外部LUN导入

Enterprise applications

NetApp
February 11, 2026

This PDF was generated from https://docs.netapp.com/zh-cn/ontap-apps-dbs/oracle/oracle-migration-fli-planning.html on February 11, 2026. Always check docs.netapp.com for the latest.

目录

外部LUN导入 1
规划 1
确定LVM LUN 1
确定ASM LUN 2
FC网络更改 2
确定外部阵列 3
确定外部LUN 3
将外部阵列LUN注册为候选导入阵列 4
创建卷以托管迁移的LUN 4
创建ONTAP LUN 4
创建导入关系 5
创建启动程序组 6
将新LUN映射到主机 6
转换 7
关闭数据库 7
关闭网格服务 8
卸载文件系统 8
停用卷组 9
FC网络更改 9
启动导入过程 9
监控导入进度 9
扫描SCSI设备更改 10
检查多路径设备 11
重新激活LVM卷组 12
重新挂载文件系统 13
重新扫描ASM设备 13
重新启动网格服务 14
重新启动数据库 14
完成 15
删除导入关系 16
取消注册外部LUN 16
协议转换 17
安装iSCSI启动程序 17
确定iSCSI启动程序名称 19
创建新启动程序组 19
关闭环境 19
取消LUN与FC网络的映射 19
将LUN重新映射到IP网络 20
发现iSCSI目标 20

发现iSCSI LUN 21
重新启动环境 21

外部LUN导入

规划

NetApp中介绍了使用FLI迁移SAN资源的过程 "ONTAP 外部 LUN 导入文档"。

从数据库和主机的角度来看、不需要执行任何特殊步骤。更新FC分区并使LUN在ONTAP上可用后、LVM应能够从LUN中读取LVM元数据。此外、卷组已准备就绪、无需执行其他配置步骤。在极少数情况下、环境可能会包含使用先前存储阵列的引用进行硬编码的配置文件。例如、包含的Linux系统 `/etc/multipath.conf` 必须更新引用给定设备的WWN的规则、以反映FLI所做的更改。

有关支持的配置的信息、请参见NetApp兼容性列表。如果您的环境未包括在其中、请与NetApp代表联系以获得帮助。

此示例显示了Linux服务器上托管的ASM和LVM LUN的迁移。FLI在其他操作系统上受支持、尽管主机端命令可能不同、但原则相同、ONTAP过程相同。

确定LVM LUN

准备工作的第一步是确定要迁移的LUN。在此处显示的示例中、两个基于SAN的文件系统挂载在上 `/orabin` 和 `/backups`。

```
[root@host1 ~]# df -k
Filesystem                          1K-blocks      Used Available Use%
Mounted on
/dev/mapper/rhel-root                52403200   8811464  43591736  17% /
devtmpfs                             65882776         0  65882776   0% /dev
...
fas8060-nfs-public:/install         199229440 119368128  79861312  60%
/install
/dev/mapper/sanvg-lvorabin           20961280  12348476   8612804  59%
/orabin
/dev/mapper/sanvg-lvbackups          73364480  62947536  10416944  86%
/backups
```

可以从设备名称中提取卷组的名称、该名称采用格式(卷组名称)-(逻辑卷名称)。在这种情况下、卷组称为 `sanvg`。

。 `pvdisplay` 命令可按如下所示来确定支持此卷组的LUN。在这种情况下、包含10个LUN `sanvg` 卷组。

```
[root@host1 ~]# pvdisplay -C -o pv_name,pv_size,pv_fmt,vg_name
  PV                                                PSize   VG
  /dev/mapper/3600a0980383030445424487556574266  10.00g sanvg
  /dev/mapper/3600a0980383030445424487556574267  10.00g sanvg
  /dev/mapper/3600a0980383030445424487556574268  10.00g sanvg
  /dev/mapper/3600a0980383030445424487556574269  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426a  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426b  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426c  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426d  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426e  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426f  10.00g sanvg
  /dev/sda2                                       278.38g rhel
```

确定ASM LUN

此外、还必须迁移ASM LUN。要以sysasm用户身份从sqlplus获取LUN和LUN路径数、请运行以下命令：

```
SQL> select path||' '||os_mb from v$asm_disk;
PATH||''||OS_MB
--------------------------------------------------------------------------------
------
/dev/oracleasm/disks/ASM0 10240
/dev/oracleasm/disks/ASM9 10240
/dev/oracleasm/disks/ASM8 10240
/dev/oracleasm/disks/ASM7 10240
/dev/oracleasm/disks/ASM6 10240
/dev/oracleasm/disks/ASM5 10240
/dev/oracleasm/disks/ASM4 10240
/dev/oracleasm/disks/ASM1 10240
/dev/oracleasm/disks/ASM3 10240
/dev/oracleasm/disks/ASM2 10240
10 rows selected.
SQL>
```

FC网络更改

当前环境包含20个要迁移的LUN。更新当前SAN、以便ONTAP可以访问当前LUN。尚未迁移数据、但ONTAP必须从当前LUN中读取配置信息、才能为该数据创建新的主目录。

至少必须将AF/FAS系统上的一个HBA端口配置为启动程序端口。此外、必须更新FC分区、以便ONTAP可以访问外部存储阵列上的LUN。某些存储阵列配置了LUN屏蔽、用于限制哪些WWN可以访问给定LUN。在这种情况下、还必须更新LUN屏蔽以授予对ONTAP WWN的访问权限。

完成此步骤后、ONTAP应能够使用查看外部存储阵列 `storage array show` 命令：它返回的关键字段是用于

标识系统上的外部LUN的前缀。在以下示例中、是外部阵列上的LUN `FOREIGN_1` 在ONTAP中显示、并使用前缀 `FOR-1`。

确定外部阵列

```
Cluster01::> storage array show -fields name,prefix
name          prefix
------------- ------
FOREIGN_1     FOR-1
Cluster01::>
```

确定外部**LUN**

通过传递、可以列出这些LUN `array-name` 到 `storage disk show` 命令：在迁移操作步骤期间、系统会多次引用返回的数据。

```
Cluster01::> storage disk show -array-name FOREIGN_1 -fields disk,serial
disk     serial-number
-------- -------------
FOR-1.1  800DT$HuVWBX
FOR-1.2  800DT$HuVWBZ
FOR-1.3  800DT$HuVWBW
FOR-1.4  800DT$HuVWBY
FOR-1.5  800DT$HuVWB/
FOR-1.6  800DT$HuVWBa
FOR-1.7  800DT$HuVWBd
FOR-1.8  800DT$HuVWBb
FOR-1.9  800DT$HuVWBc
FOR-1.10 800DT$HuVWBe
FOR-1.11 800DT$HuVWBf
FOR-1.12 800DT$HuVWBg
FOR-1.13 800DT$HuVWBi
FOR-1.14 800DT$HuVWBh
FOR-1.15 800DT$HuVWBj
FOR-1.16 800DT$HuVWBk
FOR-1.17 800DT$HuVWBm
FOR-1.18 800DT$HuVWBl
FOR-1.19 800DT$HuVWBo
FOR-1.20 800DT$HuVWBn
20 entries were displayed.
Cluster01::>
```

将外部阵列**LUN**注册为候选导入阵列

外部LUN最初归类为任何特定的LUN类型。在导入数据之前、必须将LUN标记为外部LUN、从而使其成为导入过程的候选LUN。此步骤可通过将序列号传递到来完成 `storage disk modify` 命令、如以下示例所示。请注意、此过程仅会将LUN标记为ONTAP中的外部LUN。不会向外部LUN本身写入任何数据。

```
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign true
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign true
Cluster01::*>
```

创建卷以托管迁移的**LUN**

托管迁移的LUN需要一个卷。确切的卷配置取决于利用ONTAP功能的整体计划。在此示例中、ASM LUN放置在一个卷中、而LVM LUN放置在另一个卷中。这样、您就可以将LUN作为独立的组进行管理、以实现分层、创建快照或设置QoS控制等目的。

设置 `snapshot-policy `to `none`。迁移过程中可能会涉及大量的数据周转。因此、如果由于在快照中捕获不需要的数据而意外创建快照、则空间消耗可能会大幅增加。

```
Cluster01::> volume create -volume new_asm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1152] Job succeeded: Successful
Cluster01::> volume create -volume new_lvm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1153] Job succeeded: Successful
Cluster01::>
```

创建**ONTAP LUN**

创建卷后、必须创建新的LUN。通常、创建LUN需要用户指定LUN大小等信息、但在这种情况下、外部磁盘参数会传递到命令。因此、ONTAP会从指定序列号复制当前LUN配置数据。它还会使用LUN几何结构和分区表数据来调整LUN对齐并建立最佳性能。

在此步骤中、必须对照外部阵列交叉引用序列号、以确保正确的外部LUN与正确的新LUN匹配。

```
Cluster01::*> lun create -vserver vserver1 -path /vol/new_asm/LUN0 -ostype
linux -foreign-disk 800DT$HuVWBW
Created a LUN of size 10g (10737418240)
Cluster01::*> lun create -vserver vserver1 -path /vol/new_asm/LUN1 -ostype
linux -foreign-disk 800DT$HuVWBX
Created a LUN of size 10g (10737418240)
...
Created a LUN of size 10g (10737418240)
Cluster01::*> lun create -vserver vserver1 -path /vol/new_lvm/LUN8 -ostype
linux -foreign-disk 800DT$HuVWBn
Created a LUN of size 10g (10737418240)
Cluster01::*> lun create -vserver vserver1 -path /vol/new_lvm/LUN9 -ostype
linux -foreign-disk 800DT$HuVWBo
Created a LUN of size 10g (10737418240)
```

创建导入关系

LUN现在已创建、但尚未配置为复制目标。在执行此步骤之前、必须先将LUN置于脱机状态。这一额外步骤旨在保护数据免受用户错误的影响。如果ONTAP允许对联机LUN执行迁移、则会存在一个风险、即因出现输入错误而可能会覆盖活动数据。强制用户首先使LUN脱机这一额外步骤有助于验证是否将正确的目标LUN用作迁移目标。

```
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN0
Warning: This command will take LUN "/vol/new_asm/LUN0" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN1
Warning: This command will take LUN "/vol/new_asm/LUN1" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
...
Warning: This command will take LUN "/vol/new_lvm/LUN8" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_lvm/LUN9
Warning: This command will take LUN "/vol/new_lvm/LUN9" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
```

LUN脱机后、您可以通过将外部LUN序列号传递到来建立导入关系 `lun import create` 命令：

```
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN0
-foreign-disk 800DT$HuVWBW
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN1
-foreign-disk 800DT$HuVWBX
...
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN8
-foreign-disk 800DT$HuVWBn
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN9
-foreign-disk 800DT$HuVWBo
Cluster01::*>
```

建立所有导入关系后、可以将LUN重新置于联机状态。

```
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN9
Cluster01::*>
```

创建启动程序组

启动程序组(igroGroup)是ONTAP LUN屏蔽架构的一部分。除非先授予主机访问权限、否则无法访问新创建的LUN。为此、可创建一个igrop、其中列出应授予访问权限的FC WWN或iSCSI启动程序名称。编写此报告时、只有FC LUN支持FLI。但是、迁移后转换为iSCSI是一项简单的任务、如所示 "协议转换"。

在此示例中、创建了一个igrop、其中包含两个WWN、分别对应于主机HBA上的两个可用端口。

```
Cluster01::*> igroup create linuxhost -protocol fcp -ostype linux
-initiator 21:00:00:0e:1e:16:63:50 21:00:00:0e:1e:16:63:51
```

将新**LUN**映射到主机

创建igroup后、LUN将映射到定义的igroup。这些LUN仅可供此igrop中包含的WWN使用。在迁移过程的这一阶段、NetApp会假定主机尚未分区到ONTAP。这一点非常重要、因为如果将主机同时分区到外部阵列和新的ONTAP系统、则可能会在每个阵列上发现具有相同序列号的LUN。这种情况可能会导致多路径故障或数据损坏。

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxhost
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxhost
Cluster01::*>
```

转换

由于需要更改FC网络配置、外部LUN导入期间不可避免地会发生某些中断。但是、中断的持续时间不必比重新启动数据库环境并更新FC分区以将主机FC连接从外部LUN切换到ONTAP所需的时间长。

此过程可概括如下：

1. 将外部LUN上的所有LUN活动置于静噪状态。
2. 将主机FC连接重定向到新的ONTAP系统。
3. 触发导入过程。
4. 重新发现LUN。
5. 重新启动数据库。

您无需等待迁移过程完成。给定LUN的迁移开始后、该LUN便可在ONTAP上使用、并可在数据复制过程继续期间提供数据。所有读取都会传递到外部LUN、所有写入都会同步写入到两个阵列。复制操作速度非常快、重定向FC流量的开销也非常小、因此对性能的任何影响都应该是瞬时的、并且最小化。如果有问题、您可以延迟重新启动环境、直到迁移过程完成并删除导入关系之后。

关闭数据库

在此示例中、静音环境的第一步是关闭数据库。

```
[oracle@host1 bin]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base remains unchanged with value /orabin
[oracle@host1 bin]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, Automatic Storage Management, OLAP, Advanced
Analytics
and Real Application Testing options
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

关闭网格服务

要迁移的基于SAN的文件系统之一还包括Oracle ASM服务。暂停底层LUN需要卸载文件系统、这反过来意味着停止此文件系统上具有已打开文件的所有进程。

```
[oracle@host1 bin]$ ./crsctl stop has -f
CRS-2791: Starting shutdown of Oracle High Availability Services-managed
resources on 'host1'
CRS-2673: Attempting to stop 'ora.evmd' on 'host1'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host1'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host1'
CRS-2677: Stop of 'ora.DATA.dg' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host1'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host1' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host1' succeeded
CRS-2677: Stop of 'ora.asm' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host1'
CRS-2677: Stop of 'ora.cssd' on 'host1' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed resources
on 'host1' has completed
CRS-4133: Oracle High Availability Services has been stopped.
[oracle@host1 bin]$
```

卸载文件系统

如果所有进程均已关闭、则卸载操作将成功。如果权限被拒绝、则文件系统上必须存在一个具有锁定的进程。。

`fuser` 命令有助于识别这些进程。

```
[root@host1 ~]# umount /orabin
[root@host1 ~]# umount /backups
```

停用卷组

卸载给定卷组中的所有文件系统后、可以停用该卷组。

```
[root@host1 ~]# vgchange --activate n sanvg
  0 logical volume(s) in volume group "sanvg" now active
[root@host1 ~]#
```

FC网络更改

现在、可以更新FC分区、以删除主机对外部阵列的所有访问权限、并建立对ONTAP的访问权限。

启动导入过程

要启动LUN导入过程、请运行 `lun import start` 命令：

```
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN0
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN1
...
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN8
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN9
Cluster01::lun import*>
```

监控导入进度

可以使用监控导入操作 `lun import show` 命令：如下图所示、所有20个LUN的导入正在进行中、这意味着现在可以通过ONTAP访问数据、即使数据复制操作仍在进行。

```
Cluster01::lun import*> lun import show -fields path,percent-complete
vserver   foreign-disk path              percent-complete
--------- ------------ ----------------- ----------------
vserver1  800DT$HuVWB/ /vol/new_asm/LUN4 5
vserver1  800DT$HuVWBW /vol/new_asm/LUN0 5
vserver1  800DT$HuVWBX /vol/new_asm/LUN1 6
vserver1  800DT$HuVWBY /vol/new_asm/LUN2 6
vserver1  800DT$HuVWBZ /vol/new_asm/LUN3 5
vserver1  800DT$HuVWBa /vol/new_asm/LUN5 4
vserver1  800DT$HuVWBb /vol/new_asm/LUN6 4
vserver1  800DT$HuVWBc /vol/new_asm/LUN7 4
vserver1  800DT$HuVWBd /vol/new_asm/LUN8 4
vserver1  800DT$HuVWBe /vol/new_asm/LUN9 4
vserver1  800DT$HuVWBf /vol/new_lvm/LUN0 5
vserver1  800DT$HuVWBg /vol/new_lvm/LUN1 4
vserver1  800DT$HuVWBh /vol/new_lvm/LUN2 4
vserver1  800DT$HuVWBi /vol/new_lvm/LUN3 3
vserver1  800DT$HuVWBj /vol/new_lvm/LUN4 3
vserver1  800DT$HuVWBk /vol/new_lvm/LUN5 3
vserver1  800DT$HuVWBl /vol/new_lvm/LUN6 4
vserver1  800DT$HuVWBm /vol/new_lvm/LUN7 3
vserver1  800DT$HuVWBn /vol/new_lvm/LUN8 2
vserver1  800DT$HuVWBo /vol/new_lvm/LUN9 2
20 entries were displayed.
```

如果需要脱机进程、请延迟重新发现或重新启动服务、直到命令指示所有迁移均已成功完成为止 `lun import show`。然后，您可以按中所述完成迁移过程"外部LUN导入—完成"。

如果需要联机迁移、请继续在新主目录中重新发现LUN并启动服务。

扫描SCSI设备更改

在大多数情况下、重新发现新LUN的最简单方法是重新启动主机。这样做会自动删除旧的陈旧设备、正确发现所有新LUN并构建关联的设备、例如多路径设备。此处的示例显示了一个完全联机的流程、用于演示目的。

注意：重新启动主机之前、请确保中的所有条目都已启用 `/etc/fstab` 此参考迁移的SAN资源已被注释掉。如果不执行此操作、并且LUN访问出现问题、则操作系统可能无法启动。这种情况不会损坏数据。但是、启动到救援模式或类似模式并更正可能会非常不方便 `/etc/fstab` 以便可以启动操作系统以启用故障排除。

可以使用重新扫描此示例中使用的Linux版本上的LUN `rescan-scsi-bus.sh` 命令：如果命令成功、则输出中应显示每个LUN路径。输出可能难以解释、但如果分区和igrop配置正确、则应显示许多LUN包含 `NETAPP` 供应商字符串。

```
[root@host1 /]# rescan-scsi-bus.sh
Scanning SCSI subsystem for new devices
Scanning host 0 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
 Scanning for device 0 2 0 0 ...
OLD: Host: scsi0 Channel: 02 Id: 00 Lun: 00
      Vendor: LSI      Model: RAID SAS 6G 0/1  Rev: 2.13
      Type:   Direct-Access                    ANSI SCSI revision: 05
Scanning host 1 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
 Scanning for device 1 0 0 0 ...
OLD: Host: scsi1 Channel: 00 Id: 00 Lun: 00
      Vendor: Optiarc  Model: DVD RW AD-7760H  Rev: 1.41
      Type:   CD-ROM                           ANSI SCSI revision: 05
Scanning host 2 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 3 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 4 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 5 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 6 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 7 for  all SCSI target IDs, all LUNs
 Scanning for device 7 0 0 10 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 10
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
 Scanning for device 7 0 0 11 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 11
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
 Scanning for device 7 0 0 12 ...
...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 18
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
 Scanning for device 9 0 1 19 ...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 19
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
0 new or changed device(s) found.
0 remapped or resized device(s) found.
0 device(s) removed.
```

检查多路径设备

LUN发现过程还会触发多路径设备的重新创建、但已知Linux多路径驱动程序偶尔会出现问题。的输出 `multipath - ll` 应进行检查、以验证输出是否如预期。例如、以下输出显示了与关联的多路径设备 `NETAPP` 供应商字符串。每个设备都有四个路径、其中两个路径的优先级为50、两个路径的优先级为10。尽管不同版本的Linux的确切输出可能会有所不同、但此输出看起来与预期一致。

请参考用于验证的Linux版本的Host Utilities文档 `/etc/multipath.conf` 设置正确。

```
[root@host1 /]# multipath -ll
3600a098038303558735d493762504b36 dm-5 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:4  sdat 66:208 active ready running
| `- 9:0:1:4  sdbn 68:16  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:4  sdf  8:80   active ready running
  `- 9:0:0:4  sdz  65:144 active ready running
3600a098038303558735d493762504b2d dm-10 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:8  sdax 67:16  active ready running
| `- 9:0:1:8  sdbr 68:80  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:8  sdj  8:144  active ready running
  `- 9:0:0:8  sdad 65:208 active ready running
...
3600a098038303558735d493762504b37 dm-8 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:5  sdau 66:224 active ready running
| `- 9:0:1:5  sdbo 68:32  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:5  sdg  8:96   active ready running
  `- 9:0:0:5  sdaa 65:160 active ready running
3600a098038303558735d493762504b4b dm-22 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:19 sdbi 67:192 active ready running
| `- 9:0:1:19 sdcc 69:0   active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:19 sdu  65:64  active ready running
  `- 9:0:0:19 sdao 66:128 active ready running
```

重新激活**LVM**卷组

如果已正确发现LVM LUN、则 `vgchange --activate y` 命令应成功。这是一个很好的逻辑卷管理器价值示例。更改LUN的WWN甚至序列号并不重要、因为卷组元数据会写入LUN本身。

操作系统扫描了LUN、发现写入LUN上的少量数据、将其标识为属于的物理卷 `sanvg volumegroup`。然后构建所有必需的设备。只需重新激活卷组即可。

```
[root@host1 /]# vgchange --activate y sanvg
  Found duplicate PV fpCzdLTuKfy2xDZjai1NliJh3TjLUBiT: using
/dev/mapper/3600a098038303558735d493762504b46 not /dev/sdp
  Using duplicate PV /dev/mapper/3600a098038303558735d493762504b46 from
subsystem DM, ignoring /dev/sdp
  2 logical volume(s) in volume group "sanvg" now active
```

重新挂载文件系统

重新激活卷组后、可以在挂载文件系统时保持所有原始数据完好无损。如前文所述、即使数据复制在后端组中仍处于活动状态、文件系统也能完全正常运行。

```
[root@host1 /]# mount /orabin
[root@host1 /]# mount /backups
[root@host1 /]# df -k
Filesystem                         1K-blocks      Used Available Use%
Mounted on
/dev/mapper/rhel-root               52403200   8837100  43566100  17% /
devtmpfs                            65882776         0  65882776   0% /dev
tmpfs                                6291456        84   6291372   1%
/dev/shm
tmpfs                               65898668      9884  65888784   1% /run
tmpfs                               65898668         0  65898668   0%
/sys/fs/cgroup
/dev/sda1                             505580    224828    280752  45% /boot
fas8060-nfs-public:/install       199229440 119368256  79861184  60%
/install
fas8040-nfs-routable:/snapomatic    9961472     30528   9930944   1%
/snapomatic
tmpfs                               13179736        16  13179720   1%
/run/user/42
tmpfs                               13179736         0  13179736   0%
/run/user/0
/dev/mapper/sanvg-lvorabin          20961280  12357456   8603824  59%
/orabin
/dev/mapper/sanvg-lvbackups         73364480  62947536  10416944  86%
/backups
```

重新扫描**ASM**设备

重新扫描SCSI设备时、应已重新发现ASMlib设备。可以通过重新启动ASMlib并扫描磁盘来联机验证重新发现。

此步骤仅与使用ASMlib的ASM配置相关。

注意：如果未使用ASMlib、则为 `/dev/mapper` 设备应已自动重新创建。但是、权限可能不正确。如果没有ASMlib、则必须在底层设备上为ASM设置特殊权限。通常通过任一中的特殊条目来完成此操作 `/etc/multipath.conf` 或 `udev` 规则、或者可能同时位于这两个规则集中。可能需要更新这些文件、以反映环境中的WWN或序列号变化、从而确保ASM设备仍具有正确的权限。

在此示例中、重新启动ASMlib并扫描磁盘会显示与原始环境相同的10个ASM LUN。

```
[root@host1 /]# oracleasm exit
Unmounting ASMlib driver filesystem: /dev/oracleasm
Unloading module "oracleasm": oracleasm
[root@host1 /]# oracleasm init
Loading module "oracleasm": oracleasm
Configuring "oracleasm" to use device physical block size
Mounting ASMlib driver filesystem: /dev/oracleasm
[root@host1 /]# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "ASM0"
Instantiating disk "ASM1"
Instantiating disk "ASM2"
Instantiating disk "ASM3"
Instantiating disk "ASM4"
Instantiating disk "ASM5"
Instantiating disk "ASM6"
Instantiating disk "ASM7"
Instantiating disk "ASM8"
Instantiating disk "ASM9"
```

重新启动网格服务

现在LVM和ASM设备已联机且可用、可以重新启动网格服务。

```
[root@host1 /]# cd /orabin/product/12.1.0/grid/bin
[root@host1 bin]# ./crsctl start has
```

重新启动数据库

重新启动网格服务后、可以启动数据库。在尝试启动数据库之前、可能需要等待几分钟、以便ASM服务完全可用。

```
[root@host1 bin]# su - oracle
[oracle@host1 ~]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base has been set to /orabin
[oracle@host1 ~]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> startup
ORACLE instance started.
Total System Global Area 3221225472 bytes
Fixed Size                  4502416 bytes
Variable Size            1207962736 bytes
Database Buffers         1996488704 bytes
Redo Buffers               12271616 bytes
Database mounted.
Database opened.
SQL>
```

完成

从主机角度来看、迁移已完成、但仍会从外部阵列提供I/O、直到删除导入关系为止。

在删除关系之前、必须确认所有LUN的迁移过程均已完成。

```
Cluster01::*> lun import show -vserver vserver1 -fields foreign-
disk,path,operational-state
vserver   foreign-disk path              operational-state
--------- ------------ ----------------- -----------------
vserver1 800DT$HuVWB/ /vol/new_asm/LUN4 completed
vserver1 800DT$HuVWBW /vol/new_asm/LUN0 completed
vserver1 800DT$HuVWBX /vol/new_asm/LUN1 completed
vserver1 800DT$HuVWBY /vol/new_asm/LUN2 completed
vserver1 800DT$HuVWBZ /vol/new_asm/LUN3 completed
vserver1 800DT$HuVWBa /vol/new_asm/LUN5 completed
vserver1 800DT$HuVWBb /vol/new_asm/LUN6 completed
vserver1 800DT$HuVWBc /vol/new_asm/LUN7 completed
vserver1 800DT$HuVWBd /vol/new_asm/LUN8 completed
vserver1 800DT$HuVWBe /vol/new_asm/LUN9 completed
vserver1 800DT$HuVWBf /vol/new_lvm/LUN0 completed
vserver1 800DT$HuVWBg /vol/new_lvm/LUN1 completed
vserver1 800DT$HuVWBh /vol/new_lvm/LUN2 completed
vserver1 800DT$HuVWBi /vol/new_lvm/LUN3 completed
vserver1 800DT$HuVWBj /vol/new_lvm/LUN4 completed
vserver1 800DT$HuVWBk /vol/new_lvm/LUN5 completed
vserver1 800DT$HuVWBl /vol/new_lvm/LUN6 completed
vserver1 800DT$HuVWBm /vol/new_lvm/LUN7 completed
vserver1 800DT$HuVWBn /vol/new_lvm/LUN8 completed
vserver1 800DT$HuVWBo /vol/new_lvm/LUN9 completed
20 entries were displayed.
```

删除导入关系

迁移过程完成后、删除此迁移关系。完成此操作后、I/O将专门从ONTAP上的驱动器提供。

```
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN9
```

取消注册外部**LUN**

最后、修改磁盘以删除 `is-foreign` 名称。

```
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign false
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign false
Cluster01::*>
```

协议转换

更改用于访问LUN的协议是一项常见要求。

在某些情况下、将数据迁移到云是整体战略的一部分。TCP/IP是云协议、从FC更改为iSCSI可以更轻松地迁移到各种云环境。在其他情况下、iSCSI可能是利用IP SAN降低的成本的理想选择。有时、迁移可能会使用不同的协议作为临时措施。例如、如果外部阵列和基于ONTAP的LUN不能同时位于同一HBA上、则可以使用iSCSI LUN足够长的时间来从旧阵列复制数据。从系统中删除旧LUN后、您可以将其转换回FC。

以下操作步骤演示了从FC到iSCSI的转换、但总体原则适用于从iSCSI到FC的反向转换。

安装**iSCSI**启动程序

默认情况下、大多数操作系统都包含软件iSCSI启动程序、但如果未包含、则可以轻松安装。

```
[root@host1 /]# yum install -y iscsi-initiator-utils
Loaded plugins: langpacks, product-id, search-disabled-repos,
subscription-
              : manager
Resolving Dependencies
--> Running transaction check
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.el7 will be
updated
--> Processing Dependency: iscsi-initiator-utils = 6.2.0.873-32.el7 for
package: iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.el7 will be
an update
--> Running transaction check
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.el7 will
be updated
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.el7
will be an update
--> Finished Dependency Resolution
Dependencies Resolved
```

```
================================================================================
===
 Package                         Arch   Version              Repository
Size
================================================================================
===
Updating:
 iscsi-initiator-utils           x86_64 6.2.0.873-32.0.2.el7 ol7_latest 416
k
Updating for dependencies:
 iscsi-initiator-utils-iscsiuio x86_64 6.2.0.873-32.0.2.el7 ol7_latest  84
k
Transaction Summary
================================================================================
===
Upgrade  1 Package (+1 Dependent package)
Total download size: 501 k
Downloading packages:
No Presto metadata available for ol7_latest
(1/2): iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_6 | 416 kB   00:00
(2/2): iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2. |  84 kB   00:00
--------------------------------------------------------------------------------
---
Total                                           2.8 MB/s | 501 kB
00:00Cluster01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating   : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
1/4
  Updating   : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
2/4
  Cleanup    : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
3/4
  Cleanup    : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
4/4
rhel-7-server-eus-rpms/7Server/x86_64/productid       | 1.7 kB     00:00
rhel-7-server-rpms/7Server/x86_64/productid           | 1.7 kB     00:00
  Verifying  : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
1/4
  Verifying  : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
2/4
  Verifying  : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
3/4
  Verifying  : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
```

```
4/4
Updated:
  iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.el7
Dependency Updated:
  iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.el7
Complete!
[root@host1 /]#
```

确定iSCSI启动程序名称

在安装过程中会生成一个唯一的iSCSI启动程序名称。在Linux上、它位于中 `/etc/iscsi/initiatorname.iscsi` 文件此名称用于标识IP SAN上的主机。

```
[root@host1 /]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1992-05.com.redhat:497bd66ca0
```

创建新启动程序组

启动程序组(igroGroup)是ONTAP LUN屏蔽架构的一部分。除非先授予主机访问权限、否则无法访问新创建的LUN。完成此步骤的方法是创建一个igrop、其中列出了需要访问的FC WWN或iSCSI启动程序名称。

在此示例中、创建了一个igrop、其中包含Linux主机的iSCSI启动程序。

```
Cluster01::*> igroup create -igroup linuxiscsi -protocol iscsi -ostype
linux -initiator iqn.1994-05.com.redhat:497bd66ca0
```

关闭环境

在更改LUN协议之前、必须将LUN完全置于静状态。要转换的LUN之一上的任何数据库都必须关闭、文件系统必须卸载、卷组必须停用。如果使用ASM、请确保已卸载ASM磁盘组并关闭所有网格服务。

取消LUN与FC网络的映射

在LUN完全静置后、从原始FC igrop中删除映射。

```
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxhost
...
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxhost
```

将LUN重新映射到IP网络

将对每个LUN的访问权限授予新的基于iSCSI的启动程序组。

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxiscsi
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxiscsi
Cluster01::*>
```

发现iSCSI目标

iSCSI发现分为两个阶段。第一种方法是发现目标、这与发现LUN不同。。 `iscsiadm` 下面显示的命令用于探测由指定的门户组 `-p argument` 和用于存储提供iSCSI服务的所有IP地址和端口的列表。在这种情况下、有四个IP地址在默认端口3260上提供iSCSI服务。

如果无法访问任何目标IP地址、则此命令可能需要几分钟才能完成。

```
[root@host1 ~]# iscsiadm -m discovery -t st -p fas8060-iscsi-public1
10.63.147.197:3260,1033 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
10.63.147.198:3260,1034 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.203:3260,1030 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.202:3260,1029 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
```

发现iSCSI LUN

发现iSCSI目标后、重新启动iSCSI服务以发现可用的iSCSI LUN并构建关联设备、例如多路径或ASMlib设备。

```
[root@host1 ~]# service iscsi restart
Redirecting to /bin/systemctl restart  iscsi.service
```

重新启动环境

通过重新激活卷组、重新挂载文件系统、重新启动RAC服务等方式重新启动环境。作为预防措施、NetApp建议您在转换过程完成后重新启动服务器、以确保所有配置文件均正确无误、并且所有陈旧设备均已删除。

注意：重新启动主机之前、请确保中的所有条目都已启用 `/etc/fstab` 此参考迁移的SAN资源已被注释掉。如果未执行此步骤、并且LUN访问出现问题、则可能会导致操作系统无法启动。此问题描述不会损坏数据。但是、启动到救援模式或类似模式并进行更正可能非常不方便 `/etc/fstab` 以便可以启动操作系统、以便开始故障排除工作。

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。 美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 http://www.netapp.com/TM 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。