向启用了 **NFS** 的 **SVM** 添加存储容量

ONTAP 9

NetApp
February 12, 2026

目录

向启用了 NFS 的 SVM 添加存储容量 1
了解如何向启用 ONTAP NFS 的 SVM 添加存储容量 1
创建 ONTAP NFS 导出策略 1
向 ONTAP NFS 导出策略添加规则 2
创建卷或 qtree 存储容器 7
创建 ONTAP NFS 卷 7
创建 ONTAP NFS qtree 9
使用导出策略确保 NFS 访问安全 10
了解如何使用导出策略保护 ONTAP NFS 访问 10
管理 ONTAP NFS 导出规则的处理顺序 10
将 ONTAP NFS 导出策略分配给卷 11
将 ONTAP NFS 导出策略分配给 qtree 11
验证集群中的 ONTAP NFS 客户端访问 12
从客户端系统测试 ONTAP NFS 访问 13

向启用了 NFS 的 SVM 添加存储容量

了解如何向启用 ONTAP NFS 的 SVM 添加存储容量

要向启用了 NFS 的 SVM 添加存储容量，必须创建一个卷或 qtree 以提供存储容器，并为此容器创建或修改导出策略。然后，您可以从集群验证 NFS 客户端访问，并测试客户端系统的访问。

开始之前

- 必须在 SVM 上完全设置 NFS 。
- SVM 根卷的默认导出策略必须包含允许访问所有客户端的规则。
- 必须完成对名称服务配置的所有更新。
- 必须完成对 Kerberos 配置的任何添加或修改。

创建 ONTAP NFS 导出策略

在创建导出规则之前，您必须创建一个导出策略来存放这些规则。您可以使用 `vserver export-policy create` 命令以创建导出策略。

步骤

1. 创建导出策略

   `vserver export-policy create -vserver` *`vserver_name`* `-policyname` *`policy_name`*

   策略名称最长可为 256 个字符。

2. 验证是否已创建导出策略：

   `vserver export-policy show -policyname` *`policy_name`*

示例

以下命令将在名为 vs1 的 SVM 上创建并验证是否已创建名为 exp1 的导出策略：

```
vs1::> vserver export-policy create -vserver vs1 -policyname exp1

vs1::> vserver export-policy show -policyname exp1
Vserver          Policy Name
---------------  -------------------
vs1              exp1
```

向 ONTAP NFS 导出策略添加规则

如果没有规则，导出策略将无法提供客户端对数据的访问。要创建新的导出规则，您必须标识客户端并选择客户端匹配格式，选择访问和安全类型，指定匿名用户 ID 映射，选择规则索引编号，然后选择访问协议。然后、您可以使用 `vserver export-policy rule create` 命令将新规则添加到导出策略中。

开始之前

- 要添加导出规则的导出策略必须已存在。
- 必须在数据 SVM 上正确配置 DNS ，并且 DNS 服务器必须具有适用于 NFS 客户端的正确条目。

  这是因为 ONTAP 使用数据 SVM 的 DNS 配置对某些客户端匹配格式执行 DNS 查找，如果导出策略规则匹配失败，则可能会阻止客户端数据访问。

- 如果您要使用 Kerberos 进行身份验证，则必须已确定 NFS 客户端使用以下哪种安全方法：
  - `krb5` (Kerberos V5协议)
  - `krb5i` (使用校验和进行完整性检查的Kerberos V5协议)
  - `krb5p` (具有隐私服务的Kerberos V5协议)

关于此任务

如果导出策略中的现有规则满足客户端匹配和访问要求，则无需创建新规则。

如果要使用Kerberos进行身份验证、并且SVM的所有卷都通过Kerberos进行访问、则可以设置导出规则选项 `-rorule`， `-rwrule`，和 `-superuser` 根卷的 `krb5`， `krb5i`或 `krb5p`。

步骤

1. 确定新规则的客户端和客户端匹配格式。

   。 `-clientmatch` option用于指定应用此规则的客户端。可以指定一个或多个客户端匹配值；多个值的规范必须用逗号分隔。您可以使用以下任意格式指定匹配项：

客户端匹配格式	示例
域名前面带有 "." 字符	`.example.com` 或 `.example.com,.example.net,...`
主机名	`host1` 或 `host1,host2, ...`
IPv4 地址	`10.1.12.24` 或 `10.1.12.24,10.1.12.25, ...`
IPv4 地址，子网掩码以位数表示	`10.1.12.10/4` 或 `10.1.12.10/4,10.1.12.11/4,...`

客户端匹配格式	示例
带有网络掩码的 IPv4 地址	`10.1.16.0/255.255.255.0` 或 `10.1.16.0/255.255.255.0,10.1.17.0/255.255.255.0,...`
点格式的 IPv6 地址	`::1.2.3.4` 或 `::1.2.3.4,::1.2.3.5,...`
IPv6地址、子网掩码以位数表示	`ff::00/32` 或 `ff::00/32,ff::01/32,...`
一个网络组，其网络组名称前面带有 @ 字符	`@netgroup1` 或 `@netgroup1,@netgroup2,...`

您还可以组合使用各种类型的客户端定义、例如、 `.example.com,@netgroup1`。

指定 IP 地址时，请注意以下事项：

- 不允许输入 IP 地址范围，例如 10.1.12.10-10.1.12.70 。

  此格式的条目将被解释为文本字符串，并被视为主机名。

- 在导出规则中指定单个 IP 地址以精细管理客户端访问时，请勿指定动态分配（例如 DHCP ）或临时分配（例如 IPv6 ）的 IP 地址。

  否则，当客户端的 IP 地址发生更改时，客户端将失去访问权限。

- 不允许输入带有网络掩码的 IPv6 地址，例如 ff： 12/ff：： 00 。

2. 为客户端匹配选择访问和安全类型。

   您可以为使用指定安全类型进行身份验证的客户端指定以下一种或多种访问模式：

   - `-rorule` (只读访问)
   - `-rwrule` (读写访问)
   - `-superuser` (root访问权限)

   > 只有当导出规则也允许对特定安全类型进行只读访问时，客户端才能获得该安全类型的读写访问权限。如果只读参数对于安全类型的限制性比读写参数更强，则客户端可能无法获得读写访问权限。超级用户访问也是如此。

   您可以为一个规则指定多种安全类型的逗号分隔列表。将安全类型指定为 `any` 或 `never`，请勿指定任何其他安全类型。从以下有效安全类型中进行选择：

当安全类型设置为 ...	匹配的客户端可以访问导出的数据 ...
`any`	始终，无论传入的安全类型如何。

当安全类型设置为 ...	匹配的客户端可以访问导出的数据 ...
none	如果单独列出，则具有任何安全类型的客户端将被授予匿名访问权限。如果与其他安全类型一起列出，则具有指定安全类型的客户端将被授予访问权限，而具有任何其他安全类型的客户端将被授予匿名访问权限。
never	从不，无论传入的安全类型如何。
krb5	如果通过 Kerberos 5 进行身份验证。 仅身份验证：每个请求和响应的标头都已签名。
krb5i	如果通过 Kerberos 5i 进行身份验证。 身份验证和完整性：每个请求和响应的标头和正文均已签名。
krb5p	如果使用Kerberos 5p进行身份验证。 身份验证，完整性和隐私：对每个请求和响应的标题和正文进行签名，并对 NFS 数据有效负载进行加密。
ntlm	如果通过 CIFS NTLM 进行身份验证。
sys	如果通过 NFS AUTH_SYS 进行身份验证。

建议的安全类型为 `sys`或者，如果使用`Kerberos， `krb5， krb5i`或 `krb5p。

如果要将Kerberos与NFSv3结合使用、则导出策略规则必须允许 `-rorule` 和 `-rwrule` 访问 `sys` 除了 `krb5`。这是因为需要允许 Network Lock Manager （ NLM ）访问导出。

3. 指定匿名用户 ID 映射。

。 `-anon` option用于指定映射到用户ID为0 (零)的客户端请求的UNIX用户ID或用户名、此用户ID或用户名通常与用户名root相关联。默认值为 `65534`。NFS 客户端通常会将用户 ID 65534 与用户名 nobody 相关联（也称为 *root squash* ）。在 ONTAP 中，此用户 ID 与用户 pcuser 关联。要禁止用户ID为0的任何客户端访问、请指定值 `65535`。

4. 选择规则索引顺序。

。 `-ruleindex` option用于指定规则的索引编号。规则将根据其在索引编号列表中的顺序进行评估；索引编号较低的规则将首先进行评估。例如，索引编号为 1 的规则会在索引编号为 2 的规则之前进行评估。

如果要添加 ...	那么 ...
导出策略的第一个规则	输入 … `1`。

如果要添加 ...	那么 ...
导出策略的其他规则	a. 显示策略中的现有规则： `vserver export-policy rule show -instance -policyname your_policy` b. 根据新规则的评估顺序为其选择索引编号。

5. 选择适用的NFS访问值：{nfs|nfs3|nfs4} 。

   nfs 匹配任何版本、 nfs3 和 nfs4 仅匹配这些特定版本。

6. 创建导出规则并将其添加到现有导出策略：

   ```
   vserver export-policy rule create -vserver vserver_name -policyname
   policy_name -ruleindex integer -protocol {nfs|nfs3|nfs4} -clientmatch { text |
   "text,text,…" } -rorule security_type -rwrule security_type -superuser
   security_type -anon user_ID
   ```

7. 显示导出策略的规则以验证新规则是否存在：

   ```
   vserver export-policy rule show -policyname policy_name
   ```

   命令将显示该导出策略的摘要，包括应用于该策略的规则列表。ONTAP 会为每个规则分配一个规则索引编号。知道规则索引编号后，您可以使用它显示有关指定导出规则的详细信息。

8. 验证是否已正确配置应用于导出策略的规则：

   ```
   vserver export-policy rule show -policyname policy_name -vserver vserver_name
   -ruleindex integer
   ```

示例

以下命令将在名为 RS1 的导出策略中的 SVM vs1 上创建导出规则并验证此创建过程。此规则的索引编号为 1 。此规则与域 eng.company.com 和 netgroup @netgroup1 中的任何客户端匹配。此规则将启用所有 NFS 访问。它允许使用 AUTH_SYS 进行身份验证的用户进行只读和读写访问。除非使用 Kerberos 进行身份验证，否则使用 UNIX 用户 ID 0 （零）的客户端将被匿名化。

```
vs1::> vserver export-policy rule create -vserver vs1 -policyname exp1
-ruleindex 1 -protocol nfs
-clientmatch .eng.company.com,@netgoup1 -rorule sys -rwrule sys -anon
65534 -superuser krb5

vs1::> vserver export-policy rule show -policyname nfs_policy
Virtual      Policy         Rule    Access    Client           RO
Server       Name           Index   Protocol  Match            Rule
------------ -------------- ------  --------  ---------------- ------
vs1          exp1           1       nfs       eng.company.com, sys
                                              @netgroup1

vs1::> vserver export-policy rule show -policyname exp1 -vserver vs1
-ruleindex 1

                                Vserver: vs1
                            Policy Name: exp1
                             Rule Index: 1
                          Access Protocol: nfs
Client Match Hostname, IP Address, Netgroup, or Domain:
eng.company.com,@netgroup1
                           RO Access Rule: sys
                           RW Access Rule: sys
User ID To Which Anonymous Users Are Mapped: 65534
                  Superuser Security Types: krb5
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true
```

以下命令将在名为 expol2 的导出策略中的 SVM vs2 上创建导出规则并验证此创建过程。此规则的索引编号为21。此规则会将客户端与网络组 dev_netgroup_main 中的成员匹配。此规则将启用所有 NFS 访问。它允许使用 AUTH_SYS 进行身份验证的用户进行只读访问，并要求对读写和 root 访问进行 Kerberos 身份验证。除非使用 Kerberos 进行身份验证，否则使用 UNIX 用户 ID 0 （零）的客户端将被拒绝进行 root 访问。

```
vs2::> vserver export-policy rule create -vserver vs2 -policyname expol2
-ruleindex 21 -protocol nfs
-clientmatch @dev_netgroup_main -rorule sys -rwrule krb5 -anon 65535
-superuser krb5

vs2::> vserver export-policy rule show -policyname nfs_policy
Virtual  Policy       Rule    Access    Client              RO
Server   Name         Index   Protocol  Match               Rule
-------- ------------ ------  --------  ------------------  ------
vs2      expol2       21       nfs      @dev_netgroup_main  sys

vs2::> vserver export-policy rule show -policyname expol2 -vserver vs1
-ruleindex 21

                                Vserver: vs2
                            Policy Name: expol2
                             Rule Index: 21
                          Access Protocol: nfs
Client Match Hostname, IP Address, Netgroup, or Domain:
                                         @dev_netgroup_main
                          RO Access Rule: sys
                          RW Access Rule: krb5
User ID To Which Anonymous Users Are Mapped: 65535
                  Superuser Security Types: krb5
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true
```

创建卷或 qtree 存储容器

创建 ONTAP NFS 卷

您可以使用创建卷并指定其接合点和其他属性 `volume create` 命令：

关于此任务

卷必须包含 *junction path* ，才能使其数据可供客户端使用。您可以在创建新卷时指定接合路径。如果在创建卷时未指定接合路径、则必须使用_mount_在SVM命名空间中挂载此卷 `volume mount` 命令：

开始之前

- 应设置并运行NFS。
- SVM安全模式必须为UNIX。
- 从ONTAP 9.13.1开始、您可以创建启用了容量分析和活动跟踪的卷。要启用容量或活动跟踪，请使用或 `-activity-tracking-state`将设置为 `on`发出 `volume create`命令 `-analytics-state`。

要了解有关容量分析和活动跟踪的更多信息，请参见 "启用文件系统分析"。有关的详细信息 `volume create`，请参见"ONTAP 命令参考"。

步骤

1. 创建具有接合点的卷：

   ```
   volume create -vserver svm_name -volume volume_name -aggregate aggregate_name
   -size {integer[KB|MB|GB|TB|PB]} -security-style unix -user user_name_or_number
   -group group_name_or_number -junction-path junction_path [-policy
   export_policy_name]
   ```

   的选项 `-junction-path` 包括：

   - 直接位于root下、例如、 */new_vol*

     您可以创建一个新卷并指定将其直接挂载到 SVM 根卷。

   - 在现有目录下、例如、 */existing_dir/new_vol*

     您可以创建一个新卷并指定将其挂载到现有层次结构中的现有卷，以目录的形式表示。

     例如、如果要在新目录(在新卷下的新层次结构中)中创建卷、 *`/new_dir/new_vol`* 然后，必须先创建一个与SVM根卷连接的新父卷。然后，您将在新父卷的接合路径（新目录）中创建新的子卷。

     如果您计划使用现有导出策略、则可以在创建卷时指定此策略。您也可以稍后使用添加导出策略 `volume modify` 命令：

2. 验证是否已使用所需的接合点创建卷：

   ```
   volume show -vserver svm_name -volume volume_name -junction
   ```

示例

以下命令将在 SVM vs1.example.com 和聚合 aggr1 上创建一个名为 users1 的新卷。新卷可通过访问 `/users`。此卷的大小为 750 GB ，其卷保证类型为 volume （默认值）。

```
cluster1::> volume create -vserver vs1.example.com -volume users
-aggregate aggr1 -size 750g -junction-path /users
[Job 1642] Job succeeded: Successful

cluster1::> volume show -vserver vs1.example.com -volume users -junction
                           Junction                 Junction
Vserver          Volume    Active   Junction Path   Path Source
---------------- ------- -------- --------------- -----------
vs1.example.com  users1    true     /users          RW_volume
```

以下命令会在 SVM vs1.example.com 和聚合 "aggr1" 上创建一个名为 "home4" 的新卷。目录 `/eng/` 已位于VS1 SVM的命名空间中、新卷可通过访问 `/eng/home`，将成为的主目录 `/eng/` 命名空间。此卷的大小为750 GB、其卷保证类型为 `volume` (默认情况下)。

```
cluster1::> volume create -vserver vs1.example.com -volume home4
-aggregate aggr1 -size 750g -junction-path /eng/home
[Job 1642] Job succeeded: Successful

cluster1::> volume show -vserver vs1.example.com -volume home4 -junction
                           Junction                 Junction
Vserver          Volume    Active   Junction Path   Path Source
---------------- ------- -------- --------------- -----------
vs1.example.com  home4     true     /eng/home       RW_volume
```

创建 ONTAP NFS qtree

您可以使用创建一个qtree以包含您的数据、并指定其属性 `volume qtree create` 命令：

开始之前

- 要包含新 qtree 的 SVM 和卷必须已存在。
- SVM 安全模式必须为 UNIX ，并且 NFS 应设置并运行。

步骤

1. 创建 qtree：

   ```
   volume qtree create -vserver vserver_name { -volume volume_name -qtree
   qtree_name | -qtree-path qtree path } -security-style unix [-policy
   export_policy_name]
   ```

   您可以将卷和qtree指定为单独的参数、也可以采用格式指定qtree路径参数 `/vol/volume_name/_qtree_name`。

   默认情况下， qtree 会继承其父卷的导出策略，但可以将其配置为使用自己的导出策略。如果您计划使用现有导出策略，则可以在创建 qtree 时指定该策略。您也可以稍后使用添加导出策略 `volume qtree modify` 命令：

2. 验证是否已使用所需的接合路径创建 qtree：

   ```
   volume qtree show -vserver vserver_name { -volume volume_name -qtree
   qtree_name | -qtree-path qtree path }
   ```

示例

以下示例将在SVM vs1.example.com上创建一个名为qt01的qtree、此qtree具有接合路径 `/vol/data1`：

```
cluster1::> volume qtree create -vserver vs1.example.com -qtree-path
/vol/data1/qt01 -security-style unix
[Job 1642] Job succeeded: Successful

cluster1::> volume qtree show -vserver vs1.example.com -qtree-path
/vol/data1/qt01

                      Vserver Name: vs1.example.com
                       Volume Name: data1
                        Qtree Name: qt01
  Actual (Non-Junction) Qtree Path: /vol/data1/qt01
                    Security Style: unix
                       Oplock Mode: enable
                  Unix Permissions: ---rwxr-xr-x
                          Qtree Id: 2
                      Qtree Status: normal
                     Export Policy: default
        Is Export Policy Inherited: true
```

使用导出策略确保 NFS 访问安全

了解如何使用导出策略保护 ONTAP NFS 访问

您可以使用导出策略将对卷或 qtree 的 NFS 访问限制为与特定参数匹配的客户端。配置新存储时，您可以使用现有策略和规则，向现有策略添加规则或创建新策略和规则。您还可以检查导出策略的配置

> 从 ONTAP 9.3 开始，您可以将导出策略配置检查作为后台作业来启用，以便在错误规则列表中记录任何违规。。 `vserver export-policy config-checker` 命令会调用检查程序并显示结果、您可以使用这些结果验证配置并从策略中删除错误的规则。这些命令仅验证主机名、网络组和匿名用户的导出配置。

管理 ONTAP NFS 导出规则的处理顺序

您可以使用 `vserver export-policy rule setindex` 命令以手动设置现有导出规则的索引编号。这样，您可以指定 ONTAP 将导出规则应用于客户端请求的优先级。

关于此任务

如果新索引编号已在使用中，则该命令会在指定位置插入规则并相应地对列表重新排序。

步骤

1. 修改指定导出规则的索引编号：

   ```
   vserver export-policy rule setindex -vserver virtual_server_name -policyname
   policy_name -ruleindex integer -newruleindex integer
   ```

示例

以下命令会将 SVM vs1 上名为 RS1 的导出策略中索引编号为 3 的导出规则的索引编号更改为 2：

```
vs1::> vserver export-policy rule setindex -vserver vs1
-policyname rs1 -ruleindex 3 -newruleindex 2
```

将 ONTAP NFS 导出策略分配给卷

SVM 中包含的每个卷都必须与一个导出策略相关联，该导出策略包含导出规则，客户端可以通过这些规则访问卷中的数据。

关于此任务

您可以在创建卷时或创建卷后随时将导出策略与卷关联。您可以将一个导出策略与卷关联，但一个策略可以与多个卷关联。

步骤

1. 如果在创建卷时未指定导出策略，请为此卷分配一个导出策略：

   ```
   volume modify -vserver vserver_name -volume volume_name -policy
   export_policy_name
   ```

2. 验证是否已将此策略分配给卷：

   ```
   volume show -volume volume_name -fields policy
   ```

示例

以下命令会将导出策略 nfs_policy 分配给 SVM vs1 上的卷 vol1 并验证分配情况：

```
cluster::> volume modify -v1server vs1 -volume vol1 -policy nfs_policy

cluster::>volume show -volume vol -fields policy
vserver volume       policy
------- ------------ ----------------
vs1     vol1         nfs_policy
```

将 ONTAP NFS 导出策略分配给 qtree

您还可以导出卷上的特定 qtree ，使其可供客户端直接访问，而不是导出整个卷。您可以通过为 qtree 分配导出策略来导出 qtree 。您可以在创建新 qtree 时分配导出策略，也可以通过修改现有 qtree 来分配导出策略。

开始之前

导出策略必须存在。

关于此任务

默认情况下，如果在创建时未另行指定， qtree 将继承包含卷的父导出策略。

您可以在创建 qtree 时或在创建 qtree 之后随时将导出策略与 qtree 相关联。您可以将一个导出策略与 qtree 关联，但一个策略可以与多个 qtree 关联。

步骤

1. 如果在创建 qtree 时未指定导出策略，请为此 qtree 分配一个导出策略：

   ```
   volume qtree modify -vserver vserver_name -qtree-path
   /vol/volume_name/qtree_name -export-policy export_policy_name
   ```

2. 验证是否已将此策略分配给 qtree：

   ```
   volume qtree show -qtree qtree_name -fields export-policy
   ```

示例

以下命令会将导出策略 nfs_policy 分配给 SVM vs1 上的 qtree qt1 并验证分配情况：

```
cluster::> volume modify -v1server vs1 -qtree-path /vol/vol1/qt1 -policy
nfs_policy

cluster::>volume qtree show -volume vol1 -fields export-policy
vserver volume qtree export-policy
------- ------ ----- -------------
vs1     data1  qt01  nfs_policy
```

验证集群中的 ONTAP NFS 客户端访问

您可以通过在 UNIX 管理主机上设置 UNIX 文件权限来为选定客户端授予对共享的访问权限。您可以使用检查客户端访问 `vserver export-policy check-access` 命令、根据需要调整导出规则。

步骤

1. 在集群上、使用检查客户端对导出的访问权限 `vserver export-policy check-access` 命令：

   以下命令将检查 IP 地址为 1.2.3.4 的 NFSv3 客户端对卷 Home2 的读 / 写访问权限。命令输出显示卷使用导出策略 `exp-home-dir` 而且访问被拒绝。

```
cluster1::> vserver export-policy check-access -vserver vs1 -client-ip
1.2.3.4 -volume home2 -authentication-method sys -protocol nfs3 -access
-type read-write
                                        Policy    Policy     Rule
Path                    Policy          Owner     Owner Type Index  Access
----------------------- --------------- --------- ---------- ------
----------
/                       default         vs1_root  volume          1 read
/eng                    default         vs1_root  volume          1 read
/eng/home2              exp-home-dir    home2     volume          1 denied

3 entries were displayed.
```

2. 检查输出以确定导出策略是否按预期工作以及客户端访问是否按预期进行。

   具体而言，您应验证卷或 qtree 使用的导出策略以及客户端因此具有的访问类型。

3. 如有必要，请重新配置导出策略规则。

从客户端系统测试 ONTAP NFS 访问

在验证对新存储对象的 NFS 访问之后，您应登录到 NFS 管理主机并从 SVM 读取数据并向 SVM 写入数据来测试配置。然后，您应在客户端系统上以非 root 用户身份重复此过程。

开始之前

- 客户端系统必须具有先前指定的导出规则允许的 IP 地址。
- 您必须具有 root 用户的登录信息。

步骤

1. 在集群上，验证托管新卷的 LIF 的 IP 地址：

   `network interface show -vserver` *`svm_name`*

   有关的详细信息 `network interface show`，请参见"ONTAP 命令参考"。

2. 以 root 用户身份登录到管理主机客户端系统。
3. 将目录更改为挂载文件夹：

   `cd /mnt/`

4. 使用 SVM 的 IP 地址创建并挂载新文件夹：

   a. 创建新文件夹：
      `mkdir /mnt/`*`folder`*

   b. 将新卷挂载到此新目录：

```
mount -t nfs -o hard IPAddress:/volume_name /mnt/folder
```

c. 将目录更改为新文件夹：
```
cd folder
```

以下命令将创建一个名为 test1 的文件夹，并在 test1 挂载文件夹的 192.0.2.130 IP 地址处挂载 vol1 卷，然后更改为新的 test1 目录：

```
host# mkdir /mnt/test1
host# mount -t nfs -o hard 192.0.2.130:/vol1 /mnt/test1
host# cd /mnt/test1
```

5. 创建一个新文件，验证该文件是否存在并向其写入文本：

a. 创建测试文件：
```
touch filename
```

b. 验证文件是否存在：
```
ls -l filename
```

c. 输入 ...
```
cat > filename
```

键入一些文本，然后按 Ctrl+D 将文本写入测试文件。

d. 显示测试文件的内容。
```
cat filename
```

e. 删除测试文件：
```
rm filename
```

f. 返回到父目录：
```
cd ..
```

```
host# touch myfile1
host# ls -l myfile1
-rw-r--r-- 1 root root 0 Sep 18 15:58 myfile1
host# cat >myfile1
This text inside the first file
host# cat myfile1
This text inside the first file
host# rm -r myfile1
host# cd ..
```

6. 以 root 用户身份，在挂载的卷上设置所需的任何 UNIX 所有权和权限。

7. 在导出规则中标识的 UNIX 客户端系统上，以现在有权访问新卷的授权用户之一身份登录，然后重复步骤 3 至 5 中的过程，以验证是否可以挂载卷并创建文件。