# NFS対応SVMにストレージ容量を追加する

ONTAP 9

NetApp
December 20, 2024

# 目次

# NFS対応SVMにストレージ容量を追加する

## NFS対応SVMへのストレージ容量の追加の概要

NFS 対応 SVM にストレージ容量を追加するには、ストレージコンテナを提供するボリュームまたは qtree を作成し、そのコンテナのエクスポートポリシーを作成または変更する必要があります。その後、クラスタからの NFS クライアントアクセスを確認し、クライアントシステムからのアクセスをテストできます。

必要なもの

- SVMでNFSの設定が完了している必要があります。
- SVM ルートボリュームのデフォルトのエクスポートポリシーに、すべてのクライアントへのアクセスを許可するルールが含まれている必要があります。
- ネームサービス設定に対する更新が完了している必要があります。
- Kerberos 設定への追加または変更が完了している必要があります。

## エクスポートポリシーを作成する

エクスポートルールを作成する前に、それらを保持するエクスポートポリシーを作成する必要があります。エクスポートポリシーは、コマンドを使用して作成できます `vserver export-policy create`。

手順

1. エクスポートポリシーを作成します。

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

   ポリシー名の最大文字数は256文字です。

2. エクスポートポリシーが作成されたことを確認します。

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

例

次のコマンドは、 vs1 という SVM で、 exp1 という名前のエクスポートポリシーを作成し、作成を確認します。

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

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

# エクスポートポリシーにルールを追加する

エクスポートポリシーにルールがないと、クライアントはデータにアクセスできません。新しいエクスポートルールを作成するには、クライアントを特定してクライアント照合形式を選択し、アクセスとセキュリティのタイプを選択し、匿名ユーザIDマッピングを指定し、ルールインデックス番号を選択して、アクセスプロトコルを選択する必要があります。その後、コマンドを使用して、新しいルールをエクスポートポリシーに追加できます `vserver export-policy rule create`。

必要なもの

- エクスポートルールを追加するエクスポートポリシーを用意しておく必要があります。
- データ SVM で DNS が正しく設定されている必要があり、 DNS サーバに NFS クライアント用の正しいエントリが存在する必要があります。

  その理由は、特定のクライアント照合形式で ONTAP がデータ SVM の DNS 設定を使用して DNS ルックアップを実行することと、エクスポートポリシールールの照合が失敗するとクライアントがデータにアクセスできなくなる可能性があることです。

- Kerberosで認証する場合は、NFSクライアントで次のいずれのセキュリティ方式が使用されているかを確認しておく必要があります。
  - `krb5`（Kerberos v5プロトコル）
  - `krb5i`（Kerberos v5プロトコルとチェックサムによる整合性チェック）
  - `krb5p`（Kerberos v5プロトコルとプライバシーサービス）

タスクの内容

エクスポートポリシーの既存のルールがクライアント一致とアクセスの要件を満たしている場合は、新しいルールを作成する必要はありません。

Kerberosで認証する場合に、SVMのすべてのボリュームにKerberos経由でアクセスできる場合は `-superuser`、 `krb5i`ルートボリュームのエクスポートルールオプション、 `-rwrule`、、を、または `krb5p`に `krb5`設定できます `-rorule`。

手順

1. 新しいルールのクライアントとクライアント照合形式を特定します。

   オプションは `-clientmatch`、ルールを適用するクライアントを指定します。クライアント一致の値は1つまたは複数指定できます。複数の値を指定する場合はカンマで区切る必要があります。次のいずれかの形式で指定できます。

| クライアント照合形式 | 例 |
| --- | --- |
| 先頭に文字が付いたドメイン名 | `.example.com`または `.example.com,.example.net,...` |
| ホスト名 | `host1`または `host1,host2, ...` |

| クライアント照合形式 | 例 |
| --- | --- |
| IPv4アドレス | `10.1.12.24`または `10.1.12.24,10.1.12.25, ...` |
| サブネット マスクをビット数で表したIPv4アドレス | `10.1.12.10/4`または `10.1.12.10/4,10.1.12.11/4,...` |
| IPv4アドレスとネットワークマスク | `10.1.16.0/255.255.255.0`または `10.1.16.0/255.255.255.0,10.1.17.0/255.255.255.0,...` |
| ドット付き形式のIPv6アドレス | `::1.2.3.4`または `::1.2.3.4,::1.2.3.5,...` |
| サブネットマスクをビット数で表したIPv6アドレス | `ff::00/32`または `ff::00/32,ff::01/32,...` |
| 先頭に@文字が付いた単一のネットグループ | `@netgroup1`または `@netgroup1,@netgroup2,...` |

クライアント定義のタイプを組み合わせることもできます（例：） `.example.com,@netgroup1`。

IPアドレスを指定する場合は、次の点に注意してください。

- 10.1.12.10-10.1.12.70などのIPアドレス範囲を入力することはできません。

  この形式のエントリはテキスト文字列と解釈され、ホスト名として扱われます。

- クライアントアクセスのきめ細かな管理のためにエクスポートルールで個々の IP アドレスを指定する際には、動的（ DHCP など）または一時的（ IPv6 など）に割り当てられている IP アドレスを指定しないでください。

  そうしないと、IPアドレスが変更されると、クライアントはアクセスを失います。

- ff ： 12/ff ： 00 のように、 IPv6 アドレスとネットワークマスクを入力することはできません。

2. クライアント一致のアクセスタイプとセキュリティタイプを選択します。

   指定したセキュリティタイプで認証するクライアントには、次のアクセスモードを1つ以上指定できます。

   - `-rorule`（読み取り専用アクセス）
   - `-rwrule`（読み取り/書き込みアクセス）
   - `-superuser`（ルートアクセス）

特定のセキュリティタイプの読み取り/書き込みアクセスは、エクスポートルールでそのセキュリティタイプの読み取り専用アクセスも許可されている場合にのみ許可されます。読み取り専用パラメータで読み取り/書き込みパラメータよりも限定的なセキュリティタイプを指定すると、クライアントに対して読み取り/書き込みアクセスが許可されない可能性があります。スーパーユーザアクセスについても同様です。

1つのルールに対して複数のセキュリティタイプをカンマで区切って指定できます。セキュリティタイプとしてまたはを `never`指定する場合は `any`、他のセキュリティタイプは指定しないでください。次の有効なセキュリティタイプから選択します。

| セキュリティタイプの設定 | 一致するクライアントからエクスポートされたデータへのアクセス |
|---|---|
| `any` | 受信セキュリティタイプに関係なく、常に。 |
| `none` | 単独で指定した場合、どのセキュリティタイプのクライアントにも匿名アクセスが許可されます。他のセキュリティタイプと一緒に指定すると、指定したセキュリティタイプのクライアントにアクセスが許可され、それ以外のセキュリティタイプのクライアントには匿名アクセスが許可されます。 |
| `never` | 受信セキュリティタイプに関係なく、なし。 |
| `krb5` | Kerberos 5によって認証されます。認証のみ：各要求および応答のヘッダーが署名されます。 |
| `krb5i` | Kerberos 5iによって認証されます。認証および整合性：各要求および応答のヘッダーと本文が署名されます。 |
| `krb5p` | Kerberos 5pによって認証されます。認証、整合性、およびプライバシー：各要求および応答のヘッダーと本文が署名され、 NFS データペイロードが暗号化されます。 |
| `ntlm` | CIFS NTLMによって認証されます。 |
| `sys` | NFS AUTH_SYSで認証されます。 |

推奨されるセキュリティタイプは `sys`、または（Kerberosを使用する場合） `krb5`、、 `krb5i`、、または `krb5p`です。

NFSv3でKerberosを使用している場合は `-rwrule`、に加えて `krb5`エクスポートポリシールールでアクセスを `sys`許可する必要があります `-rorule`。これは、Network Lock Manager（NLM）によるエクスポートへのアクセスを許可する必要があるためです。

3. 匿名ユーザIDマッピングを指定します。

`-anon`オプションは、ユーザIDが0（ゼロ）で到着するクライアント要求にマッピングされるUNIXユーザIDまたはユーザ名を指定します。このユーザIDは通常ユーザ名rootに関連付けられています。デフォルト値はです`65534`。NFS クライアントは通常、ユーザ ID 65534 をユーザ名 nobody と関連付けます（ _root squashing_ ） 。ONTAPでは、このユーザIDはユーザpcuserに関連付けられています。ユーザIDが0のクライアントからのアクセスを無効にするには、の値を指定し `65535`ます。

4. ルールインデックスの順序を選択します。

   オプションは -ruleindex、ルールのインデックス番号を指定します。ルールはインデックス番号のリスト内の順序に従って評価され、インデックス番号が小さいルールが最初に評価されます。たとえば、インデックス番号が1のルールは、インデックス番号が2のルールよりも先に評価されます。

| 追加対象 | そしたら...。 |
| --- | --- |
| エクスポートポリシーへの最初のルール | と入力し`1`ます。 |
| 追加のルールをエクスポートポリシーに | a. ポリシー内の既存のルールを表示します。+ `vserver export-policy rule show -instance -policyname your_policy`<br>b. 評価する順序に応じて、新しいルールのインデックス番号を選択します。 |

5. 該当するNFSアクセス値を選択します{nfs|nfs3|nfs4：}。

   nfs`任意のバージョンに一致し `nfs3、 `nfs4`特定のバージョンだけに一致します。

6. エクスポートルールを作成して既存のエクスポートポリシーに追加します。

   ```
   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. エクスポートポリシーのルールを表示して、新しいルールが存在することを確認します。

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

   このコマンドは、エクスポートポリシーに適用されているルールのリストを含む、エクスポートポリシーの概要を表示します。ONTAPは、各ルールにルールインデックス番号を割り当てます。ルールインデックス番号を確認したら、その番号を使用して、指定したエクスポートルールに関する詳細情報を表示できます。

8. エクスポートポリシーに適用されたルールが正しく設定されていることを確認します。

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

例

次のコマンドは、 rs1 というエクスポートポリシーで vs1 という SVM に対するエクスポートルールを作成し、作成を確認します。このルールのインデックス番号は1です。このルールは、ドメインeng.company.comおよびネットグループ@netgroup1内のすべてのクライアントに一致します。このルールは、すべてのNFSアクセスを有効にします。AUTH_SYSで認証されたユーザに対する読み取り専用アクセスと読み取り/書き込みアクセスを有効にします。UNIXユーザIDが0（ゼロ）のクライアントは、Kerberosで認証されないかぎり匿名化されます。

```
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
```

次のコマンドは、 expol2 というエクスポートポリシーで vs2 という SVM に対するエクスポートルールを作成し、作成を確認します。このルールのインデックス番号は21です。このルールは、クライアントをネットグループdev_netgroup_mainのメンバーと照合します。このルールは、すべてのNFSアクセスを有効にします。AUTH_SYSで認証されたユーザの読み取り専用アクセスを有効にし、読み取り/書き込みアクセスとrootアクセスにはKerberos認証を必要とします。UNIXユーザIDが0（ゼロ）のクライアントは、Kerberos以外で認証されないかぎり、ルートアクセスを拒否されます。

```
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
```

# ボリュームまたは**qtree**のストレージコンテナを作成する

## ボリュームの作成

## コマンドを使用すると、ボリュームを作成し、ジャンクションポイントやその他のプロパティを指定できます `volume create`。

### タスクの内容

クライアントがデータを使用できるようにするには、ボリュームに *junction path* を含める必要があります。ジャンクションパスは、新しいボリュームの作成時に指定できます。ジャンクションパスを指定せずにボリュームを作成する場合は、コマンドを使用して、SVMネームスペースでボリュームを_mount_theにする必要があります `volume mount`。

### 開始する前に

- NFSがセットアップされ、実行されている必要があります。
- SVMのセキュリティ形式がUNIXである必要があります。
- ONTAP 9.13.1以降では、容量分析とアクティビティ追跡を有効にしてボリュームを作成できます。容量

またはアクティビティの追跡を有効にするには、を指定してコマンドを `-analytics-state`実行する `volume create`か、 `-activity-tracking-state`に設定します `on`。

容量分析とアクティビティ追跡の詳細については、を参照してください "ファイルシステム分析を有効にする"。

手順

1. ジャンクションポイントを設定してボリュームを作成します。

   `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`*`]`

   の選択肢は `-junction-path`次のとおりです。

   - ルートの直下。例： */new_vol*

     新しいボリュームを作成し、 SVM のルートボリュームに直接マウントされるように指定することができます。

   - 既存のディレクトリの下（例： */existing_dir/new_vol*

     新しいボリュームを作成し、ディレクトリとして表現されている既存のボリューム（既存の階層内）にマウントされるように指定できます。

     たとえば、新しいディレクトリ（新しいボリュームの下の新しい階層）にボリュームを作成する場合は */new_dir/new_vol*、SVMのルートボリュームにジャンクションされている新しい親ボリュームを最初に作成する必要があります。その後、新しい親ボリューム（新しいディレクトリ）のジャンクションパスに新しい子ボリュームを作成します。

   + 既存のエクスポートポリシーを使用する場合は、ボリュームの作成時に指定できます。エクスポートポリシーは、あとからコマンドを使用して追加することもできます `volume modify`。

2. 目的のジャンクションポイントでボリュームが作成されたことを確認します。

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

例

次のコマンドは、 SVM vs1.example.com およびアグリゲート aggr1 上に、 users1 という名前の新しいボリュームを作成します。新しいボリュームは、で使用でき `/users`ます。ボリュームのサイズは750GBで、ボリュームギャランティのタイプはvolume（デフォルト）です。

```
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
```

次のコマンドは、SVM「vs1.example.com」とアグリゲート「aggr1」に「home4」という名前の新しいボリュームを作成します。ディレクトリは `/eng/`vs1 SVM`のネームスペース内にすでに存在し、新しいボリュームがで使用可能になります `/eng/home`。これがネームスペースのホームディレクトリになります。`/eng/`ボリュームのサイズは`750GB`で、ボリュームギャランティのタイプは（デフォルト）です `volume`。

```
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
```

## qtreeを作成する

コマンドを使用すると、データを含むqtreeを作成し、そのプロパティを指定できます `volume qtree create`。

必要なもの

- SVM と新しい qtree を格納するボリュームがすでに存在している必要があります。
- SVMのセキュリティ形式がUNIXで、NFSが設定されて実行されている必要があります。

手順

1. qtree を作成します。

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

   ボリュームとqtreeを別 々 の引数として指定するか、の形式でqtreeパスの引数を指定できます `/vol/volume_name/_qtree_name`。

デフォルトでは、 qtree は親ボリュームのエクスポートポリシーを継承しますが、独自のものを使用するように設定することもできます。既存のエクスポートポリシーを使用する場合は、 qtree の作成時にポリシーを指定できます。エクスポートポリシーは、あとからコマンドを使用して追加することもできます `volume qtree modify`。

2. qtree が必要なジャンクションパスで作成されたことを確認します。

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

例

次の例は、ジャンクションパスがであるSVM vs1.example.com上に、qt01という名前のqtreeを作成し`/vol/data1`ます。

```
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
```

# エクスポート ポリシーを使用したNFSアクセスの保護

## エクスポート ポリシーを使用したNFSアクセスの保護

エクスポートポリシーを使用すると、ボリュームまたはqtreeへのNFSアクセスを、特定のパラメータに一致するクライアントだけに制限できます。新しいストレージをプロビジョニングする際に、既存のポリシーとルールを使用するか、既存のポリシーにルールを追加するか、新しいポリシーとルールを作成できます。エクスポートポリシーの設定も確認できます。

ONTAP 9 .3以降では、エクスポートポリシーの設定チェックをバックグラウンドジョブとして有効にして、すべてのルール違反をエラールールリストに記録できます。 `vserver export-policy config-checker`コマンドはチェッカーを呼び出して結果を表示します。この結果を使用して、設定を検証し、エラーのあるルールをポリシーから削除できます。このコマンドで検証されるのは、ホスト名、ネットグループ、匿名ユーザのエクスポート設定のみです。

## エクスポートルールの処理順序を管理します。

コマンドを使用すると、既存のエクスポートルールのインデックス番号を手動で設定できます `vserver export-policy rule setindex`。これにより、 ONTAP がクライアント要求に対してエクスポートルールを適用する優先順位を指定できます。

タスクの内容

新しいインデックス番号がすでに使用されている場合は、指定した場所にルールが挿入され、それに応じてリストの順序が変更されます。

ステップ

1. 指定したエクスポートルールのインデックス番号を変更します。

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

例

次のコマンドは、 vs1 という SVM の rs1 というエクスポートポリシーのインデックス番号を 3 から 2 に変更します。

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

## ボリュームへのエクスポートポリシーの割り当て

SVM内の各ボリュームには、クライアントがボリューム内のデータにアクセスできるように、エクスポートルールを含むエクスポートポリシーを関連付ける必要があります。

タスクの内容

エクスポートポリシーは、ボリュームの作成時、またはボリュームの作成後にいつでも、ボリュームに関連付けることができます。1 つのボリュームに関連付けることができるのは 1 つのエクスポートポリシーだけですが、 1 つのポリシーを多数のボリュームに関連付けることができます。

手順

1. ボリュームの作成時にエクスポートポリシーを指定しなかった場合は、ボリュームにエクスポートポリシーを割り当てます。

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

2. ポリシーがボリュームに割り当てられたことを確認します。

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

例

次のコマンドは、エクスポートポリシー nfs_policy を vs1 という SVM 上のボリューム vol1 に割り当てて、割り当てを確認します。

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

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

## qtreeへのエクスポートポリシーの割り当て

ボリューム全体をエクスポートする代わりに、ボリュームの特定の qtree をエクスポートしてクライアントから直接アクセスできるようにすることもできます。qtree をエクスポートするには、 qtree にエクスポートポリシーを割り当てます。エクスポートポリシーの割り当ては、新しい qtree の作成時に行うことも、既存の qtree の変更によって行うこともできます。

必要なもの

エクスポートポリシーが存在している必要があります。

タスクの内容

qtree では、作成時に指定しなかった場合、格納先ボリュームの親のエクスポートポリシーがデフォルトで継承されます。

エクスポートポリシーは、 qtree の作成時、または qtree の作成後にいつでも、 qtree に関連付けることができます。1 つの qtree に関連付けることができるのは 1 つのエクスポートポリシーだけですが、 1 つのポリシーを多数の qtree と関連付けることができます。

手順

1. qtree の作成時にエクスポートポリシーを指定しなかった場合は、 qtree にエクスポートポリシーを割り当てます。

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

2. ポリシーが qtree に割り当てられたことを確認します。

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

例

次のコマンドは、エクスポートポリシー nfs_policy を vs1 という SVM 上の qtree qt1 に割り当てて、割り当てを確認します。

```
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
```

# クラスタからのNFSクライアントアクセスの確認

UNIX 管理ホストで UNIX ファイル権限を設定することにより、選択したクライアントに共有へのアクセスを許可できます。クライアントアクセスを確認するには、コマンドを使用し `vserver export-policy check-access`、必要に応じてエクスポートルールを調整します。

手順

1. クラスタで、コマンドを使用してエクスポートへのクライアントアクセスを確認します `vserver export-policy check-access`。

   次のコマンドは、 IP アドレスが 1.2.3.4 の NFSv3 クライアントによるボリューム home2 への読み取り / 書き込みアクセスをチェックします。コマンド出力には、ボリュームでエクスポートポリシーが使用されていること、およびアクセスが拒否されたことが示されています `exp-home-dir`。

   ```
   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. 出力を確認して、エクスポートポリシーが意図したとおりに機能してクライアントアクセスが想定どおりに動作しているかどうかを判断します。

   具体的には、ボリュームまたは qtree によって使用されたエクスポートポリシーと、結果としてクライアントが行ったアクセスのタイプを確認する必要があります。

3. 必要に応じて、エクスポートポリシールールを再設定します。

# クライアントシステムからの**NFS**アクセスをテストする

新しいストレージオブジェクトに対する NFS アクセスの確認が完了したら、設定をテストする必要があります。設定をテストするには、 NFS 管理ホストにログインし、 SVM に対するデータの読み取りと書き込みが可能かどうかを確認します。その後、 root 以外のユーザとしてクライアントシステム上で処理を繰り返します。

必要なもの

- クライアントシステムに、前に指定したエクスポートルールで許可されている IP アドレスが割り当てられている必要があります。
- root ユーザのログイン情報が必要です。

手順

1. クラスタで、新しいボリュームをホストしている LIF の IP アドレスを確認します。

   `network interface show –vserver svm_name`

2. 管理ホストクライアントシステムに root ユーザとしてログインします。
3. ディレクトリをマウントフォルダに変更します。

   `cd /mnt/`

4. 新しいフォルダを作成し、 SVM の IP アドレスを使用してマウントします。

   a. 新しいフォルダの作成：+ `mkdir /mnt/folder`

   b. この新しいディレクトリに新しいボリュームをマウントします。+ `mount -t nfs -o hard IPAddress:/volume_name /mnt/folder`

   c. ディレクトリを新しいフォルダに変更します。+ `cd folder`

   次のコマンドでは、 test1 という名前のフォルダを作成し、 IP アドレス 192.0.2.130 のボリューム vol1 をマウントフォルダ test1 にマウントして、ディレクトリを新しい test1 に変更しています。

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

5. 新しいファイルを作成し、そのファイルが存在することを確認して、テキストを書き込みます。

   a. テストファイルを作成します。+ `touch filename`

   b. ファイルが存在することを確認します。:+ `ls -l filename`

   c. 入力：+ `cat > filename`

   テキストを入力してから Ctrl+D を押してテストファイルにテキストを書き込みます。

d. テストファイルの内容を表示します。+ `cat *filename*`

e. テストファイルを削除します。+ `rm *filename*`

f. 親ディレクトリに戻ります。+ `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. root として、マウントされたボリュームに対する必要な UNIX の所有権と権限を設定します。

7. エクスポートルールで特定されている UNIX クライアントシステムで、新しいボリュームへのアクセス権を持つ許可されたユーザとしてログインし、手順 3 ~ 5 を繰り返して、ボリュームのマウントとファイルの作成が可能なことを確認します。

## 著作権に関する情報



## 商標に関する情報