



Ubuntu

ONTAP SAN Host Utilities

NetApp
January 30, 2026

目次

Ubuntu	1
ONTAPストレージを使用してUbuntu 24.04をFCPおよびiSCSI用に設定する	1
手順1：必要に応じてSANブートを有効にします。	1
手順2：Linux Host Utilitiesをインストールする	1
手順3：ホストのマルチパス構成を確認する	1
ステップ4：ホストのiSCSI構成を確認する	4
ステップ5：オプションでデバイスをマルチパスから除外する	7
ステップ6：ONTAP LUNのマルチパスパラメータをカスタマイズする	8
ステップ7：既知の問題を確認する	9
次の手順	9
ONTAPストレージを使用してUbuntu 22.04をFCPおよびiSCSI用に設定する	9
手順1：必要に応じてSANブートを有効にします。	9
手順2：Linux Host Utilitiesをインストールする	10
手順3：ホストのマルチパス構成を確認する	10
ステップ4：ホストのiSCSI構成を確認する	12
ステップ5：オプションでデバイスをマルチパスから除外する	15
ステップ6：ONTAP LUNのマルチパスパラメータをカスタマイズする	16
ステップ7：既知の問題を確認する	17
次の手順	17
ONTAPストレージを使用してUbuntu 20.04をFCPおよびiSCSI用に設定する	17
手順1：必要に応じてSANブートを有効にします。	17
ステップ2: ホストのマルチパス構成を確認する	18
ステップ3：ホストのiSCSI構成を確認する	20
手順4：必要に応じて、マルチパスからデバイスを除外する	23
手順5：ONTAP LUNのマルチパスパラメータをカスタマイズする	24
手順6：既知の問題を確認する	25
次の手順	25

Ubuntu

ONTAPストレージを使用してUbuntu 24.04をFCPおよびiSCSI用に設定する

Linux ホスト ユーティリティ ソフトウェアは、ONTAPストレージに接続された Linux ホスト用の管理および診断ツールを提供します。Ubuntu 24.04 ホストに Linux ホスト ユーティリティをインストールすると、ホスト ユーティリティを使用してONTAP LUN での FCP および iSCSI プロトコル操作を管理できるようになります。

手順1：必要に応じてSANブートを有効にします。

SANブートを使用するようにホストを設定することで、導入を簡易化し、拡張性を向上させることができます。

開始する前に

を使用["Interoperability Matrix Tool"](#)して、Linux OS、ホストバスアダプタ（HBA）、HBAファームウェア、HBAブートBIOS、およびONTAPバージョンがSANブートをサポートしていることを確認します。

手順

1. ["SANブートLUNを作成し、ホストにマップする"](#)です。
2. SAN ブート LUN がマッピングされているポートに対して、サーバ BIOS で SAN ブートを有効にします。

HBA BIOS を有効にする方法については、ベンダー固有のマニュアルを参照してください。

3. 構成が正常に完了したことを確認するために、ホストをリブートし、OSが稼働していることを確認します。

手順2：Linux Host Utilitiesをインストールする

NetAppでは、ONTAP LUN管理をサポートし、テクニカルサポートによる設定データの収集を支援するために、Linux Host Utilitiesをインストールすることを強く推奨しています。

["Linuxホストユーティリティ8.0をインストールする"](#)。



Linux Host Utilitiesをインストールしても、Linuxホストのホストタイムアウト設定は変更されません。

手順3：ホストのマルチパス構成を確認する

Ubuntu 24.04 でマルチパスを使用してONTAP LUN を管理できます。

ホストでマルチパスが正しく設定されていることを確認するには、ファイルが定義されていること、およびONTAP LUN用にNetAppの推奨設定が設定されていることを確認し `/etc/multipath.conf` ます。

手順

1. ファイルが終了することを確認し `/etc/multipath.conf` ます。ファイルが存在しない場合は、空のゼロバイトファイルを作成します。

```
touch /etc/multipath.conf
```

2. ファイルの初回作成時には `multipath.conf`、マルチパスサービスを有効にして開始し、推奨設定をロードしなければならない場合があります。

```
systemctl enable multipathd
```

```
systemctl start multipathd
```

3. ホストをブートするたびに、空のゼロバイトファイルによって `/etc/multipath.conf`、NetApp推奨のホストマルチパスパラメータがデフォルト設定として自動的にロードされます。オペレーティングシステムは、ONTAP LUNを正しく認識および管理するマルチパスパラメータでコンパイルされているため、ホスト用のファイルを変更する必要はありません `/etc/multipath.conf`。

次の表に、Linux OS標準でコンパイルされたONTAP LUNのマルチパスパラメータの設定を示します。

パラメータ設定の表示

パラメータ	設定
detect_prio	はい。
DEV_DETION_TMO	" 無限 "
フェイルバック	即時
fast_io_fail_TMO	5.
の機能	"2 pg_init_retries 50"
flush_on_last_del	はい。
hardware_handler	0
パスの再試行なし	キュー
path_checker です	" tur "
path_grouping_policy	「 group_by_prio 」
path_selector	"service-time 0"
polling_interval （ポーリング間隔）	5.
Prio	ONTAP
プロダクト	LUN
retain_attached_hw_handler	はい。
RR_weight を指定します	" 均一 "
ユーザーフレンドリ名	いいえ
ベンダー	ネットアップ

4. ONTAP LUNのパラメータ設定とパスステータスを確認します。

```
multipath -ll
```

デフォルトのマルチパス パラメータは、ASA、AFF、およびFAS構成をサポートします。これらの構成では、単一のONTAP LUN に 4 つを超えるパスは必要ありません。パスが 4 つを超えると、ストレージ障害時に問題が発生する可能性があります。

次の出力例は、ASA、AFF、またはFAS構成のONTAP LUNについて、正しいパラメータ設定とパスステータスを示しています。

ASA構成

ASA構成では、特定のLUNへのすべてのパスが最適化され、アクティブな状態が維持されます。これにより、すべてのパスを同時に経由するI/O処理が行われるため、パフォーマンスが向上します。

例を示します

```
# multipath -ll
3600a098038314559533f524d6c652f62 dm-24 NETAPP,LUN C-Mode
size=10G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
`-+- policy='service-time 0' prio=50 status=active
| - 11:0:1:13 sdm 8:192 active ready running
| - 11:0:3:13 sdah 66:16 active ready running
| - 12:0:1:13 sdbc 67:96 active ready running
`- 12:0:3:13 sdbx 68:176 active ready running
```

AFFまたはFASの設定

AFFまたはFAS構成には、優先度の高いパスと低いパスの2つのグループを設定する必要があります。優先度の高いアクティブ/最適化パスは、アグリゲートが配置されているコントローラで処理されます。優先度の低いパスはアクティブですが、別のコントローラで処理されるため最適化されていません。最適化されていないパスは、最適化されたパスを使用できない場合にのみ使用されます。

次の例は、2つのアクティブ/最適化パスと2つのアクティブ/非最適化パスがあるONTAP LUNの出力を示しています。

例を示します

```
# multipath -ll
3600a098038314837352453694b542f4a dm-0 NETAPP,LUN C-Mode
size=160G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| | - 14:0:3:0 sdbk 67:224 active ready running
| | - 15:0:2:0 sdbl 67:240 active ready running
`+- policy='service-time 0' prio=10 status=enabled
| - 14:0:0:0 sda 8:0 active ready running
`- 15:0:1:0 sdv 65:80 active ready running
```

ステップ4：ホストのiSCSI構成を確認する

ホストに対して iSCSI が正しく構成されていることを確認します。

このタスクについて

iSCSI ホストで次の手順を実行します。

手順

1. iSCSI イニシエーター パッケージ (open-iscsi) がインストールされていることを確認します：

```
$apt list |grep open-iscsi
```

次の例のような出力が表示されます。

```
open-iscsi/noble-updates,noble-updates,now 2.1.9-3ubuntu5.4 amd64
```

2. `/etc/iscsi/initiatorname.iscsi` ファイルにあるiSCSIイニシエーターノード名を確認します：

```
InitiatorName=iqn.YYYY-MM.com.<vendor>:<host_name>
```

3. `/etc/iscsi/iscsid.conf` ファイルにあるiSCSIセッションタイムアウトパラメータを設定します：

```
node.session.timeo.replacement_timeout = 5
```

iSCSI `replacement_timeout` パラメータは、タイムアウトしたパスまたはセッションが再確立されるまでのiSCSIレイヤーの待機時間を制御します。この時間が経過すると、そのパスまたはセッションに対するコマンドは失敗します。iSCSI設定ファイルで `replacement_timeout` の値を5に設定する必要があります。

4. iSCSIサービスを有効にします：

```
$systemctl enable iscsid
```

5. iSCSIサービスを開始します：

```
$systemctl start iscsid
```

6. iSCSIサービスが実行されていることを確認します：

```
$systemctl status iscsid
```

例を示します

```
●iscsid.service - iSCSI initiator daemon (iscsid)
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service;
   enabled; preset: disabled)
   Active: active (running) since Mon 2026-01-12 12:53:18 IST; 2
   days ago
   TriggeredBy: ● iscsid.socket
     Docs: man:iscsid(8)
    Main PID: 1127419 (iscsid)
      Tasks: 2 (limit: 76557)
     Memory: 4.3M (peak: 8.8M)
        CPU: 1.657s
     CGroup: /system.slice/iscsid.service
            └─1127418 /usr/sbin/iscsid
            └─1127419 /usr/sbin/iscsid
```

7. iSCSIターゲットを検出します：

```
$iscsiadm --mode discovery --op update --type sendtargets --portal
<target_IP>
```

例を表示

```
iscsiadm --mode discovery --op update --type sendtargets --portal
192.168.100.197
192.168.100.197:3260,1046 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
192.168.200.199:3260,1049 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
192.168.100.199:3260,1048 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
192.168.200.197:3260,1047 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
```

8. ターゲットにログインします：

```
$iscsiadm --mode node -l all
```

9. ホストの起動時に iSCSI が自動的にログインするように設定します：


```
$iscsiadm --mode node -T <target_name> -p <ip:port> -o update -n  
node.startup -v automatic
```

次の例のような出力が表示されます。

```
iscsiadm --mode node -T iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 -p  
192.168.100.197:3260 -o update -n node.startup -v automatic
```

10. iSCSIセッションを確認します：

```
$iscsiadm --mode session
```

例を示します

```
iscsiadm --mode session  
tcp: [1] 192.168.200.197:3260,1047 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)  
tcp: [2] 192.168.100.197:3260,1046 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)  
tcp: [3] 192.168.100.199:3260,1048 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)  
tcp: [4] 192.168.200.199:3260,1049 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)
```

ステップ5：オプションでデバイスをマルチパスから除外する

必要に応じて、不要なデバイスのWWIDをファイルの「blacklist」スタンザに追加することで、デバイスをマルチパスから除外できます `multipath.conf`。

手順

1. WWIDを確認します。

```
/lib/udev/scsi_id -gud /dev/sda
```

sdaは、ブラックリストに追加するローカルSCSIディスクです。

WWIDの例はです 360030057024d0730239134810c0cb833。

2. 「blacklist」スタンザにWWIDを追加します。

```
blacklist {
    wwid      360030057024d0730239134810c0cb833
    devnode   "(ram|raw|loop|fd|md|dm-|sr|scd|st) [0-9] *"
    devnode   "^hd[a-z] *"
    devnode   "^cciss.*"
}
```

ステップ6：ONTAP LUNのマルチパスパラメータをカスタマイズする

ホストが他のベンダーのLUNに接続されていて、マルチパスパラメータの設定が無視されている場合は、ONTAP LUNに固有のスタンザをファイルの後半の部分で追加して修正する必要があります `multipath.conf` ます。これを行わないと、ONTAP LUNが想定どおりに動作しない可能性があります。

ファイル、特にdefaultsセクションで、をオーバーライドする可能性のある設定を確認します
 /etc/multipath.conf [マルチパスパラメータデフォルトセッテイ](#)。



ONTAP LUNの推奨されるパラメータ設定は無視しないでください。これらの設定は、ホスト構成のパフォーマンスを最適化するために必要です。詳細については、NetAppサポート、OSベンダー、またはその両方にお問い合わせください。

次の例は、オーバーライドされたデフォルトを修正する方法を示しています。この例ではmultipath.conf、ファイルにONTAP LUNと互換性のないおよび `no_path_retry` の値が定義されて `path_checker` います。ONTAPストレージレイはホストに接続されたままなので、これらのパラメータを削除することはできません。代わりに、および `no_path_retry` の値を修正する `path_checker` には、ONTAP LUNに特化したファイルにデバイススタンザを追加し `multipath.conf` ます。

例を示します

```
defaults {
    path_checker      readsector0
    no_path_retry     fail
}

devices {
    device {
        vendor        "NETAPP"
        product       "LUN"
        no_path_retry  queue
        path_checker   tur
    }
}
```

ステップ7：既知の問題を確認する

既知の問題はありません。

次の手順

- ["Linux Host Utilitiesツールの使用方法"](#)。
- ASMミラーリングについて学ぶ

Automatic Storage Management (ASM) ミラーリングでは、ASMが問題を認識して別の障害グループにスイッチオーバーできるように、Linuxマルチパス設定の変更が必要になる場合があります。ONTAP上のほとんどのASM構成では、外部冗長性が使用されます。つまり、データ保護は外付けアレイによって提供され、ASMはデータをミラーリングしません。一部のサイトでは、通常の冗長性を備えたASMを使用して、通常は異なるサイト間で双方向ミラーリングを提供します。詳細については、[を参照してください"ONTAP上のOracleデータベース"](#)。

- Ubuntu Linux仮想化（KVM）について学ぶ

Ubuntu Linux は KVM ホストとして機能できます。これにより、Linux カーネルベースの仮想マシン (KVM) テクノロジーを使用して、単一の物理サーバー上で複数の仮想マシンを実行できるようになります。KVM ホストでは、ONTAP LUN に対して明示的なホスト構成設定は必要ありません。

ONTAPストレージを使用してUbuntu 22.04をFCPおよびiSCSI用に設定する

Linux ホスト ユーティリティ ソフトウェアは、ONTAPストレージに接続された Linux ホスト用の管理および診断ツールを提供します。Ubuntu 22.04 ホストに Linux ホスト ユーティリティをインストールすると、ホスト ユーティリティを使用してONTAP LUN での FCP および iSCSI プロトコル操作を管理できるようになります。

手順1：必要に応じて**SAN**ブートを有効にします。

SANブートを使用するようにホストを設定することで、導入を簡易化し、拡張性を向上させることができます。

開始する前に

を使用["Interoperability Matrix Tool"](#)して、Linux OS、ホストバスアダプタ（HBA）、HBAファームウェア、HBAブートBIOS、およびONTAPバージョンがSANブートをサポートしていることを確認します。

手順

1. ["SANブートLUNを作成し、ホストにマップする"](#)です。
2. SAN ブート LUN がマッピングされているポートに対して、サーバ BIOS で SAN ブートを有効にします。

HBA BIOS を有効にする方法については、ベンダー固有のマニュアルを参照してください。

3. 構成が正常に完了したことを確認するために、ホストをリブートし、OSが稼働していることを確認します。

手順2：Linux Host Utilitiesをインストールする

NetAppでは、ONTAP LUN管理をサポートし、テクニカルサポートによる設定データの収集を支援するために、Linux Host Utilitiesをインストールすることを強く推奨しています。

"Linuxホストユーティリティ8.0をインストールする"。



Linux Host Utilitiesをインストールしても、Linuxホストのホストタイムアウト設定は変更されません。

手順3：ホストのマルチパス構成を確認する

Ubuntu 22.04 でマルチパスを使用してONTAP LUN を管理できます。

ホストでマルチパスが正しく設定されていることを確認するには、ファイルが定義されていること、およびONTAP LUN用にNetAppの推奨設定が設定されていることを確認し`/etc/multipath.conf` ます。

手順

1. ファイルが終了することを確認し`/etc/multipath.conf` ます。ファイルが存在しない場合は、空のゼロバイトファイルを作成します。

```
touch /etc/multipath.conf
```

2. ファイルの初回作成時には`multipath.conf`、マルチパスサービスを有効にして開始し、推奨設定をロードしなければならない場合があります。

```
systemctl enable multipathd
```

```
systemctl start multipathd
```

3. ホストをブートするたびに、空のゼロバイトファイルによって`/etc/multipath.conf`、NetApp推奨のホストマルチパスパラメータがデフォルト設定として自動的にロードされます。オペレーティングシステムは、ONTAP LUNを正しく認識および管理するマルチパスパラメータでコンパイルされているため、ホスト用のファイルを変更する必要はありません`/etc/multipath.conf`。

次の表に、Linux OS標準でコンパイルされたONTAP LUNのマルチパスパラメータの設定を示します。

パラメータ設定の表示

パラメータ	設定
detect_prio	はい。
DEV_DETION_TMO	" 無限 "
フェイルバック	即時
fast_io_fail_TMO	5.
の機能	"2 pg_init_retries 50"
flush_on_last_del	はい。
hardware_handler	0
パスの再試行なし	キュー
path_checker です	" tur "
path_grouping_policy	「 group_by_prio 」
path_selector	"service-time 0"
polling_interval （ポーリング間隔）	5.
Prio	ONTAP
プロダクト	LUN
retain_attached_hw_handler	はい。
RR_weight を指定します	" 均一 "
ユーザーフレンドリ名	いいえ
ベンダー	ネットアップ

4. ONTAP LUNのパラメータ設定とパスステータスを確認します。

```
multipath -ll
```

デフォルトのマルチパス パラメータは、ASA、AFF、およびFAS構成をサポートします。これらの構成では、単一のONTAP LUN に 4 つを超えるパスは必要ありません。パスが 4 つを超えると、ストレージ障害時に問題が発生する可能性があります。

次の出力例は、ASA、AFF、またはFAS構成のONTAP LUNについて、正しいパラメータ設定とパスステータスを示しています。

ASA構成

ASA構成では、特定のLUNへのすべてのパスが最適化され、アクティブな状態が維持されます。これにより、すべてのパスを同時に経由するI/O処理が行われるため、パフォーマンスが向上します。

例を示します

```
# multipath -ll
3600a098038314559533f524d6c652f62 dm-24 NETAPP,LUN C-Mode
size=10G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
`-+- policy='service-time 0' prio=50 status=active
| - 11:0:1:13 sdm 8:192 active ready running
| - 11:0:3:13 sdah 66:16 active ready running
| - 12:0:1:13 sdbc 67:96 active ready running
`- 12:0:3:13 sdbx 68:176 active ready running
```

AFFまたはFASの設定

AFFまたはFAS構成には、優先度の高いパスと低いパスの2つのグループを設定する必要があります。優先度の高いアクティブ/最適化パスは、アグリゲートが配置されているコントローラで処理されます。優先度の低いパスはアクティブですが、別のコントローラで処理されるため最適化されていません。最適化されていないパスは、最適化されたパスを使用できない場合にのみ使用されます。

次の例は、2つのアクティブ/最適化パスと2つのアクティブ/非最適化パスがあるONTAP LUNの出力を示しています。

例を示します

```
# multipath -ll
3600a098038314837352453694b542f4a dm-0 NETAPP,LUN C-Mode
size=160G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| | - 14:0:3:0 sdbk 67:224 active ready running
| | - 15:0:2:0 sdbl 67:240 active ready running
`+- policy='service-time 0' prio=10 status=enabled
| - 14:0:0:0 sda 8:0 active ready running
`- 15:0:1:0 sdv 65:80 active ready running
```

ステップ4：ホストのiSCSI構成を確認する

ホストに対して iSCSI が正しく構成されていることを確認します。

このタスクについて

iSCSI ホストで次の手順を実行します。

手順

1. iSCSI イニシエーター パッケージ (open-iscsi) がインストールされていることを確認します：

```
$apt list |grep open-iscsi
```

次の例のような出力が表示されます。

```
open-iscsi/noble-updates,noble-updates,now 2.1.9-3ubuntu5.4 amd64
```

2. `/etc/iscsi/initiatorname.iscsi` ファイルにあるiSCSIイニシエーターノード名を確認します：

```
InitiatorName=iqn.YYYY-MM.com.<vendor>:<host_name>
```

3. `/etc/iscsi/iscsid.conf` ファイルにあるiSCSIセッションタイムアウトパラメータを設定します：

```
node.session.timeo.replacement_timeout = 5
```

iSCSI `replacement_timeout` パラメータは、タイムアウトしたパスまたはセッションが再確立されるまでのiSCSIレイヤーの待機時間を制御します。この時間が経過すると、そのパスまたはセッションに対するコマンドは失敗します。iSCSI設定ファイルで `replacement_timeout` の値を5に設定する必要があります。

4. iSCSIサービスを有効にします：

```
$systemctl enable iscsid
```

5. iSCSIサービスを開始します：

```
$systemctl start iscsid
```

6. iSCSIサービスが実行されていることを確認します：

```
$systemctl status iscsid
```

例を示します

```
●iscsid.service - iSCSI initiator daemon (iscsid)
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service;
   enabled; preset: disabled)
   Active: active (running) since Mon 2026-01-12 12:53:18 IST; 2
   days ago
   TriggeredBy: ● iscsid.socket
     Docs: man:iscsid(8)
    Main PID: 1127419 (iscsid)
      Tasks: 2 (limit: 76557)
     Memory: 4.3M (peak: 8.8M)
        CPU: 1.657s
     CGroup: /system.slice/iscsid.service
            └─1127418 /usr/sbin/iscsid
            └─1127419 /usr/sbin/iscsid
```

7. iSCSIターゲットを検出します：

```
$iscsiadm --mode discovery --op update --type sendtargets --portal
<target_IP>
```

例を表示

```
iscsiadm --mode discovery --op update --type sendtargets --portal
192.168.100.197
192.168.100.197:3260,1046 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
192.168.200.199:3260,1049 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
192.168.100.199:3260,1048 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
192.168.200.197:3260,1047 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
```

8. ターゲットにログインします：

```
$iscsiadm --mode node -l all
```

9. ホストの起動時に iSCSI が自動的にログインするように設定します：


```
$iscsiadm --mode node -T <target_name> -p <ip:port> -o update -n  
node.startup -v automatic
```

次の例のような出力が表示されます。

```
iscsiadm --mode node -T iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 -p  
192.168.100.197:3260 -o update -n node.startup -v automatic
```

10. iSCSIセッションを確認します：

```
$iscsiadm --mode session
```

例を示します

```
iscsiadm --mode session  
tcp: [1] 192.168.200.197:3260,1047 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)  
tcp: [2] 192.168.100.197:3260,1046 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)  
tcp: [3] 192.168.100.199:3260,1048 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)  
tcp: [4] 192.168.200.199:3260,1049 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)
```

ステップ5：オプションでデバイスをマルチパスから除外する

必要に応じて、不要なデバイスのWWIDをファイルの「blacklist」スタンザに追加することで、デバイスをマルチパスから除外できます `multipath.conf`。

手順

1. WWIDを確認します。

```
/lib/udev/scsi_id -gud /dev/sda
```

sdaは、ブラックリストに追加するローカルSCSIディスクです。

WWIDの例はです 360030057024d0730239134810c0cb833。

2. 「blacklist」スタンザにWWIDを追加します。

```
blacklist {
    wwid      360030057024d0730239134810c0cb833
    devnode   "(ram|raw|loop|fd|md|dm-|sr|scd|st) [0-9] *"
    devnode   "^hd[a-z]"
    devnode   "^cciss.*"
}
```

ステップ6：ONTAP LUNのマルチパスパラメータをカスタマイズする

ホストが他のベンダーのLUNに接続されていて、マルチパスパラメータの設定が無視されている場合は、ONTAP LUNに固有のスタンザをファイルの後半の部分で追加して修正する必要があります。`multipath.conf`です。これを行わないと、ONTAP LUNが想定どおりに動作しない可能性があります。

ファイル、特にdefaultsセクションで、をオーバーライドする可能性のある設定を確認します
 /etc/multipath.conf [マルチパスパラメータデフォルトセッティ](#)。



ONTAP LUNの推奨されるパラメータ設定は無視しないでください。これらの設定は、ホスト構成のパフォーマンスを最適化するために必要です。詳細については、NetAppサポート、OSベンダー、またはその両方にお問い合わせください。

次の例は、オーバーライドされたデフォルトを修正する方法を示しています。この例ではmultipath.conf、ファイルにONTAP LUNと互換性のないおよび`no_path_retry`の値が定義されて`path_checker`います。ONTAPストレージレイはホストに接続されたままなので、これらのパラメータを削除することはできません。代わりに、および`no_path_retry`の値を修正する`path_checker`には、ONTAP LUNに特化したファイルにデバイススタンザを追加し`multipath.conf`ます。

例を示します

```
defaults {
    path_checker      readsector0
    no_path_retry     fail
}

devices {
    device {
        vendor        "NETAPP"
        product       "LUN"
        no_path_retry  queue
        path_checker   tur
    }
}
```

ステップ7：既知の問題を確認する

既知の問題はありません。

次の手順

- ["Linux Host Utilitiesツールの使用方法"](#)。
- ASMミラーリングについて学ぶ

Automatic Storage Management (ASM) ミラーリングでは、ASMが問題を認識して別の障害グループにスイッチオーバーできるように、Linuxマルチパス設定の変更が必要になる場合があります。ONTAP上のほとんどのASM構成では、外部冗長性が使用されます。つまり、データ保護は外付けアレイによって提供され、ASMはデータをミラーリングしません。一部のサイトでは、通常の冗長性を備えたASMを使用して、通常は異なるサイト間で双方向ミラーリングを提供します。詳細については、[を参照してください"ONTAP上のOracleデータベース"](#)。

- Ubuntu Linux仮想化（KVM）について学ぶ

Ubuntu Linux は KVM ホストとして機能できます。これにより、Linux カーネルベースの仮想マシン (KVM) テクノLOGYを使用して、単一の物理サーバー上で複数の仮想マシンを実行できるようになります。KVM ホストでは、ONTAP LUN に対して明示的なホスト構成設定は必要ありません。

ONTAPストレージを使用してUbuntu 20.04をFCPおよびiSCSI用に設定する

Ubuntu 20.04 をマルチパス用に構成し、ONTAPストレージでの FCP および iSCSI プロトコル操作の特定のパラメータと設定を設定します。



Linux ホスト ユーティリティ ソフトウェア パッケージは、Ubuntu オペレーティング システムをサポートしていません。

ONTAP LUN はハイパーバイザーに自動的にマップされるため、カーネルベースの仮想マシン (KVM) 設定を手動で構成する必要はありません。

手順1：必要に応じて**SAN**ブートを有効にします。

SANブートを使用するようにホストを設定することで、導入を簡易化し、拡張性を向上させることができます。

開始する前に

を使用["Interoperability Matrix Tool"](#)して、Linux OS、ホストバスアダプタ（HBA）、HBAファームウェア、HBAブートBIOS、およびONTAPバージョンがSANブートをサポートしていることを確認します。

手順

1. ["SANブートLUNを作成し、ホストにマップする"](#)です。
2. SAN ブート LUN がマッピングされているポートに対して、サーバ BIOS で SAN ブートを有効にします。

HBA BIOS を有効にする方法については、ベンダー固有のマニュアルを参照してください。

3. 構成が正常に完了したことを確認するために、ホストをリブートし、OSが稼働していることを確認します。

ステップ2: ホストのマルチパス構成を確認する

Ubuntu 20.04 でマルチパスを使用してONTAP LUN を管理できます。

ホストでマルチパスが正しく設定されていることを確認するには、ファイルが定義されていること、およびONTAP LUN用にNetAppの推奨設定が設定されていることを確認し`/etc/multipath.conf` ます。

手順

1. ファイルが終了することを確認し`/etc/multipath.conf` ます。ファイルが存在しない場合は、空のゼロバイトファイルを作成します。

```
touch /etc/multipath.conf
```

2. ファイルの初回作成時には multipath.conf、マルチパスサービスを有効にして開始し、推奨設定をロードしなければならない場合があります。

```
systemctl enable multipathd
```

```
systemctl start multipathd
```

3. ホストをブートするたびに、空のゼロバイトファイルによって /etc/multipath.conf、NetApp推奨のホストマルチパスパラメータがデフォルト設定として自動的にロードされます。オペレーティングシステムは、ONTAP LUNを正しく認識および管理するマルチパスパラメータでコンパイルされているため、ホスト用のファイルを変更する必要はありません /etc/multipath.conf。

次の表に、Linux OS標準でコンパイルされたONTAP LUNのマルチパスパラメータの設定を示します。

パラメータ設定の表示

パラメータ	設定
detect_prio	はい。
DEV_DETION_TMO	" 無限 "
フェイルバック	即時
fast_io_fail_TMO	5.
の機能	"2 pg_init_retries 50"
flush_on_last_del	はい。
hardware_handler	0
パスの再試行なし	キュー
path_checker です	" tur "
path_grouping_policy	「 group_by_prio 」
path_selector	"service-time 0"
polling_interval （ポーリング間隔）	5.
Prio	ONTAP
プロダクト	LUN
retain_attached_hw_handler	はい。
RR_weight を指定します	" 均一 "
ユーザーフレンドリ名	いいえ
ベンダー	ネットアップ

4. ONTAP LUNのパラメータ設定とパスステータスを確認します。

```
multipath -ll
```

デフォルトのマルチパス パラメータは、ASA、AFF、およびFAS構成をサポートします。これらの構成では、単一のONTAP LUN に 4 つを超えるパスは必要ありません。パスが 4 つを超えると、ストレージ障害時に問題が発生する可能性があります。

次の出力例は、ASA、AFF、またはFAS構成のONTAP LUNについて、正しいパラメータ設定とパスステータスを示しています。

ASA構成

ASA構成では、特定のLUNへのすべてのパスが最適化され、アクティブな状態が維持されます。これにより、すべてのパスを同時に経由するI/O処理が行われるため、パフォーマンスが向上します。

例を示します

```
# multipath -ll
3600a098038314559533f524d6c652f62 dm-24 NETAPP,LUN C-Mode
size=10G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
`-+- policy='service-time 0' prio=50 status=active
| - 11:0:1:13 sdm 8:192 active ready running
| - 11:0:3:13 sdah 66:16 active ready running
| - 12:0:1:13 sdbc 67:96 active ready running
`- 12:0:3:13 sdbx 68:176 active ready running
```

AFFまたはFASの設定

AFFまたはFAS構成には、優先度の高いパスと低いパスの2つのグループを設定する必要があります。優先度の高いアクティブ/最適化パスは、アグリゲートが配置されているコントローラで処理されます。優先度の低いパスはアクティブですが、別のコントローラで処理されるため最適化されていません。最適化されていないパスは、最適化されたパスを使用できない場合にのみ使用されます。

次の例は、2つのアクティブ/最適化パスと2つのアクティブ/非最適化パスがあるONTAP LUNの出力を示しています。

例を示します

```
# multipath -ll
3600a098038314837352453694b542f4a dm-0 NETAPP,LUN C-Mode
size=160G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| | - 14:0:3:0 sdbk 67:224 active ready running
| | - 15:0:2:0 sdbl 67:240 active ready running
`+- policy='service-time 0' prio=10 status=enabled
| - 14:0:0:0 sda 8:0 active ready running
`- 15:0:1:0 sdv 65:80 active ready running
```

ステップ3：ホストのiSCSI構成を確認する

ホストに対して iSCSI が正しく構成されていることを確認します。

このタスクについて

iSCSI ホストで次の手順を実行します。

手順

1. iSCSI イニシエーター パッケージ (open-iscsi) がインストールされていることを確認します：

```
$apt list |grep open-iscsi
```

次の例のような出力が表示されます。

```
open-iscsi/noble-updates,noble-updates,now 2.1.9-3ubuntu5.4 amd64
```

2. `/etc/iscsi/initiatorname.iscsi` ファイルにあるiSCSIイニシエーターノード名を確認します：

```
InitiatorName=iqn.YYYY-MM.com.<vendor>:<host_name>
```

3. `/etc/iscsi/iscsid.conf` ファイルにあるiSCSIセッションタイムアウトパラメータを設定します：

```
node.session.timeo.replacement_timeout = 5
```

iSCSI `replacement_timeout` パラメータは、タイムアウトしたパスまたはセッションが再確立されるまでのiSCSIレイヤーの待機時間を制御します。この時間が経過すると、そのパスまたはセッションに対するコマンドは失敗します。iSCSI設定ファイルで `replacement_timeout` の値を5に設定する必要があります。

4. iSCSIサービスを有効にします：

```
$systemctl enable iscsid
```

5. iSCSIサービスを開始します：

```
$systemctl start iscsid
```

6. iSCSIサービスが実行されていることを確認します：

```
$systemctl status iscsid
```

例を示します

```
●iscsid.service - iSCSI initiator daemon (iscsid)
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service;
   enabled; preset: disabled)
   Active: active (running) since Mon 2026-01-12 12:53:18 IST; 2
   days ago
   TriggeredBy: ● iscsid.socket
     Docs: man:iscsid(8)
    Main PID: 1127419 (iscsid)
      Tasks: 2 (limit: 76557)
     Memory: 4.3M (peak: 8.8M)
        CPU: 1.657s
     CGroup: /system.slice/iscsid.service
            └─1127418 /usr/sbin/iscsid
            └─1127419 /usr/sbin/iscsid
```

7. iSCSIターゲットを検出します：

```
$iscsiadm --mode discovery --op update --type sendtargets --portal
<target_IP>
```

例を表示

```
iscsiadm --mode discovery --op update --type sendtargets --portal
192.168.100.197
192.168.100.197:3260,1046 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
192.168.200.199:3260,1049 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
192.168.100.199:3260,1048 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
192.168.200.197:3260,1047 iqn.1992-
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8
```

8. ターゲットにログインします：

```
$iscsiadm --mode node -l all
```

9. ホストの起動時に iSCSI が自動的にログインするように設定します：


```
$iscsiadm --mode node -T <target_name> -p <ip:port> -o update -n  
node.startup -v automatic
```

次の例のような出力が表示されます。

```
iscsiadm --mode node -T iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 -p  
192.168.100.197:3260 -o update -n node.startup -v automatic
```

10. iSCSIセッションを確認します：

```
$iscsiadm --mode session
```

例を示します

```
iscsiadm --mode session  
tcp: [1] 192.168.200.197:3260,1047 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)  
tcp: [2] 192.168.100.197:3260,1046 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)  
tcp: [3] 192.168.100.199:3260,1048 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)  
tcp: [4] 192.168.200.199:3260,1049 iqn.1992-  
08.com.netapp:sn.7cd154a7d35411f0a25ed039eaa95f59:vs.8 (non-flash)
```

手順4：必要に応じて、マルチパスからデバイスを除外する

必要に応じて、不要なデバイスのWWIDをファイルの「blacklist」スタンザに追加することで、デバイスをマルチパスから除外できます `multipath.conf`。

手順

1. WWIDを確認します。

```
/lib/udev/scsi_id -gud /dev/sda
```

sdaは、ブラックリストに追加するローカルSCSIディスクです。

WWIDの例はです 360030057024d0730239134810c0cb833。

2. 「blacklist」スタンザにWWIDを追加します。

```
blacklist {
    wwid      360030057024d0730239134810c0cb833
    devnode   "(ram|raw|loop|fd|md|dm-|sr|scd|st) [0-9] *"
    devnode   "^hd[a-z]"
    devnode   "^cciss.*"
}
```

手順5：ONTAP LUNのマルチパスパラメータをカスタマイズする

ホストが他のベンダーのLUNに接続されていて、マルチパスパラメータの設定が無視されている場合は、ONTAP LUNに固有のスタンザをファイルの後半の部分で追加して修正する必要があります。`multipath.conf`です。これを行わないと、ONTAP LUNが想定どおりに動作しない可能性があります。

ファイル、特にdefaultsセクションで、をオーバーライドする可能性のある設定を確認します
 /etc/multipath.conf [マルチパスパラメータデフォルトセッティ](#)。



ONTAP LUNの推奨されるパラメータ設定は無視しないでください。これらの設定は、ホスト構成のパフォーマンスを最適化するために必要です。詳細については、NetAppサポート、OSベンダー、またはその両方にお問い合わせください。

次の例は、オーバーライドされたデフォルトを修正する方法を示しています。この例ではmultipath.conf、ファイルにONTAP LUNと互換性のないおよび`no_path_retry`の値が定義されて`path_checker`います。ONTAPストレージレイはホストに接続されたままなので、これらのパラメータを削除することはできません。代わりに、および`no_path_retry`の値を修正する`path_checker`には、ONTAP LUNに特化したファイルにデバイススタンザを追加し`multipath.conf`ます。

例を示します

```
defaults {
    path_checker      readsector0
    no_path_retry     fail
}

devices {
    device {
        vendor        "NETAPP"
        product        "LUN"
        no_path_retry  queue
        path_checker   tur
    }
}
```

手順6：既知の問題を確認する

既知の問題はありません。

次の手順

- Ubuntu Linux仮想化（KVM）について学ぶ

Ubuntu Linux は KVM ホストとして機能できます。これにより、Linux カーネルベースの仮想マシン (KVM) テクノロジを使用して、単一の物理サーバー上で複数の仮想マシンを実行できるようになります。KVM ホストでは、ONTAP LUN に対して明示的なホスト構成設定は必要ありません。

著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。