HITACHI
Inspire the Next

aws PARTNER Premier Tier Services

Global-Active Device Quorum on AWS Cloud

iSCSI Target Using AWS Virtual Machine

Hitachi Vantara

April 2022

HITACHI
Inspire the Next

Table of Contents

Notices and Disclaimer

About This Guide

This guide provides instructions for deploying a virtual machine in the Amazon Web Services (AWS) cloud and configuring it as an iSCSI target.

After the original publication of this whitepaper, Hitachi Vantara released a virtual machine image on Amazon Marketplace that automates the process of configuring targetcli. The solution is available for free on the Amazon Marketplace. However, you must pay for various AWS fees relating to running the virtual machine. The solution can be found at https://aws.amazon.com/marketplace/pp/prodview-7yn64ltekhjus.

Intended Audience

This document is intended for Hitachi Vantara and Hitachi partner representatives who need a foundation of knowledge on this product to best represent it to potential buyers.

Document Revisions

Revision Number	Date	Details
1.0	November 2019	Initial release.
1.1	April 2022	Added recommendation to remove public IP address on the AWS virtual machine and added GAD Cloud Quorum solution. Also made wording corrections and formatting updates.

References

- Hitachi Global-Active Device User Guide

Contributors

The information included in this document represents the expertise, feedback, and suggestions of a number of skilled practitioners. The author (Dang Luong) wants to recognize and thank the following contributors and reviewers of this document (listed alphabetically by last name):

- Tom Attanese - Product Management
- Paul Romero - Global Product & Solutions Enablement

Comments

Please send your comments on this document to gpse.replicationsoftware@hitachivantara.com. Include the document title and number, including the revision level, and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara. Thank You!

Configuration and Specifications

Introduction

This guide provides instructions for deploying a virtual machine in the Amazon Web Services (AWS) cloud and configuring it as an iSCSI target. We will use the Linux package “targetcli” to create and manage block devices on the virtual machine. The objective is to leverage volumes from the iSCSI target virtual machine running on AWS as quorum volumes for Global-active device (GAD).

Only use volumes from an iSCSI target virtual machine for global-active device quorums. Do not use them as data volumes.

This guide does not include instructions for establishing a VPN connection to AWS. Refer to the AWS documentation, such as AWS Site-to-Site VPN.

Figure 1 illustrates the test environment. The on-premise datacenter is connected to the AWS cloud using a VPN tunnel. Network traffic is passed between the on-premise storage systems and the iSCSI target virtual machine in AWS using the VPN tunnel.

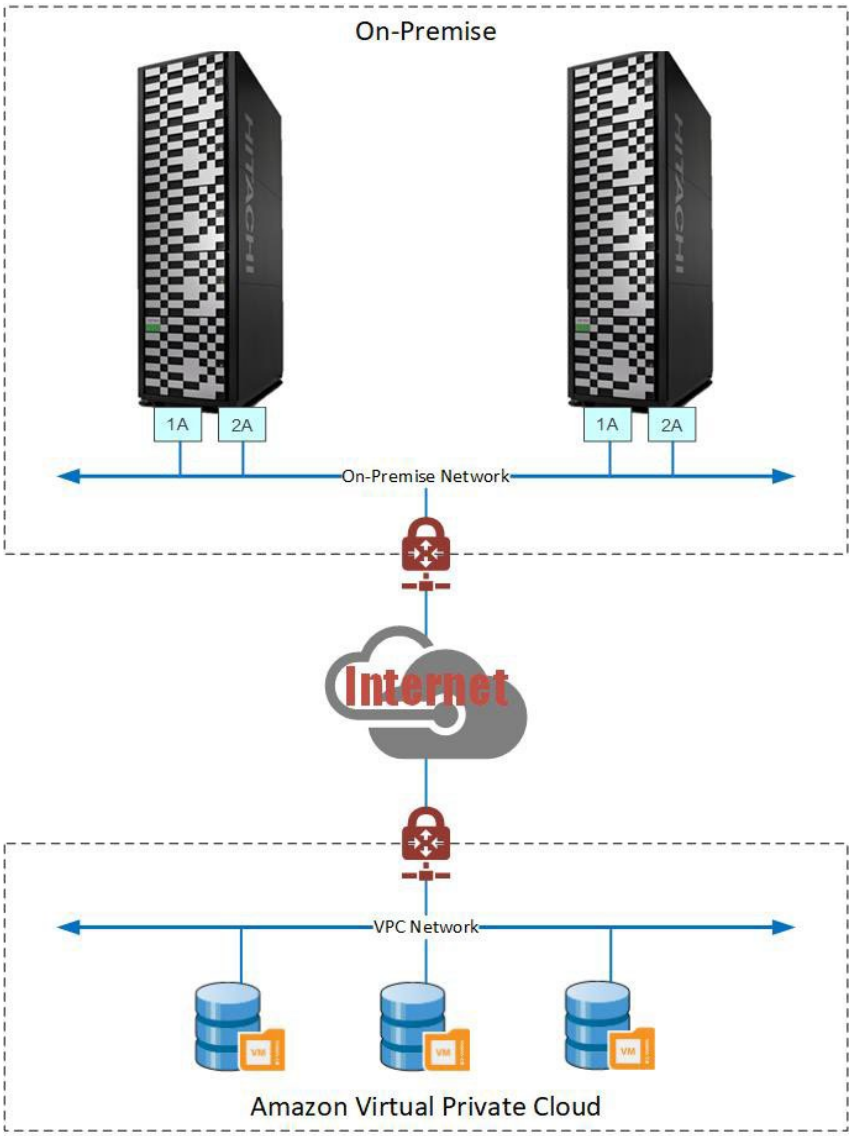


Figure 1. *Test Environment*

AWS Virtual Machine

The following settings were used for the virtual machine image:

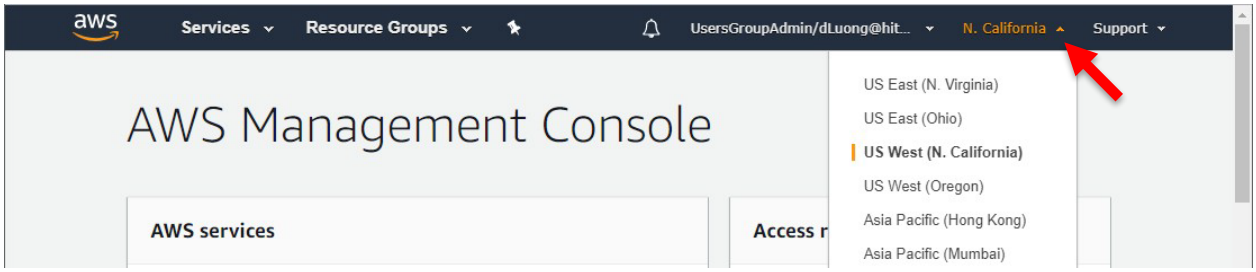
- Operating system: Amazon Linux 2
- Kernel: 4.14.123-111.109.amzn2.x86_64
- Instance type: t2.nano
  - CPU: Intel Xeon CPU E5-2676 v3 @ 2.40 GHz
  - Memory: 512 MB
- Targetcli version: targetcli-2.1.fb46-6.amzn2.noarch

Amazon Virtual Machine Instance

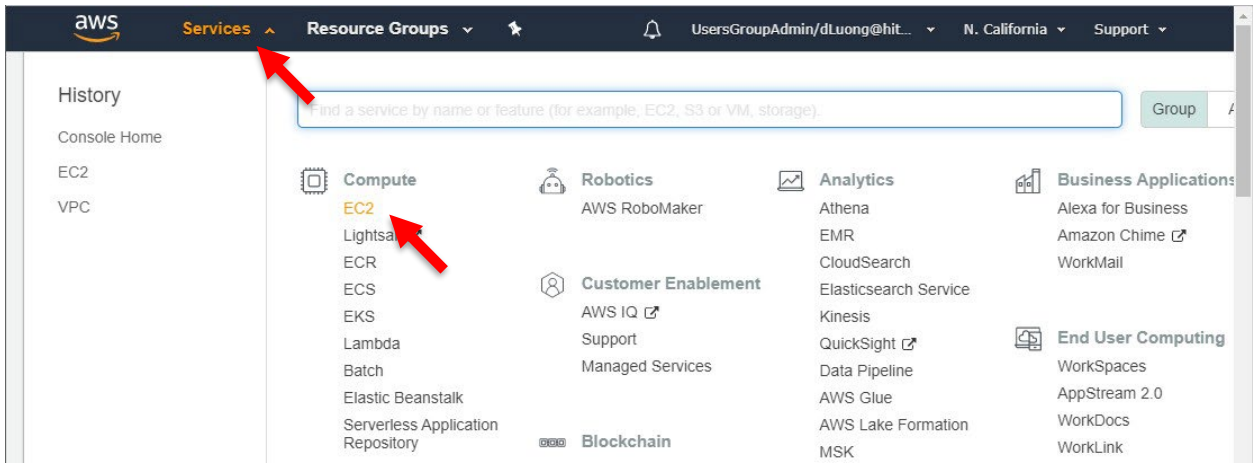
Deployment

This section provides instructions for deploying the virtual machine using an Amazon Machine Instance.

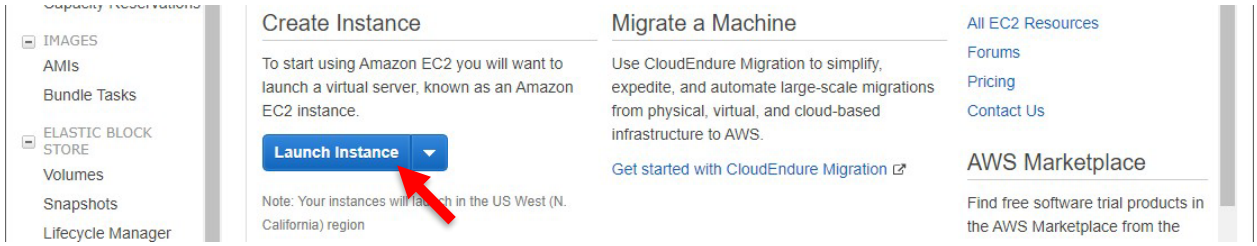
1. In the AWS Management Console, use the top-right shortcut to expand the **Region** list and select a region.



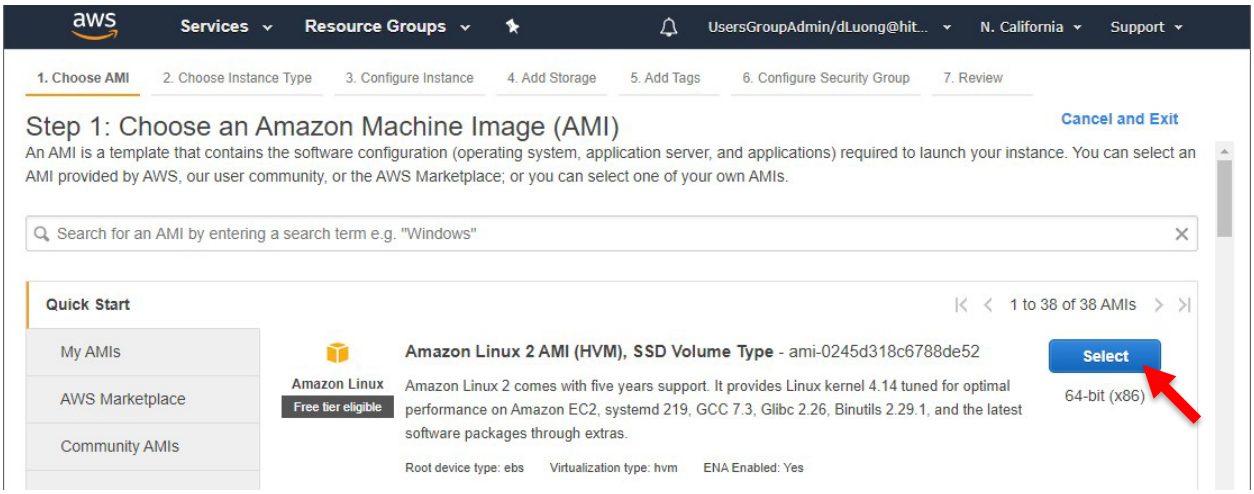
2. On the top left, select **Services** > **Compute** > **EC2**.



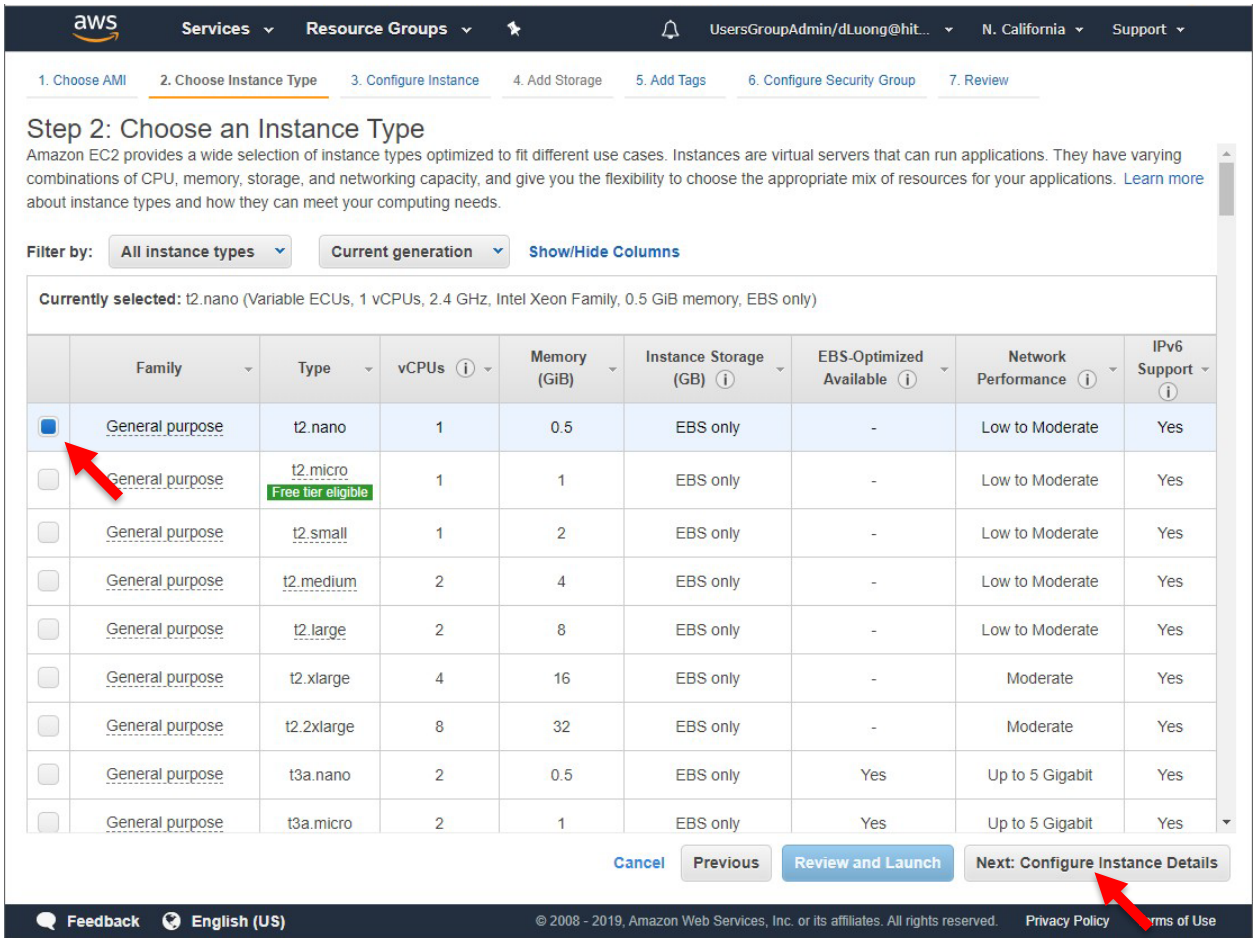
3. Click **Launch Instance**.



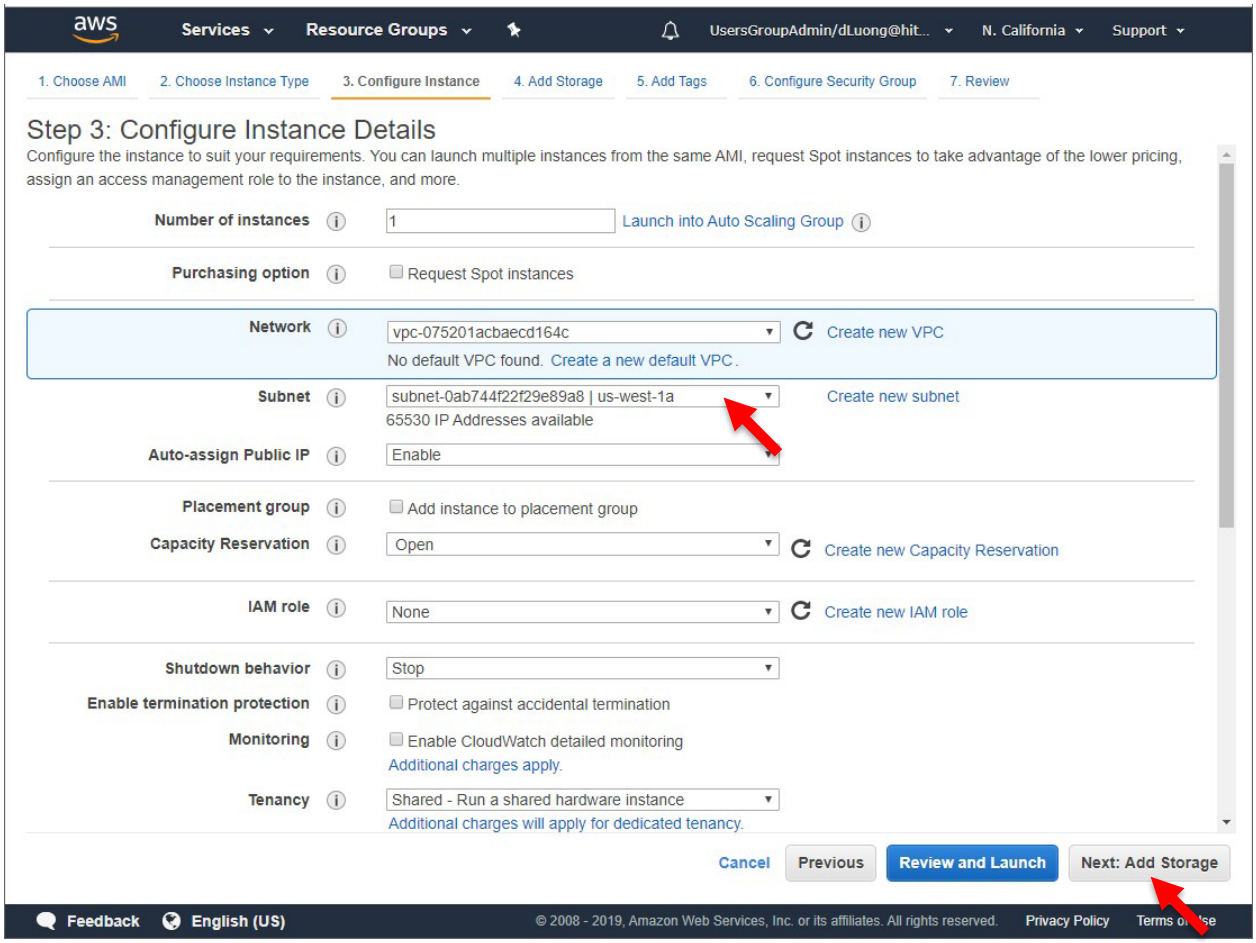
4. Locate **Amazon Linux 2 AMI** and click **Select**.



5. Select the instance type (we tested with the t2.nano type) and click **Next: Configure Instance Details**.



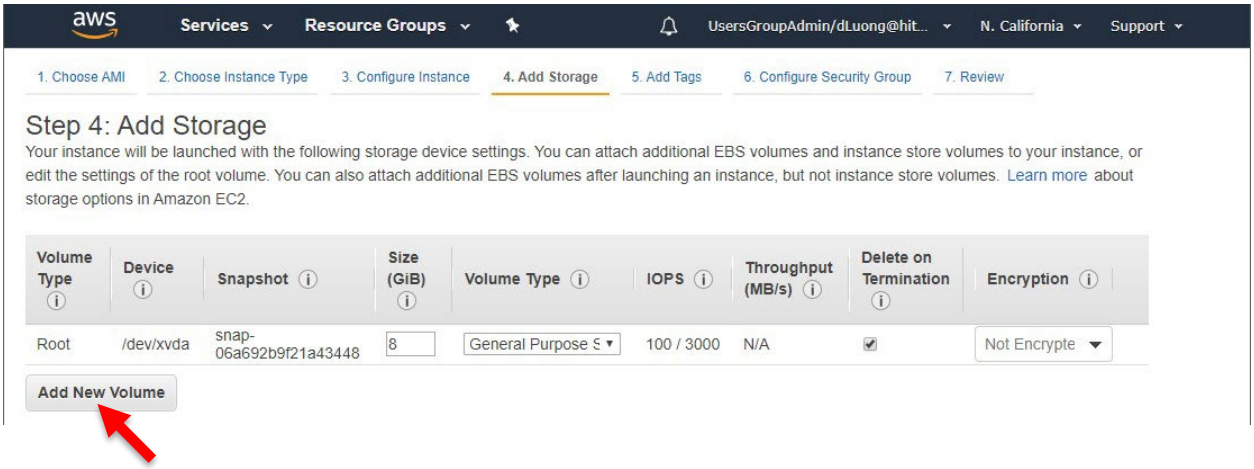
6. From the **Network** dropdown list, select a network. For the initial configuration, we enabled the **Auto-assign Public IP** option to remotely access the virtual machine and download targetcli packages. Click **Next: Add Storage**.



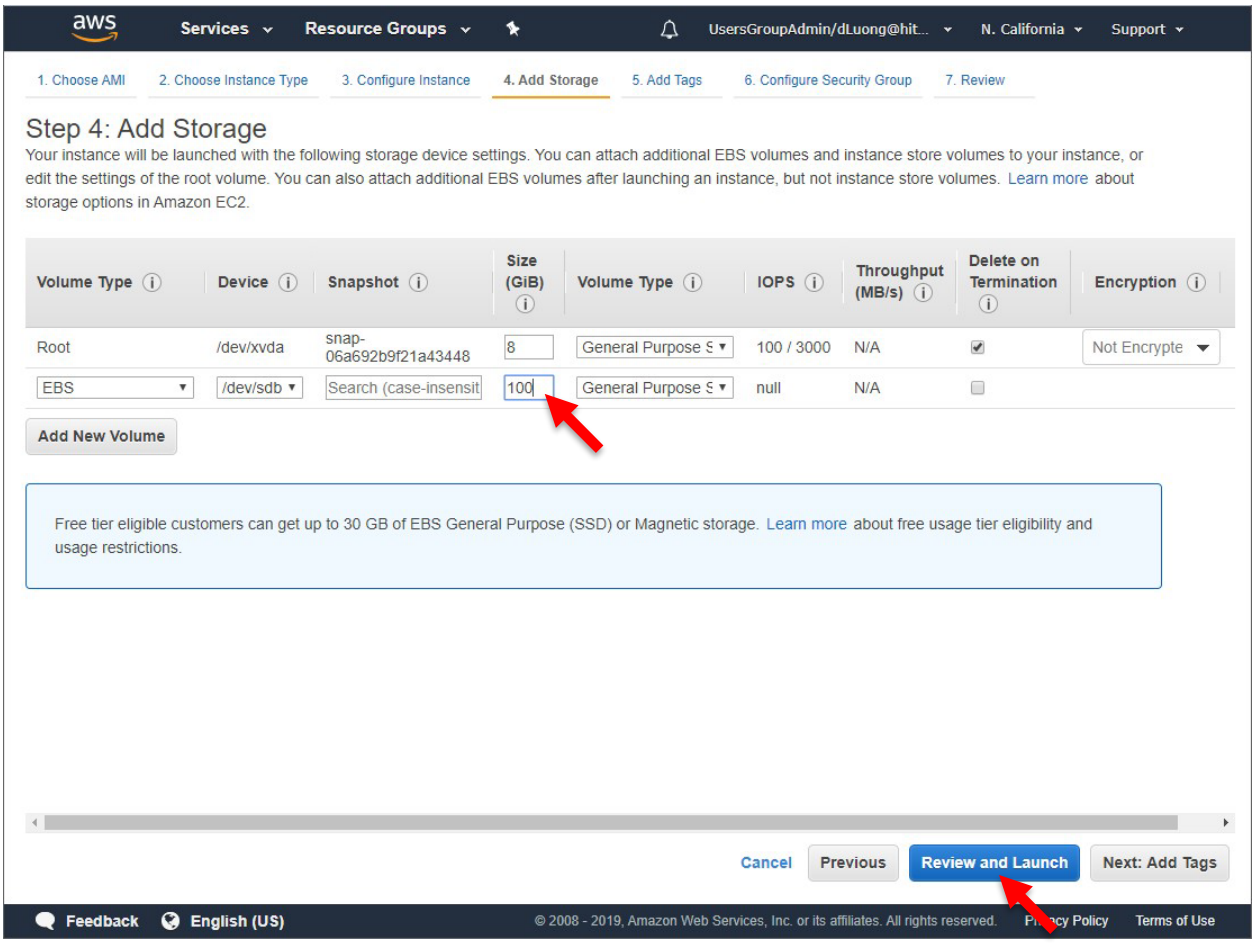
After successfully setting up the virtual machine as an iSCSI target, you must secure the solution by removing the public IP.

7. Click **Add New Volume**.

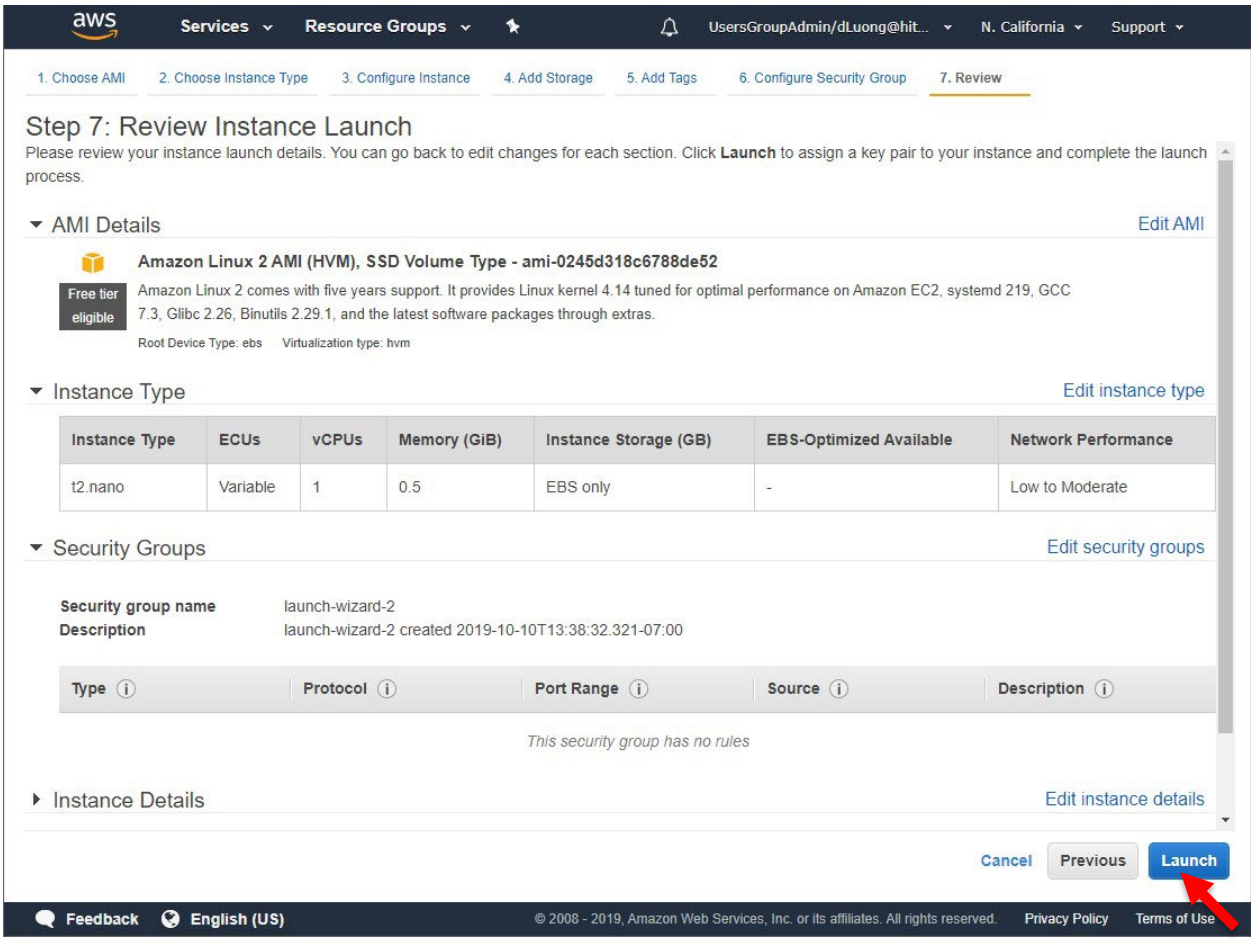
   The new volume will provide backend storage for the quorums.



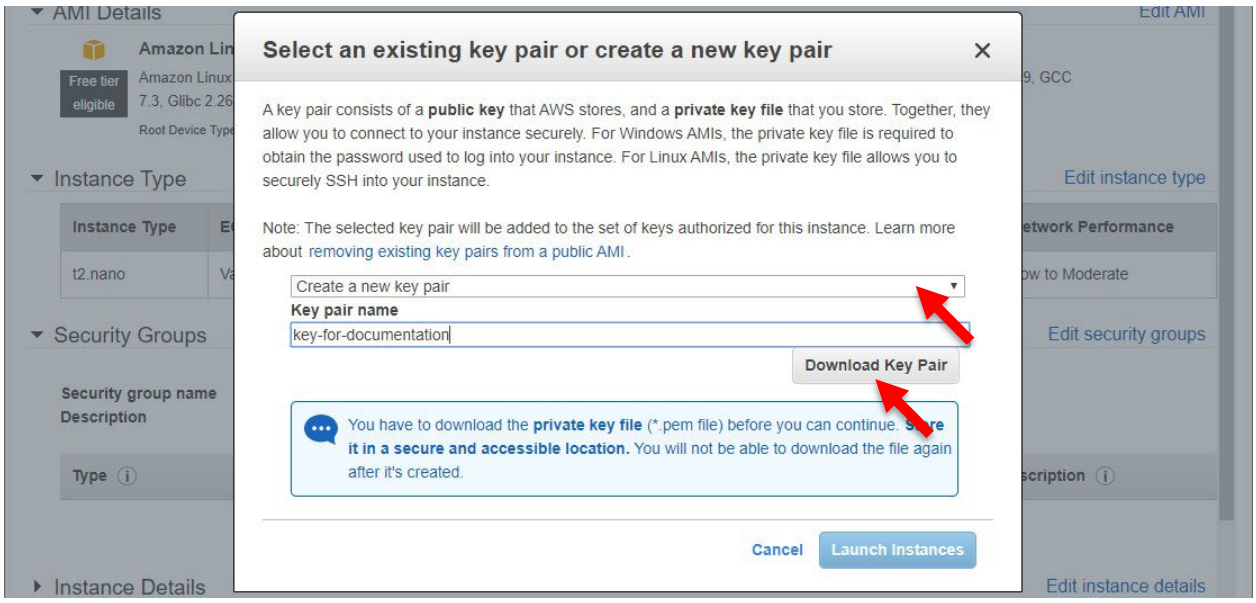
8. Enter the capacity for the new volume and click **Review and Launch**.



9. Verify the details and click **Launch**.



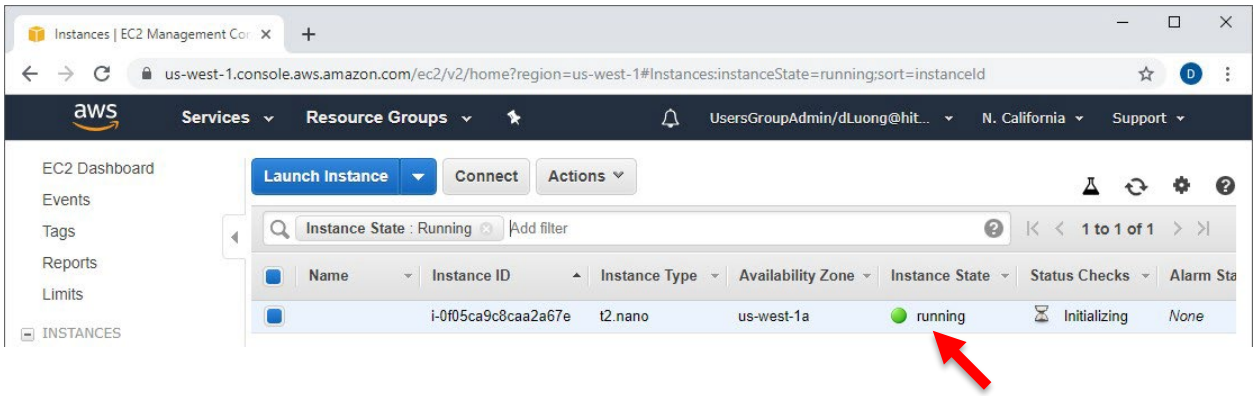
10. If you do not have an existing key pair or do not want to use an existing key pair, use the dropdown list to select **Create a new key pair**. Enter a name for the pair and click **Download Key Pair**.



11. Ensure that you download the `pem` file to your local machine and can locate it. Click **Launch Instances**.

    You can convert the pem file to ppk format, which can be used by PuTTY. For instructions to convert, see: https://tecadmin.net/convert-pem-to-ppk-via-putty/

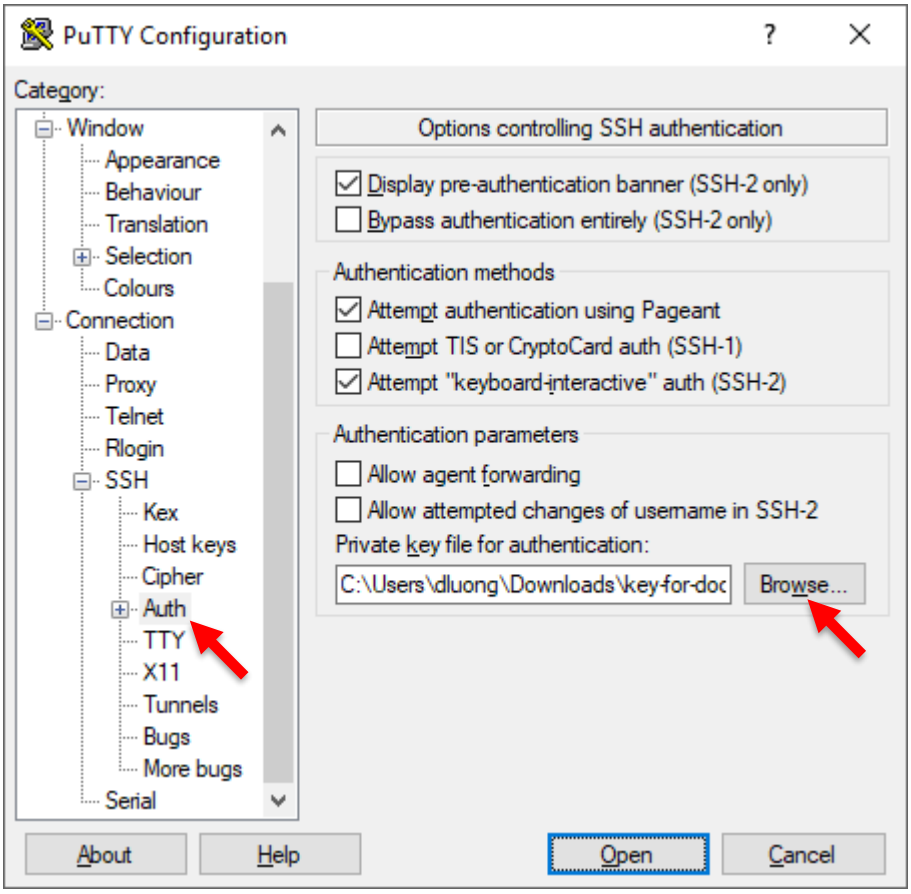
12. Verify that the **Instance State** of the new instance is running on the **Instances** screen. The new instance must be online and accessible.



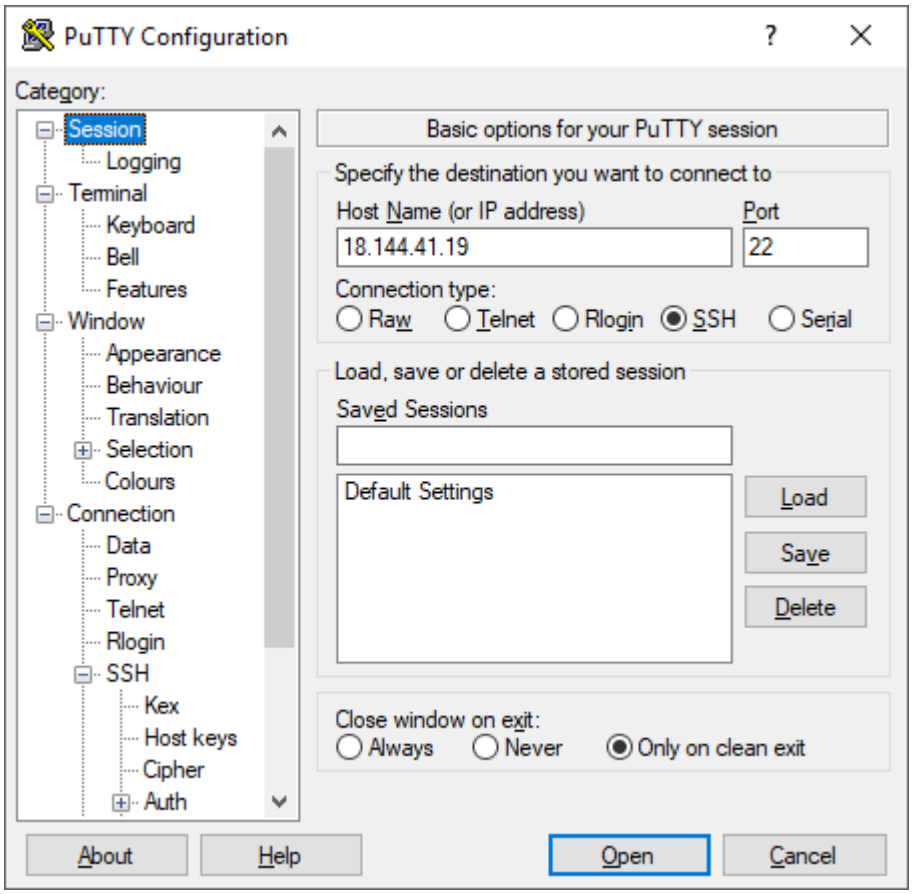
Remote Access

This section provides instructions for remotely accessing the new virtual machine using PuTTY.

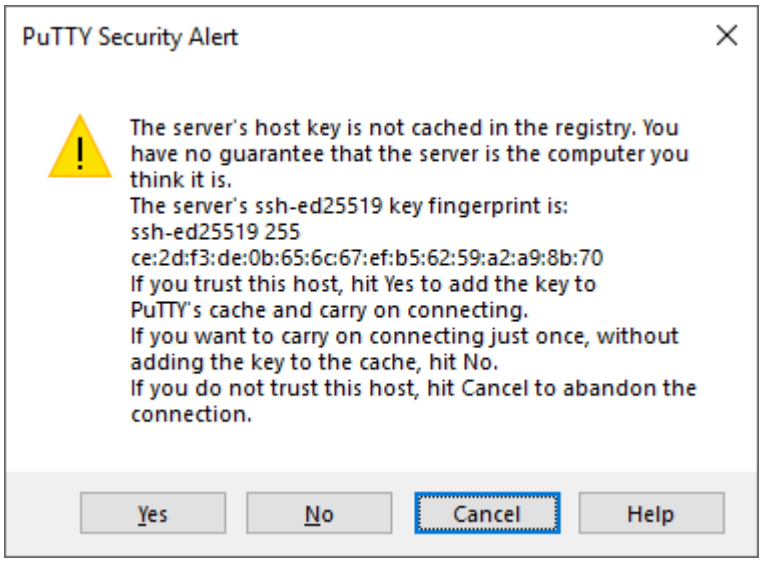
1. On the PuTTY Configuration window, under **Category**, select **Connection** > **SSH** > **Auth**. Click **Browse** to locate the `ppk` file.



2. Under **Category**, select **Session**, and enter the **IP address**. Under **Connection type**, select **SSH** and then click **Open**.

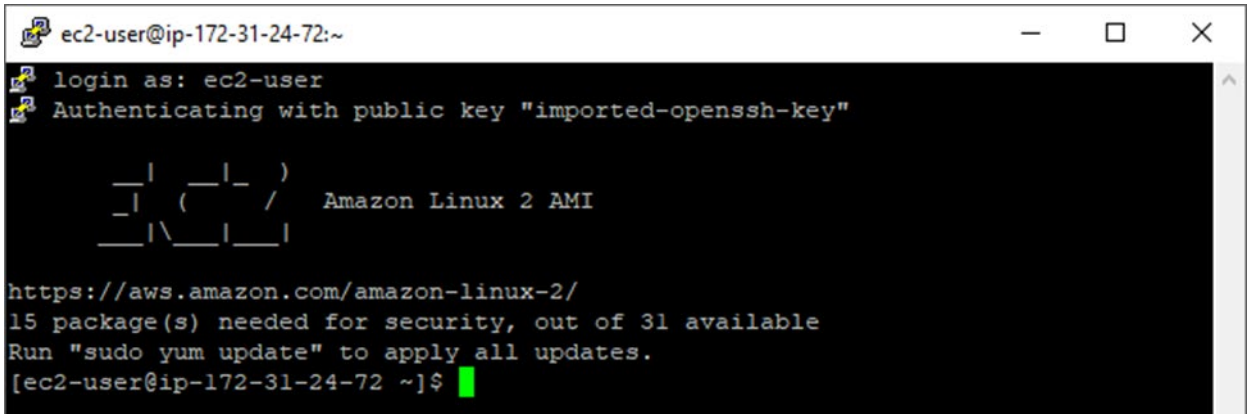


3. To accept the host key, click **Yes**.



4. For the login name, enter: **ec2-user**.

The authentication is completed with the public key.

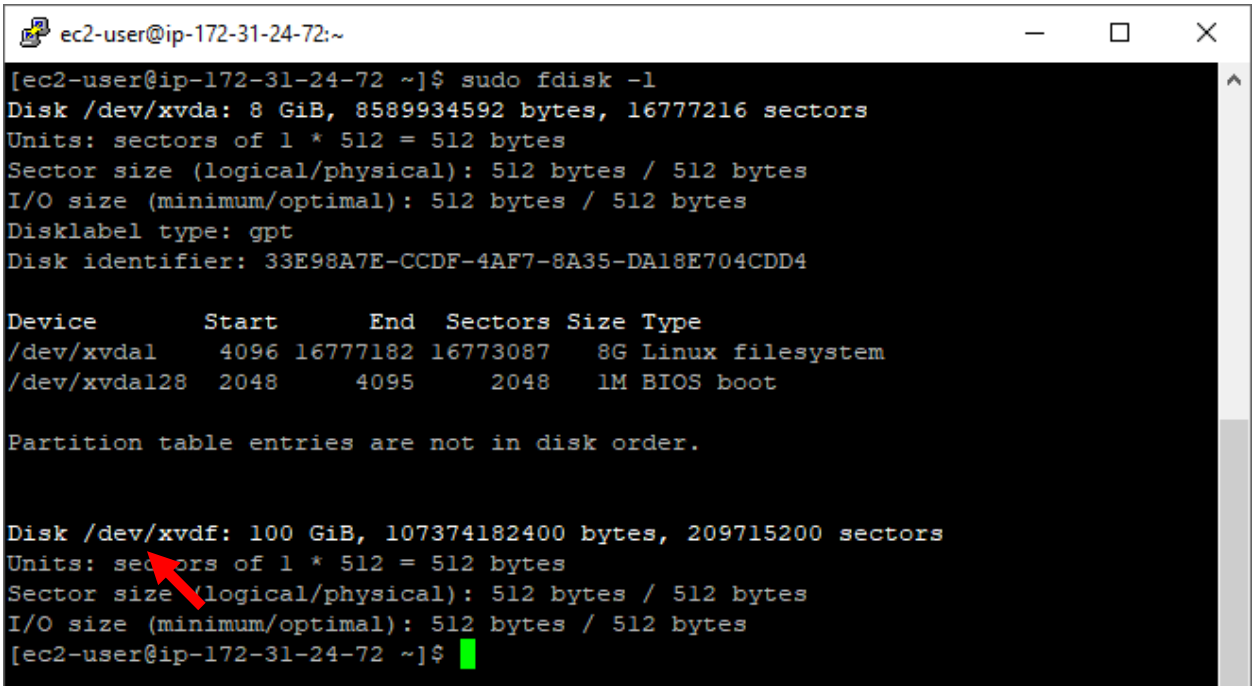


Storage Repository

This section provides instructions for creating a storage repository for storing block devices that will be presented from the virtual machine.

1. Verify that the second volume attached to the virtual machine exists by running the following command:

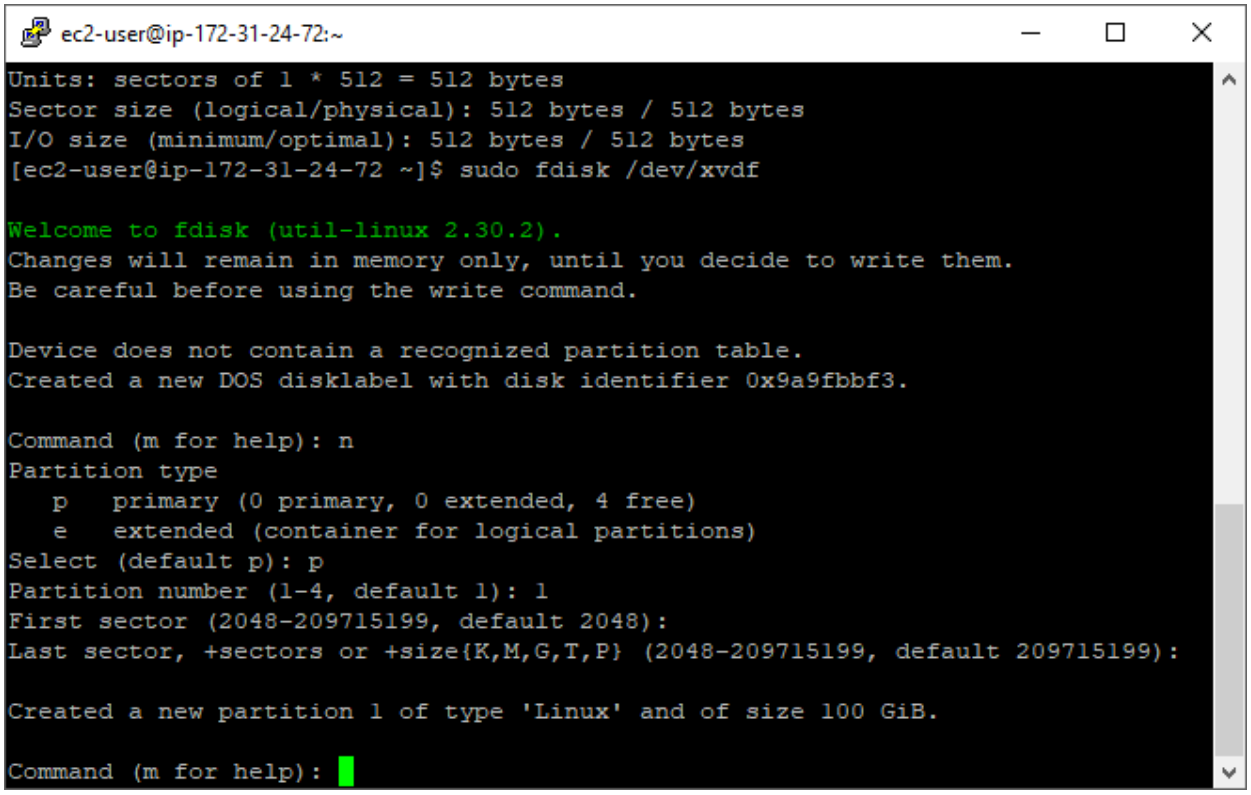
```
sudo fdisk -l
```



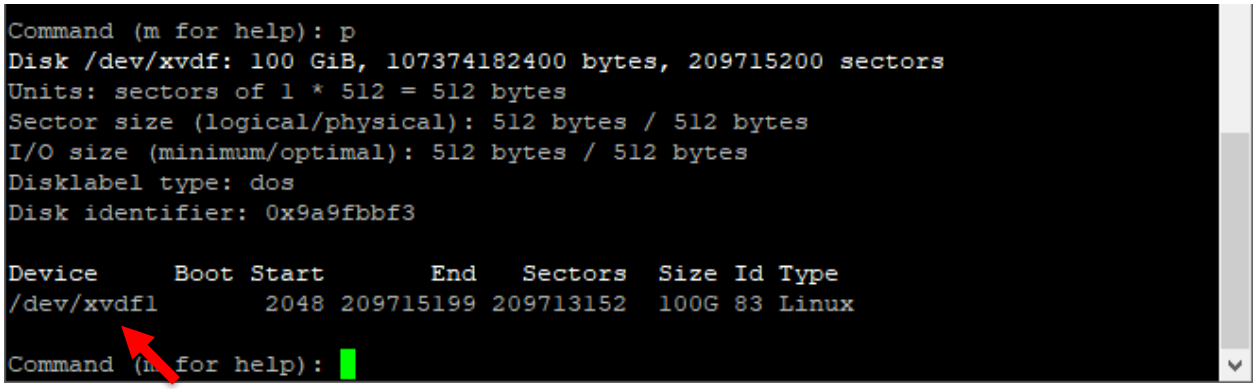
2. Create a partition on the volume by running the following command:

   ```
   sudo fdisk /dev/xvdf
   ```

3. Create a partition that fills up the entire volume:
   a. On the fdisk main menu, enter: **n**
   b. For Partition type, enter: **p**
   c. For Partition number, enter: **1**
   d. To accept default of 2048 for the first sector, press **Enter**.
   e. To accept default of max for the last sector, press **Enter**.



4. To verify the new partition, enter **p**.



5. To write changes and close fdisk, enter **w**.

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[ec2-user@ip-172-31-24-72 ~]$
```

6. To create a volume group on top of the new partition, run the following command:

```
sudo vgcreate VG_quorums /dev/xvdf1
```

```
ec2-user@ip-172-31-24-72:~
[ec2-user@ip-172-31-24-72 ~]$ sudo vgcreate VG_quorums /dev/xvdf1
  Physical volume "/dev/xvdf1" successfully created.
  Volume group "VG_quorums" successfully created
[ec2-user@ip-172-31-24-72 ~]$
```

7. Within the new volume group, to create a logical volume that spans 100% of the volume group, run the following command:

```
sudo lvcreate -l 100 VG_quorums
```

```
ec2-user@ip-172-31-24-72:~
[ec2-user@ip-172-31-24-72 ~]$ sudo lvcreate -l 100 VG_quorums
  Logical volume "lvol0" created.
[ec2-user@ip-172-31-24-72 ~]$
```

8. To create an XFS file system on top of the logical volume, run the following command:

```
sudo mkfs.xfs /dev/VG_quorums/lvol0
```

```
ec2-user@ip-172-31-24-72:~
[ec2-user@ip-172-31-24-72 ~]$ sudo mkfs.xfs /dev/VG_quorums/lvol0
meta-data=/dev/VG_quorums/lvol0  isize=512    agcount=4, agsize=25600 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=0
data     =                       bsize=4096   blocks=102400, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=855, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[ec2-user@ip-172-31-24-72 ~]$
```

9. To make a mount point, run the following command:

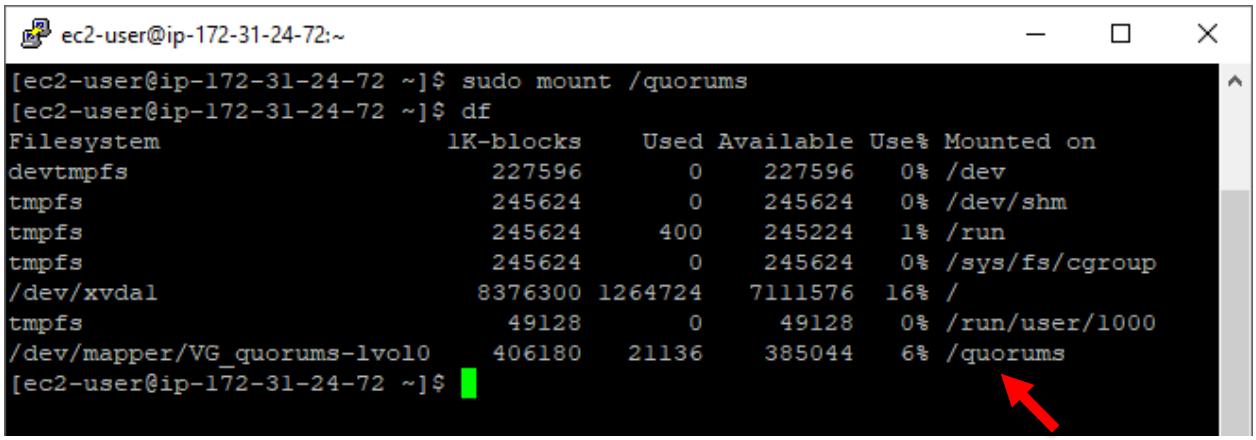
```
sudo mkdir /quorums
```

10. To mount file system automatically during a reboot, add the following line in the `/etc/fstab` file:

```
/dev/VG_quorums/lvol0 /quorums       xfs    defaults      0      0
```

11. Verify that the fstab addition works by running the following command:

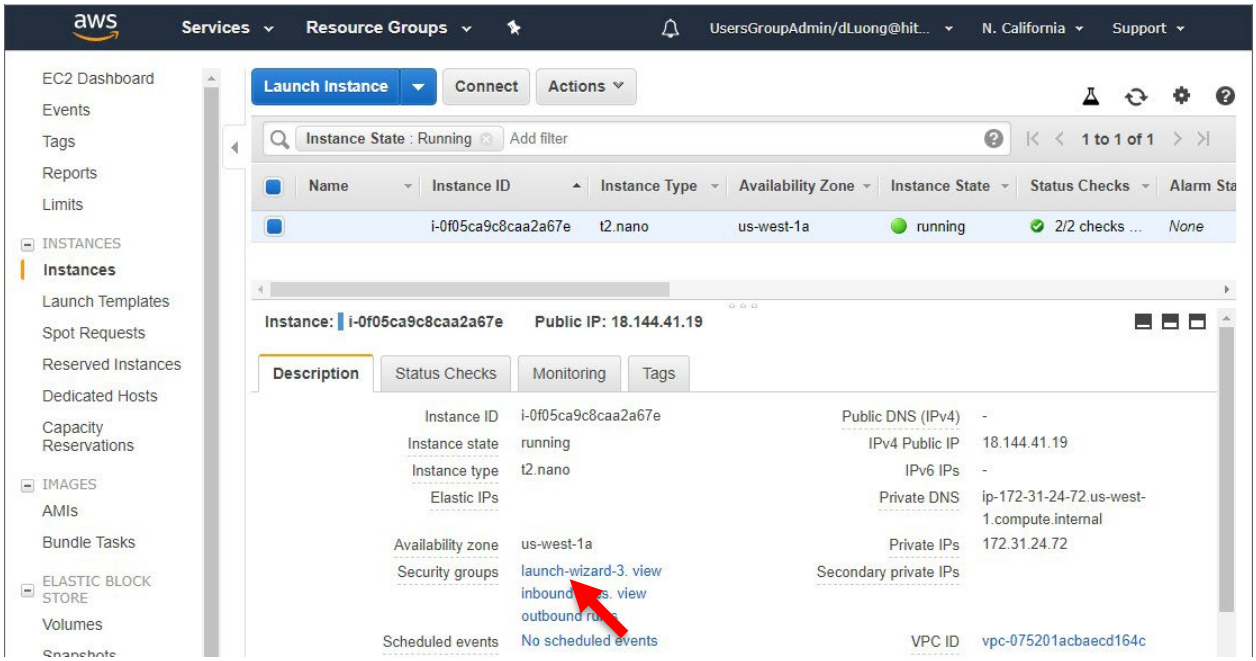
```
sudo mount /quorums
df
```



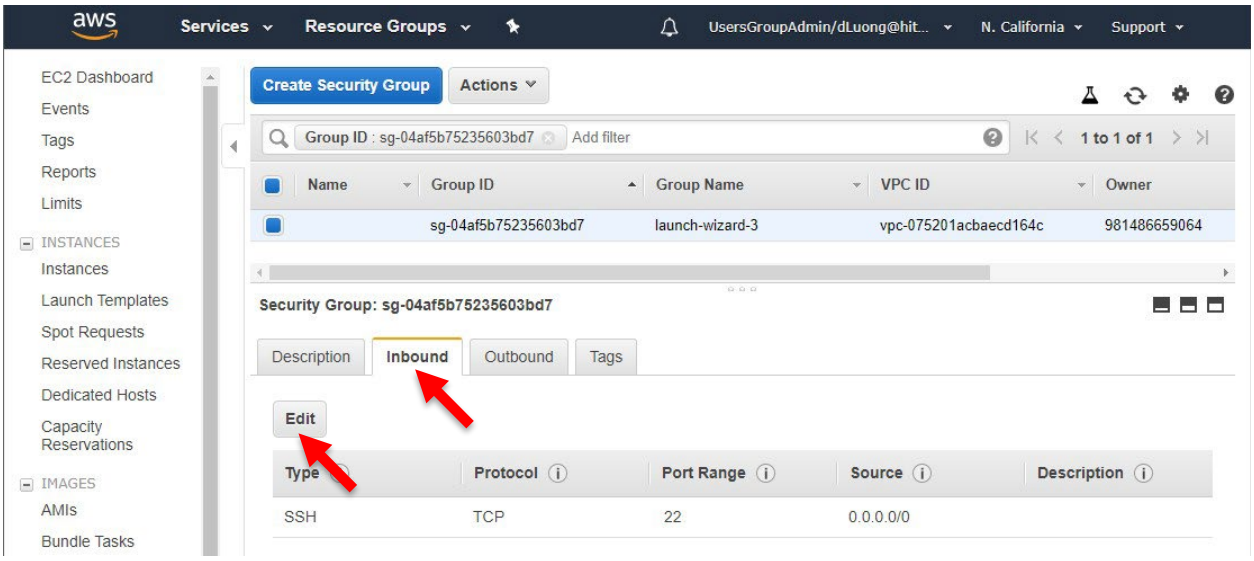
Firewall Exemption

This section provides instructions for creating a firewall exemption so TCP traffic on port 3260 can enter the Virtual Private Cloud (VPC) where the virtual machine resides. Port 3260 is the default port used for iSCSI.

1. On the **Instances** page, select the virtual machine and click the security group attached to the instance.



2. Select the **Inbound** tab and then click **Edit**.



3. Click **Add Rule**.
4. Set the new rule.
    a. For **Type**, select **Custom TCP Rule**.
    b. For **Port Range**, type: **3260**
    c. For **Source**, select **Custom,** and then enter the subnet of the storage system iSCSI ports.
    d. For **Description**, type: **iSCSI traffic**

5. Click **Save**. You do not need to create an outbound rule for TCP 3260.

Targetcli

Installation

This section provides instructions for installing targetcli on the virtual machine.

1. To install targetcli, run the following command:

```
sudo yum install -y targetcli
```

The following shows the output:

```
[ec2-user@ip-172-31-24-72 ~]$ sudo yum install -y targetcli
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package targetcli.noarch 0:2.1.fb46-6.amzn2 will be installed
--> Processing Dependency: python-rtslib >= 2.1.fb41 for package: targetcli-2.1.fb46-
6.amzn2.noarch
--> Processing Dependency: python-ethtool for package: targetcli-2.1.fb46-6.amzn2.noarch
--> Processing Dependency: python-configshell for package: targetcli-2.1.fb46-
6.amzn2.noarch
--> Running transaction check
---> Package python-configshell.noarch 1:1.1.fb23-4.amzn2 will be installed
--> Processing Dependency: python-urwid for package: 1:python-configshell-1.1.fb23-
4.amzn2.noarch
--> Processing Dependency: pyparsing for package: 1:python-configshell-1.1.fb23-
4.amzn2.noarch
---> Package python-ethtool.x86_64 0:0.8-5.amzn2.0.2 will be installed
--> Processing Dependency: libnl.so.1()(64bit) for package: python-ethtool-0.8-
5.amzn2.0.2.x86_64
---> Package python-rtslib.noarch 0:2.1.fb63-12.amzn2 will be installed
--> Processing Dependency: python-pyudev for package: python-rtslib-2.1.fb63-
12.amzn2.noarch
--> Processing Dependency: python-kmod for package: python-rtslib-2.1.fb63-12.amzn2.noarch
--> Running transaction check
---> Package libnl.x86_64 0:1.1.4-3.amzn2.0.2 will be installed
---> Package pyparsing.noarch 0:1.5.6-9.amzn2 will be installed
---> Package python-kmod.x86_64 0:0.9-4.amzn2.0.2 will be installed
---> Package python-pyudev.noarch 0:0.15-9.amzn2 will be installed
---> Package python-urwid.x86_64 0:1.1.1-3.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch       Version                  Repository      Size
================================================================================
Installing:
 targetcli              noarch     2.1.fb46-6.amzn2         amzn2-core      67 k
Installing for dependencies:
 libnl                  x86_64     1.1.4-3.amzn2.0.2        amzn2-core     129 k
 pyparsing              noarch     1.5.6-9.amzn2            amzn2-core      94 k
 python-configshell     noarch     1:1.1.fb23-4.amzn2       amzn2-core      68 k
 python-ethtool         x86_64     0.8-5.amzn2.0.2          amzn2-core      33 k
 python-kmod            x86_64     0.9-4.amzn2.0.2          amzn2-core      74 k
 python-pyudev          noarch     0.15-9.amzn2             amzn2-core      55 k
 python-rtslib          noarch     2.1.fb63-12.amzn2        amzn2-core     100 k
 python-urwid           x86_64     1.1.1-3.amzn2.0.2        amzn2-core     654 k

Transaction Summary
================================================================================
Install  1 Package (+8 Dependent packages)

Total download size: 1.2 M
```

```
Installed size: 5.3 M
Downloading packages:
(1/9): pyparsing-1.5.6-9.amzn2.noarch.rpm                  |  94 kB   00:00
(2/9): libnl-1.1.4-3.amzn2.0.2.x86_64.rpm                  | 129 kB   00:00
(3/9): python-configshell-1.1.fb23-4.amzn2.noarch.rpm      |  68 kB   00:00
(4/9): python-ethtool-0.8-5.amzn2.0.2.x86_64.rpm           |  33 kB   00:00
(5/9): python-pyudev-0.15-9.amzn2.noarch.rpm               |  55 kB   00:00
(6/9): python-kmod-0.9-4.amzn2.0.2.x86_64.rpm              |  74 kB   00:00
(7/9): python-rtslib-2.1.fb63-12.amzn2.noarch.rpm          | 100 kB   00:00
(8/9): python-urwid-1.1.1-3.amzn2.0.2.x86_64.rpm           | 654 kB   00:00
(9/9): targetcli-2.1.fb46-6.amzn2.noarch.rpm               |  67 kB   00:00
--------------------------------------------------------------------------------
Total                                           3.5 MB/s | 1.2 MB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python-pyudev-0.15-9.amzn2.noarch                              1/9
  Installing : pyparsing-1.5.6-9.amzn2.noarch                                 2/9
  Installing : python-kmod-0.9-4.amzn2.0.2.x86_64                             3/9
  Installing : python-rtslib-2.1.fb63-12.amzn2.noarch                         4/9
  Installing : libnl-1.1.4-3.amzn2.0.2.x86_64                                 5/9
  Installing : python-ethtool-0.8-5.amzn2.0.2.x86_64                          6/9
  Installing : python-urwid-1.1.1-3.amzn2.0.2.x86_64                          7/9
  Installing : 1:python-configshell-1.1.fb23-4.amzn2.noarch                   8/9
  Installing : targetcli-2.1.fb46-6.amzn2.noarch                              9/9
  Verifying  : 1:python-configshell-1.1.fb23-4.amzn2.noarch                   1/9
  Verifying  : python-ethtool-0.8-5.amzn2.0.2.x86_64                          2/9
  Verifying  : python-urwid-1.1.1-3.amzn2.0.2.x86_64                          3/9
  Verifying  : python-rtslib-2.1.fb63-12.amzn2.noarch                         4/9
  Verifying  : libnl-1.1.4-3.amzn2.0.2.x86_64                                 5/9
  Verifying  : python-kmod-0.9-4.amzn2.0.2.x86_64                             6/9
  Verifying  : pyparsing-1.5.6-9.amzn2.noarch                                 7/9
  Verifying  : python-pyudev-0.15-9.amzn2.noarch                              8/9
  Verifying  : targetcli-2.1.fb46-6.amzn2.noarch                              9/9

Installed:
  targetcli.noarch 0:2.1.fb46-6.amzn2

Dependency Installed:
  libnl.x86_64 0:1.1.4-3.amzn2.0.2
  pyparsing.noarch 0:1.5.6-9.amzn2
  python-configshell.noarch 1:1.1.fb23-4.amzn2
  python-ethtool.x86_64 0:0.8-5.amzn2.0.2
  python-kmod.x86_64 0:0.9-4.amzn2.0.2
  python-pyudev.noarch 0:0.15-9.amzn2
  python-rtslib.noarch 0:2.1.fb63-12.amzn2
  python-urwid.x86_64 0:1.1.1-3.amzn2.0.2

Complete!
```

2. To install an additional python package that is required by targetcli, run the following command:

   ```
   sudo yum install -y python-dbus
   ```

   The following shows the output:

   ```
   [ec2-user@ip-172-31-24-72 ~]$ sudo yum install -y python-dbus
   Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
   Resolving Dependencies
   --> Running transaction check
   ---> Package dbus-python.x86_64 0:1.1.1-9.amzn2.0.2 will be installed
   --> Processing Dependency: libdbus-glib-1.so.2()(64bit) for package: dbus-python-1.1.1-
   9.amzn2.0.2.x86_64
   --> Running transaction check
   ```

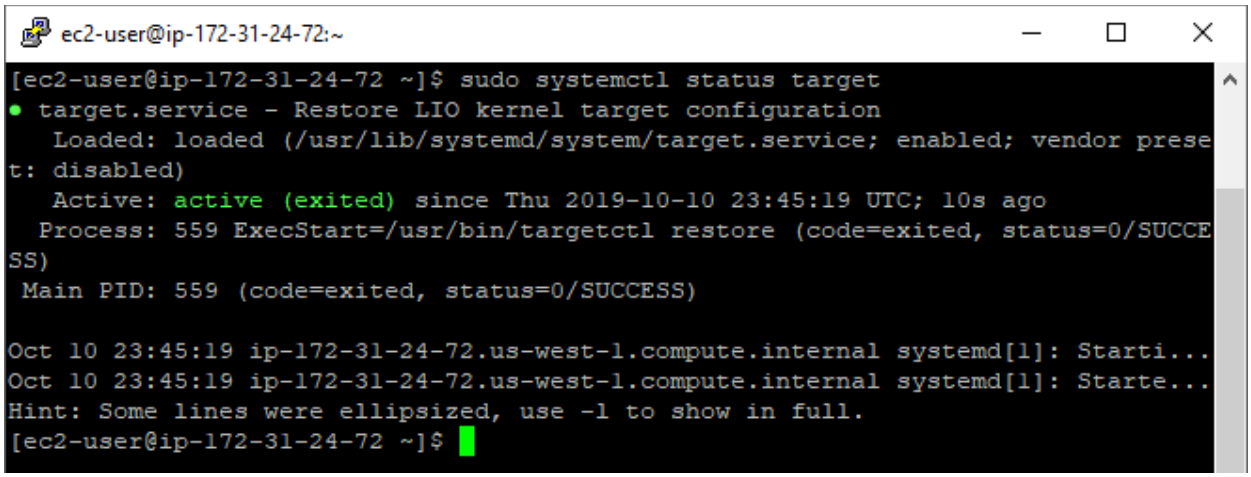
```
---> Package dbus-glib.x86_64 0:0.100-7.2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package           Arch         Version                  Repository        Size
================================================================================
Installing:
 dbus-python       x86_64       1.1.1-9.amzn2.0.2        amzn2-core       206 k
Installing for dependencies:
 dbus-glib         x86_64       0.100-7.2.amzn2          amzn2-core       103 k

Transaction Summary
================================================================================
Install  1 Package (+1 Dependent package)

Total download size: 309 k
Installed size: 1.1 M
Downloading packages:
(1/2): dbus-python-1.1.1-9.amzn2.0.2.x86_64.rpm            | 206 kB   00:00
(2/2): dbus-glib-0.100-7.2.amzn2.x86_64.rpm                | 103 kB   00:00
--------------------------------------------------------------------------------
Total                                           1.8 MB/s | 309 kB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : dbus-glib-0.100-7.2.amzn2.x86_64                             1/2
  Installing : dbus-python-1.1.1-9.amzn2.0.2.x86_64                         2/2
  Verifying  : dbus-python-1.1.1-9.amzn2.0.2.x86_64                         1/2
  Verifying  : dbus-glib-0.100-7.2.amzn2.x86_64                             2/2

Installed:
  dbus-python.x86_64 0:1.1.1-9.amzn2.0.2

Dependency Installed:
  dbus-glib.x86_64 0:0.100-7.2.amzn2

Complete!
```

3. To start the targetcli daemon, run the following command:

```
sudo systemctl start target
```

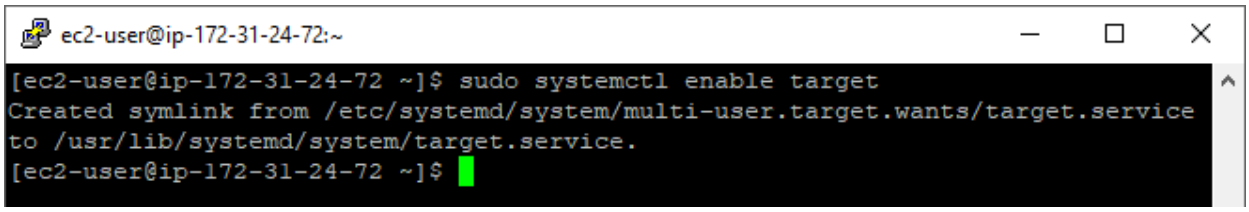
4. To verify that the daemon is running, run the following command:

```
sudo systemctl status target
```



5. To set targetcli to start automatically after it restarts, run the following command:

```
sudo systemctl enable target
```

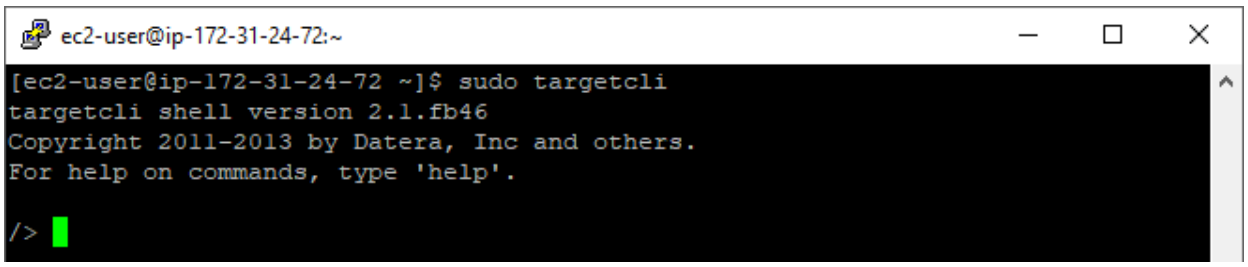


Configuration

This section provides instructions for configuring targetcli to serve three 13 GB volumes over iSCSI.

1. Log in to targetcli by running the following command:

```
sudo targetcli
```



2. Create three 13 GB volumes in the /quorums folder as follows:
    a. Create volume 1 by running the following command:

    ```
    backstores/fileio create volume1 /quorums/volume1 13G
    ```

    The following shows the output:

    ```
    Created fileio volume1 with size 13958643712
    ```

    b. Create volume 2 by running the following command:

    ```
    backstores/fileio create volume2 /quorums/volume2 13G
    ```

    The following shows the output:

    ```
    Created fileio volume2 with size 13958643712
    ```

c. Create volume 3 by running the following command:

```
backstores/fileio create volume3 /quorums/volume3 13G
```

The following shows the output:

```
Created fileio volume3 with size 13958643712
```

3. To create an iSCSI qualified name, run the following commands:

```
cd /iscsi
```

```
create
```

The following shows the output:

```
Created target iqn.2003-01.org.linux-iscsi.ip-172-31-24-72.x8664:sn.a375a63a681c. Created
TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
```

4. Change the listening IP address from all to one specific IP address.

a. Change the directory by running the following command:

```
cd iqn.2003-01.org.linux-iscsi.ip- 172-31-24-72.x8664:sn.a375a63a681c/tpg1/portals/
```

b. Delete listening on all IP addresses by running the following command:

```
delete 0.0.0.0 3260
```

The following shows the output:

```
Deleted network portal 0.0.0.0:3260
```

c. Set up listening on one specific IP address by running the following command:

```
create 172.31.24.72 3260
```

The following shows the output:

```
Using default IP port 3260
Created network portal 172.31.24.72:3260.
```

5. Map the volumes that you created earlier.

a. Change the directory by running the following command:

```
cd /iscsi/iqn.2003- 01.org.linux-iscsi.ip-172-31-24-
72.x8664:sn.a375a63a681c/tpg1/luns
```

b. Map the first LUN by running the following command:

```
create /backstores/fileio/volume1
```

The following shows the output:

```
Created LUN 0.
```

c. Map the second LUN by running the following command:

```
create /backstores/fileio/volume2
```

The following shows the output:

```
Created LUN 1.
```

d. Map the third LUN by running the following command:

```
create /backstores/fileio/volume3/
```

The following shows the output:

```
Created LUN 2.
```

6. Mask the initiator IQNs of the storage systems to allow access to the LUNs. This adds four IQNs: two ports from each storage system.

a. Change the directory by running the following command:

```
cd /iscsi/iqn.2003-01.org.linux-iscsi.ip-172-31-24-
72.x8664:sn.a375a63a681c/tpg1/acls
```

b. Create the first IQN by running the following command (your IQN will be different):

```
create iqn.1994-04.jp.co.hitachi:rsd.h8m.i.123ac6.1a
```

The following shows an example of the output:

```
Created Node ACL for iqn.1994-04.jp.co.hitachi:rsd.h8m.i.123ac6.1a
Created mapped LUN 2.
Created mapped LUN 1.
Created mapped LUN 0.
```

c. Create the second IQN by running the following command (your IQN will be different):

```
create iqn.1994-04.jp.co.hitachi:rsd.h8m.i.123ac6.2a
```

The following shows an example of the output:

```
Created Node ACL for iqn.1994-04.jp.co.hitachi:rsd.h8m.i.123ac6.2a
Created mapped LUN 2.
Created mapped LUN 1.
Created mapped LUN 0.
```

d. Create the third IQN by running the following command (your IQN will be different):

```
create iqn.1994-04.jp.co.hitachi:rsd.h8m.i.12afcd.1a
```

The following shows an example of the output:

```
Created Node ACL for iqn.1994-04.jp.co.hitachi:rsd.h8m.i.12afcd.1a
Created mapped LUN 2.
Created mapped LUN 1.
Created mapped LUN 0.
```

e. Create the fourth IQN by running the following command (your IQN will be different):

```
create iqn.1994-04.jp.co.hitachi:rsd.h8m.i.12afcd.2a
```

The following shows an example of the output:

```
Created Node ACL for iqn.1994-04.jp.co.hitachi:rsd.h8m.i.12afcd.2a
Created mapped LUN 2.
Created mapped LUN 1.
Created mapped LUN 0.
```

7. To save the changes, run the following command:

```
cd /

saveconfig
```

The following shows the output:

```
Last 10 configs saved in /etc/target/backup/. Configuration saved to
/etc/target/saveconfig.json
```

8. To view the completed configuration, run the following command:

```
ls
```

The following is an example of the output (your URL for the LUNs will be different):

```
o- / ......................................................................... [...]
  o- backstores .............................................................. [...]
  | o- block ..................................... [Storage Objects: 0]
  | o- fileio .................................... [Storage Objects: 3]
  | | o- volume1 ............. [/quorums/volume1 (13.0GiB) write-back activated]
  | | | o- alua ..................................... [ALUA Groups: 1]
  | | |   o- default_tg_pt_gp ................... [ALUA state: Active/optimized]
  | | o- volume2 ............. [/quorums/volume2 (13.0GiB) write-back activated]
```

```
  | | | o- alua ............................................ [ALUA Groups: 1]
  | | |   o- default_tg_pt_gp ................... [ALUA state: Active/optimized]
  | | o- volume3 ............. [/quorums/volume3 (13.0GiB) write-back activated]
  | |   o- alua ............................................ [ALUA Groups: 1]
  | |     o- default_tg_pt_gp ................... [ALUA state: Active/optimized]
  | o- pscsi ......................................... [Storage Objects: 0]
  | o- ramdisk ....................................... [Storage Objects: 0]
  o- iscsi ....................................................... [Targets: 1]
  | o- iqn.2003-01.org.linux-iscsi.ip-172-31-24-72.x8664:sn.a375a63a681c  [TPGs: 1]
  |   o- tpg1 ........................................... [no-gen-acls, no-auth]
  |     o- acls ...................................................... [ACLs: 4]
  |     | o- iqn.1994-04.jp.co.hitachi:rsd.h8m.i.123ac6.1a .... [Mapped LUNs: 3]
  |     | | o- mapped_lun0 ........................... [lun0 fileio/volume1 (rw)]
  |     | | o- mapped_lun1 ........................... [lun1 fileio/volume2 (rw)]
  |     | | o- mapped_lun2 ........................... [lun2 fileio/volume3 (rw)]
  |     | o- iqn.1994-04.jp.co.hitachi:rsd.h8m.i.123ac6.2a .... [Mapped LUNs: 3]
  |     | | o- mapped_lun0 ........................... [lun0 fileio/volume1 (rw)]
  |     | | o- mapped_lun1 ........................... [lun1 fileio/volume2 (rw)]
  |     | | o- mapped_lun2 ........................... [lun2 fileio/volume3 (rw)]
  |     | o- iqn.1994-04.jp.co.hitachi:rsd.h8m.i.12afcd.1a .... [Mapped LUNs: 3]
  |     | | o- mapped_lun0 ........................... [lun0 fileio/volume1 (rw)]
  |     | | o- mapped_lun1 ........................... [lun1 fileio/volume2 (rw)]
  |     | | o- mapped_lun2 ........................... [lun2 fileio/volume3 (rw)]
  |     | o- iqn.1994-04.jp.co.hitachi:rsd.h8m.i.12afcd.2a .... [Mapped LUNs: 3]
  |     |   o- mapped_lun0 ........................... [lun0 fileio/volume1 (rw)]
  |     |   o- mapped_lun1 ........................... [lun1 fileio/volume2 (rw)]
  |     |   o- mapped_lun2 ........................... [lun2 fileio/volume3 (rw)]
  |     o- luns ...................................................... [LUNs: 3]
  |     | o- lun0 ....... [fileio/volume1 (/quorums/volume1) (default_tg_pt_gp)]
  |     | o- lun1 ....... [fileio/volume2 (/quorums/volume2) (default_tg_pt_gp)]
  |     | o- lun2 ....... [fileio/volume3 (/quorums/volume3) (default_tg_pt_gp)]
  |     o- portals ................................................ [Portals: 1]
  |       o- 172.31.24.72:3260 ........................................... [OK]
  o- loopback ..................................................... [Targets: 0]
/>
```

After successfully setting up the virtual machine as an iSCSI target, you must secure the solution by removing the public IP.

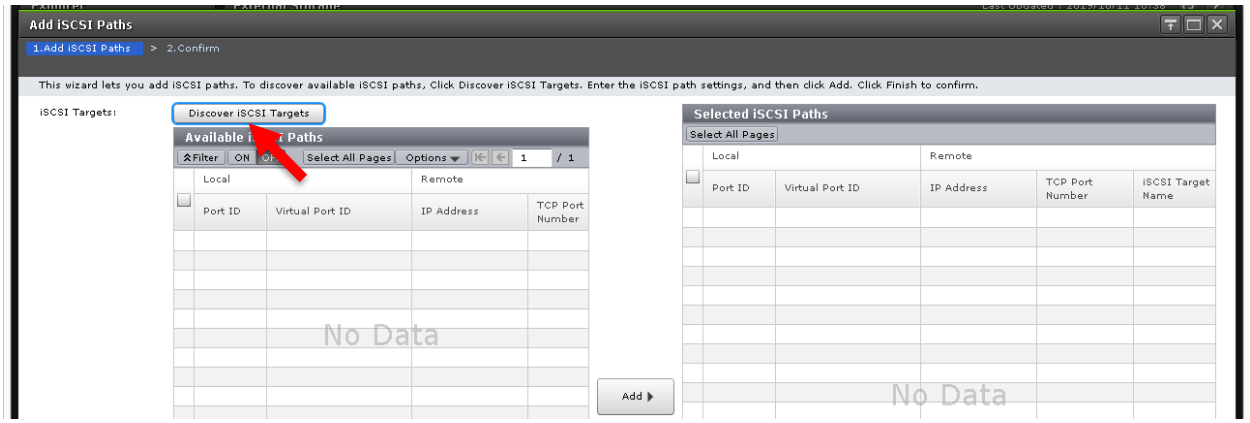
Global-Active Device Quorums

This section describes how to discover the volumes from the iSCSI target virtual machine and turn them into GAD quorums. The procedure is the same as it is to virtualize a physical Fibre Channel or iSCSI storage system.

Create iSCSI Paths

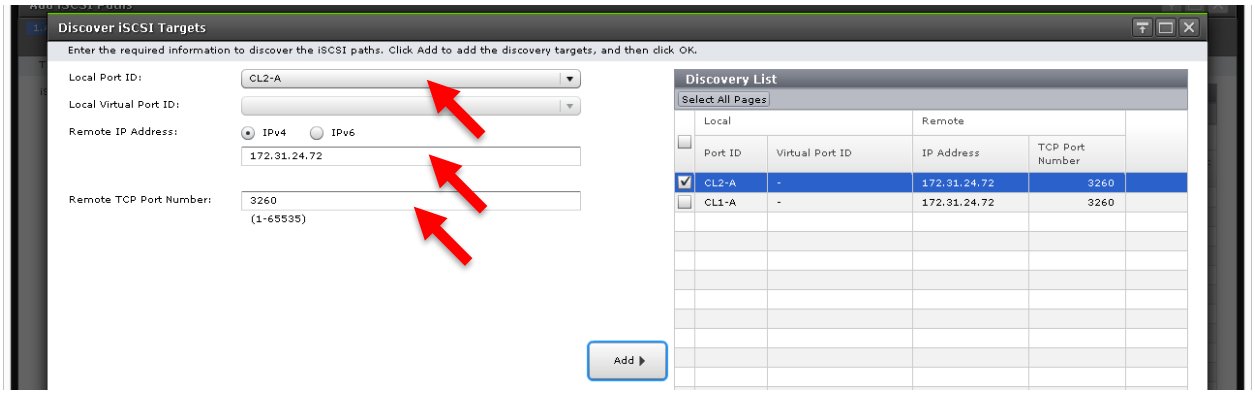
1. Log in to Storage Navigator.
2. On the left side, select **External Storage,** and then select the **iSCSI Paths** tab.

3. Click **Add iSCSI Paths**.
4. Click **Discover iSCSI Targets**.



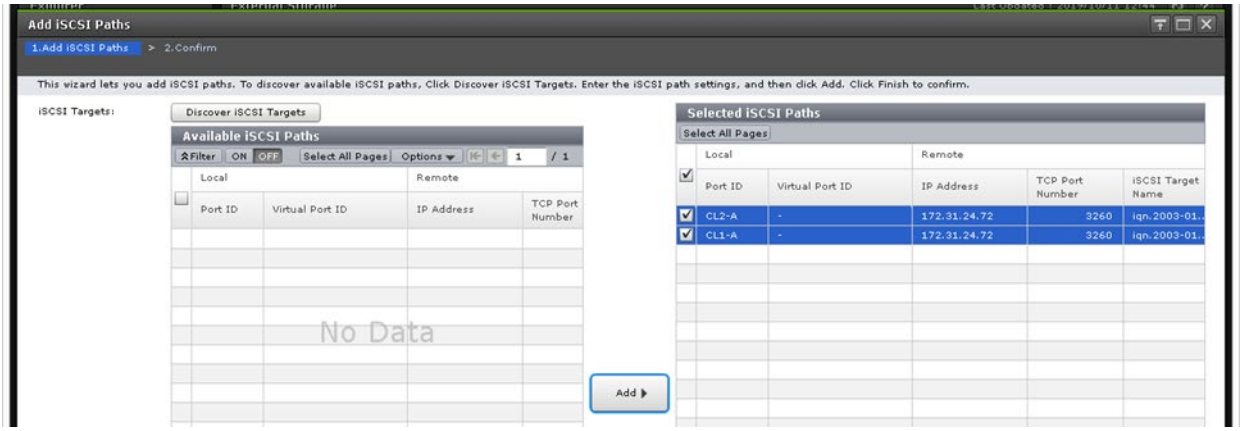
5. Add both iSCSI paths. Repeat this step for both paths.
    a. Select the storage port from the **Local Port ID** list.
    b. Enter the private IP address of the AWS virtual machine.
    c. For **Remote TCP Port Number,** enter **3260**.

d. Click **Add**.



6. After creating both iSCSI paths, click **OK**.
7. On the **Add iSCSI Paths** window, set the following:
   a. From the **Authentication Method** dropdown list, click **None**.
   b. For **Mutual CHAP**, click **Disable**.

8. Click **Add** and then click **Finish**.

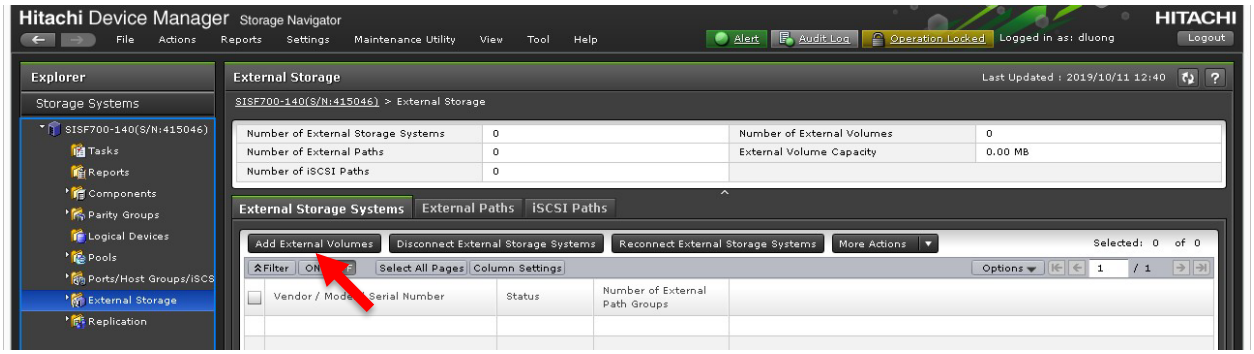


The following shows the created paths:

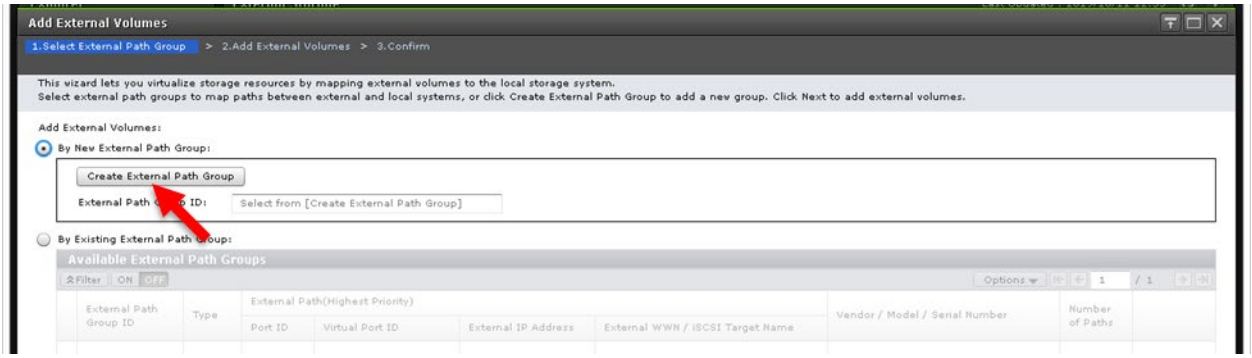
Discover External Volumes

This section describes how to discover the volumes from the iSCSI virtual machine and virtualize them.

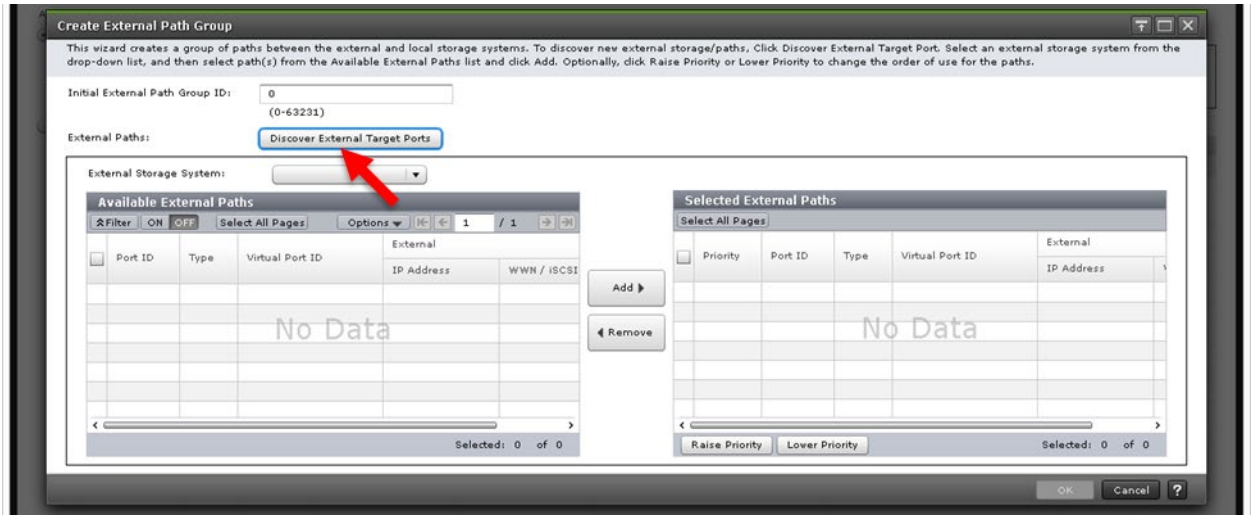
1. Select the **External Storage Systems** tab and then click **Add External Volumes**.



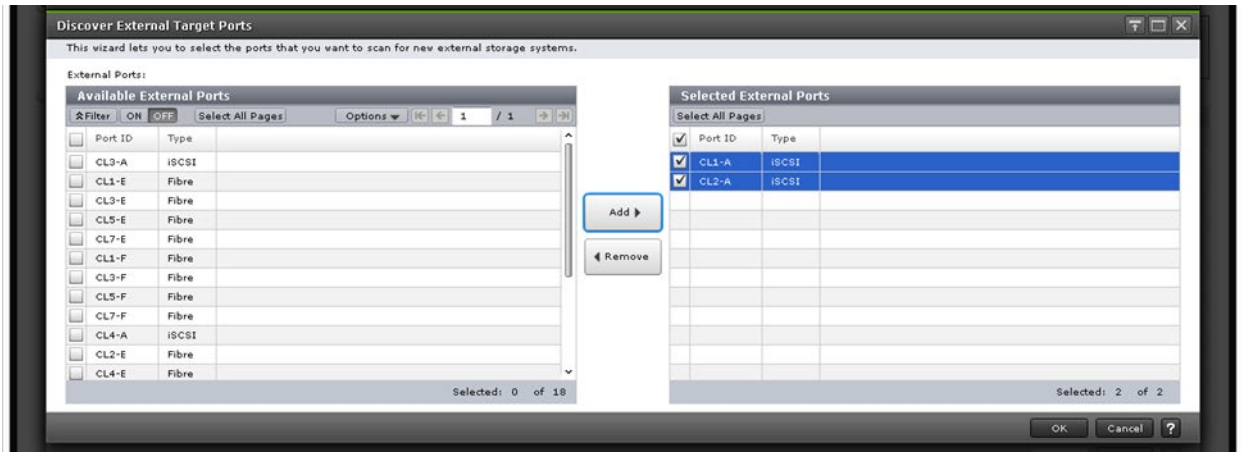
2. Click **Create External Path Group**.



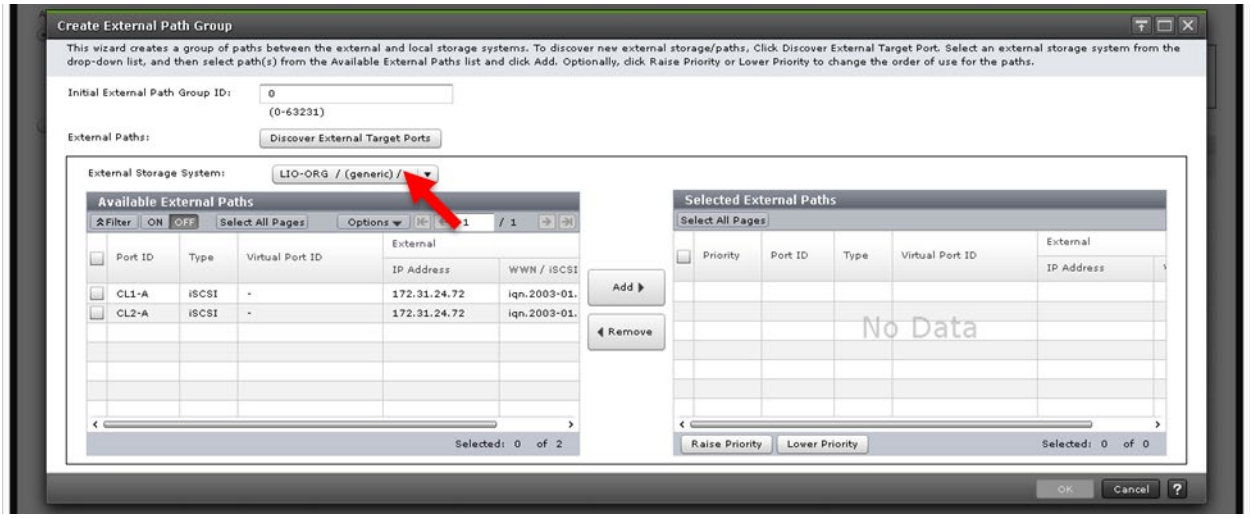
3. Click **Discover External Target Ports**.



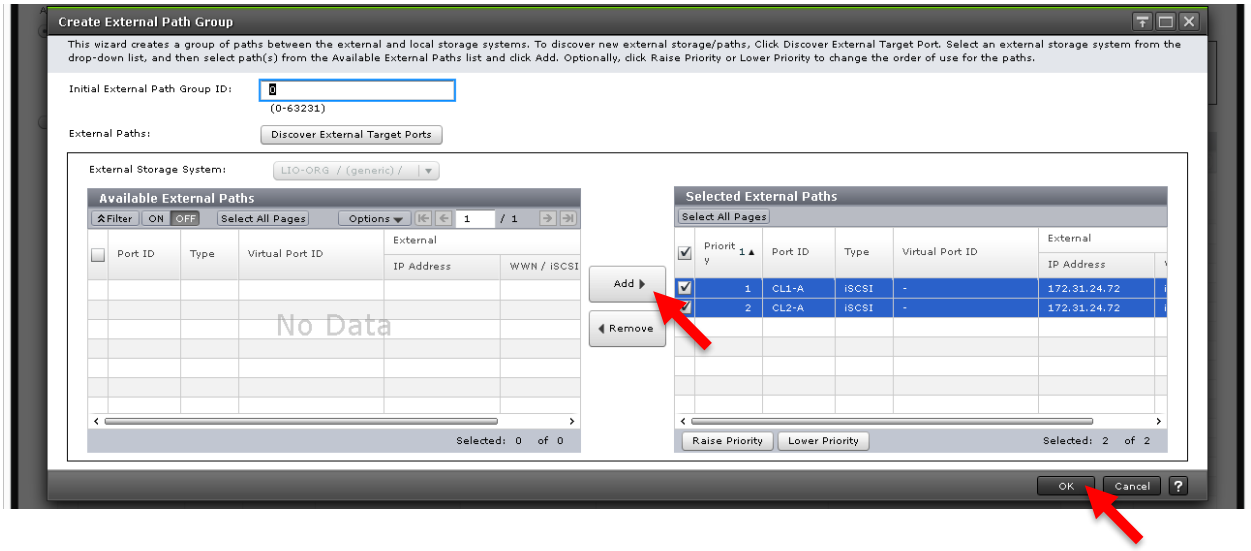
4. Select the iSCSI ports, click **Add**, and then click **OK**.



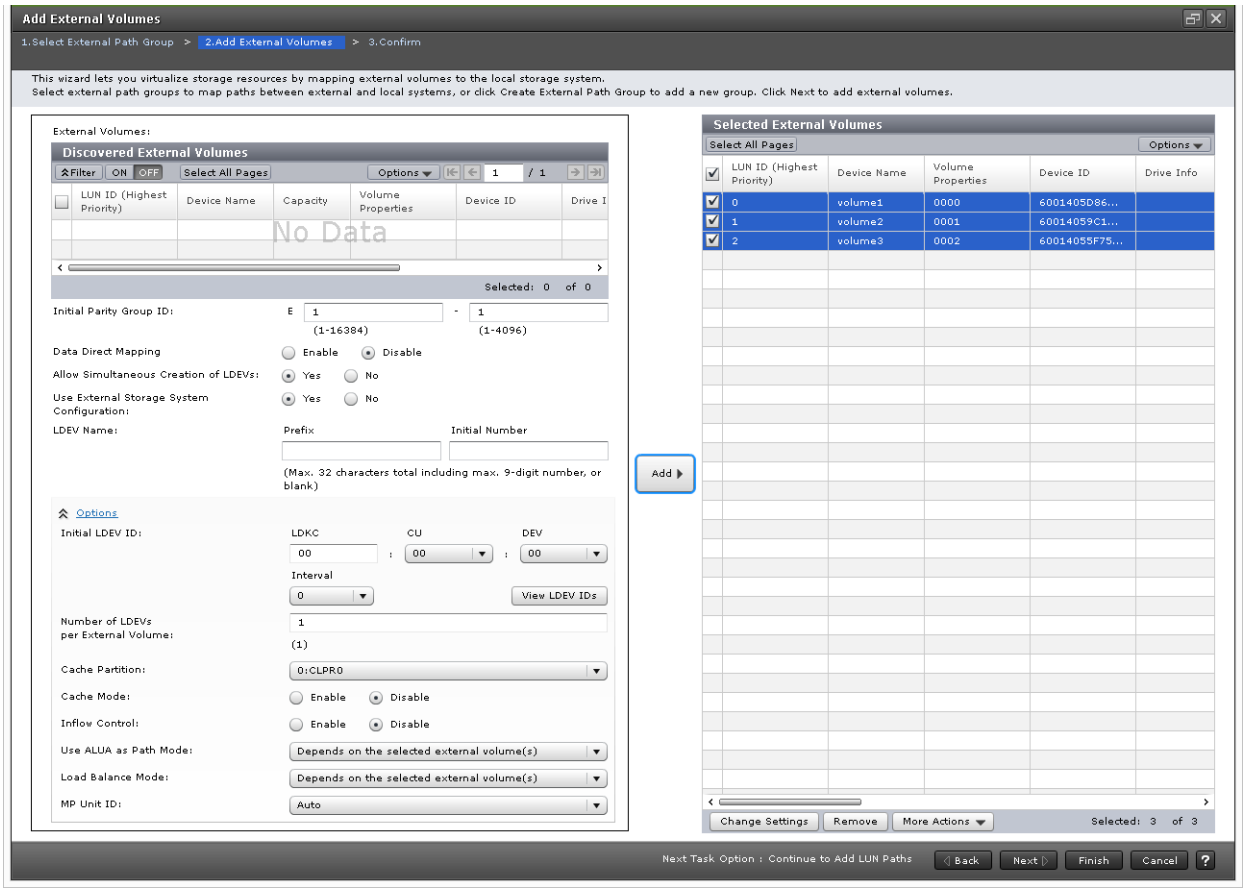
If the discovery is successful, the virtual machine shows up as LIO-ORG.



5. Select the discovered external paths, click **Add**, and then click **OK**.

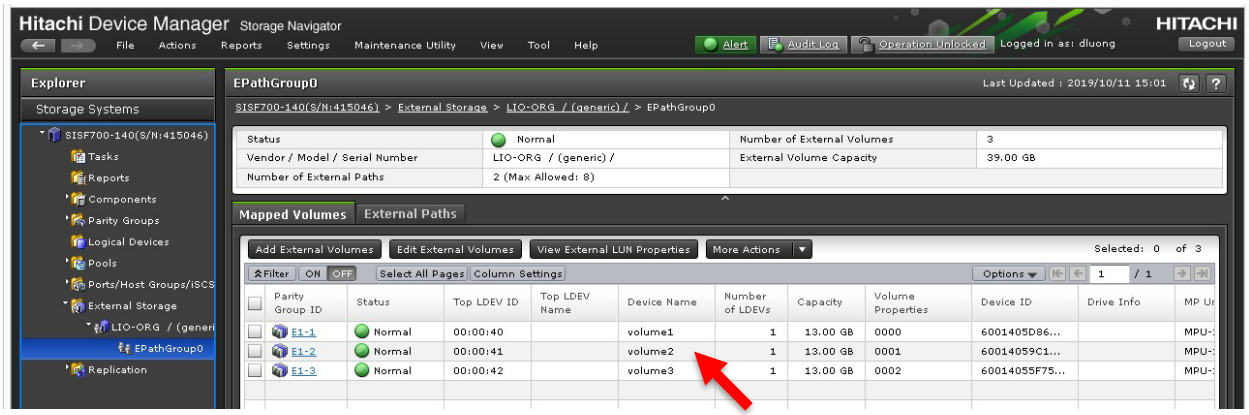


6. On the Add External Volumes window, click **Next**.
7. Select the discovered volumes and click **Add**.



8. Click **Finish** and then click **Apply**.

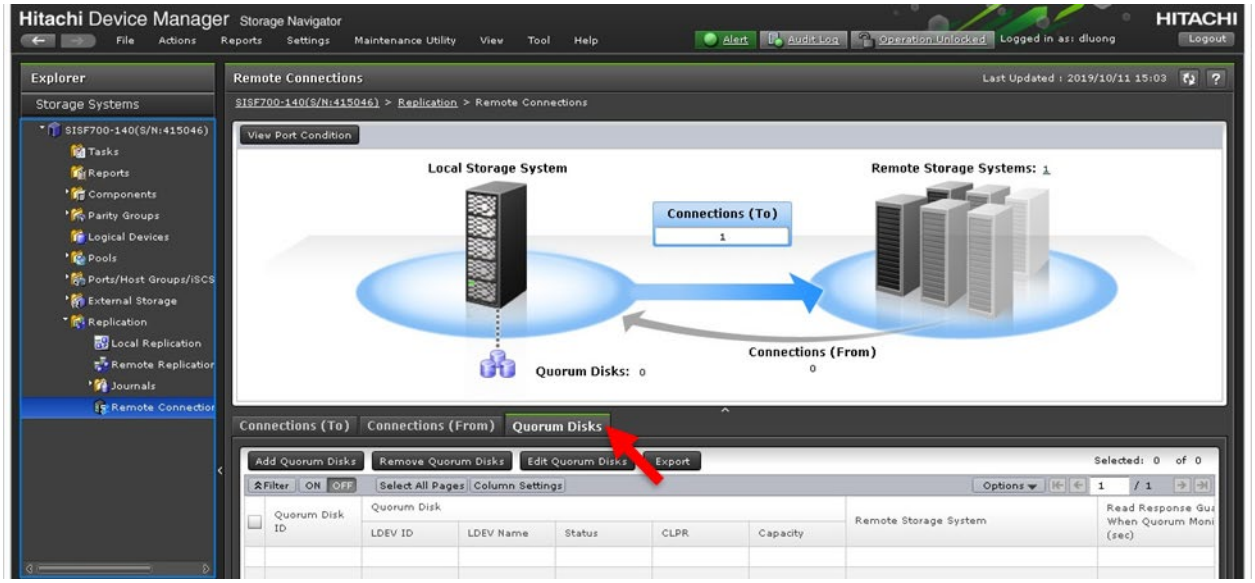
   The following shows the external volumes after they have been virtualized:



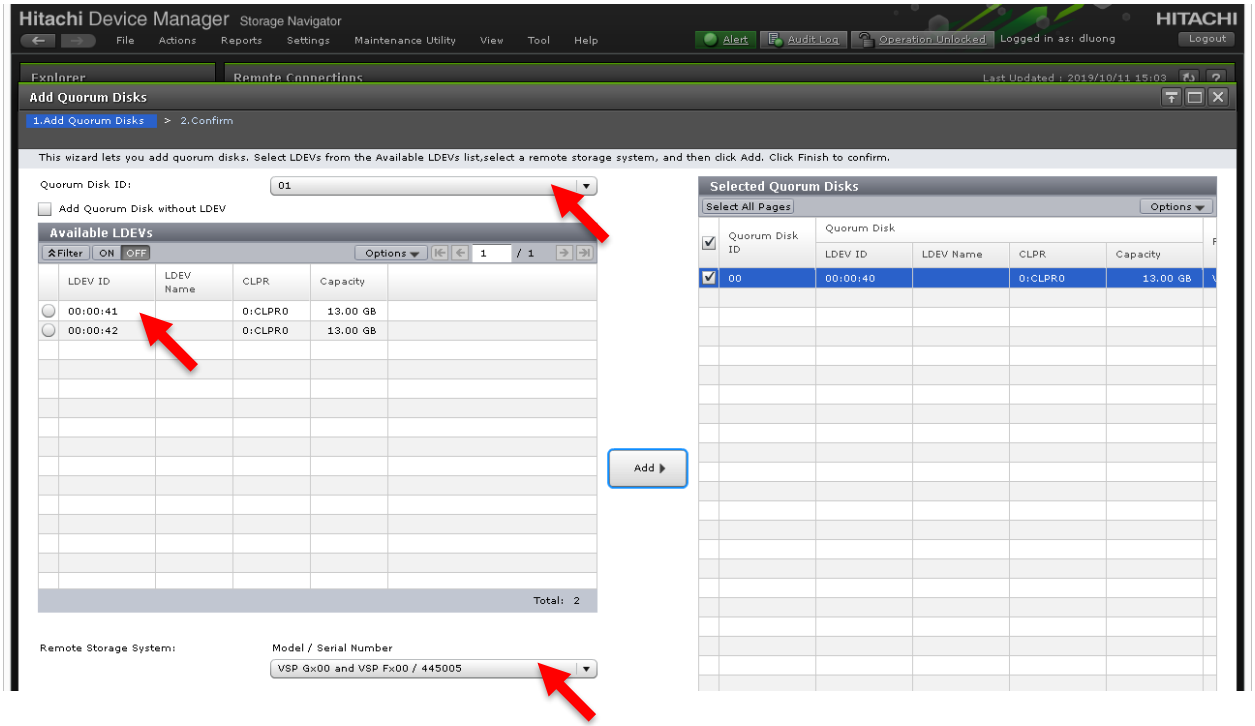
Define Global-Active Device Quorums

This section describes how to turn the external volumes into GAD quorums. The procedure is the same as it is to a virtualized physical Fibre Channel or iSCSI storage system.

1. Select **Replication** > **Remote Connections**, and then select the **Quorum Disks** tab.



2. Click **Add Quorum Disks**.
3. In the Add Quorum Disks screen, choose the appropriate option from the **Quorum Disk ID** and the **Remote Storage System** list.
4. From the Available LDEVs table, select the external volume you want to use and click **Add**.



5. Click **Finish** and then click **Apply**.

The following shows the quorum after it has been created: