Configuring iSCSI offload for QLogic QLE8442 CNA on Hitachi Storage

[Red Hat and SUSE]

v1.0

Configuration Guide

This document describes how to configure QLogic QLE8442 iSCSI offloading in various operating systems such as Red Hat Linux and SUSE. This document also includes configuring the Hitachi VSP 5000 series iSCSI storage interfaces for these operating systems.

To use this document, you must have a working knowledge of Linux.

Hitachi Vantara

March 2022

Table of Contents

Preface

About this document

This document describes how to configure QLogic QLE8442 iSCSI offloading in various operating systems such as Red Hat and SUSE. The configuration process is explained for SUSE Linux Enterprise Server 15 SP1 and Red Hat Enterprise Linux 8.2. This document also provides information for configuring the Hitachi VSP 5000 series iSCSI storage interfaces with these operating systems.

The configuration process should work for all SUSE Linux Enterprise Server 15 and Red Hat Enterprise Linux 8 implementations.

Most of the operating system configuration steps are applicable for older or newer versions. See the respective OS Guides for details.

Before installing and configuring the iSCSI initiator, you must verify which operating systems and Hitachi Storage systems are supported by referring to their respective websites.

Check the interoperability information for your operating systems and Hitachi Storage iSCSI interfaces at the following URL:

https://www.hitachivantara.com

Check the vendor websites for information about iSCSI initiators, iSCSI HBAs, and NICs (including latest HBA/NIC drivers and BIOS).

Check the vendor websites for operating systems-related issues.

Document conventions

This document uses the following typographic convention:

Convention	Description
Bold	• Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: **Click OK**. • Indicates emphasized words in list items.
Italic	Indicates a document title or emphasized words in text.
`Monospace`	Indicates text that is displayed on screen or entered by the user. Example: `pairdisplay -g oradb`

Intended audience

This document is intended for Hitachi Vantara internal.

Release notes

Read the release notes before installing and using this product. They may contain requirements or restrictions that are not fully described in this document or updates or corrections to this document.

Accessing product downloads

Product software, drivers, and firmware downloads are available on Hitachi Vantara Support Connect: https://support.hitachivantara.com/.

Log in and select Product Downloads to access the most current downloads, including updates that may have been made after the release of the product.

Getting Help

Hitachi Vantara Support Connect is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to Hitachi Vantara Support Connect for contact information: https://support.hitachivantara.com/en_us/contact-us.html.

Hitachi Vantara Community is a global online community for customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to community.hitachivantara.com, register, and complete your profile.

Comments

Please send us your comments on this document to EOSilabIndia@hitachivantara.com. Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

Introduction

QLogic QLE8442 CNA is a 10GB ethernet converge network adapter (CNA) that supports FC, FCoE, and iSCSI protocols at a 10GB line-rate speed. This card also supports full hardware offload.

The Linux driver modules for the QLogic QLE8442 CNA are bnx2i, bnx2x, CNIC, and bnx2fc. The functionality of these drivers is as follows:

- bnx2x is the Linux driver for the NetXtreme II 10GB network adapters. This driver directly controls the hardware and is responsible for sending and receiving Ethernet packets for the Linux host networking stack. This driver also receives and process device interrupts, both for itself (for L2 networking) and the bnx2fc (FCoE) and CNIC drivers.

- The CNIC driver provides the interface between the Broadcom upper-layer protocol (for example, storage) drivers and the Broadcom NetXtreme II 1GB and 10GB network adapters. The CNIC driver runs with the bnx2 and bnx2x network drives downstream, and the bnx2fc (FCoE) and bnx2i (iSCSI) drivers run upstream.

- bnx2i is the iSCSI offloading driver for Linux. This driver enables the NetXtreme II 1GB and 10GB network adapters.

- bnx2fc is a Linux FCoE kernel-mode driver that translates the layer between the Linux iSCSI stack and the Broadcom FCoE firmware/hardware. This driver is responsible for sending and receiving encapsulated FCoE frames for the libfc/libfcoe and FIP/device discovery on the networking layer.

The purpose of this article is to describe step-by-step guidelines for configuring iSCSI offloading with the QLogic QLE8442 CNA.

CNA BIOS Configuration

This configuration guide is for an iSCSI offloading setup with the Hitachi VSP 5000 series. The iSCSI offloading for local boot shows an example of a host setup with a two-port network adapter. One NIC port connects to a Hitachi iSCSI target port CL1-A, and the other NIC port connects to a CL2-A port.

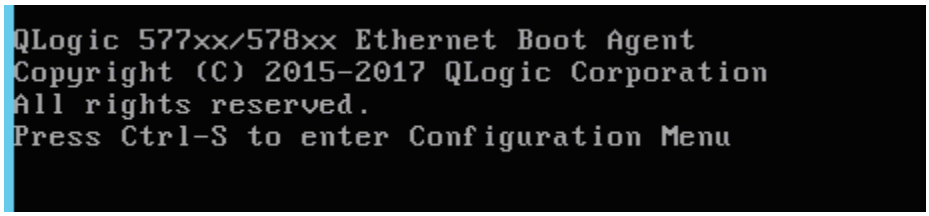
To configure the iSCSI offloading for local boot, you must change the boot protocol in CNA BIOS and note the iSCSI initiator name for the iSCSI setup configuration.

MBA Boot Protocol Configuration

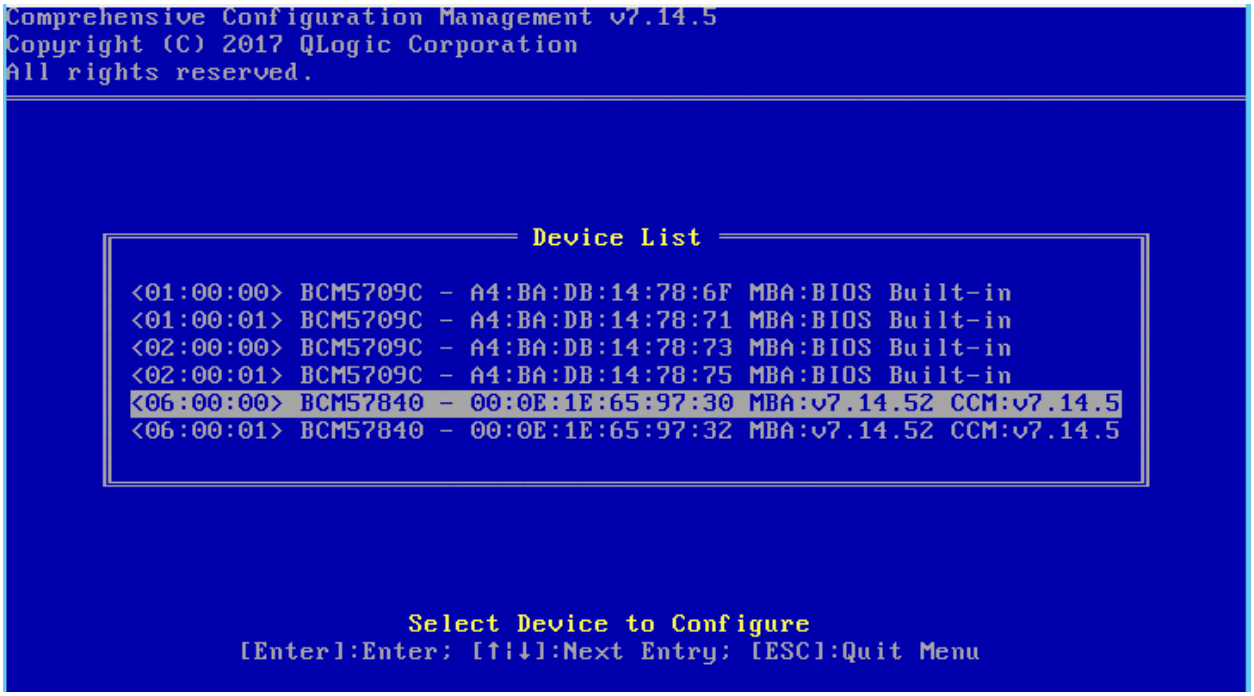
You can configure the boot protocol using the pre-boot CCM:

1. Restart the system.

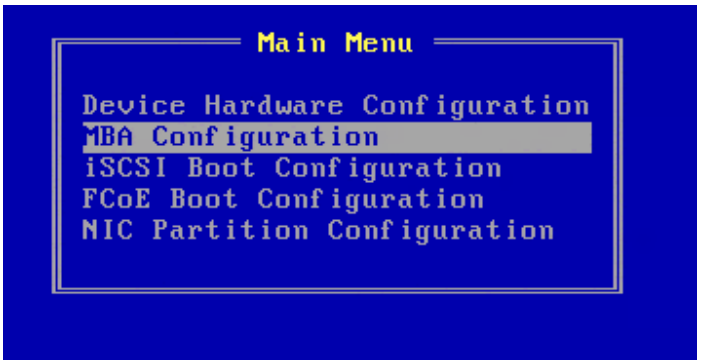
2. In the QLogic 577*xx*/578*xx* Ethernet Boot Agent banner, press the CTRL+S keys.



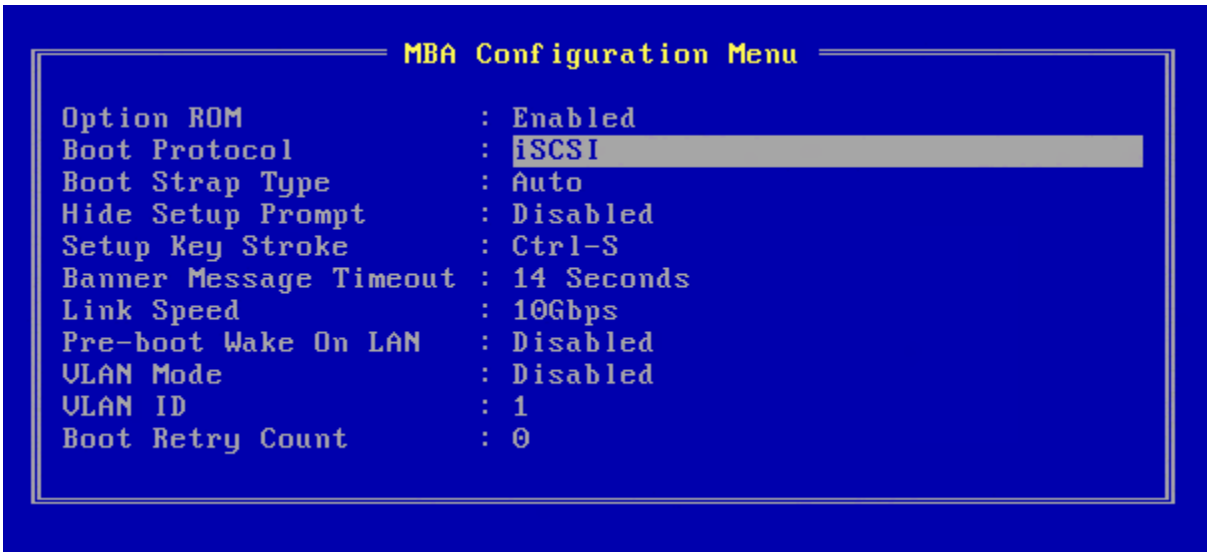
3. In the CCM **Device List,** press the UP ARROW or DOWN ARROW keys to select a device, and then press ENTER.



4. In the **Main Menu**, select **MBA Configuration** and then press ENTER.

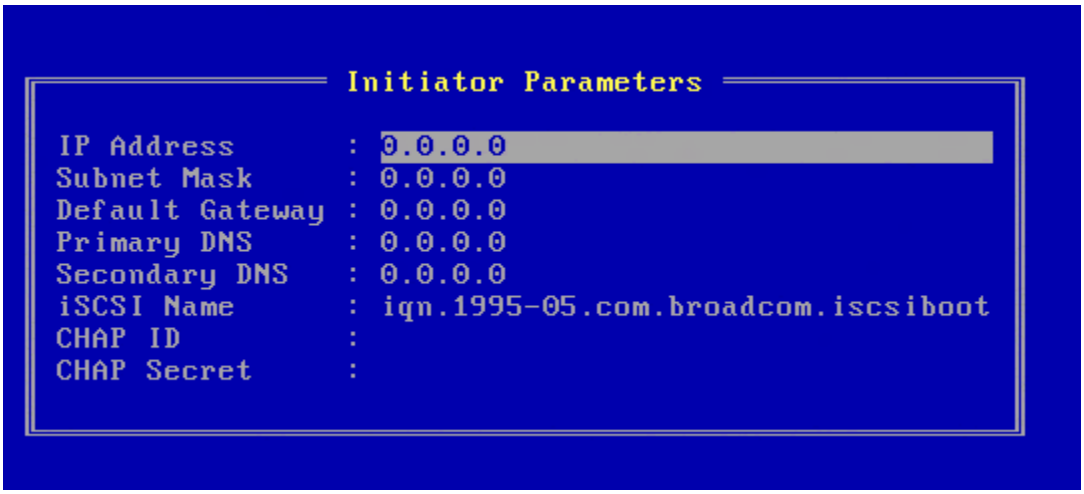


5. In the **MBA Configuration Menu,** press the UP ARROW or DOWN ARROW keys to select **Boot Protocol**. Next, press the LEFT ARROW or RIGHT ARROW keys to change the **Boot Protocol** option to **iSCSI** and then press ENTER.



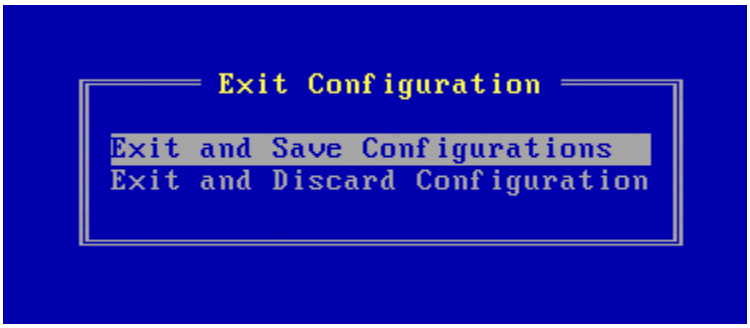
6. To check the **iSCSI Name** from the **Boot Configuration,** select **General Parameters > Initiator Parameters.**

   The **iSCSI Name** corresponds to the iSCSI initiator name to be used by the client system as follows:



   Note: For BFS configuration, we must enter the relevant fields under iSCSI Boot configuration.

7. For the local boot setup, you must use only **iSCSI Name** (hardware initiator IQN) for iSCSI target discovery and login. Select **Exit and Save Configurations** to save the iSCSI boot configuration. Otherwise, select **Exit and Discard Configuration**. Next, click ENTER.



Configuring Red Hat and SUSE Linux iSCSI Host

After installing the operating system, check that the iSCSI initiator package is installed on the Linux OS. If it's unavailable, install the package as follows:

SUSE Linux Enterprise Server:

```
localhost:~ # rpm -qa |grep iscsi
open-iscsi-2.0.876-13.26.1.x86_64
libopeniscsiusr0_2_0-2.0.876-13.26.1.x86_64
yast2-iscsi-client-4.1.7-1.12.noarch
iscsiuio-0.7.8.2-13.26.1.x86_64
localhost:~ #
```

Red Hat Enterprise Linux:

```
[root@localhost ~]# rpm -qa |grep iscsi
udisks2-iscsi-2.8.3-2.el8.x86_64
iscsi-initiator-utils-6.2.0.878-4.gitd791ce0.el8.x86_64
qemu-kvm-block-iscsi-2.12.0-99.module+el8.2.0+5827+8c39933c.x86_64
iscsi-initiator-utils-iscsiuio-6.2.0.878-4.gitd791ce0.el8.x86_64
libvirt-daemon-driver-storage-iscsi-4.5.0-
42.module+el8.2.0+6024+15a2423f.x86_64
libiscsi-1.18.0-8.module+el8.1.0+4066+0f1aadab.x86_64
[root@localhost ~]#
```

Configuring IP Addresses on iSCSI NIC Ports

```
localhost:~ # ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether a4:ba:db:14:78:6f brd ff:ff:ff:ff:ff:ff
    inet 172.17.15.32/23 brd 172.17.15.255 scope global em1
       valid_lft forever preferred_lft forever
    inet6 fe80::a6ba:dbff:fe14:786f/64 scope link
       valid_lft forever preferred_lft forever
3: em2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
qlen 1000
    link/ether a4:ba:db:14:78:71 brd ff:ff:ff:ff:ff:ff
4: em3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
qlen 1000
    link/ether a4:ba:db:14:78:73 brd ff:ff:ff:ff:ff:ff
5: em4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
qlen 1000
    link/ether a4:ba:db:14:78:75 brd ff:ff:ff:ff:ff:ff
```

```
6: p4p1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group
default qlen 1000
    link/ether 00:0e:1e:65:97:30 brd ff:ff:ff:ff:ff:ff
7: p4p2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group
default qlen 1000
    link/ether 00:0e:1e:65:97:32 brd ff:ff:ff:ff:ff:ff
localhost:~ #
```

Configure the IP addresses for both BCM57840 iSCSI network ports. The IP range should be different from the target storage IP address.

- For SUSE, the Ethernet ports for p4p1 and p4p2 are as follows:
    - Ethernet port p4p1 IP Address: 192.168.20.10
    - Ethernet port p4p2 IP Address: 192.168.21.10

```
localhost:~ # ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether a4:ba:db:14:78:6f brd ff:ff:ff:ff:ff:ff
    inet 172.17.15.32/23 brd 172.17.15.255 scope global em1
       valid_lft forever preferred_lft forever
    inet6 fe80::a6ba:dbff:fe14:786f/64 scope link
       valid_lft forever preferred_lft forever
3: em2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
qlen 1000
    link/ether a4:ba:db:14:78:71 brd ff:ff:ff:ff:ff:ff
4: em3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
qlen 1000
    link/ether a4:ba:db:14:78:73 brd ff:ff:ff:ff:ff:ff
5: em4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
qlen 1000
    link/ether a4:ba:db:14:78:75 brd ff:ff:ff:ff:ff:ff
6: p4p1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 00:0e:1e:65:97:30 brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.10/24 brd 192.168.20.255 scope global p4p1
       valid_lft forever preferred_lft forever
    inet6 fe80::20e:1eff:fe65:9730/64 scope link
       valid_lft forever preferred_lft forever
7: p4p2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 00:0e:1e:65:97:32 brd ff:ff:ff:ff:ff:ff
    inet 192.168.21.10/24 brd 192.168.21.255 scope global p4p2
       valid_lft forever preferred_lft forever
```

```
    inet6 fe80::20e:1eff:fe65:9732/64 scope link
       valid_lft forever preferred_lft forever
localhost:~ #
```

- For RHEL, the Ethernet ports are ens9f0 and ens9f1.

```
[root@localhost ~]# ifconfig
ens255f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.14.86  netmask 255.255.254.0  broadcast
172.17.15.255
        inet6 fe80::28b4:2fde:6ce9:3a81  prefixlen 64  scopeid
0x20<link>
        ether 2c:60:0c:f9:70:9b  txqueuelen 1000  (Ethernet)
        RX packets 508286  bytes 51464210 (49.0 MiB)
        RX errors 0  dropped 35549  overruns 0  frame 0
        TX packets 16524  bytes 1499859 (1.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens255f1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 2c:60:0c:f9:70:9c  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens9f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.20.10  netmask 255.255.255.0  broadcast
192.168.20.255
        inet6 fe80::4688:bd9f:13e9:4364  prefixlen 64  scopeid
0x20<link>
        ether 00:0e:1e:65:9f:20  txqueuelen 1000  (Ethernet)
        RX packets 101712  bytes 6509568 (6.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 70  bytes 8463 (8.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 61  memory 0xfb000000-fb7fffff

ens9f1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.21.10  netmask 255.255.255.0  broadcast
192.168.21.255
        inet6 fe80::1ad0:1993:d4aa:bb33  prefixlen 64  scopeid
0x20<link>
        ether 00:0e:1e:65:9f:22  txqueuelen 1000  (Ethernet)
        RX packets 101749  bytes 6511936 (6.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 62  bytes 7761 (7.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 151  memory 0xfa000000-fa7fffff

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
```

```
        RX packets 412  bytes 32988 (32.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 412  bytes 32988 (32.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast
192.168.122.255
        ether 52:54:00:1d:24:0e  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]#
```

Identifying the Host and Storage IQN

Host IQN:

For the host IQN, you can either use the hardware iSCSI initiator or software iSCSI initiator.

You can use the hardware iSCSI initiator IQN that we noted in the CNA BIOS utility.

You can also get the software initiator IQN from the following file on the Linux hosts: `/etc/iscsi/initiatorname.iscsi`.

- **SUSE Linux Enterprise Server:**

```
localhost:~ # cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1996-04.de.suse:01:672242fc424
```

- **Red Hat Enterprise Linux:**

```
[root@localhost ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1994-05.com.redhat:cceb42766a99
```

Target IQN:

Before proceeding with the iSCSI initiator login, you must verify that Hitachi iSCSI target storage is configured.

Verify that the following is configured in the Hitachi iSCSI Storage system:

- IP Addresses are configured on the iSCSI target ports.
- Configure or create iSCSI target storage in the Hitachi Storage system.

- Create a CHAP authentication user in the Hitachi Storage system for iSCSI targets
- Add the iSCSI host (iSCSI initiator) to iSCSI targets in the Hitachi Storage system.

For configuring storage, refer to the Hitachi Storage configuration section.

The iSCSI target storage ports CL1-A and CL2-A have the following IP Addresses:

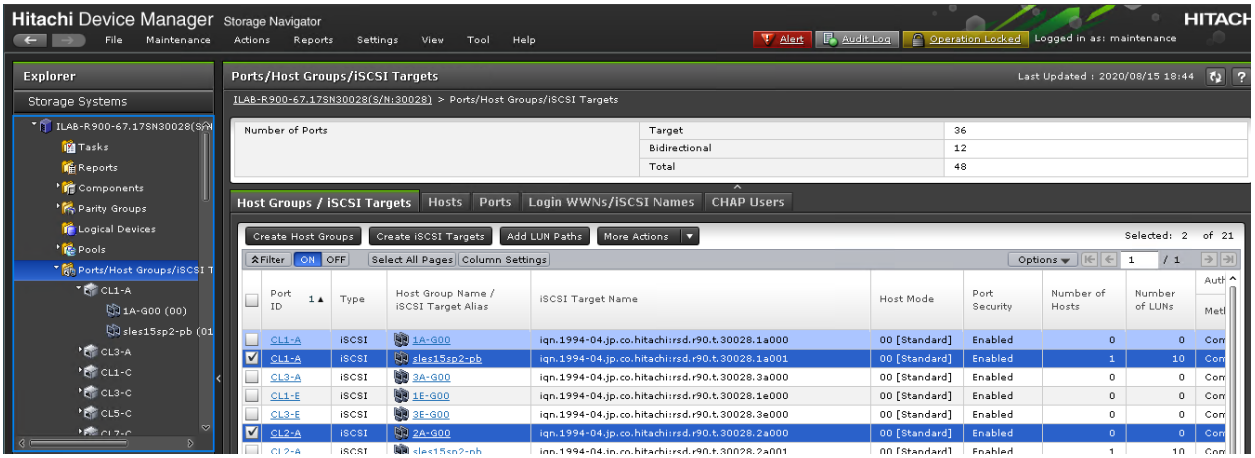
- Hitachi iSCSI storage port CL1-A IP Address: 192.168.10.10
- Hitachi iSCSI storage port CL2-A IP Address: 192.168.11.10

Note: The iSCSI initiator and storage target IQN are for later use in the configuration process.

Use the following IQN for this setup:

```
Linux Host IQN: 1995-05.com.broadcom.iscsiboot
iSCSI Storage port CL-1A target IQN: iqn.1994-
04.jp.co.hitachi:rsd.r90.t.30028.1a001
iSCSI Storage port CL-2A target IQN: iqn.1994-
04.jp.co.hitachi:rsd.r90.t.30028.2a001
```

You must verify and note the following iSCSI target IQN from storage system.



Configuring File Modifications

On a Linux host, you can edit the following file to configure the iSCSI settings:
`/etc/iscsi/iscsid.conf`

- To request that the iscsi initd scripts start a session, you must set `node.startup` to automatic:

```
node.startup = automatic
```

- To configure CHAP authentication:

- Set `node.session.auth.authmethod` to `CHAP`. The default is no authentication.

  ```
  node.session.auth.authmethod = CHAP
  ```

- To set a CHAP username and password for initiator authentication by the targets, uncomment the following lines:

  ```
  node.session.auth.username = Chap_User
  node.session.auth.password = Chap_Password
  ```

  To enable CHAP authentication, enter the CHAP username and password that was created in the iSCSI storage system for CHAP authentication.

- To allow the targets to control the digest checking settings and the initiator to request enabling digest checking, uncomment one or both of the following lines:

  ```
  node.conn[0].iscsi.HeaderDigest = None,CRC32C
  node.conn[0].iscsi.DataDigest = None,CRC32C
  ```

- To enable CRC32C digest checking for the header or the data part of the iSCSI PDUs, uncomment one or both of the following lines:

  ```
  node.conn[0].iscsi.HeaderDigest = CRC32C
  node.conn[0].iscsi.DataDigest = CRC32C
  ```

- To disable digest checking for the header or data part of iSCSI PDUs, uncomment one or both of the following lines:

  ```
  node.conn[0].iscsi.HeaderDigest = None
  node.conn[0].iscsi.DataDigest = None
  ```

In Red Hat Enterprise Linux, data digests are not supported. Set the header digest in RHEL and verify with the vendor about support.

Configuring the iSCSI Service to Start at Boot Time

SUSE Linux Enterprise Server:

```
localhost:/ # systemctl list-unit-files|grep iscsi
iscsi.service                                                 enabled
iscsid.service
      disabled
iscsiuio.service
      disabled
iscsid.socket
      enabled
iscsiuio.socket
      disabled
localhost:/ # systemctl enable iscsid.service
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/iscsid.service
â /usr/lib/systemd/system/iscsid.service.

localhost:/ # systemctl list-unit-files|grep iscsi
iscsi.service
      enabled
iscsid.service
      enabled
iscsiuio.service
      disabled
iscsid.socket
      enabled
iscsiuio.socket
      disabled
      localhost:/ #
```

Red Hat Enterprise Linux:

```
      [root@localhost ~]# systemctl list-unit-files|grep iscsi
      iscsi-onboot.service                      enabled
      iscsi-shutdown.service                    static
      iscsi.service                             enabled
      iscsid.service                            enabled
      iscsiuio.service                          disabled
      iscsid.socket                             enabled
      iscsiuio.socket                           enabled
      [root@localhost ~]#
```

Configuring the Default iface

In Linux, each iSCSI port is an interface known as *iface*. By default, the open-iscsi daemon connects to discovered targets using a software initiator (transport name = tcp) with the iface name, *default*. To offload the iSCSI connection to the C-NIC device, you use the ifaces with names that have the prefix *bnx2i*. The bnx2i ifaces are created automatically using the iscsiadm CLI utility to issue the following command:

iscsiadm -m iface

```
localhost:/ # iscsiadm -m iface
bnx2i.00:0e:1e:65:97:31 bnx2i,00:0e:1e:65:97:31,default,p4p1,<empty>
bnx2i.00:0e:1e:65:97:33 bnx2i,00:0e:1e:65:97:33,default,p4p2,<empty>
default tcp,<empty>,<empty>,<empty>,<empty>
iser iser,<empty>,<empty>,<empty>,<empty>
```

Here, p4p1 and p4p2 are two BCM57840 Ethernet controller network ports.

1. Before target discovery, you must update the IP addresses of the same range as the target IP and initiator name of both ifaces as follows:

```
localhost:~ # iscsiadm -m iface -I bnx2i.00:0e:1e:65:97:31 -n
iface.ipaddress -v 192.168.10.1 -o update
bnx2i.00:0e:1e:65:97:31 updated.
localhost:~ # iscsiadm -m iface -I bnx2i.00:0e:1e:65:97:31 -n
iface.initiatorname -v iqn.1995-05.com.broadcom.iscsiboot -o update
bnx2i.00:0e:1e:65:97:31 updated.
localhost:~ #


localhost:~ # iscsiadm -m iface -I bnx2i.00:0e:1e:65:97:33 -n
iface.ipaddress -v 192.168.11.1 -o update
bnx2i.00:0e:1e:65:97:33 updated.
localhost:~ # iscsiadm -m iface -I bnx2i.00:0e:1e:65:97:33 -n
iface.initiatorname -v iqn.1995-05.com.broadcom.iscsiboot -o update
bnx2i.00:0e:1e:65:97:33 updated.
localhost:~ #
```

2. Verify the changes by going through the path `/etc/iscsi/ifaces` as follows:

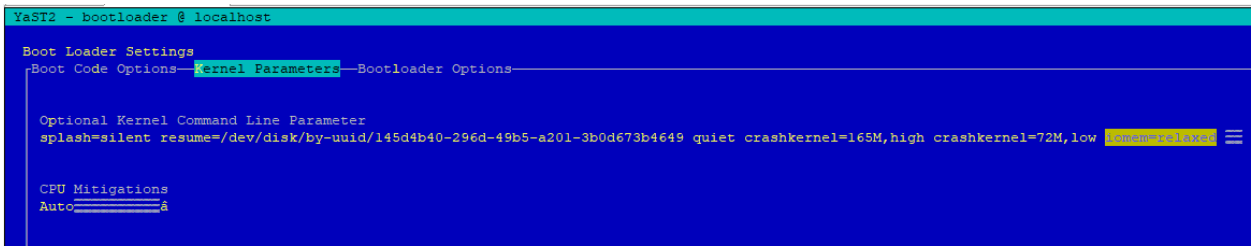
```
localhost:/etc/iscsi/ifaces # cat bnx2i.00\:0e\:1e\:65\:97\:31
# BEGIN RECORD 2.0-876-suse
iface.iscsi_ifacename = bnx2i.00:0e:1e:65:97:31
iface.net_ifacename = p4p1
iface.ipaddress = 192.168.10.1
iface.prefix_len = 0
iface.hwaddress = 00:0e:1e:65:97:31
iface.transport_name = bnx2i
iface.initiatorname = iqn.1995-05.com.broadcom.iscsiboot
iface.vlan_id = 0
iface.vlan_priority = 0
iface.iface_num = 0
iface.mtu = 0
iface.port = 0
iface.tos = 0
iface.ttl = 0
iface.tcp_wsf = 0
iface.tcp_timer_scale = 0
iface.def_task_mgmt_timeout = 0
iface.erl = 0
iface.max_receive_data_len = 0
iface.first_burst_len = 0
iface.max_outstanding_r2t = 0
iface.max_burst_len = 0
# END RECORD

localhost:/etc/iscsi/ifaces # cat bnx2i.00\:0e\:1e\:65\:97\:33
# BEGIN RECORD 2.0-876-suse
iface.iscsi_ifacename = bnx2i.00:0e:1e:65:97:33
iface.net_ifacename = p4p2
iface.ipaddress = 192.168.11.1
iface.prefix_len = 0
iface.hwaddress = 00:0e:1e:65:97:33
iface.transport_name = bnx2i
```

```
iface.initiatorname = iqn.1995-05.com.broadcom.iscsiboot
iface.vlan_id = 0
iface.vlan_priority = 0
iface.iface_num = 0
iface.mtu = 0
iface.port = 0
iface.tos = 0
iface.ttl = 0
iface.tcp_wsf = 0
iface.tcp_timer_scale = 0
iface.def_task_mgmt_timeout = 0
iface.erl = 0
iface.max_receive_data_len = 0
iface.first_burst_len = 0
iface.max_outstanding_r2t = 0
iface.max_burst_len = 0
# END RECORD
localhost:/etc/iscsi/ifaces #
```

Running iSCSI Target Discovery

For target discovery and login when using SUSE 15 SP1, you must provide workaround `iomem=relaxed` in the kernel parameter to discover the send target and successfully log in.



1. Start iSCSI target discovery and then log in to the iSCSI target using a node record ID found by the discovery as shown in the following examples:

    - iSCSI target discovery for iSCSI Storage port CL1-A (192.168.10.10) and logging in to the iSCSI target.

```
   # iscsiadm --mode discovery --type sendtargets -I
bnx2i.00:0e:1e:65:97:31 --portal 192.168.10.10:3260 --op new
192.168.10.10:3260,1 iqn.1994-04.jp.co.hitachi:rsd.r90.t.30028.1a001

   #  iscsiadm --mode node --targetname iqn.1994-
04.jp.co.hitachi:rsd.r90.t.30028.1a001 --portal 192.168.10.10:3260 -
-login

Logging in to [iface: bnx2i.00:0e:1e:65:97:31, target: iqn.1994-
04.jp.co.hitachi:rsd.r90.t.30028.1a001, portal: 192.168.10.10,3260]
```

```
Login to [iface: bnx2i.00:0e:1e:65:97:31, target: iqn.1994-
04.jp.co.hitachi:rsd.r90.t.30028.1a001, portal: 192.168.10.10,3260]
successful.
```

- iSCSI target discovery to iSCSI Storage port CL2-A (192.168.110.10) and logging in to the iSCSI target.

```
 # iscsiadm --mode discovery --type sendtargets -I
bnx2i.00:0e:1e:65:97:33 --portal 192.168.11.10:3260 --op new
192.168.11.10:3260,1 iqn.1994-04.jp.co.hitachi:rsd.r90.t.30028.2a001

   # iscsiadm --mode node --targetname iqn.1994-
04.jp.co.hitachi:rsd.r90.t.30028.2a001 --portal 192.168.11.10:3260 –
login

      Logging in to [iface: bnx2i.00:0e:1e:65:97:33, target:
iqn.1994-04.jp.co.hitachi:rsd.r90.t.30028.2a001, portal:
192.168.11.10,3260]
Login to [iface: bnx2i.00:0e:1e:65:97:33, target: iqn.1994-
04.jp.co.hitachi:rsd.r90.t.30028.2a001, portal: 192.168.11.10,3260]
successful.
```

Note: This node discovery creates `/etc/iscsi/nodes` and `/etc/iscsi/send_targets` directories in SUSE and `/var/lib/iscsi/nodes` and `/var/lib/iscsi/send_targets` in Red Hat Enterprise Server with the discovery table and node table.

2. Verify iSCSI offloading (bnx2i) by checking the path /`etc/iscsi/send_targets`.

```
#cat bnx2i.00\:0e\:1e\:65\:97\:31
# BEGIN RECORD 2.0-876-suse
node.name = iqn.1994-04.jp.co.hitachi:rsd.r90.t.30028.1a001
node.tpgt = 1
node.startup = automatic
node.leading_login = No
iface.hwaddress = 00:0e:1e:65:97:31
iface.ipaddress = 192.168.10.1
iface.iscsi_ifacename = bnx2i.00:0e:1e:65:97:31
iface.net_ifacename = p4p1
iface.prefix_len = 0
iface.transport_name = bnx2i
iface.initiatorname = iqn.1995-05.com.broadcom.iscsiboot
iface.vlan_id = 0
iface.vlan_priority = 0
iface.iface_num = 0
iface.mtu = 0
iface.port = 0
iface.tos = 0
iface.ttl = 0
iface.tcp_wsf = 0
iface.tcp_timer_scale = 0
iface.def_task_mgmt_timeout = 0
```

```
iface.erl = 0
iface.max_receive_data_len = 0
iface.first_burst_len = 0
iface.max_outstanding_r2t = 0
iface.max_burst_len = 0
node.discovery_address = 192.168.10.10
node.discovery_port = 3260
node.discovery_type = send_targets
node.session.initial_cmdsn = 0
node.session.initial_login_retry_max = 8
node.session.xmit_thread_priority = -20
node.session.cmds_max = 128
node.session.queue_depth = 32
node.session.nr_sessions = 1
node.session.auth.authmethod = CHAP
node.session.auth.username = root
node.session.auth.password = welcome123456
node.session.timeo.replacement_timeout = 120
node.session.err_timeo.abort_timeout = 15
node.session.err_timeo.lu_reset_timeout = 30
node.session.err_timeo.tgt_reset_timeout = 30
node.session.err_timeo.host_reset_timeout = 60
node.session.iscsi.FastAbort = Yes
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
node.session.iscsi.FirstBurstLength = 262144
node.session.iscsi.MaxBurstLength = 16776192
node.session.iscsi.DefaultTime2Retain = 0
node.session.iscsi.DefaultTime2Wait = 2
node.session.iscsi.MaxConnections = 1
node.session.iscsi.MaxOutstandingR2T = 1
node.session.iscsi.ERL = 0
node.session.scan = auto
node.session.reopen_max = 0
node.conn[0].address = 192.168.10.10
node.conn[0].port = 3260
node.conn[0].startup = manual
node.conn[0].tcp.window_size = 524288
node.conn[0].tcp.type_of_service = 0
node.conn[0].timeo.logout_timeout = 15
node.conn[0].timeo.login_timeout = 15
node.conn[0].timeo.auth_timeout = 45
node.conn[0].timeo.noop_out_interval = 5
node.conn[0].timeo.noop_out_timeout = 5
node.conn[0].iscsi.MaxXmitDataSegmentLength = 0
node.conn[0].iscsi.MaxRecvDataSegmentLength = 262144
node.conn[0].iscsi.HeaderDigest = None
node.conn[0].iscsi.DataDigest = None
node.conn[0].iscsi.IFMarker = No
node.conn[0].iscsi.OFMarker = No
# END RECORD
```

You can also use the `lsmod` command to verify. For example, to check the bncx2i driver version:

```
localhost:~ # modinfo bnx2i |grep version
version:        2.7.10.1
srcversion:     9AF1D90F9B62B65006D34E1
vermagic:       4.12.14-195-default SMP mod_unload modversions

localhost:~ # dmesg |grep "bnx2i"
[  114.513276] QLogic NetXtreme II iSCSI Driver bnx2i v2.7.10.1 (Jul
16, 2014)
[  114.513308] iscsi: registered transport (bnx2i)
[  114.515407] bnx2i [06:00.01]: ISCSI_INIT passed
[  114.517289] bnx2i [06:00.00]: ISCSI_INIT passed
localhost:~ #
```

3. Restart the system and verify that LUNs are detected in the host. Next, configure multipathing as follows:

```
# mpathconf --enable --find_multipaths y --with_module y --
with_multipathd y
# systemctl enable multipathd
# systemctl start multipathd


# localhost:~ # multipath -ll

360060e8008754c000050754c00000044 dm-8 HITACHI,OPEN-V
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:8 sdj 8:144 active ready running
  `- 5:0:1:8 sdt 65:48 active ready running
360060e8008754c000050754c0000001e dm-3 HITACHI,OPEN-V
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:3 sde 8:64  active ready running
  `- 5:0:1:3 sdo 8:224 active ready running
360060e8008754c000050754c0000001c dm-2 HITACHI,OPEN-V
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:2 sdd 8:48  active ready running
  `- 5:0:1:2 sdn 8:208 active ready running
360060e8008754c000050754c00000041 dm-7 HITACHI,OPEN-V
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:7 sdi 8:128 active ready running
  `- 5:0:1:7 sds 65:32 active ready running
360060e8008754c000050754c00000040 dm-6 HITACHI,OPEN-V
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:6 sdh 8:112 active ready running
  `- 5:0:1:6 sdr 65:16 active ready running
360060e8008754c000050754c0000001a dm-1 HITACHI,OPEN-V
```

```
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:1 sdc 8:32  active ready running
  `- 5:0:1:1 sdm 8:192 active ready running
360060e8008754c000050754c00000019 dm-0 HITACHI,OPEN-V
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:0 sdb 8:16  active ready running
  `- 5:0:1:0 sdl 8:176 active ready running
360060e8008754c000050754c00000033 dm-5 HITACHI,OPEN-V
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:5 sdg 8:96  active ready running
  `- 5:0:1:5 sdq 65:0  active ready running
360060e8008754c000050754c00000045 dm-9 HITACHI,OPEN-V
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:9 sdk 8:160 active ready running
  `- 5:0:1:9 sdu 65:64 active ready running
360060e8008754c000050754c0000001f dm-4 HITACHI,OPEN-V
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 6:0:0:4 sdf 8:80  active ready running
  `- 5:0:1:4 sdp 8:240 active ready running
localhost:~ #
```

Note: For details on multipath configuration, see the configuration manual.

Examples of Some Useful iSCSI Commands

- Show all records in discovery database:

  ```
  # iscsiadm -m discovery
  ```

- Display all discovered nodes from the internal persistent discovery database:

  ```
  # iscsiadm -m node
  ```

- Display all active sessions and connections:

  ```
  # iscsiadm -m session
  ```

- Report iface configurations that are set up in `/etc/iscsi/ifaces`:

  ```
  # iscsiadm -m iface
  ```

- Discover targets at with a specific IP address:

  ```
  # iscsiadm --mode discovery --type sendtargets --portal 192.168.10.10
     iscsiadm --mode discovery --type sendtargets --portal 192.168.10.201
  ```

- To log in, you must use a node record ID found in the discovery:

```
# iscsiadm --mode node --targetname
iqn.1994.04.jp.co.hitachi:rsd.r90.t.30028.1a001 --portal
192.168.10.10:3260 --login
```

- Log out from the iSCSI target:

```
# iscsiadm --mode node --targetname iqn.1994-
04.jp.co.hitachi:rsd.r90.t.30028.1a001 --portal 192.168.10.10:3260 --
logout
```

- To delete a node record, log out from the node before deleting:

```
# iscsiadm --mode node --targetname iqn.1994-
04.jp.co.hitachi:rsd.r90.t.30028.1a001 --portal 192.168.10.10:3260 --op
delete
```

- Log in to all iSCSI portals on every node and target through each interface, set in the database:

```
# iscsiadm -m node -l
```

- Log out of all iSCSI portals on every node and target through each interface set in the database:

```
# iscsiadm -m node -u
```

- Show all records in the discovery database and the targets that were discovered from each record:

```
# iscsiadm -m discovery -P 1
```

- Display session statistics:

```
# iscsiadm -m session -r 1 –stats
```

Note: This function also runs in node mode. Instead of the `-r $sid` argument, you can pass the node info such as target name, portal, and interface.

- Run an iSCSI scan on a session:

```
# iscsiadm -m session -r 1 --rescan
```

Note: This function also runs in node mode. Instead of the `-r $sid` argument, you can pass the node info such as target name, portal, and interface.

Rescanning does not delete the old LUNs. It will only create new ones.

Configuring a Hitachi VSP 5000 Series iSCSI Storage System

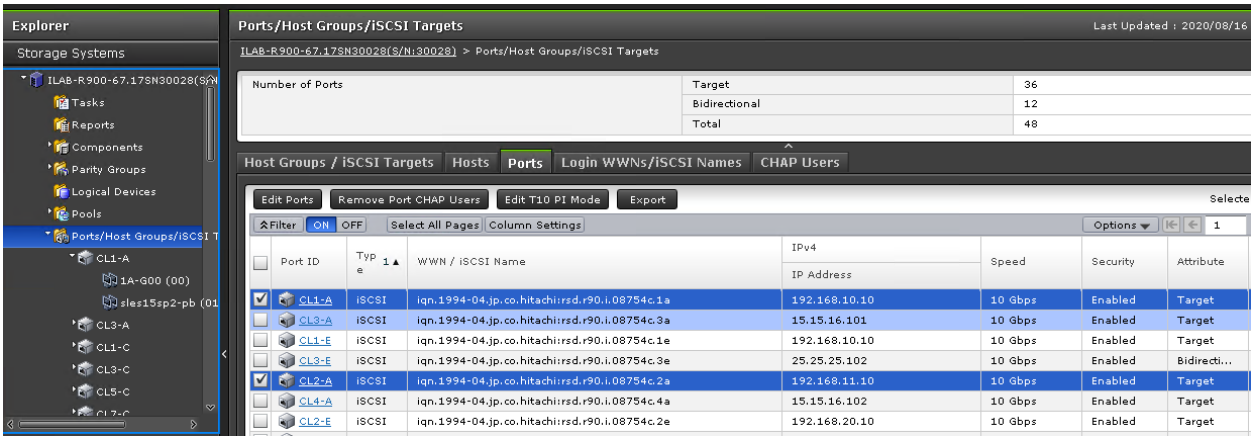
This chapter provides Hitachi iSCSI storage system configuration information. The Hitachi VSP 5000 series has an iSCSI interface that supports other FC interfaces.

- For configuring multipathing to target storage LUNs, add multiple iSCSI targets to the iSCSI host.
- To take advantage of path failure and continuity to data access, you must configure multipathing software in the iSCSI hosts.

Note: You can check support for Hitachi multipathing software Hitachi Dynamic Link Manager (HDLM) on the host operating systems. Hitachi Storage also supports most of operating system multipath software.

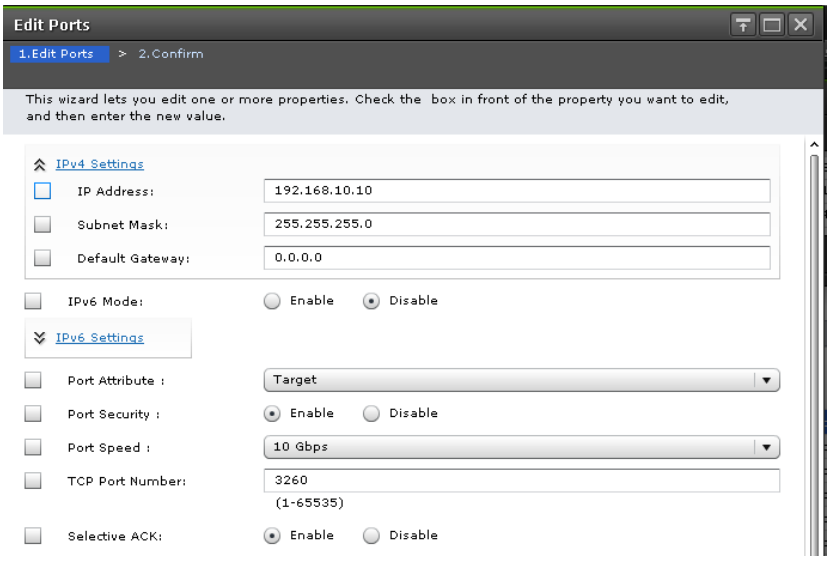
Configuring IP Addresses on iSCSI Ports

1. Set the IP addresses on the iSCSI ports that you will connect with iSCSI adapter ports. Next, select **Ports** and click **Edit**. This opens the **Edit Ports** window. Lastly, set the port IP addresses and any other port-related configurations or default settings.



2. In the **Edit Ports** window, set the IP addresses as shown in the following example:

   - Hitachi iSCSI Storage port CL1-A IP Address: 192.168.10.10
   - Hitachi iSCSI Storage port CL2-A IP Address: 192.168.11.10



Creating an iSCSI Target on the iSCSI Ports

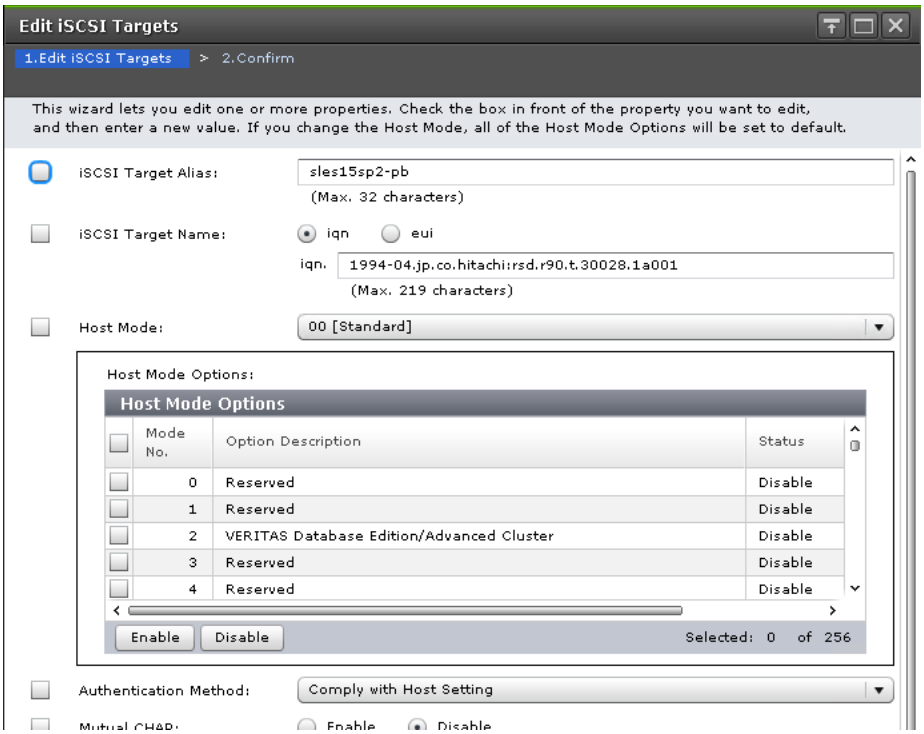
In the Storage Navigator window, click **Create iSCSI Targets**. This opens the **Create iSCSI Targets** window. Next, select the available ports and click **Add** to create new iSCSI targets. Lastly, finish the wizard to complete creating the new iSCSI targets.

Setting the Host Mode

Different **Host Mode** and **Host Mode Options** are recommended based on your operating system. In the following example, we are using Host Mode 00 [Standard] for Linux:

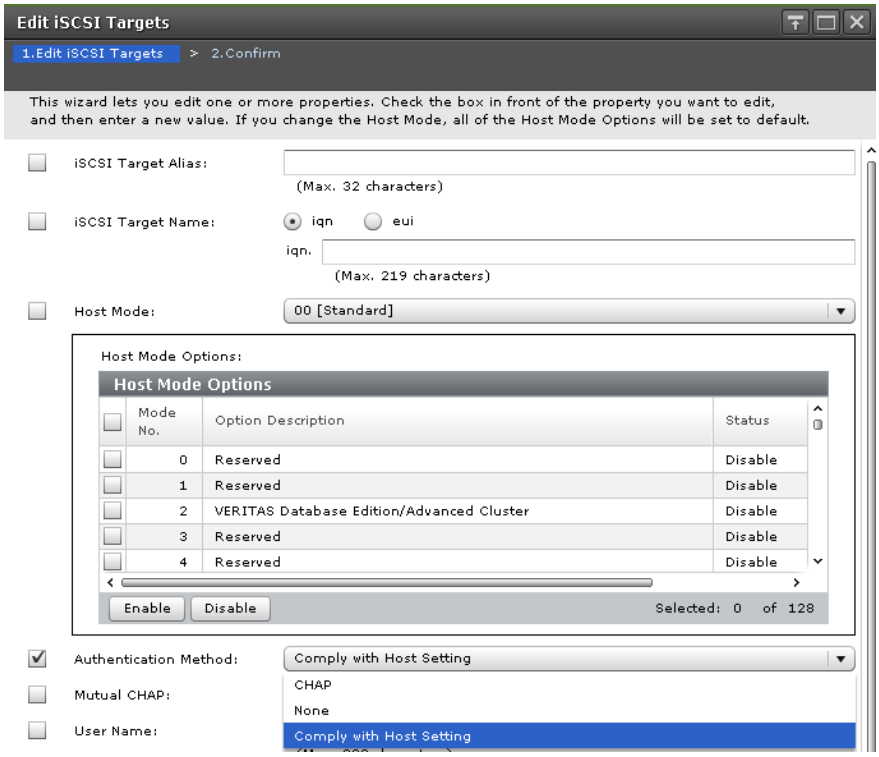
Select the **iSCSI Target** and click **Edit iSCSI Target**. Set **Host Mode** as 00 [Standard] for Linux and **Host Mode Options** according to the host operating system requirements.

The following screenshots were taken after creating the iSCSI target and setting **Host Mode**.

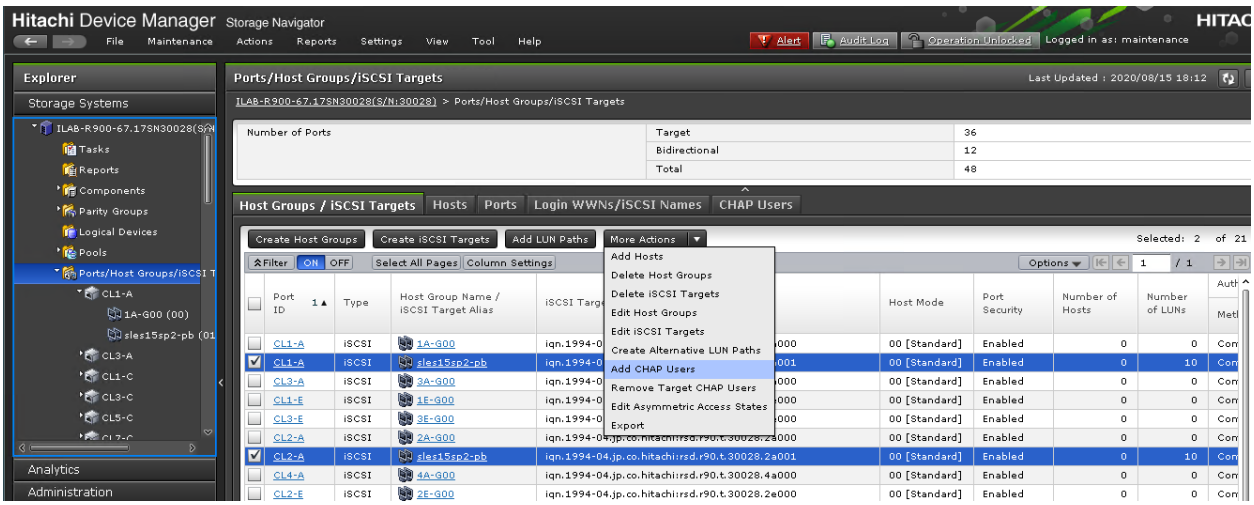


Creating CHAP Users

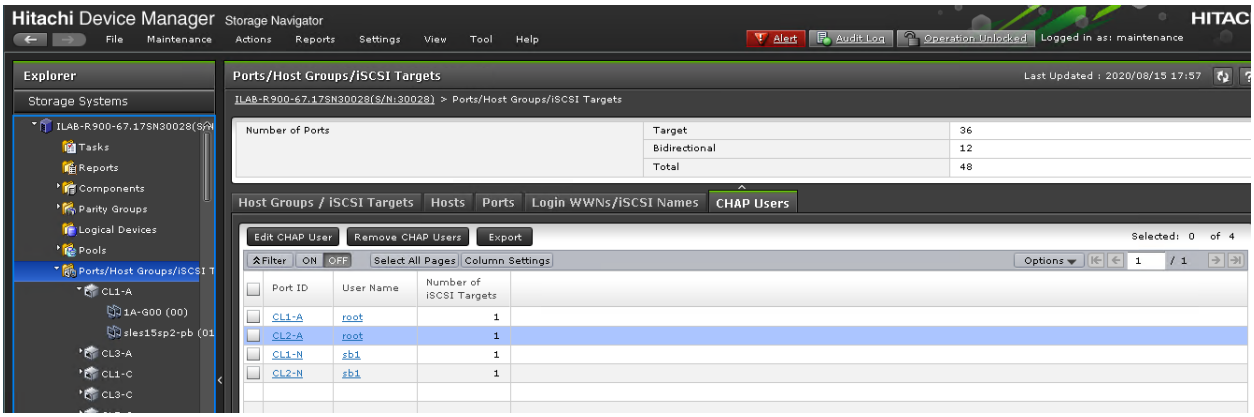
1. Verify that the **Authentication Method** is set to CHAP or **Comply with Host Setting** before creating CHAP users. Next, select the iSCSI targets, browse to **Edit iSCSI Targets,** and set the **Authentication Method** to CHAP or **Comply with Host Setting**.



2. Create a CHAP user on the iSCSI targets. The CHAP user is used for iSCSI authentication. Click **Add CHAP Users** as follows:

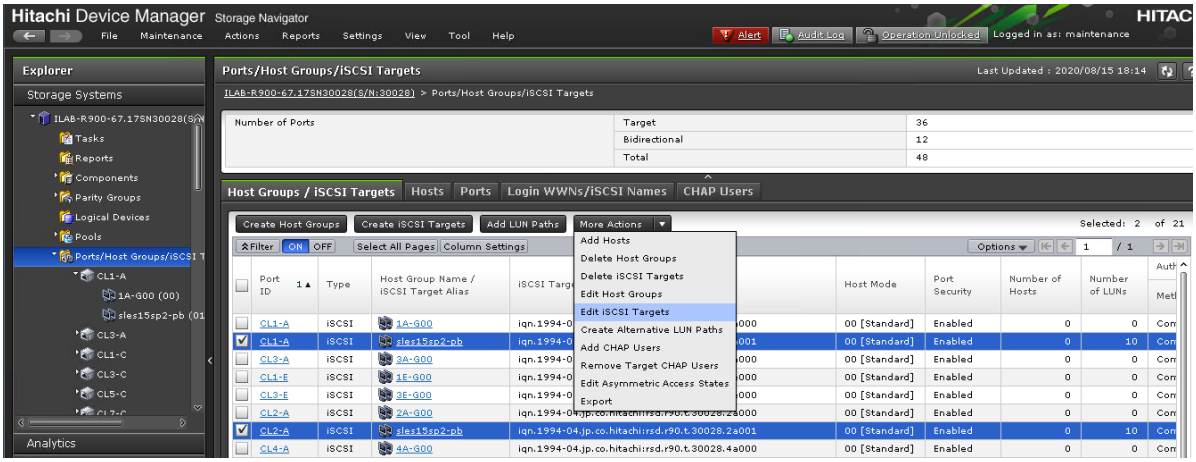


3. After a CHAP user root is created, set the password for the CHAP user to: welcome12345.

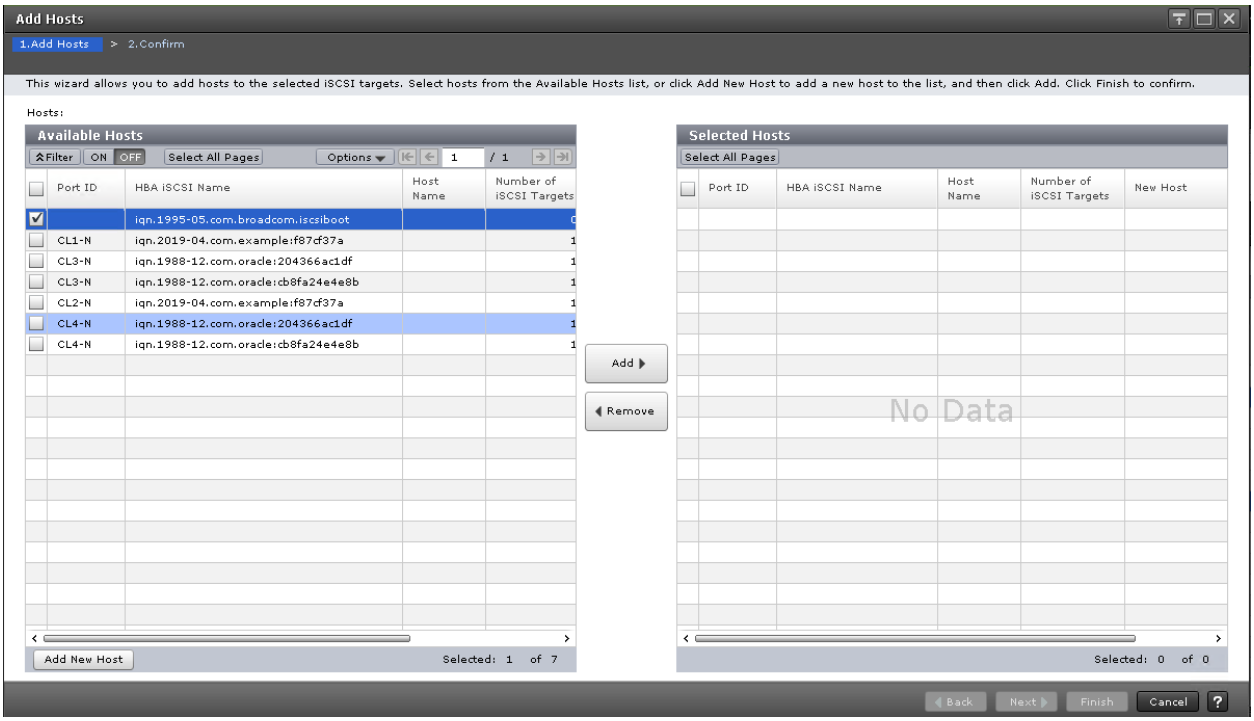


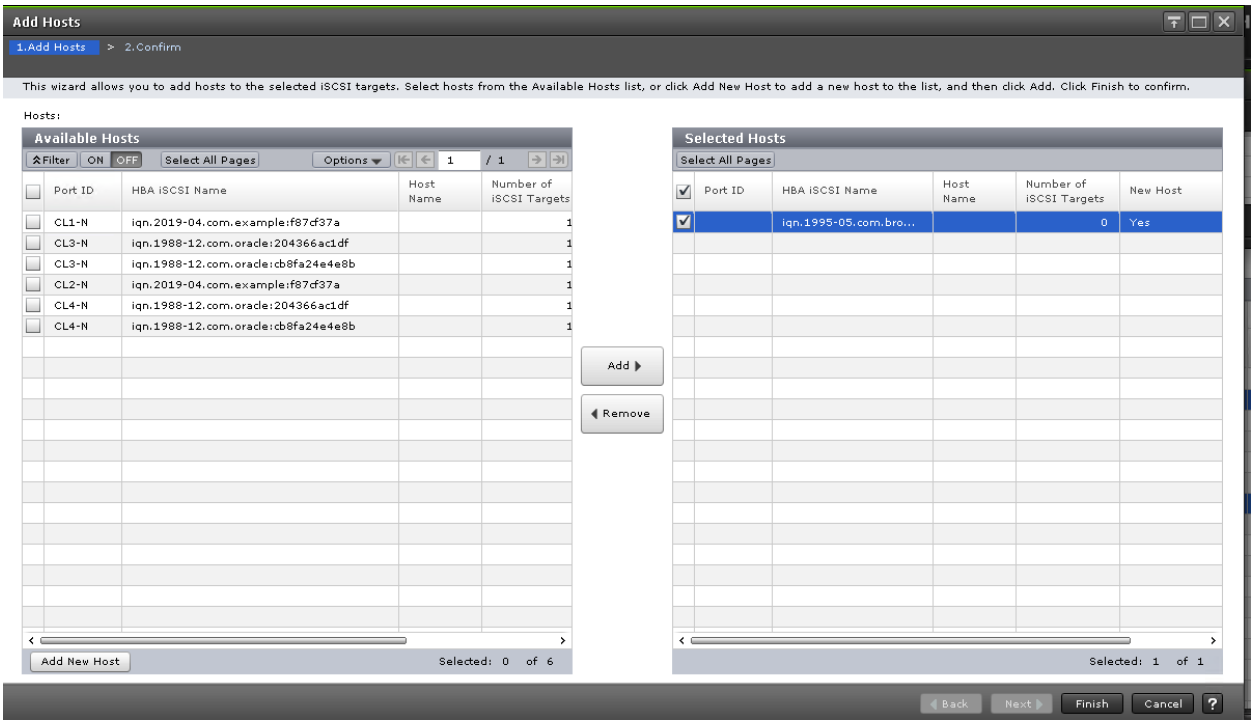
Adding a Host to the iSCSI Targets

1. Select the iSCSI targets for which you want to add the host iSCSI adapter initiator. From the More Actions drop-down list, select as follows:



2. If an adapter iSCSI name is not in available the host list, click **Add New Host** to add the iSCSI name in the following window. Add the host and then click **Finish**.


Add Hosts
1.Add Hosts > 2.Confirm
This wizard allows you to add hosts to the selected iSCSI targets. Select hosts from the Available Hosts list, or click Add New Host to add a new host to the list, and then click Add. Click Finish to confirm.
Hosts:
Available Hosts
Filter ON OFF
Select All Pages
Options
1 / 1
Port ID
HBA iSCSI Name
Host Name
Number of iSCSI Targets
iqn.1995-05.com.broadcom.iscsiboot
CL1-N iqn.2019-04.com.example:f87cf37a 1
CL3-N iqn.1988-12.com.oracle:204366ac1df 1
CL3-N iqn.1988-12.com.oracle:cb8fa24e4e8b 1
CL2-N iqn.2019-04.com.example:f87cf37a 1
CL4-N iqn.1988-12.com.oracle:204366ac1df 1
CL4-N iqn.1988-12.com.oracle:cb8fa24e4e8b 1
Add New Host
Selected: 1 of 7
Add
Remove
Selected Hosts
Select All Pages
Port ID
HBA iSCSI Name
Host Name
Number of iSCSI Targets
New Host
No Data
Selected: 0 of 0
Back
Next
Finish
Cancel


Add Hosts
1.Add Hosts > 2.Confirm
This wizard allows you to add hosts to the selected iSCSI targets. Select hosts from the Available Hosts list, or click Add New Host to add a new host to the list, and then click Add. Click Finish to confirm.
Hosts:
Available Hosts
Filter ON OFF
Select All Pages
Options
1 / 1
Port ID
HBA iSCSI Name
Host Name
Number of iSCSI Targets
CL1-N iqn.2019-04.com.example:f87cf37a 1
CL3-N iqn.1988-12.com.oracle:204366ac1df 1
CL3-N iqn.1988-12.com.oracle:cb8fa24e4e8b 1
CL2-N iqn.2019-04.com.example:f87cf37a 1
CL4-N iqn.1988-12.com.oracle:204366ac1df 1
CL4-N iqn.1988-12.com.oracle:cb8fa24e4e8b 1
Add New Host
Selected: 0 of 6
Add
Remove
Selected Hosts
Select All Pages
Port ID
HBA iSCSI Name
Host Name
Number of iSCSI Targets
New Host
iqn.1995-05.com.bro... 0 Yes
Selected: 1 of 1
Back
Next
Finish
Cancel

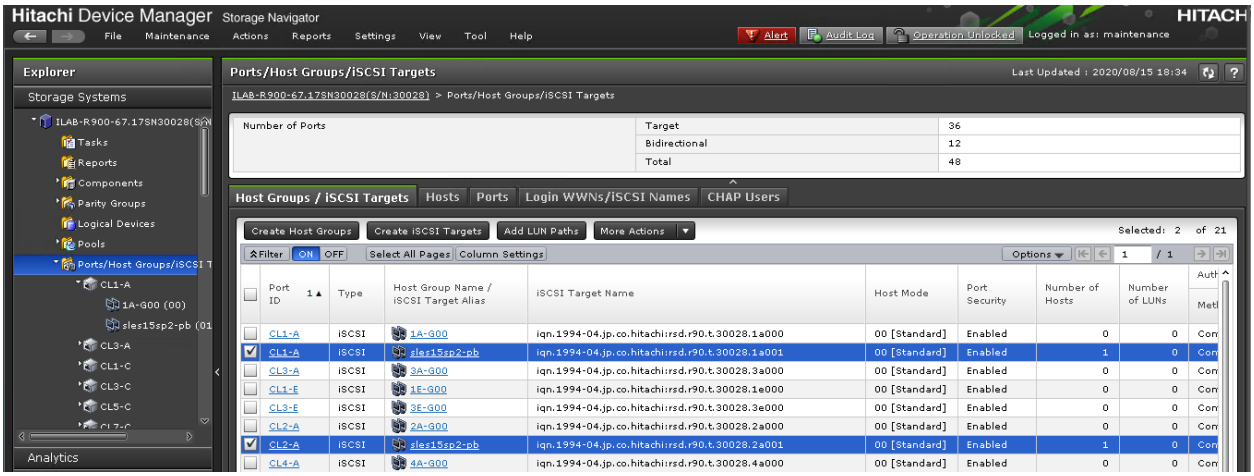
The hardware iSCSI initiator `iqn.1995-05.com.broadcom.iscsiboot` is added to the iSCSI targets as the allowed host for iSCSI login.

Note: You can also add the software iSCSI initiator `iqn.1996-04.de.suse:01:672242fc424` for target login.

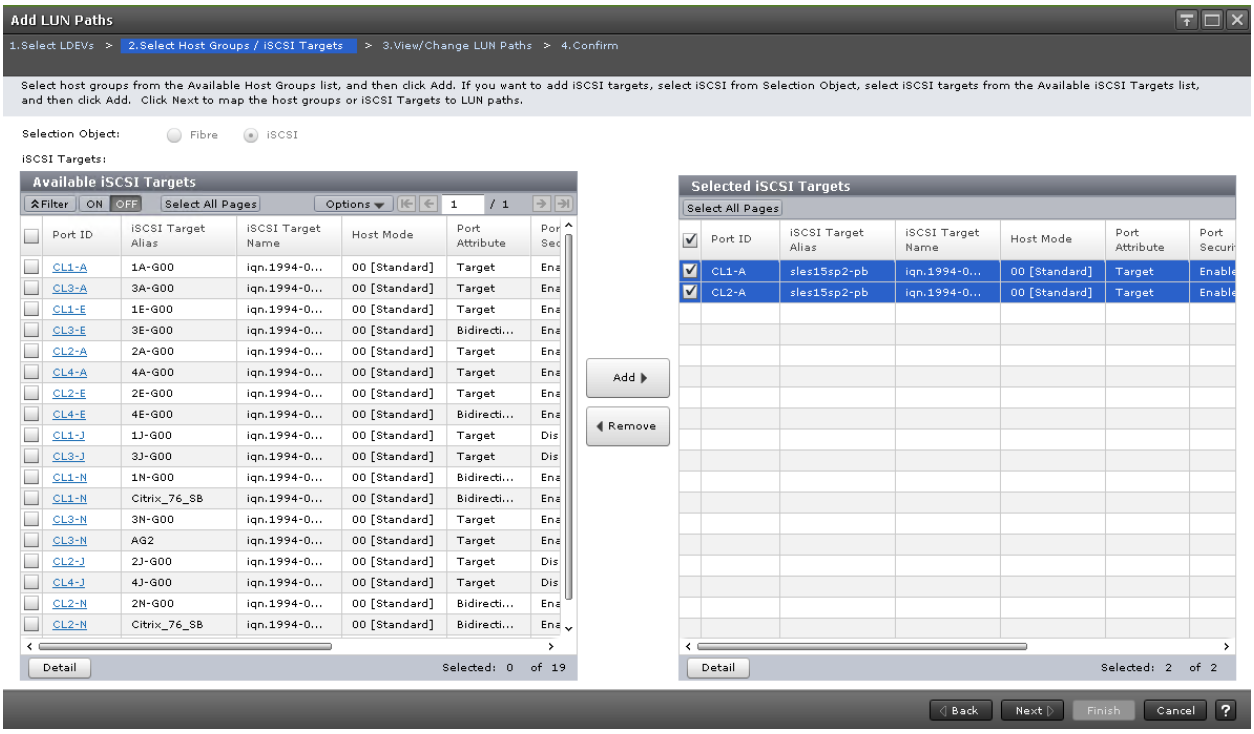
Assign LUNs to iSCSI Targets

1. To assign to iSCSI targets, create a new LDEV or use an existing LDEV.

   After you decide which LDEVs to add to the iSCSI targets, go to the Host Groups/iSCSI tab and click **Add LUN Paths**.



2. Select the iSCSI Targets for which you want to add you want to add LDEVs, and click **Add**.



3. To complete your LUN Paths addition, select the available LDEVs and click **Finish**.

Hitachi Vantara

Corporate Headquarters 2535 Augustine Drive
Santa Clara, CA 95054 USA www.HitachiVantara.com community.HitachiVantara.com

Regional Contact Information
Americas: +1 866 374 5822 or info@hitachivantara.com
Europe, Middle East and Africa: +44 (0) 1753 618000 or info.emea@hitachivantara.com
Asia Pacific: +852 3189 7900 or info.marketing.apac@hitachivantara.com