XCubeSAN Series Application Note

Achieve DR Solution with XCubeSAN on VMware with Snapshot Consistency

QSAN Technology, Inc.
www.QSAN.com

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

This XCubeSAN series white paper is applicable to the following XCubeSAN models:

XCubeSAN Storage System 4U 19” Rack Mount Models

XS5224D	Dual Controller	LFF 24-disk 4U Chassis
XS3224D	Dual Controller	LFF 24-disk 4U Chassis
XS3224S	Single Controller	LFF 24-disk 4U Chassis
XS1224D	Dual Controller	LFF 24-disk 4U Chassis
XS1224S	Single Controller	LFF 24-disk 4U Chassis

XCubeSAN Storage System 3U 19” Rack Mount Models

XS5216D	Dual Controller	LFF 16-disk 3U Chassis
XS3216D	Dual Controller	LFF 16-disk 3U Chassis
XS3216S	Single Controller	LFF 16-disk 3U Chassis
XS1216D	Dual Controller	LFF 16-disk 3U Chassis
XS1216S	Single Controller	LFF 16-disk 3U Chassis

XCubeSAN Storage System 2U 19” Rack Mount Models

XS5212D	Dual Controller	LFF 12-disk 2U Chassis
XS5212S	Single Controller	LFF 12-disk 2U Chassis
XS3212D	Dual Controller	LFF 12-disk 2U Chassis
XS3212S	Single Controller	LFF 12-disk 2U Chassis
XS1212D	Dual Controller	LFF 12-disk 2U Chassis
XS1212S	Single Controller	LFF 12-disk 2U Chassis
XS5226D	Dual Controller	SFF 26-disk 2U Chassis
XS5226S	Single Controller	SFF 26-disk 2U Chassis
XS3226D	Dual Controller	SFF 26-disk 2U Chassis
XS3226S	Single Controller	SFF 26-disk 2U Chassis
XS1226D	Dual Controller	SFF 26-disk 2U Chassis

XS1226S	Single Controller	SFF 26-disk 2U Chassis

Information contained in document has been reviewed for accuracy. But it could include typographical errors or technical inaccuracies. Changes are made to the document periodically. These changes will be incorporated in new editions of the publication. QSAN may make improvements or changes in the products. All features, functionality, and product specifications are subject to change without prior notice or obligation. All statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This application note provides technical guidance for setting up DR (Disaster Recovery) solution in VMware environment and making sure that the replicated data will be consistent with special script implemented in ESXi server, and it leads XCubeSAN products being able to achieve real DR with snapshot consistency, it is no longer necessary to install any agent in the environment before achieving this.

This document is applicable for QSAN customers and partners who are interested in learning about DR solution on VMware. It assumes the reader is familiar with QSAN products and has general IT experience, including knowledge as a system or network administrator. If there is any question, please refer to the user manuals of products, or contact QSAN support for further assistance.

Nowadays, backup has been considered as one of the most important parts of implementing Data Center environment, backing up data in a single location is no longer enough to prevent disaster, IT manager may need to prepare another copy of the important data in a remote site, DR (Disaster Recovery) solution has become the Top-1 option to be chosen. Virtualization environment may have its own DR application, but usually it is more expensive. Storage vendors have the same backup function supported natively without additional charge, but the headache here is the cached data stored in the server's memory.

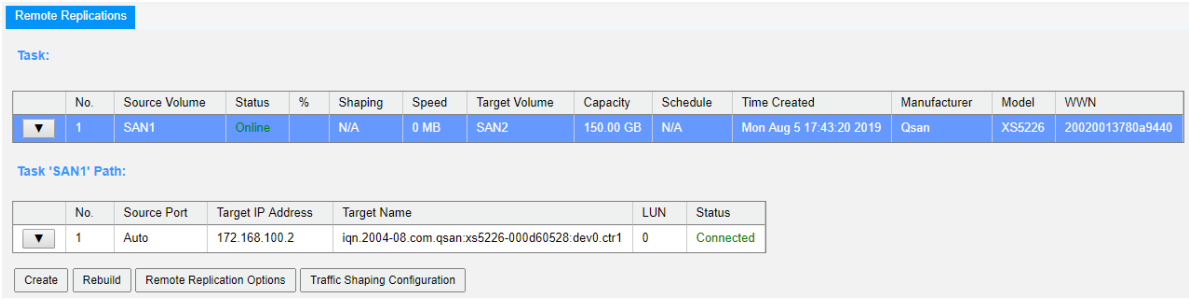
It is not a problem for those A-brand vendors as they have implemented an additional tool to be installed in the environment, supporting the feature of requesting the server to queue its I/O while the snapshot is being taken on the storage side, that makes the taken snapshot being complete with the full image of the written data, the replicated data won't be consistent without this kind of function, but the efforts to install the agent is another story.

This document will help you to set up the environment with the result as the above, but you won't need to install any agent in the environment before achieving this, this can easily be implemented with a simple script and XCubeSAN' Snapshot Replica.

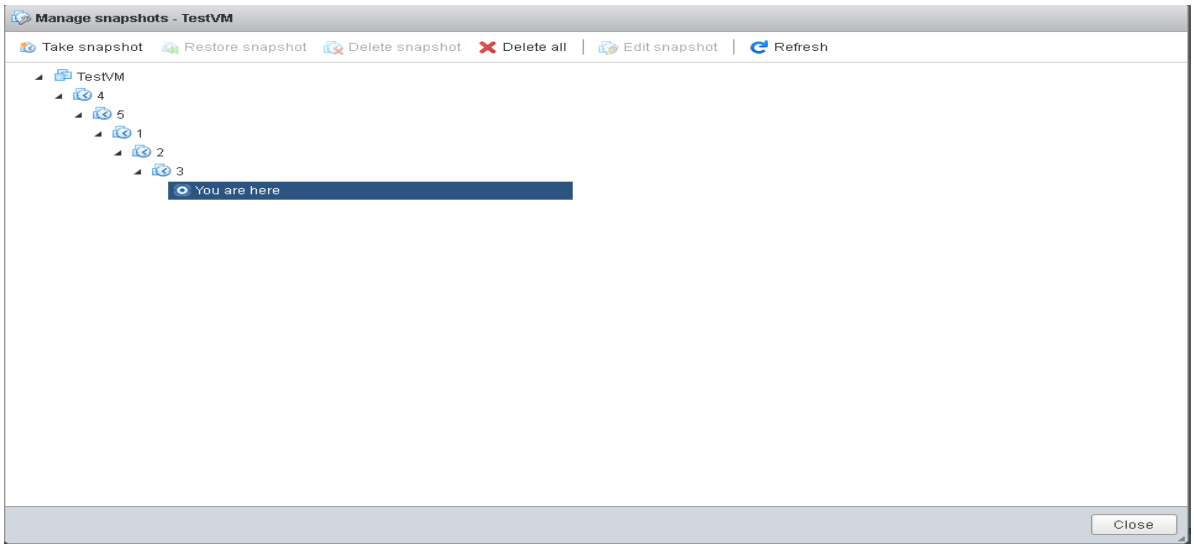
It's very simple, the environment we prepared here is an ESXi 6.5 server, installed with a 10G HBA card, connecting directly to XCubeSAN, and make sure that this ESXi server will be managed by a vCenter, and that's all!

As the purpose here is to achieve DR solution, you will need to set up two XCubeSAN systems, the available space must be the same on both units; otherwise the Snapshot Replica function may fail due to the insufficient storage space.

1. Connect one of 10G ports from SAN-a to SAN-b.
2. Create a pool and a volume on SAN-a.
3. Set the snapshot space so that the snapshot replica function can work well.
4. Repeat step2 and step3 above on SAN-b with the same size as the volume on SAN-a, you may need to set the snapshot space on the SAN-b.
5. Mount the created volume of SAN-a on the prepared ESXi server.
6. Create a VM (Virtual Machine) based on the mounted / created Datastore on the ESXi server, here we use Ubuntu16.04.
7. Visit SAN-a, go to page, create a remote replication task by choosing the volume in SAN-b.



8. Open the Console of VM in ESXi server, making a robocopy task running periodically to keep increasing data.
9. Create the schedule snapshot on this VM from vCenter UI, in this example we take 5 snapshots.



10. The preparation has been finished here.

Yes as you can see from the above actions that we are going to firstly take snapshot on the VM from ESXi server itself and replicate the .VMDK file along with the taken snapshots together to the remote site, after the data (.VMDK + snapshots) is replicated to the remote site, register and rollback the taken snapshot after mounting the Volume of SAN-b, everything will be consistent with this method. However, the taken snapshot(s) won't be automatically deleted / rotated by VMware, and keeping lots of snapshot images will lead to

the situation of terrible performance in long term, so the Script here is to specify a fixed quantity of snapshot that the ESXi server can keep for rotation, preventing from the situation that the VM performance impact due to too many snapshots.

1. Create a “Crontabs” folder in the Datastore mounted from SAN-a.
2. Upload the following script “SnapshotAutoDelete.sh” into the “Crontabs” folder.

```
# cat SnapshotAutoDelete.sh

#!/bin/sh

LOG_PATH="/var/log/Schedule_Snapshot.log"
[ -f "$LOG_PATH" ] && rm $LOG_PATH;

QTY=2 # Reserved quantity
for i in `vim-cmd vmsvc/getallvms 2>/dev/null | awk '{print $1}' | grep -e "[0-9]"`
# Grab all Vmid on esxi
do
    SNAPSHOT_COUNT=`vim-cmd vmsvc/snapshot.get $i | egrep -- '--\|-CHILD|^\|-ROOT'
| wc -l`
         GuestName=$(vim-cmd vmsvc/get.summary $i | grep name | awk '{ print $3 }'
| cut -d \" -f 2)
    if [ $SNAPSHOT_COUNT -gt $QTY ]; then  # If the number of snapshots is greater
than the number of reservations
                  DELETE_COUNT=$(($SNAPSHOT_COUNT-$QTY))
                  OLD_SNAPSHOT_ID=`vim-cmd vmsvc/snapshot.get $i | grep Id | head -
$DELETE_COUNT | awk -F: '{print $2}'`
                  for n in $OLD_SNAPSHOT_ID
                  do
                           vim-cmd vmsvc/snapshot.remove $i $n; ret=$?
                                    sleep 30s
                                             if [ $ret -eq 0 ];then
                                                      echo "$(date "+%F %T") :
$GuestName snapshot $n Delete Success.." >> $LOG_PATH # Output to log path after
deletion
                                             else
                                                      echo "$(date "+%F %T") :
$GuestName snapshot $n Delete FAILED.." >> $LOG_PATH
                                             fi
                  done
    else
        echo "$(date "+%F %T") : $GuestName snapshot not found." >> $LOG_PATH
    fi
done
```

3. Change the permission of the script to 777, from the SSH session of ESXi server.

```
[root@local:~] cd vmfs/volumes/SAN1/Crontabs/
[root@local:/vmfs/volumes/5d445d0a-fae8654e-a676-001b21d4d680/Crontabs] chmod 777 SnapshotAutoDelete.sh
[root@local:/vmfs/volumes/5d445d0a-fae8654e-a676-001b21d4d680/Crontabs] ls -al
total 1152
drwxr-xr-x    1 root     root         73728 Aug  2 16:38 .
drwxr-xr-t    1 root     root         73728 Aug  2 16:38 ..
-rwxrwxrwx    1 root     root          1088 Aug  2 18:52 SnapshotAutoDelete.sh
[root@local:/vmfs/volumes/5d445d0a-fae8654e-a676-001b21d4d680/Crontabs]
```

4. Locate the Datastore (Volume of SAN) by the below command from the SSH session.

```
# esxcli storage filesystem list
```

```
[root@local:~] esxcli storage filesystem list
Mount Point                                       Volume Name  UUID                                 Mounted  Type            Size          Free
-------------------------------------------------  -----------  -----------------------------------  -------  ------  ------------  ------------
/vmfs/volumes/5bc3fd0f-f996289d-ba94-001018edee60  datastore1   5bc3fd0f-f996289d-ba94-001018edee60     true  VMFS-6  492042190848  442177159168
/vmfs/volumes/5d445d0a-fae8654e-a676-001b21d4d680  SAN1         5d445d0a-fae8654e-a676-001b21d4d680     true  VMFS-6  160792838144   88226136064
/vmfs/volumes/5ceb8d20-96976e3b-25ef-08606e151c65               5ceb8d20-96976e3b-25ef-08606e151c65     true  vfat       299712512      80486400
/vmfs/volumes/9bfaa77a-a157614d-7923-8cc7a16bcdea               9bfaa77a-a157614d-7923-8cc7a16bcdea     true  vfat       261853184     261844992
/vmfs/volumes/3d40c777-b5b2f4fb-b003-5dfeca8c4b86               3d40c777-b5b2f4fb-b003-5dfeca8c4b86     true  vfat       261853184     113819648
/vmfs/volumes/5ceb8d28-4a26e650-7a8a-08606e151c65               5ceb8d28-4a26e650-7a8a-08606e151c65     true  vfat      4293591040    4264230912
[root@local:~]
```

5. Using below command to add a cron job to execute the script at 23:30 everyday (you may specify the time point based on your scenario, this time point should be earlier than the scheduled snapshot task created via vCenter mentioned above), or you may edit this file directly.

```
# echo "30    23    *    *    *    sh /vmfs/volumes/5d445d0a-fae8654e-a676-
00lb2ld4d680/Crontabs/SnapshotAutoDelete.sh" >> /var/spool/cron/crontabs/root
```

The YELLOW'ed word above is the UUID of the NFS Datastore, please check yours by the above command.

6. Edit the native cron job of ESXi server (vi /etc/rc.local.d/local.sh), and add below commands at the end of the configuration file.

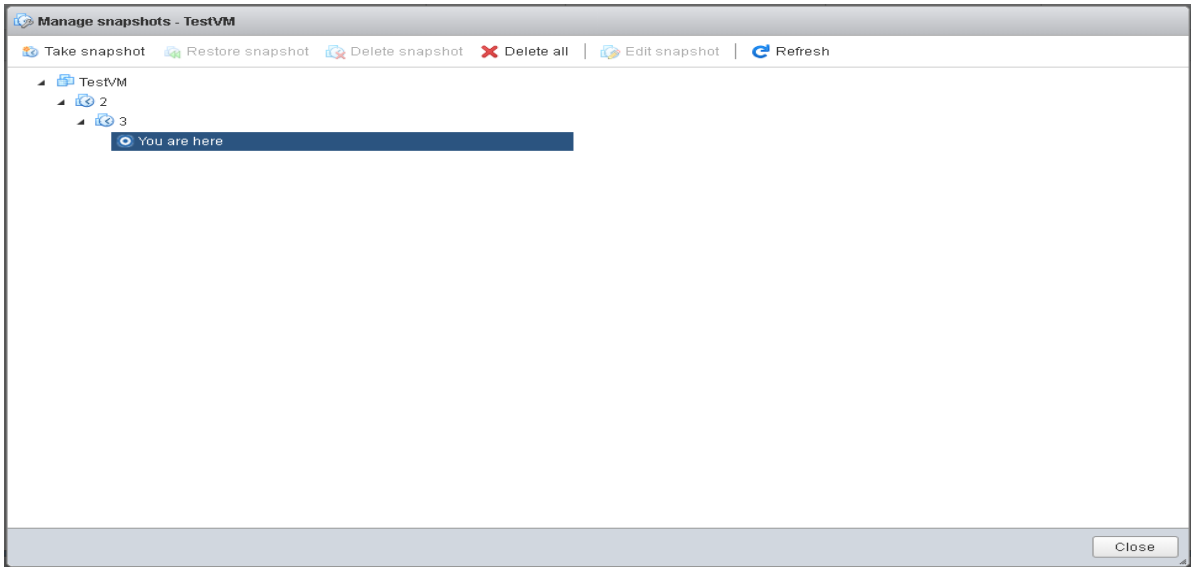
```
# /bin/echo "30    23    *    *    *    sh /vmfs/volumes/5d445d0a-fae8654e-a676-
00lb2ld4d680/Crontabs/SnapshotAutoDelete.sh" >>/var/spool/cron/crontabs/root
# /bin/kill $(cat /var/run/crond.pid)
# /usr/lib/vmware/busybox/bin/busybox crond
```

```
[root@local:~] esxcli storage filesystem list
Mount Point                                       Volume Name  UUID                                 Mounted  Type            Size          Free
-------------------------------------------------  -----------  -----------------------------------  -------  ------  ------------  ------------
/vmfs/volumes/5bc3fd0f-f996289d-ba94-001018edee60  datastore1   5bc3fd0f-f996289d-ba94-001018edee60     true  VMFS-6  492042190848  442177159168
/vmfs/volumes/5d445d0a-fae8654e-a676-001b21d4d680  SAN1         5d445d0a-fae8654e-a676-001b21d4d680     true  VMFS-6  160792838144   88226136064
/vmfs/volumes/5ceb8d20-96976e3b-25ef-08606e151c65               5ceb8d20-96976e3b-25ef-08606e151c65     true  vfat       299712512      80486400
/vmfs/volumes/9bfaa77a-a157614d-7923-8cc7a16bcdea               9bfaa77a-a157614d-7923-8cc7a16bcdea     true  vfat       261853184     261844992
/vmfs/volumes/3d40c777-b5b2f4fb-b003-5dfeca8c4b86               3d40c777-b5b2f4fb-b003-5dfeca8c4b86     true  vfat       261853184     113819648
/vmfs/volumes/5ceb8d28-4a26e650-7a8a-08606e151c65               5ceb8d28-4a26e650-7a8a-08606e151c65     true  vfat      4293591040    4264230912
[root@local:~]
```

```
#!/bin/sh

# local configuration options

# Note: modify at your own risk!  If you do/use anything in this
# script that is not part of a stable API (relying on files to be in
# specific places, specific tools, specific output, etc) there is a
# possibility you will end up with a broken system after patching or
# upgrading.  Changes are not supported unless under direction of
# VMware support.

# Note: This script will not be run when UEFI secure boot is enabled.

/bin/echo "30    11   *   *   *   sh /vmfs/volumes/5d445d0a-fae8654e-a676-001b21d4d680/Crontabs/SnapshotAutoDelete.sh" >>/var/spool/cron/crontabs/root
/bin/kill $(cat /var/run/crond.pid)
/usr/lib/vmware/busybox/bin/busybox crond


exit 0
```

As the configuration will be clear after a reboot of ESXi server, you need to add the above commands to save the configuration permanently.

7. Check the reserved snapshot quantity from ESXi UI, confirmed that the snapshots have been kept with the latest two.



8. And you may check the log by the below command.

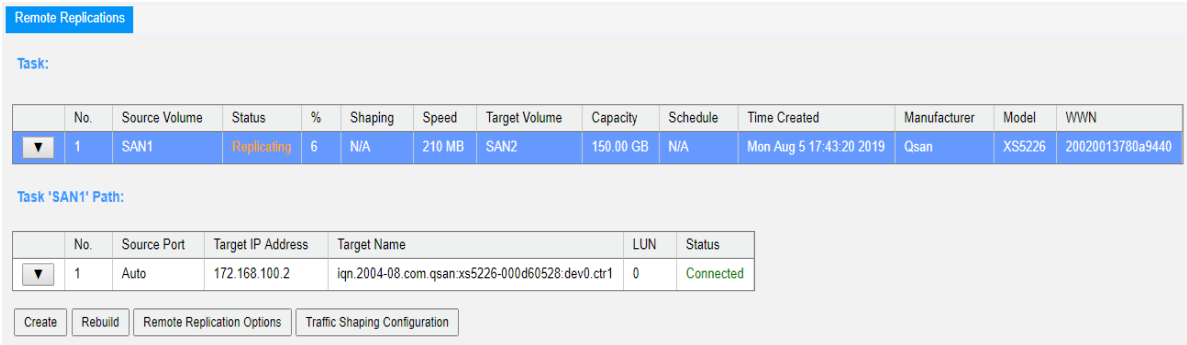
```
# cat /var/log/Schedule_Snapshot.log
```

```
[root@local:~] cat /var/log/Schedule_Snapshot.log
2019-08-05 11:30:38 : 2012R2-SAN1 snapshot 1 Delete Success..
2019-08-05 11:31:12 : 2012R2-SAN1 snapshot 2 Delete Success..
2019-08-05 11:32:10 : 2012R2-SAN1 snapshot 3 Delete Success..
[root@local:~]
```

9. The configuration from ESXi server side has been completed.

Now we are going to set up the remote replication by schedule from one to another unit, so that the VM (.VMDK file) can be replicated to the remote location along with the taken snapshot on the ESXi server.

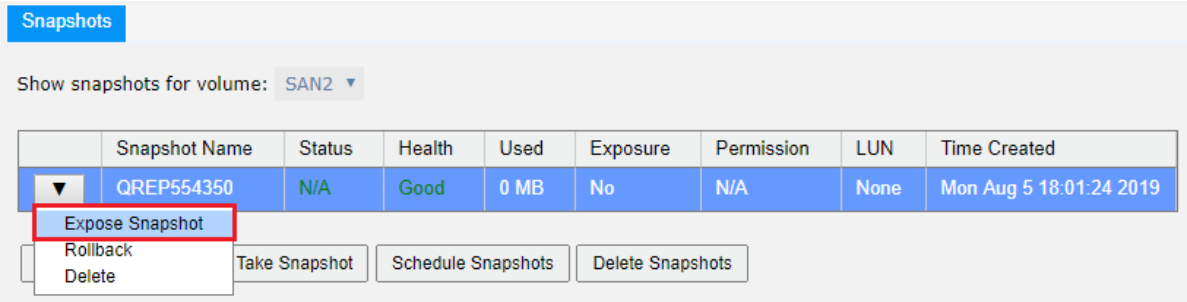
1. Visit on web UI of SAN-a, and execute the created remote replication task.



	No.	Source Volume	Status	%	Shaping	Speed	Target Volume	Capacity	Schedule	Time Created	Manufacturer	Model	WWN
▼	1	SAN1	Replicating	6	N/A	210 MB	SAN2	150.00 GB	N/A	Mon Aug 5 17:43:20 2019	Qsan	XS5226	20020013780a9440

	No.	Source Port	Target IP Address	Target Name	LUN	Status
▼	1	Auto	172.168.100.2	iqn.2004-08.com.qsan:xs5226-000d60528:dev0.ctr1	0	Connected

2. You may need to umount the original Datastore (of SAN-a) from the ESXi server to simulate the disaster occurring on the SAN-a.
3. Visit on web UI of SAN-b, and the replicated snapshot as a writable volume, this is called Writable Snapshot function.



4. After the expose is completed, please map the volume with read-write permission LUN, and visit vCenter UI (of the ESXi server), mount the exposed snapshot volume to be a Datastore.
5. During the mounting process of Datastore, the ESXi system will require you to assign a New Signature or use the Existing one, please choose to use the one.
6. Right-click on the Datastore, you shall be able to see VM replicated from SAN-a, and you may then register this VM and try to boot it up after a rollback of the taken snapshot on VM