QSAN Official Document

How to implement iSCSI multipath on Linux OS

XCubeSAN and FAS Series White Paper

August 2022

PREFACE

Information, Tip and Caution

This manual uses the following symbols to draw attention to important safety and operational information.

INFORMATION:

INFORMATION provides useful knowledge, definition, or terminology for reference.

TIP:

TIP provides helpful suggestions for performing tasks more effectively.

CAUTION:

CAUTION indicates that failure to take a specified action could result in damage to the system.

TABLE OF CONTENT

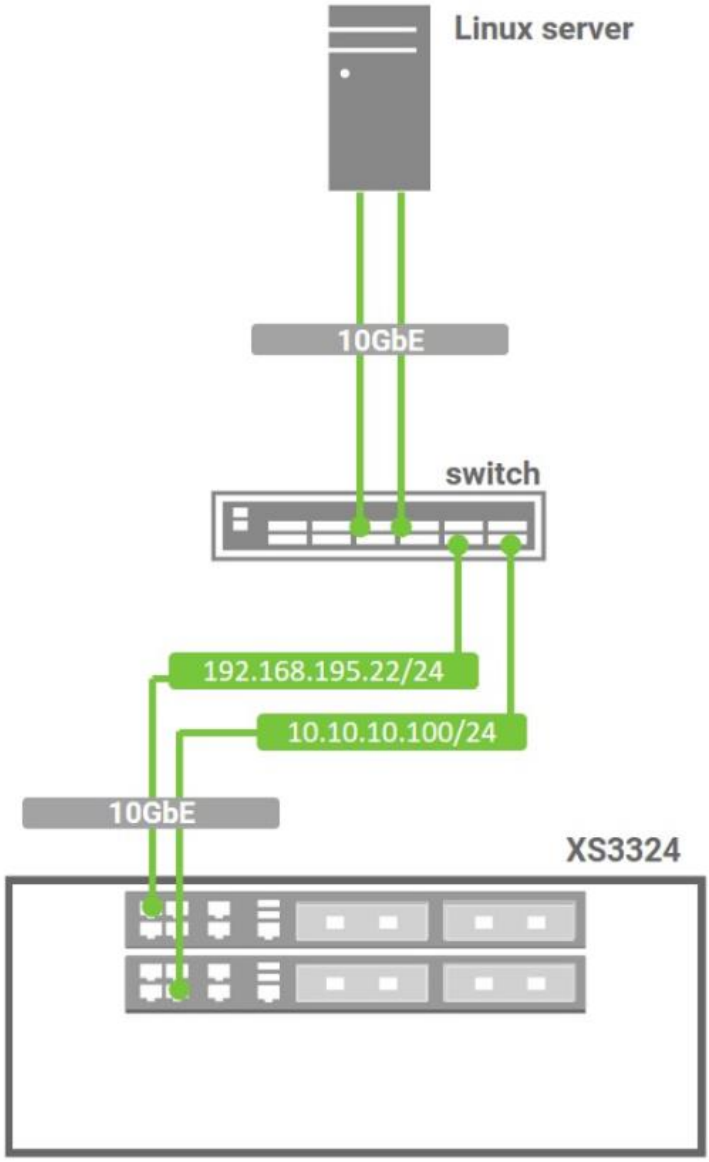
AUDIENCE

In this document, it describes how to connect an iSCSI target in Linux OS via software iSCSI initiator and use the device-mapper-multipath package to create a multipath device which is presented by QSAN XS3324 via iSCSI. All Qsan iSCSI dual controller models can be used within this configuration.

Environment

Host OS	Red Hat Enterprise Linux
Storage	XS3324D
Firmware	V2.2.0
RAM	8GB
iSCSI data port	10.10.10.100; 192.168.195.22
RPM packages	iscsi-initiator-utils-6.2.0.873-10.el6.x86_64 device-mapper-1.02.79-8.el6.x86_64 device-mapper-multipath-0.4.9-72.el6.x86_64

Diagram



.

INSTALLATION

Before configuring the iSCSI multipath, you have to install the following rpm packages and source files (.tar.gz), so that the iSCSI service could run smoothly and without any compatible issues.

Here is the order to install the packages we need:

- iscsi-initiator-utils-6.2.0.873-10.el6.x86_64.rpm
- device-mapper-1.02.79-8.el6.x86_64.rpm
- device-mapper-multipath-0.4.9-72.el6.x86_64.rpm

All the necessary rpm packages can be found in the Linux OS DVD, install them as follows:

```
# rpm -ivh /media/"RHEL x86_64 Disc 1"/Packages/iscsi-initiator-utils-6.2.0.873-10.el6.x86_64.rpm
# rpm -ivh /media/"RHEL x86_64 Disc 1"/Packages/device-mapper-1.02.79-8.el6.x86_64.rpm
# rpm -ivh /media/"RHEL x86_64 Disc 1"/Packages/device-mapper-multipath-0.4.9-72.el6.x86_64.rpm
```

CONFIGURATION

Usage of iSCSI initiator

The iSCSI initiator name can be specified in the configuration file /etc/iscsi/initiatorname.iscsi.

```
# vi /etc/iscsi/initiatorname.iscsi
InitiatorName = Your_initiator_name
```

Edit the configuration file of iSCSI initiator in /etc/iscsi/iscsid.conf, the iscsi session timeout value has to be changed to a proper value. The default value is 120 seconds, but it is too long to keep the I/O wait before the path is judged as fail and it may cause the I/O failure. Please set a shorter and proper timeout value in this configuration file.

```
# vi /etc/iscsi/iscsid.conf
node.session.timeo.replacement_timeout = 30
 (Please set a proper timeout value)
```

In /etc/iscsi/iscsid.conf, it also provides others settings, such as below showing:

```
# vi /etc/iscsi/iscsid.conf
node.startup = Automatic
(Set auto-login when discover target)
node.session.auth.authmethod = CHAP
(Enable CHAP auth)
node.session.auth.username = username
(Set CHAP username)
node.session.auth.password = password
(Set CHAP password)
```

Please restart the iSCSI service to make these changes work.

```
# service iscsi restart
```

Login the iSCSI target

The rpm package iscsi-initiator-utils provides a command line tool called iscsiadm. It can manage the connections to iSCSI target. The iscsiadm tool has three operational modes - discovery, node, and session. The following will introduce these modes.

1. Discovery the all port and target name by # iscsiadm -m discovery.

   Operational mode -discovery is used to discover the target, the usage is

   # iscsiadm -m discovery -t st -p target_ip

```
# iscsiadm -m discovery -t st -p 10.10.10.100
192.168.1.1:3260,0 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1
192.168.2.1:3260,0 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1
10.10.10.100:3260,1 iqn.2004-08.com.qsan:xx3324-000d2340:dev0.ctr1
192.168.4.1:3260,1 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1

# iscsiadm -m discovery -t st -p 192.168.195.22
192.168.5.1:3260,0 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2
192.168.6.1:3260,0 iqn.2004-08.com.qsan:xs3324-000d23400:dev0.ctr2
192.168.195.22:3260,1 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2
192.168.8.1:3260,1 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2
```

2. Users can login and logout by # iscsiadm -m node with the ip and target name.

   Operational mode -node is used to login/logout, the usage is

   # iscsiadm -m node -T target_iqn -p target_ip -l

   # iscsiadm -m node -T target_iqn -p target_ip -u

```
# iscsiadm -m node -T iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1 -p
10.10.10.100 -l
(login 10.10.10.100)
# iscsiadm -m node -T iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2 -p
192.168.195.22 -l
(login 192.68.195.22)
# iscsiadm -m node -T iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1 -p
10.10.10.100 -u
(logout 10.10.10.100)
# iscsiadm -m node -T iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2 -p
192.168.195.22 –u
(logout 192.168.195.22)
```

3. Query the list of nodes, the usage is

   # iscsiadm -m node.

```
# iscsiadm -m node
192.168.1.1:3260,0 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1
192.168.2.1:3260,0 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1
10.10.10.100:3260,1 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1
192.168.4.1:3260,1 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1
192.168.5.1:3260,0 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2
192.168.6.1:3260,0 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2
192.168.195.22:3260,1 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2
192.168.8.1:3260,1 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2
```

4. If users want to clear the node list, the usage is

   # iscsiadm -m node -0 delete

5. This command will list the connected iSCSI session, it can be expressed as

   # iscsiadm -m session

```
# iscsiadm -m session
tcp: [3] 10.10.10.100:3260,1 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1
tcp: [4] 192.168.195.22:3260,1 iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr2
```

6. In session mode, the iSCSI session can be logout, the usage is

   # iscsiadm -m session -r session_id –u

```
# iscsiadm -m session -r 3 -u
Logging out of session [sid: 3, target: iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1,
portal: 10.10.10.100]
Logout of [sid: 3 target: iqn.2004-08.com.qsan:xs3324-000d2340:dev0.ctr1, portal:
10.10.10.100,3260]: successful
```

7. To log out all sessions, the usage is

   # iscsiadm -m session –u

How to enable and DM-Multipath and create multipath device

The procedures of setup a multipath DM-Multipath are as the following.

1. To enable mpathcof first, and then start the multipathd service

```
# mpathconf -h
 usage: /sbin/mpathconf <command>
Commands:
Enable: --enable
Disable: --disable
Set user_friendly_names (Default n): --user_friendly_names <y|n>
Set find_multipaths (Default n): --find_multipaths <y|n>
Load the dm-multipath modules on enable (Default y): --with_module <y|n>
start/stop/reload multipathd (Default n): --with_multipathd <y|n>
chkconfig on/off multipathd (Default y): --with_chkconfig <y|n>

# mpathconf --enable
(It will create multipath.conf file as the configuration of multipath)
# service multipathd start
(To enable multipath)
```

2. Flush the existing device-maps and then create the dev-maps for the multipath device

```
# multipath -F
(Clear existing device-maps)
# multipath
(Create devicemaps)
 create: mpathb (3203300137890ad00) undef Qsan,xs3324
[size=500g feature='0' hwhandler='0' wp=undef
|-+- policy='round-robin 0' prio=1 status=undef
| '- 12:0:0:0 sdb 8:16 undef ready running
'-+- policy='round-robin 0' prio=1 status=undef
'- 13:0:0:0 sdc 8:32 undef ready running
```

TIP:

The device A as follow means failover. And another one means round-robin.

A. |-+- policy='round-robin 0' prio=1 status=undef
| '- 12:0:0:0 sdb 8:16 undef ready running
'-+- policy='round-robin 0' prio=1 status=undef
'- 13:0:0:0 sdc 8:32 undef ready running
B. |-+- policy='round-robin 0' prio=1 status=active
'- 12:0:0:0 sdb 8:16 active ready running
- 13:0:0:0 sdc 8:32 active ready running

How to exclude local disks (Optional)

There are two ways that the local disks can be excluded when generating multipath devices.

1. Determine which WWN of local disks will be ignored. In this example, using the command "multipath" can find out the WWN of local disk /dev/sda

```
# multipath –F
(Clear all multipath device maps)
# multipath
(Create multipath)
create: mpatha(1ATA ST31000528AS 9V)undef ATA,ST31000528A
[size=932G feature='0' hwhandler='0' wp=undef
'-+- policy='round-robin 0' prio=1 status=undef
'- 2:0:0:0 sda8:0 undef ready running
create: mpathb (3203300137890ad00) undef Qsan,xs3324
[size=500g feature='0' hwhandler='0' wp=undef
|-+- policy='round-robin 0' prio=1 status=undef
| '- 12:0:0:0 sdb 8:16 undef ready running
'-+- policy='round-robin 0' prio=1 status=undef
'- 13:0:0:0 sdc 8:32 undef ready running
```

2. The WWN of local disk /dev/sda is in the parenthesis followed by the word "mpatha".Edit /etc/multipath.conf, and insert the WWN of local disk into the blacklist.

```
# vi /etc/multipath.conf
blacklist {
 wwid 1ATA ST31000528AS 9V
}
# service multipthd restart
```

3. User can also change the find_multipths to block the local disk

```
# multipath -find_multipaths y

or

# vi /etc/multipath.conf
defaults{
find_multipaths yes
}
```

TIP:
If you change the value of multipath.conf, you must restart multipath to take effect.

How to change the alias of iSCSI device (Optional)

The alias name will help iSCSI device to be identified easily. Find the UUID of iSCSI device in Red(mpathb) below:

```
# multipath -ll
mpathb (32033001378901d00) dm-3 Qsan,xs3324
[size=500g feature='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| '- 12:0:0:0 sdb 8:16 active ready running
'-+- policy='round-robin 0' prio=1 status=enabled
'- 13:0:0:0 sdc 8:32 active ready running
```

1. Edit the /etc/multipath.conf again, and change the alias as "qsan"

```
# vi /etc/multipath.conf
multipaths {
    multipath {
        wwid 32033001378901d00
        alias qsan
        path_grouping_policy multibus
#       path_checker direction
        (This line may cause multipath be invalid in different device)
        path_selector "round-robin 0"
        failback manual
        rr_weight priorities
        no_path_retry 5
}
```

2. Confirm that the persistent name to iSCSI device has been modified.

```
# multipath -ll
qsan (32033001378901d00) dm-3 Qsan,xs3324
[size=500g feature='1 queue_if_no_path' hwhandler='0' wp=ro
|-+- policy='round-robin 0' prio=1 status=active
'- 12:0:0:0 sdb 8:16 active ready running
'- 13:0:0:0 sdc 8:32 active ready running
# ls -l /dev/mapper
total 0
crw-rw---- 1 root root 10, 58 jul 28 18:34 control
lrwxrwxrwx 1 root root 7 jul 28 18:34 qsan -> ../dm-3
lrwxrwxrwx 1 root root 7 jul 28 18:34 VolGroup00-lv_home -> ../dm-2
lrwxrwxrwx 1 root root 7 jul 28 18:34 VolGroup00-lv_root -> ../dm-0
```

TIP:
Usually it uses the command multipath to manage the multipath devices.
Here is the parameter manual.

Multipath	Without parameters, create the devmaps for the multipath devices.
-h	Print this usage text.
-l	Show multipath topology. (sysfs and DM info)
-ll	Show multipath topology. (maximum info)
-f	Flush a multipath device map.
-F	Flush all multipath device maps.
-c	Check if advice should be a path in a multipath device.
-q	Allow queue_if_no_path when multipathd is not running.
-d	Dry run, do not create or update devmaps
-r	Force devmap reload.
-p	Policy failover|multibus|group_by_serial|group_by_prio.
-b fil	fil Bindings file location.
-p pol	pol Force all maps to specified path grouping policy:
	failover 1 path per priority group
	multibus all paths in 1 priority group
	group_by_serial 1 priority group per serial
	group_by_prio 1 priority group per priority level
	group_by_node_name 1 priority group per target node

-v lvl	Verbosity level: 0 no output 1 print created devmap names only 2 default verbosity 3 print debug information
Dev	Action limited to: Multipath named 'dev' (ex: mpath0) or Multipath whose wwidis 'dev' (ex:60051..) Multipath including the path named 'dev' (ex: /dev/sda) Multipath including the path with maj:min 'dev' (ex:8:0)

How to set the priority configuration to make the LUN accessed by pool owner controller (Optional)

Due to our dual controller design, the pool would have the controller owner concept. Access through pool controller owner would have the best performance. You can refer the below configuration to add red part to achieve this.

```
multipath {
   wwid 320370013780d2000
   alias Phil_test
   path_grouping_policy "group_by_prio"
   prio "alua"
   prio_args "exclusive_pref_bit"
}
```

PRACTICE

Here is an example of how to create a multipath device and change the MPIO policy between failover and round-robin.

```
# multipath -F
(Clear existing device-maps)
# multipath
(Create devicemaps)
creat: qsan (32033001378901d00) undef Qsan,xs3324
[size=500g feature='0' hwhandler='0' wp=undef
|-+- policy='round-robin 0' prio=1 status=undef
'- 12:0:0:0 sdb 8:16 undef ready running
'- 13:0:0:0 sdc 8:32 undef ready running

# fdisk -l
Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000da7c4

Device Boot Start End Blocks Id System
/dev/sda1 * 1 64 512000 83 Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2 64 121602 976248832 8e Linux LVM

Disk /dev/sdb: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xa3872e75

Device Boot Start End Blocks Id System
/dev/sdb1 1 13054 104854528 7 HPFS/NTFS

Disk /dev/sdc: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0xa3872e75

Device Boot Start End Blocks Id System
/dev/sdc1 1 13054 104854528 7 HPFS/NTFS

Disk /dev/mapper/qsan: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xa3872e75

# multipath -l
(List the multipath topology, default MPIO policy is multibus)
qsan (32033001378901d00) dm-3 Qsan,xs3324
[size=500g feature='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=0 status=active
'- 12:0:0:0 sdb 8:16 active ready running
'- 13:0:0:0 sdc 8:32 active ready running
```

You can change policy (ex:failover, multibus) by editing multipath.conf

```
# vi /etc/multipath.conf
multipath {
        wwid 32033001378901d00
        alias qsan
        path_grouping_policy multibus
        (You can change failover by this line)
#       path_checker directio
        path_selector "round-robin 0"
        failback manual
        rr_weight priorities
        no_path_retry 5
}
```

Now, the abstract device /dev/mapper/qsan which groups /dev/sdb and /dev/sdc are ready to be used.

Mount the file system on iSCSI device at boot time (Optional)

1. Format the iSCSI device as EXT4 file system.

```
# mkfs.ext4 /dev/mapper/qsan
```

2. In order to mount a file system that exits on an iSCSI device connected through the open-iSCSI software initiator, you need to add a line to the /etc/fstab file.

```
# vi /etc/fstab
/dev/mapper/qsan /mnt/DataCoreSAN ext4 defaults ,_netdev 0 0
```

3. The _netdev option will delay the time when mounting the file system on listed devices until the network has been started. Also ensures that the file system is unmounted before stopping the network at shutdown

Performance

Test XS3324 performance with 10 x SAS disks. And the configure setting is as follow.
bs=1M・Count=20480

```
# dd if=/dev/zero of=/dev/mapper/qsan bs=2M count=20480
(I/O configure setting)
20480+0 records in
20480+0 records out
42949672960 bytes (43 GB) copied, 45.942 s, 934 MB/s
```

Conclusion

Follow the procedures; it's easy to implement iSCSI and multipath I/O in Linux OS. Take advantage of these two convenient packages cooperate with QSAN XCubeSAN series controllers, enjoy a fault tolerance and performance enhancement technique.

Apply To

XCubeSAN XS3300

XCubeFAS series

References

XCubeSAN XEVO Software Manual

ANNOUNCEMENT

Copyright

August 2022

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

Trademarks

- QSAN, the QSAN logo, XCubeSAN, and QSAN.com are trademarks or registered trademarks of QSAN Technology, Inc.
- Microsoft, Windows, Windows Server, and Hyper-V are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.
- Linux is a trademark of Linus Torvalds in the United States and/or other countries.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Mac and OS X are trademarks of Apple Inc., registered in the U.S. and other countries.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- VMware, ESXi, and vSphere are registered trademarks or trademarks of VMware, Inc. in the United States and/or other countries.
- Citrix and Xen are registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries.
- Other trademarks and trade names used in this document to refer to either the entities claiming the marks and name or their products are the property of their respective owners.

APPENDIX

Related Documents

There are related documents which can be downloaded from the website.

- All XCubeSAN Documents
- XCubeSAN QIG (Quick Installation Guide)
- XCubeSAN Hardware Manual
- XCubeSAN Configuration Worksheet
- XCubeSAN XEVO Software Manual
- Compatibility Matrix
- White Papers
- Application Notes

Technical Support

- Do you have any questions or need help troubleshooting a problem? Please contact QSAN Support, we will reply to you as soon as possible.
- Via the Web: https://www.qsan.com/en/contact_support.php
- Via Telephone: +886-2-7720-6355 (Service hours: 09:30 - 18:00, Monday - Friday, UTC+8)
- Via Skype Chat, Skype ID: qsan.support (Service hours: 09:30 - 02:00, Monday - Friday, UTC+8, Summertime: 09:30 - 01:00)
- Via Email: support@qsan.com