



HCI Utility CLI Quick Reference

- Purpose

- HCI Utility CLI will enable NetApp internal users to easily add Compute and Storage nodes to existing HCI clusters.
- Longer term, this functionality will be made available in the NetApp Deployment Engine (NDE).

- General

- The CLI is a binary written in Python that has been 'frozen'. This means that you do not need to install a Python environment on the client you are using. You can simply invoke the tool per the instructions in this document.
- The CLI will do the following things:
 - i. Configure the network for the HCI nodes being added
 - ii. Configure the storage node into a new or existing HCI storage cluster
 - iii. Configure the compute node into a new or existing vCenter Data Center
 - iv. Mount the existing iSCSI HCI datastores on the compute node.

- Known limitations

- Storage node additions
 - i. If you try to add a 1.x node to a HCI 1.0 cluster.
 - In this case the addition will be rejected with a message that the HCI 1.0 cluster should be updated.
- Compute node additions
 - i. If you try to add a 1.0 compute node and specify 6.5 as the version it will still be added as a 6.0 ESXi node.
 - ii. Adding nodes to clusters with different versions have issues to be aware of:
 - 1.0 Compute node addition to HCI 1.1 cluster.
 - ⊖ In this case, the 'inspect' option will not identify the IP address(es). You will need to manually enter the entire compute json structure and specify the IP address, serial

number, etc. of the Compute node in the **inventory.json** file prior to running the 'deploy' .

- Addresses in use will not be detected with HCI 1.0
 - i. IP addresses you enter in the **inventory.json** file should be available for use.
 - HCI 1.1 will validate that the IP addresses are free; HCI 1.0 does not have this validation rule.

● Workflow

- At a high level, the workflow comprises of 3 steps –
 - i. Inspect - discover the HCI node information and create **inventory.json**
 - ii. Modify **inventory.json** file to update the HCI node configuration
 - iii. Deploy - install the software and add the HCI node to an existing HCI cluster.
- Overview of each step is as follows:
 - i. Execute the tool, per the command line options shown in Example usage, with the **inspect** command in order to discover information about the HCI node that has been physically connected but has not yet been configured from a software perspective, e.g. networking, vCenter, storage.
 - 1. Redirect the output of this command to a file called **inventory.json**.
 - 2. Use the Management IP of one of the storage nodes in the existing storage cluster as the **storage_ip**.
 - ii. Edit the **inventory.json** file created by the tool in step (i) above using your favorite editor.
 - 1. Review all the values generated by the tool for IP addresses, DNS, gateway, NTP, netmask, etc. and update them to the desired configuration.
 - Note: If you are attempting either of these mixed node/device version additions
 - 1.0 Compute node addition to HCI 1.1 cluster
 - 1.1 Compute node addition to HCI 1.0 cluster
 - The Compute node will not be discovered during inspect and you will need to manually enter the entire compute json structure and specify the IP address, serial number, etc. of the Compute node in the **inventory.json** file prior to running the 'deploy' .
 - 2. Confirm the 'deploy' field is set to 'true' within the **hciSettings** sections of the **inventory.json** file for any compute or storage node that you want to add to the HCI cluster when the tool is launched with the 'deploy' command.

- iii. (Optional) Perform 'deploy' with the modified **inventory.json** file and all of the required command line parameters and with the --validate_only flag to verify the json entries are correct.

- 1. JSON Error Handling - If modifications to the **inventory.json** make the json format invalid you can use a tool on the system called 'jq'
 - 2. 'jq' will typically highlight an error in the json format down to within +-5 lines based upon our testing.
 - 3. Here is an example use of the jq tool:

```
jmeares-LT@[jmeares-lt]:sfprime$ jq . inspect.json
parse error: Invalid string: control characters from U+0000 through U+001F must be escaped at line 320, column 17
```

- 4. Once the issue is identified, then update the **inventory.json** file and proceed.
- iv. Execute the tool again, per the command line options shown in the Example usage, with the **deploy** command in order to complete the configuration and addition of the HCI nodes you identified within the **inventory.json** file and set the 'deploy' field to 'true' under hciSettings.
 - 1. The **deploy** parameter can be used with and without vCenter credential information. If you are going to configure Compute nodes and join a vCenter then the vCenter credential information is required, otherwise not, that's why examples below show it with and without vCenter credential information.
 - 2. Note: The tool will install ESXi based on the 'version' specified in the Compute section of **inventory.json**; independent of you providing the command line parameters for vCenter. Providing the parameters for vCenter access just enables the tool to join the Compute node to vCenter.

• Usage

- Platform appropriate tool names

- i. Linux

- hciutil

- ii. Windows

- hciutil.exe

- Commands

- i. inspect Discover information about the nodes to potentially be added
 - ii. deploy Perform the configuration of nodes identified within the inventory json file
 - iii. manual List usage information and details about the command line parameters

- Command line options

- -s, --storage_ip=STORAGE_IP An existing storage node IP address in an HCI cluster
 - -u, --username=USER The username for the storage node
 - -p, --password=PASS The password for the storage node
 - -i, --inventory=INVENTORY The node inventory to deploy (from inspect)

- `--esxipassword=ESXIPASS` Optional password for new compute nodes (`--password` otherwise)
- `--vcenter=VCENTER_IP_OR_NAME` The IP address or FQDN of the vCenter server
- `--vcusername=VCUSER` The username for the vCenter server
- `--vcpasspassword=VCPASS` The password for the vCenter server
- `--datacenter=DC` The vCenter server datacenter
- `--cluster=CLUSTER` Optional cluster for compute nodes
- `-v, --version` Show the version
- `--skip_validation` CLI will not perform network validation
- `--validate_only` Performs network validation, does not deploy

• Environment variable usage

- The environment variables below can be used to override default values built into the CLI.
 - # seconds to wait prior to timing out a SolidFire API call
 - `SF_API_TIMEOUT`', default=13
 - # seconds to wait prior to timing out an NDE API call
 - `NDE_API_TIMEOUT`', default=31
 - # seconds to wait prior to timing out a call specifically to gather inventory
 - `GET_INVENTORY_TIMEOUT_SECS`', default=60
 - # seconds to wait prior to timing out a call specifically waiting for storage nodes to come out of SolidFire pending state
 - `PENDING_STORAGE_TIMEOUT_SECS`', default=4 * 60
 - # seconds to wait prior to timing out a call specifically for installing a Storage node
 - `STORAGE_INSTALL_TIMEOUT_SECS`', default=12 * 60
 - # seconds to wait prior to timing out a call specifically for installing a Compute node
 - `COMPUTE_INSTALL_TIMEOUT_SECS`', default=120 * 60
 - # seconds to wait prior to timing out a call specifically for configuring a Compute node
 - `COMPUTE_CONFIGURATION_TIMEOUT_SECS`', default=5 * 60
 - # how long to wait prior to timing out the monitoring portion of the CLI process
 - `MONITOR_UPDATE_PERIOD`', default=23
 - # Log file name
 - `HCIUTIL_LOG_NAME`, default="hciutil_log"
 - # Log file path
 - `HCIUTIL_LOG_PATH`, default="."
 - # Log to console
 - `HCIUTIL_LOG_TO_CONSOLE`, default=False
 - # Compute node user name
 - `ESXI_USERNAME`, default="root"

• Example usage

- To obtain the usage information and details about the command line parameters to use. You are encouraged to invoke the tool with the *manual* parameter first.
 - `hciutil manual`
- To discover information about the nodes to potentially be added
 - `hciutil inspect (--storage_ip=<STORAGE_IP>) (--username=<USER>) (--password=<PASS>) >inventory.json`

- To configure storage nodes using the *deploy* command using the modified *inventory.json* file.
 - `hciutil deploy (--storage_ip=<STORAGE_IP>) (--username=<USER>) (--password=<PASS>) (--inventory= inventory.json)`
- To configure compute nodes using the *deploy* command using the modified *inventory.json* file.
 - `hciutil deploy (--storage_ip=<STORAGE_IP>) (--username=<USER>) (--password=<PASS>) (--inventory= inventory.json) (--vcenter=<VCENTER_IP>) (--vcusername=<VCUSER>) (--vcpass=<VCPASS>) (--datacenter=<DC>) [--cluster=<CLUSTER>]`

- Example *inventory.json* settings

- Compute hciSettings entry

```
"hciSettings": {
  "deploy": true,
  "hostName": "Host-Comp-2",
  "iscsi-a": {
    "address": "11.111.111.71",
    "gateway": "11.111.111.254",
    "netmask": "255.255.248.0",
    "vlan": null
  },
  "iscsi-b": {
    "address": "11.111.111.72",
    "gateway": "11.111.111.254",
    "netmask": "255.255.248.0",
    "vlan": null
  },
  "management": {
    "address": "11.111.111.106",
    "dns": ["11.111.11.11"],
    "gateway": "11.111.111.254",
    "ntp": ["11.111.11.50"],
    "netmask": "255.255.248.0",
    "vlan": null
  },
  "version": "6.0",
  "vmotion": {
    "address": "11.111.111.107",
    "gateway": "11.111.111.254",
    "netmask": "255.255.248.0",
    "vlan": null
  }
},
```

- Storage hciSettings entry

```
"hciSettings": {
  "deploy": true,
  "hostName": "storage-04",
  "iscsi": {
```

```
        "address": "11.111.111.77",
        "gateway": "11.111.111.254",
        "netmask": "255.255.248.0",
        "vlan": 23
    },
    "management": {
        "address": "11.111.111.112",
        "dns": ["11.111.11.11"],
        "gateway": "11.111.111.254",
        "netmask": "255.255.248.0",
        "vlan": 24
    }
},
```