach Volume Group to VM : `attach-to-vm`
- Clone VM Disk for Volume Group : `clone-disk`
- Closes Volume Group for iSCSI initiators : `close` | `detach-external`
- Create Volume Group : `create`
- Create VM Disk for Volume Group : `create-disk`
- Delete Volume Group : `delete`
- Delete VM Disk from Volume Group : `delete-disk`
- Detach Volume Group to VM : `detach-from-vm`
- Inspect Volume Group : `get`
- List all Volume Groups : `list` | `ls`
- Show unprotected Volume Groups : `list-unprotected`
- Opens Volume Group for iSCSI initiators : `open` | `attach-external`
- Update Volume Group : `update`
- Update VM Disk for Volume Group : `update-disk`

Attach Volume Group to VM

```
ncli> volume-group { attach-to-vm } uuid="uuid" vm-uuid="vm_uuid" [ index="index" ]
```

Required arguments

`uuid`

UUID of the Volume Group

`vm-uuid`

Virtual machine UUID

Optional arguments

`index`

Volume Group index of the disk

Clone VM Disk for Volume Group

```
ncli> volume-group { clone-disk }uuid="uuid"[ vmdisk-uuid="vmdisk_uuid" ][ adsf-
filepath="adsf_filepath" ][ index="index" ][ size-mib="size_mib" ]
```

Required arguments

`uuid`

UUID of the Volume Group

Optional arguments

`vmdisk-uuid`

VM Disk UUID to clone from

`adsf-filepath`

ADSF file path to VM Disk to clone from

`index`

Volume Group index of the disk

`size-mib`

Size (MiB) to create a VM Disk for Volume Group

Closes Volume Group for iSCSI initiators

```
ncli> volume-group { close  | detach-external}uuid="uuid"[ iscsi-
initiator="iscsi_initiator" ][ iscsi-client-ip="iscsi_client_ip" ][ iscsi-client="iscsi_client"
]
```

Required arguments

`uuid`

UUID of the Volume Group

Optional arguments

`iscsi-initiator`

iSCSI qualified name

`iscsi-client-ip`

iSCSI client ip address

`iscsi-client`

List of supported authentication types

Create Volume Group

```
ncli> volume-group { create }name="name"[ description="description" ][
shared="shared" ][ iscsi-target="iscsi_target" ][ iscsi-target-prefix="iscsi_target_prefix"
][ iscsi-initiator-name-list="iscsi_initiator_name_list" ][ attached-
clients="attached_clients" ][ enabled-authentications="{ true | false }" ]
```

Required arguments

`name`

Volume Group name

Optional arguments

`description`

Volume Group description

`shared`

True or false indicating whether volume is shared across multiple iSCSI initiators

`iscsi-target`

iSCSI target name

`iscsi-target-prefix`

iSCSI target prefix name

`iscsi-initiator-name-list`

iSCSI qualified name list

`attached-clients`

List of the attached clients

`enabled-authentications`

List of supported authentication types

Create VM Disk for Volume Group

```
ncli> volume-group { create-disk }uuid="uuid" size-mib="size_mib"[ container-
id="container_id" ][ container-uuid="container_uuid" ][ index="index" ]
```

Required arguments

`uuid`

UUID of the Volume Group

`size-mib`

Size (MiB) to create a VM Disk for Volume Group

Optional arguments

`container-id`

ID of Storage Container to create VM Disk

`container-uuid`

UUID of the Storage Container

`index`

Volume Group index of the disk

Delete Volume Group

```
ncli> volume-group { delete }uuid="uuid"
```

Required arguments

`uuid`

UUID of the Volume Group

Delete VM Disk from Volume Group

```
ncli> volume-group { delete-disk }uuid="uuid" index="index"
```

Required arguments

`uuid`

UUID of the Volume Group

`index`

Volume Group index of the disk

Detach Volume Group to VM

```
ncli> volume-group { detach-from-vm } uuid="uuid" vm-uuid="vm_uuid" [ index="index" ]
```

Required arguments

`uuid`

UUID of the Volume Group

`vm-uuid`

Virtual machine UUID

Optional arguments

`index`

Volume Group index of the disk

Inspect Volume Group

```
ncli> volume-group { get }[ uuid="uuid" ][ name="name" ][ include-disk-size="{ true | false }" ]
```

Required arguments

None

Optional arguments

`uuid`

UUID of the Volume Group

`name`

Name of the Volume Group

`include-disk-size`

Whether to include disk sizes, true by default

List all Volume Groups

```
ncli> volume-group { list | ls }[ include-disk-size="{ true | false }" ]
```

Required arguments

None

Optional arguments

`include-disk-size`

Whether to include disk sizes, true by default

Show unprotected Volume Groups

```
ncli> volume-group { list-unprotected }[ uuids="uuids" ]
```

Required arguments

None

Optional arguments

`uuids`

Volume Group UUIDs

Opens Volume Group for iSCSI initiators

```
ncli> volume-group { open | attach-external } uuid="uuid" [ iscsi-initiator="iscsi_initiator" ][ iscsi-client-ip="iscsi_client_ip" ][ iscsi-client="iscsi_client" ]
```

Required arguments

`uuid`

UUID of the Volume Group

Optional arguments

`iscsi-initiator`

iSCSI qualified name

`iscsi-client-ip`

iSCSI client ip address

`iscsi-client`

List of supported authentication types

Update Volume Group

```
ncli> volume-group { update }uuid="uuid"[ name="name" ][ description="description"
][ shared="shared" ][ iscsi-target="iscsi_target" ][ iscsi-target-
prefix="iscsi_target_prefix" ][ attached-clients="attached_clients" ][ enabled-
authentications="{ true | false }" ]
```

Required arguments

`uuid`

UUID of the Volume Group

Optional arguments

`name`

Volume Group name

`description`

Volume Group description

`shared`

True or false indicating whether volume is shared across multiple iSCSI initiators

`iscsi-target`

iSCSI target name

`iscsi-target-prefix`

iSCSI target prefix

`attached-clients`

List of the attached clients

`enabled-authentications`

List of supported authentication types

Update VM Disk for Volume Group

```
ncli> volume-group { update-disk }uuid="uuid" index="index" size-
mib="size_mib"[ preserve-data="preserve_data" ][ vmdisk-uuid="vmdisk_uuid" ][ adsf-
filepath="adsf_filepath" ]
```

Required arguments

`uuid`

UUID of the Volume Group

`index`

Volume Group index of the disk

`size-mib`

Size (MiB) to create a VM Disk for Volume Group

Optional arguments

`preserve-data`

Whether to preserve data of the volume disk, true by default

`vmdisk-uuid`

VM Disk UUID to clone from

`adsf-filepath`

ADSF file path to VM Disk to clone from

`vstore`: VStore

Description A file namespace in a Storage Container

Alias

Operations

- List VStores : `list` | `ls`
- Protect a VStore : `protect`
- Unprotect a VStore : `unprotect`

List VStores

```
ncli> vstore {  list  | ls }[ id="id" ][ name="name" ]
```

Required arguments

None

Optional arguments

`id`

ID of a VStore

`name`

Name of a VStore

Protect a VStore. Files in a protected VStore are replicated to a Remote Site at a defined frequency and these protected files can be recovered in the event of a disaster

```
ncli> vstore {  protect  }[ id="id" ][ name="name" ]
```

Required arguments

None

Optional arguments

`id`

ID of a VStore

`name`

Name of a VStore

Unprotect a VStore

```
ncli> vstore {  unprotect  }[ id="id" ][ name="name" ]
```

Required arguments

None

Optional arguments

`id`

ID of a VStore

`name`

Name of a VStore

vzone: vZone

Description A vZone

Alias

Operations

- Add Physical Hosts to a vZone : `add-hosts`
- Add Virtual Disks to a vZone : `add-vdisks`
- Create a new vZone : `create` | `add`
- List vZones : `list` | `ls`
- Delete avZone : `remove` | `rm`
- Remove allVirtual Disks from a vZone : `remove-all-vdisks`
- Remove Virtual Disks from a vZone : `remove-vdisks`

Add Physical Hosts to a vZone

```
ncli> vzone { add-hosts } name="name" host-ids="host_ids"
```

Required arguments

`name`

Name of the vZone

`host-ids`

A comma-separated list of the ids of the Physical Hosts

Add Virtual Disks to a vZone

```
ncli> vzone { add-vdisks } name="name" vdisk-names="vdisk_names"
```

Required arguments

`name`

Name of the vZone

`vdisk-names`

A comma-separated list of the names of the Virtual Disks

Create a new vZone

```
ncli> vzone { create  | add } name="name"
```

Required arguments

`name`

Name of the vZone

List vZones

```
ncli> vzone { list  | ls }[ name="name" ]
```

Required arguments

None

Optional arguments

`name`

Name of the vZone

Delete avZone

```
ncli> vzone { remove  | rm } name="name"
```

Required arguments

`name`

Name of the vZone

Remove allVirtual Disks from a vZone

```
ncli> vzone { remove-all-vdisks } name="name"
```

Required arguments

`name`

Name of the vZone

Remove Virtual Disks from a vZone

```
ncli> vzone { remove-vdisks } name="name" vdisk-names="vdisk_names"
```

Required arguments

`name`

Name of the vZone

`vdisk-names`

A comma-separated list of the names of the Virtual Disks

3

Controller VM Commands

Specifying Credentials

When specifying a password on the command line, always enclose the password in single quotes. For example:`--hypervisor_password='nutanix/4u'`

- To display all user name and password options for `diagnostics.py`, type `/home/nutanix/diagnostics/diagnostics.py --help | egrep -A1 'password|user'`

```
--hypervisor_password: Default hypervisor password.
   (default: 'nutanix/4u')

 --hyperv_hypervisor_username: The username to use when logging into the local
   Hyper-V node.

 --hypervisor_username: The username to use when logging into the local
   hypervisor.

 --ipmi_password: The password to use when logging into the local IPMI device.
   (default: 'ADMIN')
 --ipmi_username: The username to use when logging into the local IPMI device.
   (default: 'ADMIN')

 --kmip_user_id: Username to insert into certificate signing request.
   (default: 'nutanix')

 --nsc_username: User name that provides services through SSH at Nutanix
   service center.
```

- You can find all user name and password options for `cluster`, `genesis`, and `setup_hyperv.py` by also typing `--help | egrep -A1 'password|user'` as part of the command. For example, `setup_hyperv.py --help | egrep -A1 'password|user'`

cluster

Usage

```
Usage: /usr/local/nutanix/cluster/bin/cluster [flags] [command]

commands:

 add_public_key
 convert_cluster
 create
 destroy
 disable_auto_install
 enable_auto_install
 firmware_upgrade
 foundation_upgrade
 host_upgrade
 ipconfig
 lite_upgrade
```

```
migrate_zeus
pass_shutdown_token
reconfig
remove_all_public_keys
remove_public_key
reset
restart_genesis
set_two_node_cluster_leader
start
status
stop
upgrade
upgrade_node
```

/usr/local/nutanix/cluster/bin/cluster

`--add_dependencies`

Include Dependencies.

Default: false

`--backplane_netmask`

Backplane netmask

`--backplane_network`

Backplane network config

Default: false

`--backplane_subnet`

Backplane subnet

`--backplane_vlan`

Backplane VLAN id

Default: -1

`--block_aware`

Set to True to enable block awareness.

Default: false

`--bundle`

Bundle for upgrading host in cluster.

`--clean_debug_data`

If 'clean_debug_data' is True, then when we destroy a cluster we will also remove the logs, binary logs, cached packages, and core dumps on each node.

Default: false

`--cluster_external_ip`

Cluster ip to manage the entire cluster.

`--cluster_function_list`

List of functions of the cluster (use with create). Accepted functions are ['minerva', 'multicluster', 'two_node_cluster', 'jump_box_vm', 'ags_cluster', 'one_node_cluster', 'xi_vm', 'iam_cluster', 'ndfs', 'extension_store_vm', 'witness_vm', 'cloud_data_gateway']

Default: ndfs

`--cluster_name`

Name of the cluster (use with create).

`--cluster_uuid`

Cluster uuid for cluster in string format.

`--config`

Path to the cluster configuration file.

`--container_name`

Name of the default container on the cluster.

`--dns_servers`

Comma separated list of one or more DNS servers.

`--domain_password`

Domain password of the hosts.

`--domain_username`

Domain username of the hosts.

`--enable_lite_upgrade`

Set to False to disable lite upgrade before it is ready.

Default: true

`--firmware_upgrade`

Operation specified will be done in context of firmware upgrade

Default: false

`--force_install_genesis`

Installs the infrastructure package on all of the nodes.

Default: true

`--foundation_upgrade`

Operation specified will be done in context of foundation upgrade

Default: false

`--hardware_device_type`

Type of hardware device. Please specify one of the following: disk nic hba bios bmc. (Currently supported: disk, bios, bmc.)

`--help`

show this help

Default: 0

`--helpfull`

Show flags for all modules

Default: false

`--helpshort`

show usage only for this module

Default: 0

`--helpxml`

like --help, but generates XML output

Default: false

`--host_upgrade`

Operation specified will be done in context of host_upgrade

Default: false

`--hyperv_sku`

Hypervisor sku to which the HyperV host is being upgraded.

`--hypervisor`

Hypervisor that needs to be upgraded. Possible values: esx, kvm, hyperv.

`--ignore_preupgrade_tests`

Perform preupgrade tests

Default: false

`--ignore_vm_conversion_warnings`

Ignore vm conversion errors during cluster conversion.

Default: false

`--inline_firmware_upgrade`

Firmware upgrade will be done inline

Default: false

`--installer_dir_path`

Location of the Nutanix installer directory.

`--ip_specification_json`

JSON file with IP configuration.

`--key_file`

Nutanix default SSH public key.

Default: `/home/nutanix/ssh_keys/nutanix.pub`

`--key_name`

Identifier of the public ssh key in the cluster.

Default: `legacy_nos_compatibility`

`--license_file_zknode`

Path to the zookeeper node that contains the cluster license information.

Default: `/appliance/logical/license/license_file`

`--lockdown_mode`

Flag for lockdown mode

Default: false

`--manual_upgrade`

Manual upgrade method.

Default: false

`--md5sum`

Md5sum of the bundle.

`--migrate_from`

The old zeus node IP address for Zeus migration.

`--migrate_to`

The new zeus node IP address for Zeus migration.

`--no_verification`

Skip verification for package integrity.

Default: true

`--ntp_servers`

Comma separated list of one or more NTP servers.

`--nutanix_default_password_salt`

Seed prism admin password only if the password hash does not contain this salt.

Default: `$6$Mkd8T74/$`

`--password_lockdown_mode`

Flag for password lockdown mode

Default: false

`--python_proto_egg_path`

Path of protobuf egg relative to the install dir.

Default: `lib/py/protobuf-2.6.1-py2.6-linux-x86_64.egg`

`--rack_aware`

Set to True to enable rack awareness.

Default: false

`--rack_config_json_path`

Path to the json file containing svm_ips to rack namemapping. Json file will contain svm_ips as the keys and the name of the rack they belong to, as values

`--redundancy_factor`

Max redundancy factor supported by the cluster. 'redundancy_factor - 1' determines the number of node failures that the cluster should be able to tolerate.

Default: -1

`--remove_installer_dir`

Whether or not to remove the installer directory automatically when finished.

Default: true

`--reset_gflags_on_destroy`

When performing cluster destroy, remove all gflag files.

Default: false

`--seed_prism_password`

Seed Prism admin password to be the same as the nutanix user if nutanix user has non-default password.

Default: true

`--shutdown_token_ip`

IP address of intended shutdown token holder SVM.

`--skip_discovery`

Skip mdns discovery.

Default: false

`--skip_reconfig`

Skip CVM reconfig.

Default: false

`--skip_upgrade`

Skip actual upgrade.

Default: false

`--stand_alone_upgrade_timeout`

Timeout for stand-alone upgrade.

Default: 600

`--svm_ips`

Comma separated list of IP addresses of one or more SVMs in the target cluster. Not required if cluster.cfg is being used.

`--svm_login`

User name for logging into SVM.

Default: `nutanix`

`--target_hypervisor`

Target hypervisor type for cluster conversion. Valid types esx/kvm.

`--timeout`

Number of seconds each command to SVMs should take.

Default: 180

`--upgrade_node_ip`

Ipv4 or IPv6 address of node to be upgraded.

`--vcenter_json_file`

File containing vcenter details for dial workflow. The json has host, username and password keys.

`--vcenter_not_required`

Set it to true if vcenter is not used to manage ESX cluster.

Default: false

`--verification_file`

Metadata file for package integrity, upgrade info.

`--version`

Version to which upgrade needs to be performed.

`--wait`

Wait for action to complete.

Default: true

cluster.ce_helper

`--ce_version_map_znode_path`

Zookeeper node containing the CE version mapping.

Default: `/appliance/logical/community_edition/version_map`

cluster.cluster_upgrade

`--svm_reboot_timeout`

Maximum time expected for SVM to reboot/shutdown.

Default: 420

cluster.consts

`--allow_hetero_sed_node`

Flag that can be set by an SRE to let a node have a mix of sed and non-sed disks.

Default: true

`--app_deployment_progress_zknode`

Zknode to use for deployment state machine

Default: `/appliance/logical/app_deployment_progress`

`--app_deployment_proto_zknode`

Zknode to use for deployment state machine

Default: `/appliance/logical/app_deployment_info`

`--authorized_certs_file`

Path to file containing list of permitted SSL certs.

Default: `/home/nutanix/ssh_keys/AuthorizedCerts.txt`

`--auxiliary_config_json_path`

Path to the auxiliary_config.json file

Default: `/etc/nutanix/auxiliary_config.json`

`--build_last_commit_date_path`

Path to the file that contains the local release version's last commit date.

Default: `/etc/nutanix/build_last_commit_date`

`--cassandra_health_znode`

Zookeeper node where each cassandra creates an ephmeral node indicating it is currently available.

Default: `/appliance/logical/health-monitor/cassandra`

`--cluster_disabled_services`

Zookeeper node where a service profile is represented asthe set of services to disable.

Default: `/appliance/logical/cluster_disabled_services`

`--command_timeout_secs`

Number of seconds to spend retrying an RPC request.

Default: 180

`--compute_only_enabled`

Boolean signifying CO feature support in current NOS

Default: true

`--convert_cluster_zknode`

Holds information about cluster conversion operations and current status for each node.

Default: `/appliance/logical/genesis/convert_cluster`

`--csr_cn_entry`

Common name to use instead of <node_uuid>.nutanix.com

Default: `None`

`--csr_cn_suffix`

Suffix to use instead of nutanix.com when creating CSR

Default: `nutanix.com`

`--cvm_certs_file`

File containing SSL certs of all CVMs in the cluster.

Default: `/home/nutanix/ssh_keys/cvm_certs`

`--cx4_rdma_support`

Boolean to control whether to enable/disable CX4 RDMA feature

Default: true

`--default_cvm_password`

Default password for the CVM.

Default: `nutanix/4u`

`--default_disable_services_file`

Path to the default_disabled_services.json file.

Default: `/home/nutanix/config/genesis/service_profiles/default_disabled_services.json`

`--default_host_password`

Default password for the hypervisor.

Default: `nutanix/4u`

`--default_remote_shell_receive_timeout_secs`

The default timeout for idle connections after which the connection is terminated if there is no activity.

Default: 3600

`--default_remote_shell_socket_timeout_secs`

Timeout for the socket connecting to NutanixHostAgent.

Default: 300

`--default_remote_shell_timeout_secs`

The default timeout for completion of a powershell command made by calling the execute method.

Default: 3600

`--default_upgrade_info_zknode`

Location in a zookeeper where we keep the Upgrade node information.

Default: `/appliance/logical/upgrade_info/nos`

`--degraded_node_policy_dir`

Zookeeper directory where we store the degraded_node_policy

Default: `/appliance/logical/zookeeper-monitor`

`--degraded_node_policy_path`

Zookeeper location to store the degraded node shutdown policy.

Default: `/appliance/logical/zookeeper-monitor/degraded_node_policy`

`--dell_ptagent_password`

Password for dell ptagent user 'ptuser'

Default: `Dellam123`

`--dell_ptagent_port`

Port of dell PTagent REST service

Default: `8086`

`--dell_ptagent_user_name`

User name for dell ptagent

Default: `ptuser`

`--deployment_container_name`

Name for the private container for app images.

Default: `NutanixManagementShare`

`--deployment_info_zknode`

Zknode to keep download info related to uvm deployments

Default: `/appliance/logical/deployment_info`

`--disable_cluster_sync_marker_path`

Path of the marker to disable Cluster Sync.

Default: `/home/nutanix/.disable_cluster_sync`

`--disk_diagnostics_asup_path`

Path for disk diagnostics ASUP data.

Default: `/home/nutanix/data/serviceability/disk_diagnostics_run.json`

`--enable_hyperv_internal_switch_ha_monitoring`

Flag that can be set by an SRE to disable the HyperV HA algorithm of monitoring the internal switch health.

Default: false

`--enable_legacy_aplos_uwsgi_stack`

If True, then the legacy Aplos stack with uwsgi is started. If False, v3 APIs are routed through Mercury.

Default: true

`--ergon_register_name`

Name of component to be registered with ergon service.

Default: `Genesis`

`--esx_cvm_backplane_portgroup`

Portgroup for backplane interface created in CVM.

Default: `CVM Backplane Network`

`--esx_external_vswitch`

External vswitch name in ESX host.

Default: `vSwitch0`

`--esx_host_backplane_portgroup`

Portgroup for backplane interface created in ESXi host.

Default: `Backplane Network`

`--expand_cluster_info_znode`

Location in Zookeeper where we keep the info for nodes to be added.

Default: `/appliance/logical/expand_cluster_info`

`--expand_cluster_status_znode`

Location in Zookeeper where the list for nodes which are currently being added by Genesis are kept.

Default: `/appliance/logical/genesis/expand_cluster_status`

`--expand_cluster_whitelist_znode`

Location in Zookeeper where we keep the whitelist containing allowed hypervisors for imaging new nodes.

Default: `/appliance/logical/expand_cluster_whitelist`

`--factory_config_json_path`

Path to the factory_config.json file.

Default: `/etc/nutanix/factory_config.json`

`--firmware_config_json_path`

Path to json file storing information about firmware of hardware devices such as BIOS and BMC.

Default: `/etc/nutanix/firmware_config.json`

`--firmware_installer_storage`

Directory where the firmware installer is located.

Default: `/home/nutanix/data/installer/firmware`

`--firmware_upgrade_fatal_failure`

Firmware upgrade failed and hardware is not working.

Default: `fatal`

`--firmware_upgrade_params_znode`

Zookeeper location to store firmware upgrade parameters.

Default: `/appliance/logical/upgrade_info/firmware_upgrade_params`

`--firmware_upgrade_recoverable_failure`

Firmware upgrade failed but hardware is functioning ok.

Default: `recoverable`

`--firmware_znode`

Zookeeper node where we keep firmware upgrade information.

Default: `/appliance/logical/genesis/firmware`

`--foundation_port`

Port on which foundation service listens.

Default: 8000

`--foundation_upgrade_flock_path`

Path to the flock in order that must be acquired before running the foundation upgrade script.

Default: `/home/nutanix/foundation_upgrade_lock`

`--foundation_upgrade_info_znode`

Location in a zookeeper where we keep the foundationUpgrade node information.

Default: `/appliance/logical/upgrade_info/foundation`

`--foundation_upgrade_installer_dir`

Directory where all foundation upgrade related packages are stored.

Default: `/home/nutanix/data/installer/foundation`

`--foundation_upgrade_params_znode`

Zookeeper location to store foundation upgrade paramteters.

Default: `/appliance/logical/upgrade_info/foundation_upgrade_params`

`--foundation_url_port`

Foundation service port.

Default: `8000`

`--foundation_version`

Zookeeper location to store foundation version.

Default: `/appliance/logical/foundation/version`

`--frozen_mounts_file`

Path to the file that indicates that Nutanix mountpoints are currently frozen.

Default: `/tmp/frozen_nutanix_mounts`

`--genesis_bin_dir`

Directory where all of the Genesis scripts are located.

Default: `/home/nutanix/cluster/bin`

`--genesis_jsonrpc_url`

URL of the JSON RPC handler on the Genesis HTTP server.

Default: `/jsonrpc`

`--genesis_libs`

Directory where genesis libraries are located.

Default: `/home/nutanix/cluster/lib`

`--genesis_path`

Path to genesis command on SVMs.

Default: `/home/nutanix/cluster/bin/genesis`

`--genesis_port`

Port that Genesis listens on.

Default: 2100

`--genesis_rpc_timeout_secs`

Timeout for each Genesis RPC.

Default: 60

`--genesis_wal_zk_path`

Path to zknode that will be used as a wal to implement any genesis operations idempotently.

Default: `/appliance/logical/genesis/_genesis_wal`

`--hades_config_znode_dir`

Parent zookeeper directory for hades config per CVM.

Default: `/appliance/physical/hades/configuration`

`--hades_location`

Location of Hades binary.

Default: `/usr/local/nutanix/bootstrap/bin/hades`

`--hades_znode_dir`

Parent zookeeper directory for hades information.

Default: `/appliance/physical/hades`

`--hardware_config_json_path`

Path to the hardware_config.json file.

Default: `/etc/nutanix/hardware_config.json`

`--hardware_config_znode_path`

Zookeeper node base path containing the hardware configurations of each node.

Default: `/appliance/physical/hardware_configs`

`--hcl_json_path`

Path to the hcl.json file.

Default: `/etc/nutanix/hcl.json`

`--health_monitor_znode`

Zookeeper node where ephmeral nodes of services are created.

Default: `/appliance/logical/health-monitor`

`--host_bootdisk_repair_preprocess_znode`

Location in Zookeeper where the status of the host bootdisk repair preprocess is kept.

Default: `/appliance/logical/genesis/host_bootdisk_preprocess`

`--host_bootdisk_repair_status_znode`

Location in Zookeeper where the status of the host bootdisk repair is kept.

Default: `/appliance/logical/genesis/host_bootdisk_repair_status`

`--host_bundle_name`

Base name of the host bundle to be copied to other CVM and hosts.

Default: `host-bundle`

`--host_ssh_key`

Location of ssh key for accessing the CVM's host.

Default: `/home/nutanix/ssh_keys/host`

`--host_ssl_cert`

Location of ssl cer for accessing the CVM's Hyperv host.

Default: `/home/nutanix/ssh_keys/host.cer`

`--host_upgrade_flock_path`

Path to the flock in order that must be acquired before running host upgrade

Default: `/tmp/host_upgrade_lock`

`--host_upgrade_installer_dir`

Directory where all hypervisor upgrade related bundle are stored.

Default: `/home/nutanix/data/installer/hypervisor/`

`--host_znode`

Zookeeper node where we keep the host related dataof the cluster.

Default: `/appliance/logical/genesis/hypervisor`

`--hpssacli_location`

Location of hpssacli utility.

Default: `/usr/local/nutanix/cluster/lib/hp/hpssacli`

`--hyperv_details_znode`

Location in Zookeeper where we keep the HyperV details, like domain info, failover cluster info, used when adding new hyperv nodes to cluster.

Default: `/appliance/logical/genesis/hyperv_details`

`--hyperv_external_vswitch_name`

Default name of the network switch for Hyper-V's external network.

Default: `ExternalSwitch`

`--hyperv_failed_internal_switch_health_znode`

Zookeeper node where genesis on each node monitors the health of their internal switch and publishes if bad.

Default: `/appliance/logical/genesis/hyperv_failed_internal_switch_health`

`--hyperv_host_backplane_adapter`

Backplane Adapter created in Hyperv host.

Default: `Backplane`

`--hyperv_hypervisor_username`

The username to use when logging into the local Hyper-V node.

Default: `Administrator`

`--hyperv_internal_adapter_macaddr`

MAC address of the internal network adapter connected to the CVM in Hyper-V nodes.

Default: `00:15:5D:00:00:80`

`--hyperv_internal_vswitch_name`

Default name of the network switch for Hyper-V's internal network.

Default: `InternalSwitch`

`--hyperv_ipmicfg_relative_path`

Path of the ipmicfg utility in a Hyper-V node relative to the Nutanix directory.

Default: `\ipmicfg\IPMICFG-Win.exe`

`--hyperv_lsiutil_relative_path`

Path of the LSIUtil utility in a Hyper-V node relative to the Nutanix directory.

Default: `\Utils\LSIUtil.exe`

`--hyperv_megacli64_relative_path`

Path of the MegaCli64 utility in a Hyper-V node relative to the Nutanix directory.

Default: `\Utils\MegaCli64.exe`

`--hyperv_powershell_path`

The path to the powershell binary on the Hyper-V host.

Default: `${env:windir}\System32\WindowsPowerShell\v1.0\powershell.exe`

`--hyperv_sas2ircu_relative_path`

Path of the sas2ircu utility in a Hyper-V node relative to the Nutanix directory.

Default: `\Utils\sas2ircu.exe`

`--hyperv_sas3ircu_relative_path`

Path of the sas3ircu utility in a Hyper-V node relative to the Nutanix directory.

Default: `\Utils\sas3ircu.exe`

`--hyperv_sg_raw_path`

Path to the utility to allow sg_raw on Hyper-V clusters.

Default: `/home/nutanix/bin/winsg_raw`

`--hyperv_sg_scan_relative_path`

Path of the sg_scan utility in a Hyper-V node relative to the Nutanix directory.

Default: `\Utils\sg_scan.exe`

`--hyperv_sg_ses_relative_path`

Path of the sg_ses utility in a Hyper-V node relative to the Nutanix directory.

Default: `\Utils\sg_ses.exe`

`--hyperv_upgrade_min_space`

Minimum space required for hyperv host upgrade.

Default: 6291456

`--hyperv_upgrade_status_znode`

Location where the status of the hyperv upgrade is kept.

Default: `/appliance/logical/genesis/hyperv_upgrade_status`

`--hypervisor_esxi_smartctl_path`

Path to the smartctl binary on ESX.

Default: `/scratch/smartctl`

`--hypervisor_esxi_smartctl_wrapper_path`

Path to the smartctl wrapper script on ESX.

Default: `/smartctl`

`--hypervisor_imaging_foundation_zknode`

Holds IP address of CVM which needs to be reimaged using foundation service.

Default: `/appliance/logical/genesis/hypervisor_imaging/foundation`

`--hypervisor_imaging_zknode`

Holds details of the node which needs to be reimaged using foundation service.

Default: `/appliance/logical/genesis/hypervisor_imaging`

`--hypervisor_installer_dir`

Path at which various hypervisor installer tarballs are kept.

Default: `/home/nutanix/software_downloads/hypervisor`

`--hypervisor_installer_info_znode`

Location in Zookeeper where we keep the info about the hypervisor installers.

Default: `/appliance/logical/software_info/hypervisor_installer`

`--hypervisor_internal_ip`

Internal IP address of the hypervisor.

Default: `192.168.5.1`

`--hypervisor_ipmicfg_menu_path`

Path to the ipmicfg Menu.dat on hypervisor.

Default: `/Menu.dat`

`--hypervisor_ipmicfg_path`

Path to the ipmicfg binary on hypervisor.

Default: `/ipmicfg`

`--hypervisor_release_version_path`

File on hypervisor that contains the local AHV version.

Default: `/etc/nutanix-release`

`--hypervisor_username`

The username to use when logging into the local hypervisor.

Default: `root`

`--installer_storage`

Directory where the installer is located.

Default: `/home/nutanix/data/installer`

`--ip6tables_backup_path`

Path to the backup of ip6tables file

Default: `/home/nutanix/config/salt_templates/ip6tables.backup`

`--ip6tables_path`

Path to the ip6tables file

Default: `/etc/sysconfig/ip6tables`

`--ipmi_password`

The password to use when logging into the local IPMI device.

Default: `ADMIN`

`--ipmi_username`

The username to use when logging into the local IPMI device.

Default: `ADMIN`

`--iptables_backup_path`

Path to the backup of iptables file

Default: `/home/nutanix/config/salt_templates/iptables.backup`

`--iptables_path`

Path to the iptables file

Default: `/etc/sysconfig/iptables`

`--key_management_server_test_result`

Path to zknode that will store key management server test results.

Default: `/appliance/logical/genesis/kms_test_result`

`--kmip_user_id`

Username to insert into certificate signing request.

Default: `nutanix`

`--kvm_external_vswitch`

External vswitch name in KVM host.

Default: `br0`

`--kvm_host_backplane_interface`

Backplane interface created in KVM host.

Default: `br0-backplane`

`--la_jolla_destroy_info_znode`

Location in Zookeeper where we keep La Jolla destroy specific info.

Default: `/appliance/logical/genesis/la_jolla_destroy_info`

`--la_jolla_destroy_mount_path`

Location of mount path for la_jolla_destroy.

Default: `/tmp/la_jolla_destroy`

`--la_jolla_directory_name`

Name of the directory where LaJolla contents will besaved on disk.

Default: `lajolla`

`--la_jolla_info_znode`

Location in Zookeeper where we keep La Jolla specific info.

Default: `/appliance/logical/genesis/la_jolla_info`

`--la_jolla_marker_file`

Marker file name for la_jolla.

Default: `workload.json`

`--la_jolla_mount_path`

Location of mount path for la_jolla.

Default: `/tmp/la_jolla`

`--la_jolla_pre_deploy_info_znode`

Location in Zookeeper where we keep La Jolla pre deployment specific info.

Default: `/appliance/logical/genesis/la_jolla_pre_deploy_info`

`--la_jolla_share`

Name of the share containing La Jolla contents.

Default: `/ManagementShare`

`--local_esxi_smartctl_path`

Local path to the smartctl binary for ESX.

Default: `/usr/local/nutanix/bootstrap/lib/smartctl.esx`

`--local_esxi_smartctl_wrapper_path`

Local path to the smartctl wrapper for ESX.

Default: `/usr/local/nutanix/bootstrap/lib/smartctl.esx.sh`

`--local_ipmicfg_menu_path`

Path to the ipmicfg Menu.dat on the local host.

Default: `/usr/local/nutanix/cluster/lib/esx5/Menu.dat`

`--local_ipmicfg_path`

Path to the ipmicfg binary on the local host.

Default: `/usr/local/nutanix/cluster/lib/esx5/IPMICFG-Linux.x86_64`

`--local_timezone_path`

Path to the file that contains the local timezone.

Default: `/etc/localtime`

`--logpipe_path`

Path to logpipe.

Default: `/home/nutanix/bin/logpipe`

`--low_water_mark_file`

File where the time manager logs the time every minute

Default: `/home/nutanix/.low_water_mark.json`

`--maintenance_mode_history_file`

The path of maintenance mode history file which records the time when CVM entered maintenance mode.

Default: `/home/nutanix/config/maintenance_mode.history`

`--manual_discovery_timeout_secs`

Timeout for manual discovery API.

Default: 30

`--mount_dir`

Path of storage dir where disk should be mounted

Default: `/home/nutanix/data/stargate-storage/disks`

`--ndp_timeout_secs`

Timeout for hitting NDP manually.

Default: 20

`--network_segmentation_enable_znode`

Location in Zookeeper where the network segmentation enable parameters and status are kept.

Default: `/appliance/logical/genesis/network_segmentation_params`

`--node_in_firmware_upgrade_znode`

This node exists if some node is undergoing firmware upgrade of hardware components. It stores information about that node.

Default: `/appliance/logical/genesis/firmware/upgrade_node`

`--node_shutdown_token_priority_list`

Location in Zookeeper where we keep list of svm ips which are given preference for token.

Default: `/appliance/logical/genesis/node_shutdown_priority_list`

`--node_ssh_key`

Location of ssh key for accessing remote CVMs and hosts.

Default: `/home/nutanix/.ssh/id_rsa`

`--node_ssl_cert`

Location of ssl cer for accessing remote Hyperv hosts.

Default: `/home/nutanix/.ssh/id_rsa.cer`

`--non_ha_vm_info`

Location where the non-HA VMs configuration file path is stored

Default: `/appliance/logical/genesis/non_ha_vm_info`

`--nos_installer_dir`

Path at which NOS installer tarballs are kept.

Default: `/home/nutanix/software_downloads/nos`

`--ntp_external_netif`

Network interface of SVM for communicating with NTP server.

Default: `all`

`--ntpd_driftfile_path`

Path to the NTPD daemon's drift file.

Default: `/var/lib/ntp/drift`

`--nutanix_bin_dir`

Path to the nutanix bin folder.

Default: `/home/nutanix/bin`

`--nutanix_binary_log_dir`

Directory containing binary logs.

Default: `/home/nutanix/data/binary_logs`

`--nutanix_core_dir`

Directory containing core dumps of all Nutanix binaries.

Default: `/home/nutanix/data/cores`

`--nutanix_data_dir`

Path to the nutanix data folder.

Default: `/home/nutanix/data`

`--nutanix_default_ssh_key`

Nutanix default SSH key used for logging into SVM.

Default: `/home/nutanix/ssh_keys/nutanix`

`--nutanix_home`

Nutanix home directory.

Default: `/home/nutanix`

`--nutanix_host_agent_port`

The port on which the NutanixHostAgent service listens on the Hyper-V host.

Default: 3071

`--nutanix_lock_dir`

Directory containing Nutanix service lock files.

Default: `/home/nutanix/data/locks`

`--nutanix_storage_dir`

Nutanix storage directory

Default: `/home/nutanix/data/stargate-storage/disks`

`--pc_deployment_ephemeral`

Zknode to use for task tracking

Default: `/appliance/logical/pc_deployment_ephemeral`

`--persistent_foundation_cvms_zk_path`

Zookeeper node to track which cvms have foundation started in a 'persistent' mode

Default: `/appliance/logical/genesis/persistent_foundation`

`--phoenix_firmware_installer_path`

Path to the module in phoenix image which does firmware install.

Default: `/phoenix/firmware_installer.py`

`--phoenix_firmware_installer_sh_path`

Path to the shell script in phoenix image which triggers firmware install.

Default: `/phoenix/firmware_installer.sh`

`--planned_outage_zk_path`

Expect ZK child nodes under this path that are IP addresses of nodes that are going down.

Default: `/appliance/logical/genesis/planned_outage`

`--prism_monitor_port`

Default port used by Prism Monitor.

Default: 2019

`--release_tag_path`

Path to the file that contains the local release tag.

Default: `/etc/nutanix/release_tag`

`--release_version_path`

Path to the file that contains the local release version.

Default: `/etc/nutanix/release_version`

`--rolling_restart_znode`

Location in Zookeeper where the intent of rollingrestart is committed.

Default: `/appliance/logical/genesis/rolling_restart_znode`

`--rolling_svm_update_znode`

Location in Zookeeper where the intent of rollingsvm update is committed.

Default: `/appliance/logical/genesis/rolling_svm_update_znode`

`--salt_command_path`

Path to the script to run salt related state changes.

Default: `/srv/salt/statechange`

`--salt_state_configuration_directory`

Directory where the current salt states are maintained.

Default: `/home/saltstates`

`--sas3ircu_location`

Location of sas3ircu utility.

Default: `/home/nutanix/cluster/lib/lsi-sas/sas3ircu`

`--sasircu_location`

Location of sas2ircu utility.

Default: `/home/nutanix/cluster/lib/lsi-sas/sas2ircu`

`--secure_files_zkpath`

Path of the Zookeeper node storing Azure certificates.

Default: `/appliance/logical/secure_files`

`--sed_certs_file_name`

Name of the file containing list of signed certs of the node.

Default: `SEDCerts.txt`

`--sed_flock_path`

File to use as a lock to ensure orderly trusted_storage_device calls to the drives.

Default: `/tmp/sed_tool_lock`

`--sed_status_flush_seconds`

Number of seconds to wait before flushing the status of a self encrypting drive operation.

Default: 10

`--sed_tool_path`

Path to the self_encrypting_drive binary.

Default: `/usr/local/nutanix/bootstrap/bin/self_encrypting_drive`

`--self_encrypting_drive_config_zkpath`

Path to the zookeeper node where self encrypting drive configuration is stored.

Default: `/appliance/logical/genesis/sed_state`

`--self_encrypting_drives_operation_status`

Path to the zookeeper node where each drive will store it's status after an operation was issued

Default: `/appliance/logical/genesis/sed_operation_status`

`--service_vm_config_json_path`

Path to the service_vm_config.json file.

Default: `/home/nutanix/data/stargate-storage/service_vm_config.json`

`--services_definition`

Zookeeper node where a service profile stored.

Default: `/appliance/logical/services_definition`

`--sg_raw_path`

Path to the sg_raw binary.

Default: `/usr/bin/sg_raw`

`--sg_ses_location`

Location of sg_ses utility.

Default: `/usr/local/nutanix/cluster/lib/sg3utils/bin/sg_ses`

`--single_ssd_repair_status_znode`

Location in Zookeeper where the status for node which is currently being repaired by Genesis is kept.

Default: `/appliance/logical/genesis/single_ssd_repair_status`

`--ssd_threshold_to_enable_multiqueue_nic`

Number of SSDs needed to be on the node to enable features like multi-queue

Default: 4

`--ssh_key_dir`

Location of ssh keys and ssl certificates

Default: `/home/nutanix/ssh_keys`

`--stargate_disk_config`

JSON file on stargate disk holding disk config

Default: `disk_config.json`

`--stargate_ha_term_conn_ack_timeout_msecs`

The stargate NfsTerminate ACK timeout value to be set. This is used only when there are hyper-v nodes in the cluster. This value is set on all stargates before upgrade and also set persistently in stargate.gflags

Default: 30000

`--stargate_health_znode`

Zookeeper node where each stargate creates an ephmeral node indicating it is currently available.

Default: `/appliance/logical/health-monitor/stargate`

`--stargate_port`

Port for Stargate.

Default: 2009

`--storcli64_location`

Path to storcli utility.

Default: `/home/nutanix/cluster/lib/storcli/storcli64`

`--svm_backplane_netif`

Backplane network interface of SVM to communicate with other nodes within cluster.

Default: `eth2`

`--svm_ca_bundle_path`

Path to the CA certificate bundle file.

Default: `/home/nutanix/certs/ca.pem`

`--svm_ca_certs_dir`

Directory where CA certificates are cached on the filesystem.

Default: `/home/nutanix/certs/CA_certs`

`--svm_certs_dir`

Directory where certificates are cached on the filesystem.

Default: `/home/nutanix/certs`

`--svm_csr_path`

Path to local Svm's certificate signing request file.

Default: `/home/nutanix/certs/svm.csr`

`--svm_eth1_mtu_size`

eth1 MTU size configured in the SVMs.

Default: 1500

`--svm_external_netif`

External network interface of SVM to communicate with other SVMs.

Default: `eth0`

`--svm_internal_ip`

IP address of the SVM IP address on the internal Nutanix vSwitch.

Default: `192.168.5.2`

`--svm_internal_netif`

Internal network interface of the SVM to communicate with local hypervisor.

Default: `eth1`

`--svm_internal_sub_netif`

Internal network sub-interface of the SVM to communicate with local hypervisor.

Default: `eth1:1`

`--svm_mtu_size`

eth0 MTU size configured in the SVMs.

Default: 1500

`--svm_non_ha_internal_ip`

Internal IP address of SVM that is not redirected for HA.

Default: `192.168.5.254`

`--svm_os_version`

The default version of SVM os

Default: `centos`

`--svm_private_key_path`

Path to local Svm's digital certificates private key.

Default: `/home/nutanix/certs/svm.key`

`--svm_self_signed_cert_path`

Path to local Svm's self signed certificate.

Default: `/home/nutanix/certs/svm.crt`

`--svmboot_iso_upgraded`

Name of the customized svmboot iso used during AOS upgrade

Default: `svmboot.iso.upgraded`

`--svmboot_xenserver_iso`

xenserver specific svmboot -- to workaround kexec issues

Default: `svmboot-415-xenserver.iso`

`--tentative_backplane_ips`

Location in Zookeeper where the node tentative ips used for network segmentation would be kept.

Default: `/appliance/logical/genesis/tentative_backplane_ips`

`--test_lcm_rpc_enable`

Flag that can be set to allow lcm to expose rpc call to automation test.

Default: false

`--upgrade_finish_flock_path`

Path to the flock in order that must be acquired before running the finish script.

Default: `/tmp/upgrade_finish_lock`

`--upgrade_flock_path`

Path to the flock in order that must be acquired before running the upgrade install script.

Default: `/tmp/upgrade_lock`

`--upgrade_info_firmware_znode`

Location in a zookeeper where we keep the upgrade information for firmware upgrades published by Prism.

Default: `/appliance/logical/upgrade_info/firmware`

`--upgrade_info_znode`

Location in a zookeeper where we keep the Upgrade node information.

Default: `/appliance/logical/upgrade_info/nos`

`--upgrade_info_znode_dir`

Parent Directory of location in a zookeeper where we keep the Upgrade node information.

Default: `/appliance/logical/upgrade_info`

`--upgrade_params_znode`

Zookeeper location to store upgrade paramteters.

Default: `/appliance/logical/upgrade_info/upgrade_params`

`--vcenter_info`

zk node of host to vcenter info

Default: `/appliance/logical/genesis/vcenter_info`

`--vmxnet3_rx_queue_size`

RX buffer size for vmxnet3.

Default: 2048

`--vmxnet3_tx_queue_size`

TX buffer size for vmxnet3.

Default: 512

`--witness_config_cache_path`

Path to the witness configuration cache kept in sync by witness manager

Default: `/home/nutanix/config/witness_config_proto.dat`

`--witness_config_zknode`

Zookeeper node containing the witness config once it is created.

Default: `/appliance/witness_config`

`--xen_details_znode`

Location in Zookeeper where we keep the Xen details, like credentials, etc when adding new xen nodes to cluster.

Default: `/appliance/logical/genesis/xen_details`

`--xi_cluster_ssh_key`

Xi Cluster SSH key used to login using the cluster vipcreated by DCM

Default: `/home/nutanix/xi_ssh_key_dir/nutanix`

`--xi_ssh_key_dir`

Location of ssh keys specific to the cluster created by DCM

Default: `/home/nutanix/xi_ssh_key_dir`

`--zeus_config_cache_path`

Path to the zeus configuration cache kept in sync by Genesis.

Default: `/home/nutanix/config/configuration_proto.dat`

`--zkmigration_wal_path`

Path to the zookeeper migration local WAL.

Default: `/home/nutanix/data/zookeeper_monitor/zookeeper_migration.wal`

`--zkquorum_change_wal_path`

Path to the zookeeper quorum change local WAL.

Default: `/home/nutanix/data/zookeeper_monitor/quorum_change.wal`

`--zookeeper_ca_certs_zkdir`

Path to the zookeeper directory node where CA certificates are stored.

Default: `/appliance/logical/certs/CA_certs`

`--zookeeper_certs_zkdir`

Path to the zookeeper directory node where Svm certificates are stored.

Default: `/appliance/logical/certs`

`--zookeeper_host_port_list`

Host port pair list for zookeeper.

Default: `zk1:9876,zk2:9876,zk3:9876`

`--zookeeper_migration_zknode`

Path to the zookeeper migration status zknode.

Default: `/appliance/logical/zookeeper_migration`

cluster.container.docker.utils

`--default_volume_plugin_name`

Name of default docker nutanix volume plugin

Default: `pc/nvp`

`--default_volume_plugin_type`

Default docker volume plugin type

Default: `default`

`--docker_systemd_service`

Name of the systemd docker service

Default: `docker-latest`

`--docker_volume_plugin_binary_path`

Path to volume plugin install script

Default: `/home/nutanix/bin/create_plugin_from_tar.sh`

`--docker_volume_plugin_image_path`

Path to docker volume plugin image

Default: `/usr/local/nutanix/volume-plugin/dvp.tar.gz`

`--volume_plugin_install_timeout_secs`

Timeout in secs for volume plugin installation

Default: 60

`--volume_plugin_version_znode`

Path to docker volume plugin image

Default: `/appliance/logical/genesis/volume_plugin`

cluster.deployment.deployment_utils

`--default_password_reset_timeout`

Timeout in seconds for the executing the password reset script on the PC VM.

Default: 90

cluster.disk_flags

`--clean_disk_log_path`

Path to the logs from the clean_disks script.

Default: `/home/nutanix/data/logs/clean_disks.log`

`--clean_disk_script_path`

Path to the clean_disks script.

Default: `/home/nutanix/cluster/bin/clean_disks`

`--disk_partition_margin`

Limit for the number of bytes we will allow to be unpartitioned on a disk.

Default: 2147483648

`--disk_size_threshold_percent`

Percentage of available disk space to be allocated to stargate

Default: 95

`--enable_all_ssds_for_oplog`

DEPRECATED: Use all ssds attached to this node for oplog storage.

Default: true

`--enable_fio_realtime_scheduling`

Use realtime scheduling policy for fusion io driver.

Default: false

`--fio_realtime_priority`

Priority for fusion io driver, when realtime scheduling policy is being used.

Default: 10

`--format_fusion_percent`

The percentage of total capacity of fusion-io drives that should be formatted as usable

Default: 60

`--max_ssds_for_oplog`

Maximum number of ssds used for oplog per node. If value is -1, use all ssds available. If only_select_nvme_disks_for_oplog gflag is true and NVMe disks are present, only NVMe disks are used for selecting oplog disks.

Default: 8

`--metadata_maxsize_GB`

Maximum size of metadata in GB

Default: 30

`--only_select_nvme_disks_for_oplog`

If true and NVMe disks are present, only use NVMe disks for selecting oplog disks.

Default: true

`--path_to_setscheduler_binary`

Path to setscheduler binary, which is used to set realtime priority for fusion io driver.

Default: `/home/nutanix/bin/setscheduler`

`--skip_metadata_link_setup`

Skip creation of metadata links (Use rootfs for storing metadata

Default: false

`--skip_scsi_bus_rescan`

Skip rescanning scsi bus while running disk prep

Default: false

`--striped_models_csv`

A comma-separated list of disk models that should be striped together to form a single logical device.

Default: `SSD_910_200GB`

cluster.esx_upgrade_helper

`--esx_vib_extraction_dir`

Directory where ESXi vibs are extracted on CVM before copying to host.

Default: `/home/nutanix/tmp/.esx_upgrade`

`--foundation_esx_vib_path`

Path in foundation package where ESX VIBs are stored.

Default: `/home/nutanix/foundation/lib/driver/esx/vibs`

`--poweroff_uvms`

Power off UVMs during hypervisor upgrade if Vmotion is not enabled or Vcenter is not configured for cluster.

Default: false

`--update_foundation_vibs`

Update VIBS which are present in foundation during ESX hypervisor upgrade.

Default: true

cluster.firewall.consts

`--cluster_function_temp_file`

Path to temporary file which has the cluster function

Default: `/home/nutanix/tmp/cluster_function`

`--consider_salt_framework`

Whether to consider salt framework or not

Default: true

`--execute_concurrent_salt_call`

Indicate if salt call should be executed concurrently.

Default: true

`--iptables_command_timeout_secs`

Timeout in seconds for any iptables command.

Default: 10

`--iptables_salt_blacklist_path`

Path to salt config for iptables state.

Default: `/srv/pillar/blacklist.sls`

`--iptables_salt_config_path`

Path to salt config for iptables state.

Default: `/srv/pillar/iptables.sls`

`--iptables_temp`

Path to temporary iptables file

Default: `/home/nutanix/tmp/iptables.temp`

`--jumpbox_vm_deployments`

Comma separated list of deployments where Jumpbox rules are applied

Default: `512`

`--prism_client_key`

Client key value for prism

Default: `prism.nfs_whitelists`

`--salt_call_command_path`

Path to the salt-call command.

Default: `/usr/bin/salt-call`

`--salt_call_command_timeout_secs`

Timeout in seconds for the salt-call command.

Default: 30

`--salt_dynamic_blacklist`

Path to the dynamic template

Default: `/home/nutanix/config/salt_templates/dynamic_blacklist.json`

`--salt_dynamic_blacklist_temp`

Path to the temporary dynamic blacklist template

Default: `/home/nutanix/config/salt_templates/dynamic_blacklist_temp.json`

`--salt_dynamic_template`

Path to the dynamic template

Default: `/home/nutanix/config/salt_templates/dynamic_open_list.json`

`--salt_dynamic_template_temp`

Path to the temporary dynamic template

Default: `/home/nutanix/config/salt_templates/dynamic_open_list_temp.json`

`--salt_jinja_template`

Path to salt ipv4 jinja template.

Default: `/srv/salt/security/CVM/iptables/iptables4.jinja`

`--salt_pc_jinja_template`

Path to salt ipv4 jinja template on PC.

Default: `/srv/salt/security/PC/iptables/iptables4.jinja`

`--salt_states_templates_dir`

Path to dir holding the salt templates.

Default: `/home/nutanix/config/salt_templates`

`--unsupported_firewall_deployments`

Comma separated list of deployments where firewall will FATAL out

Default: `128,64,8`

`--valid_firewall_deployments`

Comma separated list of deployments where firewall uses salt

Default: `1,16,2,4,256,512,32,4096`

cluster.genesis.breakfix.host_bootdisk_graceful

`--clone_bootdisk_default_timeout`

The default timeout for completion of cloning of bootdisk.

Default: 28800

`--restore_bootdisk_default_timeout`

The default timeout for completion of restore of bootdisk.

Default: 14400

`--wait_for_phoenix_boot_timeout`

The maximum amount of time for which the state machine waits after cloning for the node, to be booted in phoenix environment.

Default: 36000

cluster.genesis.breakfix.host_bootdisk_utils

`--host_boot_timeout`

The maximum amount of time for which the state machine waits for host to be up.

Default: 36000

cluster.genesis.breakfix.ssd_breakfix

`--ssd_repair_copy_svmrescue_timeout`

Timeout for copying svmrescue.iso from CVM to host

Default: 600

cluster.genesis.breakfix.ssd_breakfix_esx_helper

`--svm_regex`

Regular expression used to find the SVM vmx name.

Default: `ServiceVM`

cluster.genesis.cluster_manager

`--agave_dir`

Identify if agave is running on cluster.

Default: `/home/nutanix/agave`

`--cloud_start_dynamic_ring_changer`

Whether to start Dynamic ring changer for cloud nodes.Dynamic Ring changer is required to run for some time to add new metadata disk. This may be required in customer escalations if existing metadata disk becomes full.

Default: false

`--cluster_firmware_upgrade_params`

Zookeeper location to store upgrade parameters for firmware upgrades.

Default: `/appliance/logical/genesis/firmware_upgrade_params`

`--cluster_state_znode`

Location in Zookeeper where we keep whether a node start or stop.

Default: `/appliance/logical/genesis/cluster_state`

`--cluster_upgrade_method`

Location in Zookeeper where we keep upgrade method.

Default: `/appliance/logical/genesis/cluster_upgrade_method`

`--cluster_versions_znode`

Location in Zookeeper where we keep the desired software versions map.

Default: `/appliance/logical/genesis/cluster_versions`

`--cvm_reboot_wait`

Timeout for waiting for cvm reboot.

Default: 100

`--firmware_preupgrade_path`

Location to store script that runs preupgrade tests before running firmware upgrade.

Default: `/home/nutanix/cluster/bin/`

`--force_disable_blackbox`

File to disable blackbox mode completely.

Default: `/home/nutanix/.force_disable_blackbox`

`--foundation_uncompress_path`

Location for uncompressing foundation package.

Default: `/home/nutanix/software_uncompressed/foundation/`

`--node_shutdown_token_state_znode`

Location in Zookeeper where we keep state of which node has currently requested to go down for maintenance.

Default: `/appliance/logical/genesis/node_shutdown_token`

`--node_upgrade_status`

Location in Zookeeper where we store upgrade status of nodes.

Default: `/appliance/logical/genesis/node_upgrade_status`

`--prism_user_repository_znode`

Zookeeper node of the Prism user repository

Default: `/appliance/physical/userrepository`

`--rdma_config_timeout`

Timeout for waiting for RDMA configuration to complete.

Default: 300

`--release_tag_znode`

Zookeeper node where we keep the current release tag of the cluster.

Default: `/appliance/logical/genesis/release_tag`

`--release_version_znode`

Zookeeper node where we keep the current release version of the cluster.

Default: `/appliance/logical/genesis/release_version`

`--shutdown_token_timeout`

Timeout for waiting for shutdown token.

Default: 60

`--uncompress_path`

Location for uncompressing nutanix binaries.

Default: `/home/nutanix/software_uncompressed/nos/`

cluster.genesis.compute_only.client

`--configured_marker_file`

Path to the marker file containing cluster id if node is part of a cluster

Default: `/root/configured`

cluster.genesis.compute_only.consts

`--factory_config_json_path_on_host`

Path to factory_config.json on the CO host

Default: `/root/factory_config.json`

`--hardware_config_json_path_on_host`

Path to hardware_config.json on the CO host

Default: `/root/hardware_config.json`

cluster.genesis.convert_cluster.utils

`--cluster_conversion_preserve_mac`

Preserve MAC addresses of VM NICs in conversion

Default: true

`--convert_cluster_blacklisted_vms`

List of VM UUIDs which won't be converted during cluster conversion

Default: `/appliance/logical/genesis/convert_cluster/blacklisted_vms`

`--convert_cluster_disable_marker`

Marker file to disable hypervisor conversion on node.

Default: `/home/nutanix/.convert_cluster_disable`

`--convert_cluster_node_ids`

List of node ids which will be converted to target hypervisor

Default: `/appliance/logical/genesis/convert_cluster/converting_node_ids`

`--converting_vm_info`

Path to zk node where the reg info of all VMs undergoing conversion is stored

Default: `/appliance/logical/genesis/convert_cluster/converting_vm`

`--default_vcenter_port`

Default port to register with Vcenter.

Default: 443

`--fail_vm_uuids_conversion`

Comma separated list of VM UUIDs which will fail vm conversion

`--fail_vm_uuids_power_off`

Comma separated list of VM UUIDs which will fail vm power off operation during conversion

`--fail_vm_uuids_power_on`

Comma separated list of VM UUIDs which will fail vm power on operation during conversion

`--ignore_reverse_conversion_checks`

Ignore checks performed in reverse conversion eg. presence of ESXi cluster name in Vcenter and existence of metadata zknode, etc.

Default: false

cluster.genesis.convert_cluster.vm_migration

`--disable_vm_migration`

Disable VM migration for the node. This is used for error injection and testing.

Default: false

cluster.genesis.expand_cluster.expand_cluster

`--node_up_retries`

Number of retries for node genesis rpcs to be up after reboot

Default: 40

cluster.genesis.expand_cluster.utils

`--nos_packages_file`

File containing packages present in the nos software

Default: `install/nutanix-packages.json`

`--nos_tar_timeout_secs`

Timeout in secs for tarring nos package

Default: 3600

cluster.genesis.la_jolla.la_jolla

`--add_la_jolla_disk`

Flag to add La Jolla disk back

Default: true

cluster.genesis.la_jolla.la_jolla_utils

`--nfs_buf_size`

NFS buffer size

Default: 8388608

cluster.genesis.network_segmentation_helper

`--disable_wait_time`

Time to wait (in seconds), between removing network segmentation configuration in zeus and removing the interface configuration on cvms

Default: 5

`--ns_state_machine_timeout`

The timeout for completion of network segmentation state machine.

Default: 600

`--retry_count_zk_map_publish`

Retry count for publishing new zk mapping.

Default: 3

`--revert_ns_config_on_failure`

Revert the network segmentation configuration in the case of a failure.

Default: true

cluster.genesis.node_manager

`--auto_discovery_interval_secs`

Number of seconds to sleep when local node can't join any discovered cluster.

Default: 5

`--co_nodes_unconfigure_marker`

Path to marker file to indicate that node has to unconfigure CO nodes as part of unconfiguring itself. The contents of the marker file containsspace seperated IPs of the CO nodes to unconfigure

Default: `/home/nutanix/.co_nodes_unconfigure`

`--download_staging_area`

Directory where we will download directories from other SVMs.

Default: `/home/nutanix/tmp`

`--firmware_disable_auto_upgrade_marker`

Path to marker file to indicate that automatic firmware upgrade should not be performed on this node.

Default: `/home/nutanix/.firmware_disable_auto_upgrade`

`--foundation_disable_auto_upgrade_marker`

Path to marker file to indicate that automatic foundation upgrade should not be performed on this node.

Default: `/home/nutanix/.foundation_disable_auto_upgrade`

`--genesis_restart_required_path`

Marker file to indicate that genesis restart is required during upgrade.

Default: `/home/nutanix/.genesis_restart_required_path`

`--genesis_restart_timeout`

Time we wait for the genesis to restart.

Default: 120

`--gold_image_version_path`

Path to the file that contains the version of the gold image.

Default: `/etc/nutanix/svm-version`

`--hcl_znode_path`

Zookeeper node containing the hcl.

Default: `/appliance/physical/hcl`

`--la_jolla_unconfigure_marker`

Path to marker file to indicate that LaJolla is ready to be unconfigured.

Default: `/home/nutanix/.la_jolla_unconfigure`

`--mdns_service_type`

Avahi service type to publish under.

Default: `_nutanix._tcp`

`--mdns_timeout_secs`

Timeout for mdns browse service.

Default: 1

`--move_time_back`

The flag is set to allow time to be moved back by more than time_rollback_tolerance_secs.

Default: false

`--nagios_config_path`

Path to the nagios configuration file.

Default: `/home/nutanix/serviceability/config/nagios3/nutanix_nagios.cfg`

`--node_disable_auto_upgrade_marker`

Path to marker file to indicate that automatic software upgrade should not be performed on this node.

Default: `/home/nutanix/.node_disable_auto_upgrade`

`--node_reconfigure_marker`

Path to marker file to indicate that this node should be reconfigured.

Default: `/home/nutanix/.node_reconfigure`

`--node_ssh_key_dir`

Path for node specific ssh keys on local disk

Default: `/home/nutanix/ssh_keys/.blackbox`

`--node_unconfigure_marker`

Path to marker file to indicate that node is ready to be unconfigured.

Default: `/home/nutanix/.node_unconfigure`

`--raid_sync_default_wait`

Default retry count for firmware upgrade.

Default: 60

`--raid_sync_retry_count`

Default retry count for firmware upgrade.

Default: 1000

`--rpm_genesis_log_file`

Path to rpm log file during Genesis self install.

Default: `/home/nutanix/data/logs/rpm.genesis.out`

`--sshd_config_path`

Path to sshd config file.

Default: `/etc/ssh/sshd_config`

`--stop_service_zknode`

Start services upto but not including the service specified in this zk node.

Default: `/appliance/logical/genesis/stop_service`

`--svm_internal_ips`

Internal IP addresses on eth1. The first one is the primary IP address, and the remaining are aliases.

Default: `192.168.5.2,192.168.5.254`

`--svm_internal_netif_netmask`

The netmask for the internal IP addresses on eth1.

Default: `255.255.255.128`

`--svm_non_ha_internal_netmask`

The netmask for the non-data internal IP aliases on eth1.

Default: `255.255.255.0`

`--timezones_dir`

Directory where all of the valid timezones exist.

Default: `/usr/share/zoneinfo`

`--vlan_sniffer_log`

Path to vlan_sniffer log.

Default: `/home/nutanix/data/logs/vlan_sniffer.log`

`--vlan_sniffer_path`

Path to vlan sniffer proxy.

Default: `/home/nutanix/cluster/bin/vlan_sniffer_proxy`

`--zookeeper_server_config_file`

zookeeper server config file.

Default: `/home/nutanix/data/zookeeper_monitor/zk_server_config_file`

`--zookeeper_start_timeout_secs`

Timeout for waiting on Zookeeper connection on startup.

Default: 120

cluster.genesis.rdma_helper

`--check_rdma_switch_config_script`

Script to check RDMA interface and port config

Default: `/usr/local/nutanix/cluster/bin/check_rdma_switch_config`

`--mellanox_tc_wrap`

Path to the Mellanox's tc_wrap.py script

Default: `/usr/local/nutanix/bin/tc_wrap.py`

`--rdma_nic_config_file`

Path to json containing the mac of the nic to be used for rdma

Default: `/etc/nutanix/nic_config.json`

cluster.genesis.resource_management.rm_helper

`--common_pool_map`

Mapping of node with its common pool memory in kb

Default: `/appliance/logical/genesis/common_pool_map`

`--common_pool_mem_for_low_mem_nodes_gb`

Common pool memory reservation for nodeswith cvm memory less than 20gb

Default: 8

`--default_common_pool_memory_in_gb`

Stargate default common pool memory reservation

Default: 12

`--memory_update_history`

File containing history of memory update on node

Default: `/home/nutanix/config/memory_update.history`

`--memory_update_resolution`

Minumum amount of memory difference for update

Default: 2097152

`--rolling_restart_memory_update_reason`

Reason set in rolling restart for memory update

Default: `cvm_memory_update`

`--target_memory_zknode`

CVM target memory map zk node

Default: `/appliance/logical/genesis/target_memory_map`

cluster.genesis.resource_management.rm_prechecks

`--cushion_memory_in_kb`

Cushion Memory required in nodes before update

Default: 2097152

`--delta_memory_for_nos_upgrades_kb`

Amount of CVM memory to be increased during NOS upgrade

Default: 4194304

`--host_memory_threshold_in_kb`

Min host memory for memory update , set to 62 Gb

Default: 65011712

`--max_cvm_memory_upgrade_kb`

Maximum allowed CVM memory for update during upgrade

Default: 31457280

cluster.genesis.resource_management.rm_tasks

`--cvm_reconfig_component`

Component for CVM reconfig

Default: `kGenesis`

`--cvm_reconfig_operation`

Component for CVM reconfig

Default: `kCvmreconfig`

cluster.genesis.service_management.service_mgmt_utils

`--core_services_managed`

This Flag will be used to force service mgmt to enable/disable core services.

Default: false

cluster.genesis_utils

`--orion_config_path`

Path to orion config

Default: `/appliance/logical/orion/config`

`--svm_default_login`

User name for logging into SVM.

Default: `nutanix`

`--timeout_HA_route_verification`

Timeout for setting HA route.

Default: 180

`--timeout_zk_operation`

Timeout for zk operation like write

Default: 120

`--upgrade_fail_marker`

Marker to indicate upgrade has failed.

Default: `/appliance/logical/genesis/upgrade_failed`

cluster.hades.client

`--hades_jsonrpc_url`

URL of the JSON RPC handler on the Hades HTTP server.

Default: `/jsonrpc`

`--hades_port`

Port that Hades listens on.

Default: 2099

`--hades_rpc_timeout_secs`

Timeout for each Hades RPC.

Default: 30

cluster.hades.disk_diagnostics

`--hades_retry_count`

Default retry count.

Default: 5

`--max_disk_offline_count`

Maximum error count for disk after which disk is to removed.

Default: 3

`--max_disk_offline_timeout`

Maximum time value where disk offline events areignored.

Default: 3600

cluster.hades.disk_manager

`--aws_cores_partition`

Partition in which core files are stored on AWS.

Default: `/dev/xvdb1`

`--boot_part_size`

The size of a regular boot partition in 512-byte sectors.

Default: 20969472

`--device_mapper_name`

A name of the device mapper that is to be created in case striped devices are discovered.

Default: `dm0`

`--disk_unmount_retry_count`

Number of times to retry unmounting the disk.

Default: 60

`--firmware_upgrade_default_wait`

Default wait time after issuing firmware upgrade.

Default: 10

`--hdparm_path`

Location of hdparm.

Default: `/sbin/hdparm`

`--home_nutanix_size`

Size of the /home/nutanix partition in 512-byte sectors.

Default: 83884032

`--lsiutil_location`

Location of lsiutil utility.

Default: `/home/nutanix/cluster/lib/lsi-sas/lsiutil`

`--sas3flash_location`

Location of sas3flash utility.

Default: `/home/nutanix/cluster/lib/lsi-sas/sas3flash`

`--sas3flash_target_runtime`

Approximate time taken by sas3flash for upgrading HBA

Default: `300`

`--sas_disk_firmware_upgrade_retry_count`

Numter of times to retry SAS disk firmware upgrade.

Default: 60

`--sata_disk_firmware_upgrade_retry_count`

Default retry count for firmware upgrade.

Default: 12

`--sg_start_path`

Location of sg_start binary.

Default: `/usr/local/nutanix/cluster/lib/sg3utils/bin/sg_start`

`--sg_write_buffer_path`

Location of sg_write_buffer. The version of the current binary used is 1.18 20141107.

Default: `/usr/local/nutanix/cluster/lib/sg3utils/bin/sg_write_buffer`

`--skip_disk_remove_reboot`

Skip reboot for bad or removed disk.

Default: false

`--skip_disk_remove_reboot_marker`

Skip reboot marker for bad or removed disk.

Default: `/home/nutanix/.skip_disk_remove_reboot_marker`

`--ssd_read_ahead_size_kb`

Read ahead size in KB for SSDs. Min: 0 Max: 256 KB

Default: 16

`--ssd_rules_path`

Location of the SSD udevadm rules file.

Default: `/etc/udev/rules.d/99-ssd.rules`

`--stargate_directory`

Directory where stargate disks should be mounted.

Default: `/home/nutanix/data/stargate-storage`

`--unmount_timeout`

Default wait time for unmounting to succeed.

Default: 120

cluster.hades.raid_utils

`--min_raid_sync_speed`

Minimum raid sync speed

Default: 50000

cluster.host_upgrade_common

`--host_poweroff_uvm_file`

File containing list of UVM which are powered off for host upgrade.

Default: `/home/nutanix/config/.host_poweroff_vm_list`

`--upgrade_delay`

Seconds to wait before runninguprgade script in Esx Host.

Default: 0

cluster.host_upgrade_helper

`--host_disable_auto_upgrade_marker`

Path to marker file to indicate that automatic host upgrade should not be performed on this node.

Default: `/home/nutanix/.host_disable_auto_upgrade`

`--hypervisor_upgrade_history_file`

File path where hypervisor upgrade history is recorded.

Default: `/home/nutanix/config/hypervisor_upgrade.history`

`--hypervisor_upgrade_info_znode`

Location in a zookeeper where we keep the Hypervisor upgrade information.

Default: `/appliance/logical/upgrade_info/hypervisor`

cluster.hyperv_upgrade

`--vmms_setting_info`

Location where the vmms settings are stored

Default: `/appliance/logical/genesis/vmms_setting_info`

cluster.hyperv_upgrade_helper

`--perpetual_container`

Container name to test if hypervisors have access to storage.

Default: `NutanixManagementShare`

`--upgrade_helper_rshell_timeout`

The default timeout for RemoteShell connection made by HyperVHostObject.

Default: 600

cluster.ipv4config

`--end_linklocal_ip`

End of the range of link local IP4 addresses.

Default: `169.254.254.255`

`--esx_cmd_timeout_secs`

Default timeout for running a remote command on an ESX host.

Default: 120

`--hyperv_cmd_timeout_secs`

Default timeout for running a remote command on an hyperv host.

Default: 120

`--ipmi_apply_config_retries`

Number of times to try applying an IPMI IPv4 configuration before failing.

Default: 6

`--kvm_cmd_timeout_secs`

Default timeout for running a remote command on an KVM host.

Default: 120

`--kvm_external_network_interface`

Default name of the network device for KVM's external network.

Default: `br0`

`--linklocal_netmask`

End of the range of link local IP4 addresses.

Default: `255.255.0.0`

`--start_linklocal_ip`

Start of the range of link local IP4 addresses.

Default: `169.254.1.0`

`--xen_external_network_interface`

Default name of the network device for Xen's external network.

Default: `xapi1`

cluster.kvm_upgrade_helper

`--ahv_enter_maintenance_mode_retry_max_delay_secs`

Max delay time for exponential backoff retries to enter maintenance

Default: 720

`--ahv_enter_maintenance_mode_retry_slot_time_secs`

Slot time for exponential backoff retries to enter maintenance

Default: 120

`--ahv_enter_maintenance_mode_timeout_secs`

Seconds to wait before retrying enter maintenance mode operation on failures

Default: 7200

`--ahv_remove_unknown_packages_on_upgrade`

Remove RPMs not in AHV manifest file on upgrade

Default: false

`--kvm_reboot_delay`

Seconds to delay before reboot KVM host

Default: 30

cluster.license_config

`--license_config_file`

Zookeeper path where license configuration is stored.

Default: `/appliance/logical/license/configuration`

`--license_config_proto_file`

License configuration file shipped with NOS.

Default: `configuration.cfg`

`--license_dir`

License feature set files directory shipped with NOS.

Default: `/home/nutanix/serviceability/license`

`--license_public_key`

License public key string shipped with NOS.

Default: `/appliance/logical/license/public_key`

`--license_public_key_str`

License public key string shipped with NOS.

Default: `public_key.pub`

`--zookeeper_license_root_path`

Zookeeper path where license information is stored.

Default: `/appliance/logical/license`

cluster.lite_upgrade.core.consts

`--cluster_sync_path`

Path to cluster_sync command on SVMs.

Default: `/home/nutanix/cluster/bin/cluster_sync`

`--hades_path`

Path to hades command on SVMs.

Default: `/home/nutanix/cluster/bin/hades`

`--lite_upgrade_running_zknode`

Zk node that tracks if lite upgrade is running.

Default: `/appliance/logical/genesis/lite_upgrade/active`

cluster.lite_upgrade.interfaces.genesis_interface

`--genesis_lu_intent_zknode`

Lite upgrade zk node, set with target version.

Default: `/appliance/logical/genesis/lite_upgrade/genesis_intent`

cluster.multihome_utils

`--multihome_zkpath`

Marker to indicate if any node of cluster is multihome.

Default: `/appliance/logical/genesis/multihome`

cluster.ncc_upgrade_helper

`--cluster_health_shutdown_max_retries`

Max number of retries to shutdown cluster health.

Default: 5

`--cluster_health_shutdown_threshold_ms`

Time threshold (ms) between cluster health shutdown retries.

Default: 2000

`--ncc_installation_path`

Location where NCC is installed on a CVM.

Default: `/home/nutanix/ncc`

`--ncc_num_nodes_to_upload`

Number of nodes to upload the NCC installer directory to.

Default: 2

`--ncc_uncompress_path`

Location for uncompressing nutanix NCC binaries.

Default: `/home/nutanix/data/ncc/installer`

`--ncc_upgrade_info_znode`

Location in a zookeeper where we keep the Upgrade node information.

Default: `/appliance/logical/upgrade_info/ncc`

`--ncc_upgrade_params_znode`

Zookeeper location to store NCC upgrade parameters.

Default: `/appliance/logical/upgrade_info/ncc_upgrade_params`

`--ncc_upgrade_status`

Location in Zookeeper where we store upgrade status of nodes.

Default: `/appliance/logical/genesis/ncc_upgrade_status`

`--ncc_upgrade_timeout_secs`

Timeout in seconds for the NCC upgrade module.

Default: 30

`--ncc_version_znode`

Zookeeper node where we keep the current release version of NCC.

Default: `/appliance/logical/genesis/ncc_version`

cluster.preupgrade_checks

`--arithmos_binary_path`

Path to the arithmos binary.

Default: `/home/nutanix/bin/arithmos`

`--cluster_external_state`

Cluster external state zk path

Default: `/appliance/physical/clusterexternalstate`

`--connected_cluster_path`

Connected cluster zk path

Default: `/appliance/physical/zeusconfig`

`--license_config_path`

The path to the license configuration.

Default: `/appliance/logical/license/configuration`

`--min_disk_space_for_upgrade`

Minimum space (KB) required on /home/nutanix for upgrade to proceed.

Default: 3600000

`--min_replication_factor`

Minimum replication factor required per container.

Default: 2

`--minimum_memory_for_prism_pro_in_kb`

Minimum memory needed for prism pro features.

Default: 16777216

`--mountsfile`

Path to the mounts file in proc.

Default: `/proc/mounts`

`--prism_gateway_port`

The port on which prism gateway is running.

Default: 9440

`--role_mapping_path`

Role mapping zk path

Default: `/appliance/logical/prism/rolemapping`

`--signature_file_extension`

Extension of the signature file.

Default: `.asc`

`--signature_verification_available_version`

Release version in which gpg signature validation is supported.

Default: `5.8`

`--stargate_mount_prefix`

Mount point prefix for stargate disks

Default: `/home/nutanix/data/stargate-storage/disks/`

cluster.preupgrade_checks_ncc_helper

`--ncc_temp_location`

Location to extract NCC.

Default: `/home/nutanix/ncc_preupgrade`

cluster.rsyslog_helper

`--lock_dir`

Default path for nutanix lock files.

Default: `/home/nutanix/data/locks/`

`--log_dir`

Default path for nutanix log files.

Default: `/home/nutanix/data/logs`

`--module_level`

Level of syslog used for sending module logs

Default: `local0`

`--rsyslog_conf_file`

Default Configuration file for Rsyslog service.

Default: `/etc/rsyslog.d/rsyslog-nutanix.conf`

`--rsyslog_configure_queue`

Boolean indicating whether to configure action queue for rsyslog.

Default: true

`--rsyslog_queue_memory_size`

Size of rsyslog remote logging action queue in bytes.

Default: 104857600

`--rsyslog_rule_header`

Nutanix specified rsyslog rules are appended only below this marker.

Default: `# Nutanix remote server rules`

`--rsyslog_rule_header_end`

Nutanix specified rsyslog rules are added above this marker.

Default: `# Nutanix remote server rules end`

`--rsyslog_work_dir`

Default path for syslog state files. This stores thestate of rsyslog across restarts.

Default: `/var/lib/rsyslog`

`--syslog_module`

Dummy module name to denote non nutanix default syslog.

Default: `syslog_module`

cluster.service.cluster_config_service

`--cluster_config_path`

Path to the Cluster Config binary.

Default: `/home/nutanix/bin/cluster_config`

`--cluster_config_server_rss_mem_limit`

Maximum amount of resident memory Cluster Config may use on an Svm with 8GB memory configuration.

Default: 268435456

cluster.service.curator_service

`--curator_config_json_file`

JSON file with curator configuration

Default: `curator_config.json`

`--curator_data_dir_size`

Curator data directory size in MB (80 GB).

Default: 81920

`--curator_data_dir_symlink`

Path to curator data directory symlink.

Default: `/home/nutanix/data/curator`

`--curator_data_disk_subdir`

Path to curator subdirectory in data disk.

Default: `curator`

`--curator_oom_score`

If -1, OOM is disabled for this component. If in [1, 1000], it is taken as the OOM score to be applied for this component.

Default: -1

`--curator_path`

Path to the curator binary.

Default: `/home/nutanix/bin/curator`

`--curator_rss_mem_limit`

Maximum amount of resident memory Curator may use on an Svm with 8GB memory configuration.

Default: 536870912

cluster.service.foundation_service

`--foundation_path`

Path to the foundation service script.

Default: `/home/nutanix/foundation/bin/foundation`

`--foundation_rss_mem_limit`

Maximum amount of resident memory Foundation may use on a CVM with 8GB memory configuration.

Default: 1073741824

cluster.service.ha_service

`--def_stargate_stable_interval`

Default number of seconds a stargate has to be alive to be considered asstable and healthy.

Default: 30

`--hyperv_internal_switch_health_timeout`

Timeout for how long we should wait before waking up the thread that monitors internal switch health on HyperV.

Default: 30

`--num_worker_threads`

The number of worker threads to use for running tasks.

Default: 8

`--old_stop_ha_zk_node`

When this node is created the old ha should not take any actions on the cluster.

Default: `/appliance/logical/genesis/ha_stop`

`--stargate_aggressive_monitoring_secs`

Default number of seconds a stargate is aggressively monitored after it is down.

Default: 3

`--stargate_aggressive_monitoring_two_node_multiplier`

Multiplier to the --stargate_aggressive_monitoring_secsfor the two node cluster.

Default: 4

`--stargate_exit_handler_aggressive_timeout_secs`

Aggressive timeout for accessing the Stargate exit handler page during an unplanned failover.

Default: 3

`--stargate_exit_handler_timeout_secs`

Default timeout for accessing the Stargate exit handler page.

Default: 10

`--stargate_health_watch_timeout`

Timeout for how long we should wait before waking up the thread that monitors stargate health.

Default: 30

`--stargate_initialization_secs`

Number of seconds to wait for stargate to initialize.

Default: 30

`--stop_ha_zk_node`

When this node is created ha 2.0 should not take any actions on the cluster.

Default: `/appliance/logical/genesis/ha_2_stop`

`--tcpkill_el6_path`

Path to el6 tcpkill binary.

Default: `/home/nutanix/cluster/lib/el6/tcpkill`

`--tcpkill_log`

Path to tcpkill binary.

Default: `/home/nutanix/data/logs/tcpkill.log`

`--tcpkill_path`

Path to el7 tcpkill binary.

Default: `/home/nutanix/cluster/lib/el7/tcpkill`

cluster.service.kafka_service

`--kafka_bootstrap_binary`

Path to binary which will start kafka on PC in docker container

Default: `/usr/local/nutanix/bin/bootstrap_kafka`

`--kafka_data_volume_mode`

Volume mode to provide data volume to kafka. If host attached, vmdisk is attached to PC VM. Else, docker volume plugin is used to conenct to remote PE cluster iscsi endpoint

Default: `host_attached`

`--kafka_disk_size_factor_wrt_data_disk`

Kafka disk size = data disk size / <this constant>

Default: 7

`--kafka_disks_base_dir`

Base directory of kafka data disks. Each disk is mounted with directory name being disk serial on this base dir. Applicable to host attached disks

Default: `/home/nutanix/data/kafka/disks/`

`--kafka_pc_deployed_disk_scsi_id`

When PC is deployed, kafka disk is attached at this scsi id

Default: 4

`--kafka_rss_mem_limit`

Maximum amount of resident memory Kafka may use on an svm with 8 GB memory configuration.

Default: 268435456

cluster.service.service_utils

`--auto_set_cloud_gflags`

If true, recommended gflags will be automatically set for cloud instances.

Default: true

`--cgroup_subsystems`

Default subsystems used for cgroup creation.

Default: `cpu,cpuacct,memory,freezer,net_cls`

`--containersvcmon_path`

Path to container service monitor binary.

Default: `/home/nutanix/bin/containersvcmon`

`--enable_service_monitor`

If true, C based service_monitor will re-spawn the service process upon exit, else Python based self_monitor will be used.

Default: true

`--memory_limits_base_size_kb`

Total memory size of the standard Svm based on which memory limits are derived.

Default: 8388608

`--path_to_cgclassify_binary`

Path to cgclassify binary, which is used to add a process into cgroup.

Default: `/bin/cgclassify`

`--path_to_cgcreate_binary`

Path to cgcreate binary, which is used to create a cgroup.

Default: `/bin/cgcreate`

`--path_to_cgset_binary`

Path to cgset binary, which is used to set parameter for cgroup.

Default: `/bin/cgset`

`--path_to_chrt_binary`

Path to chrt binary, which is used to set realtime priority for a process.

Default: `/usr/bin/chrt`

`--path_to_root_cgroup`

Path to root cgroup directory.

Default: `/dev/cgroup`

`--path_to_taskset_binary`

Path to taskset binary, which is used to set cpu affinity for a process.

Default: `/bin/taskset`

`--service_monitor_path`

Path to the service monitor binary.

Default: `/home/nutanix/bin/service_monitor`

`--service_start_as_container`

Flag to indicate service can start as containers

Default: true

`--service_stop_wait`

Wait time for how long we should wait before we start issuing SIGKILL instead of SIGTERM.

Default: 5

cluster.service.zookeeper_service

`--zookeeper_client_port`

TCP port number for zookeeper service clients.

Default: 9876

`--zookeeper_config_template`

Path to zookeeper configuration template.

Default: `/home/nutanix/cluster/config/zoo.cfg.template`

`--zookeeper_cpuset_exclude_cpu_default`

Default CPU affinity. This is a comma seperated list of cpus to be excluded for this component. If specified as -1 then cpuset exclude cpu is disabled for this component.

Default: `0`

`--zookeeper_data_dir`

Path to zookeeper data directory.

Default: `/home/nutanix/data/zookeeper`

`--zookeeper_env_path`

Path to shell script with zookeeper environment.

Default: `/etc/profile.d/zookeeper_env.sh`

`--zookeeper_init`

Path to the zookeeper_init tool.

Default: `/home/nutanix/bin/zookeeper_init`

`--zookeeper_init_max_retry_attempts`

The number of times zookeeper_init would retry if it fails. -1 means that it would retry indefinitely

Default: 10

`--zookeeper_leader_election_port`

TCP port number for zookeeper service leader election.

Default: 3888

`--zookeeper_lib_dir`

Path to zookeeper jars.

Default: `/home/nutanix/lib/zookeeper`

`--zookeeper_max_snapshots`

Maximum number of snapshots to retain when compacting the zookeeper data

Default: 5

`--zookeeper_monitor`

Path to the zookeeper monitor binary.

Default: `/home/nutanix/bin/zookeeper_monitor`

`--zookeeper_monitor_component_id_file_path`

Path to file where the local zookeeper component id is stored.

Default: `/home/nutanix/data/zookeeper_monitor/zookeeper_monitor_config.json`

`--zookeeper_monitor_data_dir`

Path to zookeeper monitor data directory.

Default: `/home/nutanix/data/zookeeper_monitor`

`--zookeeper_monitor_self_configure_marker`

Marker file to indicate that Zookeeper monitor is updated to configure itself.

Default: `/home/nutanix/data/zookeeper_monitor/.zkmonitor_self_configure`

`--zookeeper_myid_path`

Path to zookeeper id file.

Default: `/home/nutanix/data/zookeeper/myid`

`--zookeeper_oom_score`

If -1, OOM is disabled for this component. If in [1, 1000], it is taken as the OOM score to be applied for this component.

Default: -1

`--zookeeper_peer_port`

TCP port number for zookeeper service peers.

Default: 2888

`--zookeeper_realtime_priority_default`

Default priority. If specified as -1 then realtime priority is disabled and default scheduling policy is applied. If specified as positive number between 1 to 99 then that is taken as realtime priority to be applied to this component.

Default: 6

`--zookeeper_rss_mem_limit`

Maximum amount of resident memory Zookeeper may use on an Svm with 8GB memory configuration.

Default: 268435456

`--zookeeper_zoo_cfg_path`

Path to zookeeper config, zoo.cfg.

Default: `/home/nutanix/config/zookeeper/zoo.cfg`

cluster.sshkeys_helper

`--authorized_keys_file`

Path to file containing list of permitted RSA keys

Default: `/home/nutanix/.ssh/authorized_keys2`

`--authorized_keys_file_admin`

Path to file containing list of permitted RSA keys for admin.

Default: `/home/admin/.ssh/authorized_keys2`

`--id_rsa_path`

Nutanix default SSH key used for logging into SVM.

Default: `/home/nutanix/.ssh/id_rsa`

`--id_rsa_public_pem_path`

PEM formatted public key for this node.

Default: `/home/nutanix/.ssh/id_rsa.pub.pem`

`--ssh_client_configuration`

Location of ssh client configuration file.

Default: `/home/nutanix/.ssh/config`

`--ssh_config_server_alive_count_max`

Max SSH keepalive messages missed before declaring connection dead.

Default: 3

`--ssh_config_server_alive_interval`

SSH keepalive message interval.

Default: 10

`--ssh_path`

Location of ssh folder for nutanix.

Default: `/home/nutanix/.ssh`

`--ssh_path_admin`

Location of ssh folder for admin.

Default: `/home/admin/.ssh`

cluster.time_manager.time_manager_utils

`--ntpdate_timeout_secs`

Timeout to wait for ntp server to return a valid time.

Default: 10

cluster.two_node.cluster_manager

`--minimum_pass_pings_to_allow_fails`

Default minimum number of pings to peer node that must pass before we allow some pings to fail (controlled by number_of_ping_fails_to_reset_pass_counter flag) and not reset pass counter.

Default: 10

`--minimum_uptime_mins_before_starting_pings`

Minimum system uptime before starting pinging the peer node.

Default: 4

`--minutes_of_successful_pings_for_transition`

Default number of minutes for successful pings beforeconsidering peer node stable and moving to kSwitchToTwoNode mode.

Default: 15

`--node_health_check_ping_interval_secs`

Default time (in seconds) for which health thread sleeps before checking on another node.

Default: 2

`--number_of_ping_fails_to_reset_pass_counter`

Default number of pings to peer node that should failto reset the pass counter.

Default: 4

`--rpc_retry_timeout_seconds`

Default time till when to keep retrying RPCs to peer node before claiming failure.

Default: 600

`--seconds_to_wait_before_checking_ha_forwarding`

Default number of seconds to wait before checking again if host is forwarding storage traffic.

Default: 30

`--seconds_to_wait_before_pinging_upgrading_node`

Default number of seconds to wait before pinging a node that separated due to upgrade.

Default: 600

`--threshold_number_of_consecutive_failed_pings`

Default threshold for number of times a ping to a nodecan fail before it is assumed down.

Default: 4

`--transition_to_standalone_due_to_degraded_cassandra_on_peer`

If True, allow transitions to kStandAlone if peer cassandra is degraded. transition_to_standalone_due_to_degraded_service_on_peer also needs to be true to support this.

Default: true

`--transition_to_standalone_due_to_degraded_service_on_peer`

If True, allow transitions to kStandAlone if peer services are degraded. This gflag is only used to control global support for transition due to any degraded services.

Default: true

`--transition_to_standalone_due_to_degraded_stargate_on_peer`

If True, allow transitions to kStandAlone if peer stargate is degraded. transition_to_standalone_due_to_degraded_service_on_peer also needs to be true to support this.

Default: true

`--unit_test_mode`

If True, running in unit test mode

Default: false

cluster.two_node.state_transitions

`--witness_state_history_size`

Maximum length of witness state history to keep in zeus configuration.

Default: 10

cluster.two_node.witness_manager

`--witness_ping_interval_seconds`

Frequency (in seconds) to check the availability of the witness VM.

Default: 60

`--witness_refresh_cluster_name`

If True, will check whether to update the witness with the current cluster info.

Default: true

cluster.upgrade_helper

`--arithmos_rpc_timeout`

Timeout for arithmos RPCs retries

Default: 180

`--cluster_name_update_timeout`

Default timeout for updating the cluster name in zeus.

Default: 5

`--num_nodes_to_upload`

Number of nodes to upload the installer directory to.

Default: 2

`--nutanix_packages_json_basename`

Base file name of the JSON file that contains the list of packages to expect in the packages directory.

Default: `nutanix-packages.json`

`--uncompress_buffer_ratio`

Space Buffer ratio to uncompress a compressed file.

Default: 0.2

`--upgrade_genesis_restart`

Location in Zookeeper where we store if genesis restart is required or not.

Default: `/appliance/logical/upgrade_info/upgrade_genesis_restart`

cluster.utils.device_mapper_utils

`--stripe_size_sectors`

Stripe chunk size in number of 512-byte sectors.

Default: 128

cluster.utils.foundation_rest_client

`--foundation_ipv6_interface`

Ipv6 interface corresponding to eth0

Default: 2

cluster.utils.foundation_utils

`--foundation_root_dir`

Root directory for foundation in CVM

Default: `/home/nutanix`

cluster.utils.hyperv_ha_utils

`--default_internal_switch_monitor_interval`

Default polling period for monitoring internal switch health

Default: 30

`--default_ping_success_percent`

Default percentage of success which is used to determine switch health

Default: 100

`--default_total_ping_count`

Default number of pings sent to determine switch health

Default: 10

cluster.utils.hypervisor_ha

`--internal_nutanix_portgroup_name`

Name of the internal Nutanix portgroup configured on ESX.

Default: `vmk-svm-iscsi-pg`

`--terminate_connection_timeout_secs`

Timeout in seconds for the NfsTerminateConnection RPC issued to stargate on a failback.

Default: 15

cluster.utils.new_node_nos_upgrade

`--stand_alone_upgrade_log`

Log file for stand-alone upgrade.

Default: `/home/nutanix/data/logs/stand_alone_upgrade.out`

cluster.xen_upgrade_helper

`--xen_maintenance_mode_retries`

Seconds to delay before reboot Xen host

Default: 5

`--xen_reboot_delay`

Seconds to delay before reboot Xen host

Default: 30

`--xen_uvm_no_migration_counter`

Number of retries to wait for UVMs to migrate from xen host after it is put in maintenance mode

Default: 7

`--xen_webserver_port`

Port for webserver that will serve files required during xen upgrades

Default: 8999

util.cluster.consts

`--azure_cert_dir`

Directory in which Azure certificates are stored.

Default: `/home/cloud/azure`

`--cloud_credentials_zkpath`

Zookeeper node path to the cloud credentials node.

Default: `/appliance/logical/cloud_credentials`

`--disable_hades_marker_path`

Path of the marker to disable Hades.

Default: `/home/nutanix/.disable_hades`

`--disk_location_json_path`

Path to the disk_location.json file.

Default: `/etc/nutanix/disk_location.json`

`--hyperv_nutanix_path`

The path of the Nutanix directory on the Hyper-V host.

Default: `${env:ProgramFiles}\Nutanix`

`--megacli64_location`

Location of MegaCli64 utility.

Default: `/usr/local/nutanix/cluster/lib/MegaCli/MegaCli64`

`--nested_esx_marker_path`

Is this nested ESX?

Default: `/home/nutanix/.nested_esx_marker`

`--nutanix_log_dir`

Directory containing logs.

Default: `/home/nutanix/data/logs`

`--zookeeper_timeout_secs`

Zookeeper connection timeout in seconds.

Default: 60

util.infrastructure.cluster

`--infra_service_vm_config_json_path`

Path to the service_vm_config.json file with the svm id.

Default: `/home/nutanix/data/stargate-storage/service_vm_config.json`

diagnostics.py

Usage

```
Usage: /home/nutanix/diagnostics/diagnostics.py [command]

commands:

 cleanup
 delete_disks
 drain_oplog
 list_runtime_test_args
 recreate_disks
 run
 run_iperf
```

/home/nutanix/diagnostics/diagnostics.py

`--add_vms_to_pd`

Whether to add Diagnostic VMs to pd.

Default: false

`--cluster_external_data_services_ip`

Cluster external data services IP

`--collect_cassandra_latency_stats`

Collect cassandra latency stats for each test.

Default: true

`--collect_fio_logs`

Collects fio bw and iops logs

Default: false

`--collect_illuminati_stats`

Grab collect_perf stats to be upload to illuminati during every test.

Default: false

`--collect_iostat_info`

Reads and writes to disk

Default: false

`--collect_sched_stats`

Collect stats related to Linux scheduling in SVM

Default: false

`--collect_stargate_stats`

Grab snapshot of 2009 stargate stats page before and after every test.

Default: false

`--collect_stargate_stats_interval`

Internal in secs to collect stargate stats.

Default: 10

`--collect_stargate_stats_timeout`

Max timeout in secs for stargate stats collection.

Default: 720

`--collect_top_stats`

Collect top stats for each test.

Default: false

`--collect_uvm_stats`

Collects uvm cpu and latency stats.

Default: false

`--ctr_name`

This flag is deprecated. Please use ctr_name_prefix

Default: `NTNX-NFS-DEFAULT`

`--ctr_name_prefix`

Prefix for the container name to be created for the disks of the UVM

Default: `NTNX-NFS-DEFAULT`

`--default_aggregator`

Default script to aggregate results.

Default: `fio_aggregator.py`

`--diagnostics_image_dir`

Local directory for storing images.

Default: `/usr/local/nutanix/data/images/diagnostics`

`--display_latency_stats`

Display latency stats for each test (available in fio tests only).

Default: false

`--dry_run`

If True, only prints the tests it would have executed.

Default: false

`--esx_disk_bus_type`

Which bus to attach ESX vdisks to.

Default: `pvscsi`

`--ext4`

Tests on EXt4 for Minerva.Uses iscsi_in_guest and enables it by default.

Default: false

`--ext4_stargate`

Tests on EXt4 basic stargateSets up formatted vdisks.

Default: false

`--external_uvm_ips`

IP addresses of external UVMs. If specified, the setup step is skipped and the tests are run on these UVMs

`--fingerprint_on_write`

Whether to enable fingerprint on write on container.

Default: false

`--flush_svm_caches`

Whether to flush svm cache before each test.

Default: true

`--frodo`

Turn FRODO ON (AHV Only)

Default: false

`--help`

show this help

Default: 0

`--helpshort`

show usage only for this module

Default: 0

`--helpxml`

like --help, but generates XML output

Default: false

`--hypervisor_password`

Default hypervisor password.

Default: `nutanix/4u`

`--ignore_oplog_flush_failure`

Proceed even if oplog is not completely flushed.

Default: false

`--inline_compression`

Whether to enable inline compression on container.

Default: false

`--iperf_sessions`

Number of parallel iperf sessions.

Default: 1

`--iscsi_in_guest`

Use iscsi attached disks for the test. The virtual IP is used for discovery and attaching targets. 4 vcpu UVMs are used with one target per UVM and 6 VGs per target.

Default: false

`--kvm_disk_bus_type`

Which bus to attach KVM vdisks to.

Default: `scsi`

`--kvm_use_vg`

Whether to use VG for UVM data disks on KVM.

Default: false

`--kvm_vm_extra_flags`

List of semicolon separated key=value pairs for KVM

`--max_oplog_flush_wait_secs`

Maximum time waiting for the oplogs to flush before running a test.

Default: 1800

`--memory_mb`

If non-zero, override the memory size for the uvm.

Default: 0

`--mix_mode_type`

<esx|ahv|hyperv>: Run Diagnostics on a mixed hypervisor cluster. Specify subset of hypervisors to create vm's on.

Default: esx

`--name_server`

DNS Server

Default: `10.4.8.15`

`--nfs_master`

Run the test on the nfs master only. This is intended to be used with the metadata test. NFS master and NFS slave have different performance characteristics. If the --nodes parameter is set, this parameter is ignored.

Default: true

`--nodes`

Comma-separated list of nodes (IP addresses of the hosts) where the tests are run. If not specified, tests are run on all nodes.

`--on_disk_dedup`

Whether to enable on disk dedup on a container.

Default: false

`--pd_name`

Name of pd to add Diagnostics VMs.

Default: `diagnostics_pd`

`--ppc64le_uvm_image_directory`

Directory having the right diagnostics uvm images

Default: `diagnostics-1.0`

`--print_cassandra_latency_stats`

Print cassandra latency stats for each test.

Default: false

`--replication_factor`

Replication factor of the container.

Default: 2

`--results_dir_name`

Name of the results directory. Leave empty for an automatically generated name.

`--retry_delay_seconds`

Number of seconds to wait between successive polling or command attempts.

Default: 1

`--run_gxprof`

Whether to run gxprof to track CPU & heap data.

Default: false

`--run_iperf`

Whether to run Iperf bandwidth test. Cleanup will be run before Iperf, so this incompatible with --skip_setup, since the remaining tests will have to perform setup after Iperf is run.

Default: false

`--runtime_test_args`

Comma-separated list of test arguments of the form key=value, to be passed to the test scripts. Use 'diagnostics.py list_test_args' for a list of supported test args.

`--scripts_dir`

Path to the directory where the uvm tool scripts reside

Default: `/home/nutanix/diagnostics`

`--setup_only`

Only setup the test UVMs without running any actual tests. This option is mutually exclusive with --skip_setup.

Default: false

`--skip_cleanup`

Skip entire diagnostics cleanup. Useful for testing purposes.

Default: false

`--skip_full_clean`

Skips cleaning up container and storage pool.

Default: false

`--skip_setup`

Skip setting up the UVMs. Useful when the setup has already been performed by a previous run. This option is mutually exclusive with --setup_only. This option is also incompatible with --run_iperf because Iperf requires cleanup before it can be run and remaining tests will then require setup to be run.

Default: false

`--sp_name`

Name of the storage pool to be created for the disks of the UVM

Default: `NTNX-SP-DEFAULT`

`--ssh_command_timeout_sec`

Timeout in seconds, for ssh commands to finish.

Default: 90

`--terse`

Terse output - just prints the test name and the BW/IOPs as applicable. Note that any pre-req cluster/uvm setup logs are still emitted.

Default: false

`--test_groups`

Comma-separated list of test groups to run. If not specified, the 'default' test group is run (unless --tests is specified).

Default: `default`

`--tests`

Comma-separated list of tests to run. If not specified, all tests in the specified test groups are run.

`--tunnels_start_port`

Starting port for setting up tunnels to the UVMs

Default: 17000

`--uvm_identity_file`

SSH identity file of the diagnostics VM

Default: `diagnostics_id_rsa`

`--uvm_image_server`

IP of the server having the diagnostics uvm images

Default: `10.4.64.11:8080`

`--uvm_ip`

IP address of the UVM

Default: `192.168.5.253`

`--uvm_name_prefix`

Common prefix of the names of the UVMs

Default: `NTNX_diagnostics`

`--uvm_os_type`

OS type of the diagnostics UVM to run.

Default: `linux`

`--uvms_per_node`

Number of UVMs per node.

Default: 1

`--vcpus`

If non-zero, override the number of vcpus for the uvm.If iscsi_in_guest flag is used, vpcus are set to 4.

Default: 0

`--verbose`

If True, print verbose information to stdout.

Default: false

`--x86_64_uvm_image_directory`

Directory having the right diagnostics uvm images

Default: `diagnostics-2.4`

`--zero_fill`

Boolean whether to zero_fill before first seq write. Zero fill is enabled by default.

Default: true

`--zfs`

Tests on ZFS for Minerva.Uses iscsi_in_guest and enables it by default.

Default: false

`--zfs_flush_caches`

Whether to flush ZFS cache before each test.

Default: true

`--zfs_trim_min_ext_sz`

ZFS trim minimum extent size.

Default: 4096

`--zfs_txgs_per_trim`

ZFS transactions per trim.

Default: 32

genesis

Usage

```
Usage: /usr/local/nutanix/cluster/bin/genesis start|stop [all|<service1>
 [<service2> ...]]|restart|status
```

/usr/local/nutanix/cluster/bin/genesis

`--foreground`

Run Genesis in foreground.

Default: false

`--genesis_debug_stack`

Flag to indicate whether signal handler need to be registered for debugging greenlet stacks.

Default: true

`--genesis_self_monitoring`

Genesis to do self monitoring.

Default: true

`--genesis_upgrade`

Flag to indicate that genesis restarted because it is upgrading itself.

Default: false

`--help`

show this help

Default: 0

`--helpshort`

show usage only for this module

Default: 0

`--helpxml`

like --help, but generates XML output

Default: false

cluster.genesis.compute_only.client

`--configured_marker_file`

Path to the marker file containing cluster id if node is part of a cluster

Default: `/root/configured`

cluster.genesis.compute_only.consts

`--factory_config_json_path_on_host`

Path to factory_config.json on the CO host

Default: `/root/factory_config.json`

`--hardware_config_json_path_on_host`

Path to hardware_config.json on the CO host

Default: `/root/hardware_config.json`

cluster.genesis.convert_cluster.utils

`--cluster_conversion_preserve_mac`

Preserve MAC addresses of VM NICs in conversion

Default: true

`--convert_cluster_blacklisted_vms`

List of VM UUIDs which won't be converted during cluster conversion

Default: `/appliance/logical/genesis/convert_cluster/blacklisted_vms`

`--convert_cluster_disable_marker`

Marker file to disable hypervisor conversion on node.

Default: `/home/nutanix/.convert_cluster_disable`

`--convert_cluster_node_ids`

List of node ids which will be converted to target hypervisor

Default: `/appliance/logical/genesis/convert_cluster/converting_node_ids`

`--converting_vm_info`

Path to zk node where the reg info of all VMs undergoing conversion is stored

Default: `/appliance/logical/genesis/convert_cluster/converting_vm`

`--default_vcenter_port`

Default port to register with Vcenter.

Default: 443

`--fail_vm_uuids_conversion`

Comma separated list of VM UUIDs which will fail vm conversion

`--fail_vm_uuids_power_off`

Comma separated list of VM UUIDs which will fail vm power off operation during conversion

`--fail_vm_uuids_power_on`

Comma separated list of VM UUIDs which will fail vm power on operation during conversion

`--ignore_reverse_conversion_checks`

Ignore checks performed in reverse conversion eg. presence of ESXi cluster name in Vcenter and existence of metadata zknode, etc.

Default: false

cluster.genesis.convert_cluster.vm_migration

`--disable_vm_migration`

Disable VM migration for the node. This is used for error injection and testing.

Default: false

cluster.genesis.la_jolla.la_jolla_utils

`--nfs_buf_size`

NFS buffer size

Default: 8388608

cluster.genesis.node_manager

`--auto_discovery_interval_secs`

Number of seconds to sleep when local node can't join any discovered cluster.

Default: 5

`--co_nodes_unconfigure_marker`

Path to marker file to indicate that node has to unconfigure CO nodes as part of unconfiguring itself. The contents of the marker file containsspace seperated IPs of the CO nodes to unconfigure

Default: `/home/nutanix/.co_nodes_unconfigure`

`--download_staging_area`

Directory where we will download directories from other SVMs.

Default: `/home/nutanix/tmp`

`--firmware_disable_auto_upgrade_marker`

Path to marker file to indicate that automatic firmware upgrade should not be performed on this node.

Default: `/home/nutanix/.firmware_disable_auto_upgrade`

`--foundation_disable_auto_upgrade_marker`

Path to marker file to indicate that automatic foundation upgrade should not be performed on this node.

Default: `/home/nutanix/.foundation_disable_auto_upgrade`

`--genesis_restart_required_path`

Marker file to indicate that genesis restart is required during upgrade.

Default: `/home/nutanix/.genesis_restart_required_path`

`--genesis_restart_timeout`

Time we wait for the genesis to restart.

Default: 120

`--gold_image_version_path`

Path to the file that contains the version of the gold image.

Default: `/etc/nutanix/svm-version`

`--hcl_znode_path`

Zookeeper node containing the hcl.

Default: `/appliance/physical/hcl`

`--la_jolla_unconfigure_marker`

Path to marker file to indicate that LaJolla is ready to be unconfigured.

Default: `/home/nutanix/.la_jolla_unconfigure`

`--mdns_service_type`

Avahi service type to publish under.

Default: `_nutanix._tcp`

`--mdns_timeout_secs`

Timeout for mdns browse service.

Default: 1

`--move_time_back`

The flag is set to allow time to be moved back by more than time_rollback_tolerance_secs.

Default: false

`--nagios_config_path`

Path to the nagios configuration file.

Default: `/home/nutanix/serviceability/config/nagios3/nutanix_nagios.cfg`

`--node_disable_auto_upgrade_marker`

Path to marker file to indicate that automatic software upgrade should not be performed on this node.

Default: `/home/nutanix/.node_disable_auto_upgrade`

`--node_reconfigure_marker`

Path to marker file to indicate that this node should be reconfigured.

Default: `/home/nutanix/.node_reconfigure`

`--node_ssh_key_dir`

Path for node specific ssh keys on local disk

Default: `/home/nutanix/ssh_keys/.blackbox`

`--node_unconfigure_marker`

Path to marker file to indicate that node is ready to be unconfigured.

Default: `/home/nutanix/.node_unconfigure`

`--raid_sync_default_wait`

Default retry count for firmware upgrade.

Default: 60

`--raid_sync_retry_count`

Default retry count for firmware upgrade.

Default: 1000

`--rpm_genesis_log_file`

Path to rpm log file during Genesis self install.

Default: `/home/nutanix/data/logs/rpm.genesis.out`

`--sshd_config_path`

Path to sshd config file.

Default: `/etc/ssh/sshd_config`

`--stop_service_zknode`

Start services upto but not including the service specified in this zk node.

Default: `/appliance/logical/genesis/stop_service`

`--svm_internal_ips`

Internal IP addresses on eth1. The first one is the primary IP address, and the remaining are aliases.

Default: `192.168.5.2,192.168.5.254`

`--svm_internal_netif_netmask`

The netmask for the internal IP addresses on eth1.

Default: `255.255.255.128`

`--svm_non_ha_internal_netmask`

The netmask for the non-data internal IP aliases on eth1.

Default: `255.255.255.0`

`--timezones_dir`

Directory where all of the valid timezones exist.

Default: `/usr/share/zoneinfo`

`--vlan_sniffer_log`

Path to vlan_sniffer log.

Default: `/home/nutanix/data/logs/vlan_sniffer.log`

`--vlan_sniffer_path`

Path to vlan sniffer proxy.

Default: `/home/nutanix/cluster/bin/vlan_sniffer_proxy`

`--zookeeper_server_config_file`

zookeeper server config file.

Default: `/home/nutanix/data/zookeeper_monitor/zk_server_config_file`

`--zookeeper_start_timeout_secs`

Timeout for waiting on Zookeeper connection on startup.

Default: 120

cluster.genesis.resource_management.rm_helper

`--common_pool_map`

Mapping of node with its common pool memory in kb

Default: `/appliance/logical/genesis/common_pool_map`

`--common_pool_mem_for_low_mem_nodes_gb`

Common pool memory reservation for nodeswith cvm memory less than 20gb

Default: 8

`--default_common_pool_memory_in_gb`

Stargate default common pool memory reservation

Default: 12

`--memory_update_history`

File containing history of memory update on node

Default: `/home/nutanix/config/memory_update.history`

`--memory_update_resolution`

Minumum amount of memory difference for update

Default: 2097152

`--rolling_restart_memory_update_reason`

Reason set in rolling restart for memory update

Default: `cvm_memory_update`

`--target_memory_zknode`

CVM target memory map zk node

Default: `/appliance/logical/genesis/target_memory_map`

cluster.genesis.resource_management.rm_prechecks

`--cushion_memory_in_kb`

Cushion Memory required in nodes before update

Default: 2097152

`--delta_memory_for_nos_upgrades_kb`

Amount of CVM memory to be increased during NOS upgrade

Default: 4194304

`--host_memory_threshold_in_kb`

Min host memory for memory update , set to 62 Gb

Default: 65011712

`--max_cvm_memory_upgrade_kb`

Maximum allowed CVM memory for update during upgrade

Default: 31457280

cluster.genesis.resource_management.rm_tasks

`--cvm_reconfig_component`

Component for CVM reconfig

Default: `kGenesis`

`--cvm_reconfig_operation`

Component for CVM reconfig

Default: `kCvmreconfig`

cluster.genesis.server.server

`--genesis_document_root`

Document root where static files are served from.

Default: `/home/nutanix/cluster/www`

`--genesis_server_timeout_secs`

Timeout for rpc made through http server.

Default: 30

cluster.genesis_utils

`--orion_config_path`

Path to orion config

Default: `/appliance/logical/orion/config`

`--svm_default_login`

User name for logging into SVM.

Default: `nutanix`

`--timeout_HA_route_verification`

Timeout for setting HA route.

Default: 180

`--timeout_zk_operation`

Timeout for zk operation like write

Default: 120

`--upgrade_fail_marker`

Marker to indicate upgrade has failed.

Default: `/appliance/logical/genesis/upgrade_failed`

ncc

Usage

```
nutanix@cvm$ /home/nutanix/ncc/bin/ncc [flags]
```

/home/nutanix/ncc/bin/ncc

`--generate_plugin_config_template`
Generate plugin config for the plugin
Default: false

`--health_checks_and_log_collector`
Flag to run health_checks and log_collector together.
Default: false

`--help`
show this help
Default: 0

`--helpshort`
show usage only for this module
Default: 0

`--helpxml`
like --help, but generates XML output
Default: false

`--ncc_logging_dir`
Directory where script log files are stored.
Default: `/home/nutanix/data/logs`

`--ncc_run_on_dev_vm`
Whether NCC is running on dev vm or not.
Default: false

`--ncc_version`
Show script version.
Default: false

`--show_version`
Show detailed NCC version.
Default: false

`--version`
Show script version.
Default: false

ncc.ncc_logger

`--enable_plugin_wise_logging`
Enabling this flag will add plugin name to log record during plugin run
Default: true

ncc.plugins.consts

`--HDD_latency_threshold_ms`

HDD await threshold (ms/command) to determine Disk issues.

Default: 500

`--SSD_latency_threshold_ms`

SSD await threshold (ms/command) to determine Disk issues.

Default: 50

`--avg_io_latency_threshold_ms`

Average I/O Latency threshold (ms) for a VM.

Default: 5000

`--cas_failures_threshold`

Threshold count for cas failure log warnings.

Default: 5

`--cassandra_column_families`

Comma separated list of column families to check against size thresholds.

Default: `historical_stats`

`--cassandra_crashes_threshold`

Threshold for number of cassandra crashes.

Default: 5

`--cassandra_gc_activity_threshold`

Threshold for garbage collection related messages.

Default: 5

`--cassandra_high_heap_usage_threshold`

Threshold for number of heap usage log messages.

Default: 5

`--cassandra_init_json_file_path`

Path to find cassandra_init.json file.

Default: `/home/nutanix/data/cassandra/conf/cassandra_init.json`

`--cassandra_load_threshold`

Threshold for cassandra load-related messages.

Default: 5

`--cassandra_progress_timeout_threshold`

Threshold for number of cassandra progress timeout log messages.

Default: 5

`--cassandra_restarts_threshold`

Threshold for cassandra restarts.

Default: 5

`--cassandra_retries_threshold`

Threshold count for cassandra retry log warnings.

Default: 5

`--cassandra_ring_cluster_version`

Cluster version of the nodetool_ring_file

Default: `None`

`--cassandra_stage_counters_threshold`

Threshold for number of missing stage counter messages.

Default: 5

`--cassandra_token_precision`

Maximum chars in the cassandra token that are expected to be unique.

Default: 8

`--cassandra_unavailability_threshold`

Threshold count for cassandra unavailability log warnings.

Default: 5

`--check_cloud_cvm_services`

Comma-separated list of services which are checked on cloud CVM.

Default: cassandra,pithos,hera,stargate,insights_server,ergon,cerebro,chronos,curator,alert_manager,cluster_

`--check_max_failure_score`

Health score to indicate check failure.

Default: 24

`--check_max_info_score`

Health score to indicate check info.

Default: 98

`--check_max_warning_score`

Health score to indicate check warning.

Default: 74

`--config_file_dir`

The base directory contains all plugin configs

Default: `/home/nutanix/ncc/plugin_config`

`--ctr_ec_ideal_delay_threshold`

Current ideal EC delay value setting for better Erasure coding benefits.

Default: 604800

`--cvm_avg_mem_util_threshold`

CVM memory average usage threshold (%) to determine memory contention.

Default: 90

`--cvm_commit_memory_threshold_pct`

CVM Memory Commit Threshold(%).

Default: 90

`--cvm_load_average_threshold`

CVM Load level.

Default: 10

`--cvm_load_average_threshold_critical`

CVM Load level.

Default: 100

`--cvm_mem_min_threshold`

Minimum threshold for CVM memory for NOS > 2.6.4

Default: 15728640

`--cvm_peak_cpu_util_threshold_pct`

Peak utilization threshold (%) to determine CVM CPU contention.

Default: 90

`--detect_rogue_plugin`

Flag to decide if the rogue plugin detection should get triggered.

Default: true

`--disk_abort_rate_threshold`

Disk read/write abort threshold. (aborts/sec)

Default: 0

`--disk_corruptions_threshold`

Threshold count for disk corruption log warnings.

Default: 5

`--disk_read_latency_threshold`

Latency threshold (ms/command) to determine storage issues.

Default: 10

`--disk_read_write_errors_threshold`

Threshold count for disk read/write log warnings.

Default: 5

`--disk_usage_threshold_pct_critical`

Threshold for disk usage (percentage) from being labeled as critical.

Default: 90

`--disk_usage_threshold_pct_warning`

Threshold for disk usage (percentage) from being labeled as a warning.

Default: 75

`--disk_write_latency_threshold`

Latency threshold (ms/command) to determine storage issues.

Default: 10

`--display_details_for_num_moves`

Displays the effect on the ring by applying the suggested node moves. The default is not to display any, the value of 0 is to display all until max_number_of_node_moves.

Default: -1

`--email_asup_external_contact_list`

External contact list to send email asups.

Default: `nos-asups@nutanix.com`

`--email_from_address`

Default senders email addresses.

Default: `cluster@nutanix.com`

`--email_human_readable_format`

Whether emails are to be sent in text or binary format.

Default: true

`--end_time_secs`

End point of time range.

Default: 1542615938

`--entity_id`

The id of the entity.

Default: `None`

`--entity_type`

The type of entity.

Default: `cluster`

`--error_id_json_path`

The file containing the error ids

Default: `/home/nutanix/ncc/plugin_config/error_ids.json`

`--ext4_journal_sequence_threshold`

Threshold for high ext4 journal sequence id as it approaches 2^31.

Default: 1879048192

`--field`

Field of data to be trained. E.g. CPU usage, disk usage.

Default: `hypervisor_cpu_usage_ppm`

`--file_path`

Full path of file to copy.

`--forced_reschedule_interval_sec`

Reschedule interval if the check execution falls in maintenance window.

Default: 300

`--future_end_time_secs`

End point of a future time range.

Default: 1543220738

`--future_start_time_secs`

Start Point of a future time range.

Default: 1542615938

`--garbage_egroups_size_critical_threshold_pct`

Default threshold for reporting size (percentage) occupied by garbage egroups as critical event

Default: 20

`--garbage_egroups_size_warning_threshold_pct`

Default threshold for reporting size (percentage) occupied by garbage egroups as warning event

Default: 10

`--generate_node_add_order`

Generate the order for node adds specified in the operations file, which results in the minimal token skew.

Default: false

`--gx_cpuprofile_frequency`

Number of interrupts per second the pprof cpu-profiler uses to sample.

Default: 100

`--gx_mysql_db`

MySQL database that is used to store the profiles.

Default: `profiles`

`--gx_mysql_host`

MySQL server's IP address.

Default: `10.1.56.13`

`--gx_mysql_hostname`

MySQL server's host name.

Default: `gxprof.eng.nutanix.com`

`--gx_mysql_passwd`

MySQL password for the given 'gx_mysql_user'.

Default: `password`

`--gx_mysql_user`

MySQL user name that was granted access to 'gx_mysql_db'.

Default: `nutanix`

`--gx_output_file`

Output filename without extension where gx write profiling data.

Default: `None`

`--gx_output_type`

Specify whether the output file is in `html` or `text` format.

Default: `html`

`--gx_pprof_file`

Specify the pprof file that is to be translated to gxprof format.

Default: `None`

`--gx_profile_description`

A text string of other important notes about the profile.

Default: `None`

`--gx_profile_detail_json`

A dictionary json of additional details about the profile.

Default: `None`

`--gx_profile_id`

Numeric identifier of the profile in the database.

Default: None

`--gx_profile_name`

Name of the profile in the database.

Default: `no name`

`--gx_profile_recorder`

Name of the person who run the profiler.

Default: `unknown`

`--gx_program`

Specify the program that is being run.

Default: `None`

`--gx_program_build_date`

Version timestamp of the program that was being profiled

Default: `Build date is unspecified`

`--gx_program_build_id`

Version number of the program that was being profiled

Default: `Build version is unspecified`

`--gx_regression_id`

Numeric identifier of a regression in the database.

Default: None

`--gx_run_duration`

The duration it takes to complete a run of the program or test. The unit of the duration is dependent on which program is being run. It could be second, mili-second, micro-second or others.

Default: 0

`--gx_run_status`

Current status of the execution being profiled.

Default: `Running pprof.`

`--gx_sample_excluded_symbols`

Comma-separated list of symbols to be excluded from profiling data.

Default: __restore_rt,std::_Function_handler::_M_invoke,std::_Bind::__call,std::function::operator

`--gx_total_seconds`

The pprof sampling duration in seconds.

Default: None

`--gx_verbose`

Log the progress for all gx commands to help trouble-shooting.

Default: false

`--health_disk_latency_threshold_ms`

Latency threshold (ms/command) to determine Disk issues.

Default: 50

`--helpshort`

show usage only for this module

Default: 0

`--high_disk_usage_threshold`

Threshold count for disk usage log warnings.

Default: 5

`--home_nutanix_usage_threshold_pct_critical`

Critical threshold for high disk usage on /home/nutanix /, as well as other disks that are not in Arithmos (percentage).

Default: 90

`--home_nutanix_usage_threshold_pct_warning`

Warn threshold for high disk usage on /home/nutanix /, as well as other disks that are not in Arithmos (percentage).

Default: 80

`--host_avg_cpu_util_threshold_pct`

Host utilization threshold (%) to determine host CPU contention.

Default: 75

`--host_balloon_threshold`

Balloon threshold to determine host memory bottlenecks.

Default: 0

`--host_disk_usage_threshold_pct_critical`

Threshold for host disk usage (percentage) being labeled as critical.

Default: 95

`--host_disk_usage_threshold_pct_warning`

Threshold for host disk usage (percentage) being labeled as warning.

Default: 90

`--host_peak_cpu_util_threshold_pct`

Host utilization threshold (%) to determine host CPU contention.

Default: 90

`--host_rcv_drop_threshold`

Receive packets drop threshold at host to determine network issues.

Default: 0

`--host_swap_in_threshold`

Swap in threshold (mbytes/sec) to determine host memory bottlenecks.

Default: 0

`--host_swap_out_threshold`

Swap out threshold (mbytes/sec) to determine host memory bottlenecks.

Default: 0

`--host_swap_threshold_mbps`

Swap rate threshold (mbytes/sec) to determine host memory bottlenecks.

Default: 0

`--host_swap_used_threshold_bytes`

Swap used threshold (bytes) to determine host memory bottlenecks.

Default: 0

`--host_tx_drop_threshold`

Transmitted packets drop threshold at host to determine network issues.

Default: 0

`--hyperv_host_external_vswitch_name`

Default name of the network switch for Hyper-V's external network.

Default: `ExternalSwitch`

`--hyperv_host_internal_ip_address`

IP address of internal interface on the Hyper-V host

Default: `192.168.5.1`

`--hyperv_user_name`

User name to use to connect to the Hyper-V host

Default: `administrator`

`--hyperv_user_password`

Password to use to connect to the Hyper-V host

Default: `nutanix/4u`

`--hypervisor_sample_period`

Hypervisor monitoring sampling interval.

Default: 300

`--ignore_negative_slope`

Whether ignore the negative slope or not.

Default: true

`--inode_usage_threshold_pct_critical`

Threshold for inode usage (percentage) being labeled as critical.

Default: 90

`--inode_usage_threshold_pct_warning`

Threshold for inode usage (percentage) being labeled as a warning.

Default: 75

`--insights_max_memory_usage_MB`

Maximum amount of memory in MB that the insights collector can use.

Default: 256

`--intel_ssd_num_bytes_written_critical_limit`

Critical markers for number of bytes written into the Intel SSDs.

Default: 7e+15

`--intel_ssd_num_bytes_written_warning_limit`

Warning markers for number of bytes written into the Intel SSDs.

Default: 6.5e+15

`--intel_ssd_warning_marker_position_below_max_celcius`

Temperature warning marker position for the Intel 910 SSDs. Since maximum temperature is retrieved from the device automatically, this value is actually the number of celcuis below the maximum temperature allowed storagepool

Default: 7

`--inter_cvm_bw_mb_sec_threshold`

Minimum bandwidth between CVMs (MB/s).

Default: 800

`--ip_list`

List of IPs to copy the file to.

`--iperf_server_timeout_secs`

Timeout (in seconds) for the iperf server per client.

Default: 60

`--is_periodic_run`

Flag to specify if the current ncc task is async orperiodic. True if the current task is periodic else False.

Default: false

`--log_collection_duration`

Time duration for which logs are parsed.

Default: 300

`--log_plugin_output`

Logs the protobug generated by each plugin execution - used mainly for debugging.

Default: false

`--log_types`

Log types to include within the compression.

Default: `None`

`--log_util_ip_list`

List of IPs to gather logs from.

Default: `None`

`--max_cassandra_restart_time_secs`

Maximum time for cassandra daemon to be killed and restarted.

Default: 15

`--max_health_score`

Maximum health score for any plugin.

Default: 100

`--max_number_of_node_moves`

The number of iterations to find minimum token skew. Default is to use 3 node moves. The time taken is exponential in the number of node moves and the number of nodes in cluster.

Default: 3

`--max_retry_on_not_master`

Number of retries when a rpc to cerebro returns kNotMaster

Default: 3

`--max_ring_load_balance_ratio`

Maximum ratio between least and greatest loaded node in cassandra ring.

Default: 3

`--max_ring_token_balance_ratio`

Maximum token skew allowed in cassandra ring.

Default: 2

`--minimum_container_rf`

Minimum replication factor for containers.

Default: 2

`--ncc_autocomplete`

If true, recreate autocomplete bash script.

Default: false

`--ncc_autocomplete_dir`

Directory to store the auto-generated completion file.

Default: `/home/nutanix/ncc`

`--ncc_canvas_json_dump_file`

File to which canvas json of latest NCC run will be serialized.

`--ncc_enable_intrusive_plugins`

If true, run plugins with intrusive impacts also.

Default: false

`--ncc_execution_token`

Execution token for the task.

Default: `None`

`--ncc_factory_config_path`

Path to factory config file.

Default: `/etc/nutanix/factory_config.json`

`--ncc_global`

If true, any local plugins invoked will be run across all nodes in the cluster.

Default: true

`--ncc_global_install`

Install ncc on all nodes.

Default: false

`--ncc_hardware_config_path`

Path to hardware config file.

Default: `/etc/nutanix/hardware_config.json`

`--ncc_interactive`

If true, the plugins will be run in an interactive mode, where the result will be available as soon as its ready.

Default: true

`--ncc_list_modules`

List all available ncc modules.

Default: false

`--ncc_master_ip`

IP of the node from where the command is run.

Default: `None`

`--ncc_monitor_progress`

Monitor progress for long running tasks.

Default: true

`--ncc_plugin_dir`

Directory from where plugin should be loaded.

Default: `/home/nutanix/ncc/bin/plugins`

`--ncc_plugin_output_file`

File to which raw output of latest NCC run will be written (use '' to disable).

Default: `/home/nutanix/data/logs/ncc-output-latest.log`

`--ncc_plugin_output_history_file`

File to which raw output of all NCC runs will be written.

Default: `/home/nutanix/data/logs/ncc-output.log`

`--ncc_print_config`

Print the plugin config and exit without running.

Default: false

`--ncc_show_hidden`

If true, display hidden plugins/modules.

Default: false

`--ncc_show_tree`

If true, display all the plugins in a tree format.

Default: false

`--ncc_via_rpc`

Indicates if NCC is running via RPC.

Default: false

`--ncli_password`

Use a non-default ncli password (will be prompted for password input).

Default: true

`--nfs_abort_rate_threshold`

NFS read/write abort threshold (aborts/sec).

Default: 0

`--nfs_command_latency_threshold`

Latency threshold (ms/command) to determine storage issues.

Default: 10

`--nfs_file_count_threshold`

Upper threshold on the number of NFS files and directories per datastore.

Default: 20000

`--nic_error_check_period_secs`

Period over which to monitor NIC errors (seconds).

Default: 3600

`--nic_error_threshold_info`

Maximum errors permissible for any NIC.

Default: 5

`--nic_error_threshold_warning`

Maximum errors permissible for any NIC before warning is issued.

Default: 100

`--nodetool_ring_file`

Path to the 'nodetool' ring output file. Default is to use the nodetool command on the cluster

Default: `None`

`--non_shell_vdisk_count_safe_threshold`

Safe threshold for max number of regular vdisks in versions > 5.5

Default: 600000

`--non_shell_vdisk_count_threshold`

Maximum number of non-shell vdisks in cluster.

Default: 200000

`--nutanix_model_config_file_dir`

The base directory contains all nutanix models

Default: `/home/nutanix/ncc/config`

`--nvm_peak_cpu_util_threshold_pct`

Peak utilization threshold (%) to determine FSVM CPU contention.

Default: 90

`--operations_file`

Path to node operations add/remove/no-op file

Default: `None`

`--oplog_episode_count_threshold`

Threshold for the number of episodes per vdisk.

Default: 1200

`--oplog_episode_count_threshold_pre_4_6`

Threshold for the number of episodes for a vdisk in a cluster which has a version lower than 4.6.

Default: 100

`--override_existing_plugin_config`

Over-ride the plugin configuration stored in Zookeeper with the default NCC plugin configuration.

Default: false

`--ping_time_threshold_ms`

Maximum ping time (in milli-seconds) between any two nodes.

Default: 15

`--pre_upgrade_check_enable`

The flag to specify if the current ncc run is for pre-upgrade

Default: false

`--pre_upgrade_check_list_path`

The flag to specify the path of all pre_upgrade checks

Default: `/home/nutanix/ncc/config/pre_upgrade_check_list.json`

`--preupgrade_log_path`

The log path of all preupgrade checks.

Default: `/home/nutanix/data/logs/preupgrade.out`

`--progress_timeout`

Timeout for getting progress updates from plugins by progress monitor.

Default: 2500

`--remote_site_ping_warn_threshold_ms`

Threshold for ping latency to remote beyond which the remote will be labeled as warning.

Default: 300

`--remote_site_socket_timeout_secs`

Timeout for creating a socket to Stargate running on the remote site.

Default: 5

`--remote_site_sync_critical_threshold_sec`

Threshold for remote cluster time (in seconds) being out of sync with source cluster to be labeled as critical.

Default: 300

`--remote_site_sync_warn_threshold_sec`

Threshold for remote cluster time (in seconds) being out of sync with source cluster to be labeled as warning.

Default: 180

`--replication_lag_critical_threshold`

Maximum number of pending replications beyond which a protection domain will be labeled as critical.

Default: 5

`--replication_lag_warn_threshold`

Maximum number of pending replications beyond which a protection domain will be labeled as warning.

Default: 2

`--rpc_timeouts_threshold`

Threshold count for rpc timeout log warnings.

Default: 5

`--run_using_gevent`

Run scheduler using gevent

Default: true

`--sampling_interval_secs`

The length of sampling interval.

Default: 600

`--shell_vdisk_count_threshold`

Maximum number of shell vdisks in cluster.

Default: 1000000

`--smtp_tunnel_port`

Local port number for the SMTP tunnel on the zookeeper leader.

Default: 2525

`--snapshot_chain_height_threshold`

Default snapshot chain height threshold to determine if snapshot chains are not getting severed.

Default: 25

`--snapshot_usage_bytes_stat`

Generic stat name to get the space used by snapshots of a protection domain

Default: `dr.exclusive_snapshot_usage_bytes`

`--snapshot_usage_crit_threshold_pct`

Default threshold for reporting size (percentage) occupied by snapshots as critical event

Default: 40

`--snapshot_usage_warn_threshold_pct`

Default threshold for reporting size (percentage) occupied by snapshots as warning event

Default: 30

`--sstable_size_thresholds_bytes`

Comma separated list of SSTable file size threshold.

Default: `2147483648`

`--stargate_retries_threshold`

Threshold count for stargate retry log warnings.

Default: 5

`--stargate_vars`

List of stargate variables to store in arithmos (empty list returns all). Stats will only be stored if they are supported by the relevant plugin.

`--stargate_vars_url`

Url where stargate stats can be queried.

Default: `http://127.0.0.1:2009/h/vars`

`--start_time_secs`

Start Point of time range.

Default: 0

`--stats_collection_interval`

Default stats collection gathering interval.

Default: 300

`--stats_default_email_addresses`

Default target email addresses for sending stats.

Default: `stats@nutanix.com`

`--test_data_config_file`

The path to the file containing the test data config

Default: `None`

`--transport_errors_threshold`

Threshold count for transport error log warnings.

Default: 5

`--update_plugin_config`

Update the plugin configuration stored in Zookeeper with the default NCC plugin configuration.

Default: true

`--use_esxtop`

Use esxtop to get stats. Otherwise use Arithmos

Default: false

`--vdisk_count_threshold`

Maximum number of vdisks in cluster.

Default: 80000

`--vm_avg_cpu_util_threshold`

Average utilization threshold (%) to determine VM CPU contention.

Default: 75

`--vm_avg_mem_util_threshold_pct`

VM memory average usage threshold (%) to determine memory contention.

Default: 80

`--vm_balloon_reclaim_threshold_mb`

Threshold to determine memory pressurefor a VM.

Default: 0

`--vm_balloon_threshold`

Balloon threshold to determine memory bottlenecks.

Default: 0

`--vm_cpu_util_threshold`

Utilization threshold (%) to determine VM CPU contention.

Default: 75

`--vm_peak_cpu_util_threshold_pct`

Peak utilization threshold (%) to determine VM CPU contention.

Default: 90

`--vm_peak_mem_util_threshold`

Peak utilization threshold (%) to determine VM memory bottlenecks.

Default: 90

`--vm_rcv_drop_threshold`

Receive packets drop threshold at a VM to determine network issues.

Default: 4096

`--vm_swap_in_threshold`

Swap in threshold (mbytes/sec) to determine memory bottlenecks.

Default: 0

`--vm_swap_out_threshold`

Swap out threshold (mbytes/sec) to determine memory bottlenecks.

Default: 0

`--vm_swap_threshold_mbps`

Swap out threshold (mbytes/sec) to determine memory bottlenecks.

Default: 0

`--vm_tx_drop_threshold`

Transmitted packets drop threshold at a VM to determine network issues.

Default: 0

`--vm_vcpu_ready_time_threshold`

Vcpu ready time threshold (%) to determine CPU contention.

Default: 10

`--zeus_config_file`

Path to the zeus config file. Default is to use the zeus config printer on the cluster

Default: `None`

`--zeus_configuration_path`

Zookeeper path where zeus configuration is stored.

Default: `/appliance/physical/configuration`

`--zookeeper_plugin_config_path`

Zookeeper path where plugin configuration is stored.

Default: `/appliance/logical/analytics/plugin_schema`

ncc.plugins.firstimport

`--ncc_base_dir`

The base NCC directory

Default: `/home/nutanix/ncc`

setup_hyperv.py

Usage

```
Usage: setup_hyperv.py [FLAGS] [command]

commands:
 register_shares
```

```
setup_metro
setup_scvmm
```

/usr/local/nutanix/bin/setup_hyperv.py

`--configure_library_share`

Whether a library share should be configured

Default: None

`--default_host_group_path`

The default SCVMM host group

Default: `All Hosts`

`--help`

Print detailed help

Default: false

`--library_share_name`

The name of the container that will be registered as a library share in SCVMM

`--metro_smb_account_password`

Password for the new metro cluster pair fqdn.

`--metro_smb_name`

This is the name that identifies a unique pair of AOS clusters. This name must be used for the name of the SMB server when provisioning virtual machines and virtual hard disks on any metro container stretched between the pair of AOS clusters.

`--ncli_password`

Password to be used when running ncli

`--nutanix_management_share`

The storage container for nutanix cluster management

Default: `NutanixManagementShare`

`--password`

Domain account password for the host

`--scvmm_host_group_path`

Host Group to which this cluster should be added

`--scvmm_password`

SCVMM account password - defaults to <password>

`--scvmm_server_name`

Name of the server running SCVMM

`--scvmm_username`

SCVMM account username (with the FQDN) - defaults to <host_fqdn>\<username>

`--setup_metro_no_ad`

Skips adding AD and DNS, but verifies it is added before proceeding further. This does not require the domain username and password and requires manual AD and DNS setup prior to running the script.

Default: false

`--storage_pool_name`

The name of the storage pool created if one doesn't already exist

Default: `NTNX-SP-DEFAULT`

`--username`

Domain account username (without domain name) which has/will have administrative rights on the host

`--verbose`

Print verbose output

Default: false

`--zknode`

Zookeeper node for Secure Keys

Default: `/appliance/logical/prismsecurekey`