



MetroClusterおよびSnapMirrorアクティブ同 期用のONTAP Mediator

ONTAP 9

NetApp
March 02, 2026

目次

MetroClusterおよびSnapMirrorアクティブ同期用のONTAP Mediator	1
ONTAP Mediatorについて学ぶ	1
システム管理者向けのツール	1
注意事項	2
ONTAP Mediatorの新機能	2
機能強化	2
OSのサポート対応表	4
SCSTのサポート対応表	7
インストールまたはアップグレード	8
ONTAP Mediatorのインストールワークフローの概要	8
ONTAPメディアーターをインストールまたはアップグレードする	9
ホストOSとONTAP Mediatorをアップグレードする	15
ONTAP Mediatorのインストールにリポジトリアクセスを提供する	20
ONTAPメディアーターのインストールパッケージをダウンロードします	27
ONTAP Mediatorのコード署名の検証	28
ONTAP Mediatorのインストール パッケージのインストール	30
ONTAP Mediatorのインストールステータスを確認する	46
インストール後のONTAP Mediator の設定	47
ONTAP Mediatorの管理	52
ユーザー名を変更する	52
パスワードの変更	53
ONTAP Mediatorを停止する	54
ONTAPメディアーターを再度有効にする	54
ONTAPメディアーターが正常であることを確認する	55
ONTAP Mediatorのアンインストール	56
一時的な自己署名証明書の再生成	57
信頼できるサードパーティの証明書への自己署名証明書の置き換え	58
証明書関連の問題のトラブルシューティング	76
ONTAP MediatorのホストOSを保守する	83
ホストのリブート	83
ホスト パッケージの更新	83
ホストOSカーネルをアップグレードする	83
ホストのメンテナンスを実行する	83
ホストのホスト名またはIPの変更	86

MetroClusterおよびSnapMirrorアクティブ同期用のONTAP Mediator

ONTAP Mediatorについて学ぶ

このドキュメントは、オンプレミス版のONTAP Mediatorについて説明しています。ONTAP 9.17.1以降で利用可能なONTAP Cloud Mediatorの詳細については、"[SnapMirror アクティブ同期のドキュメント](#)"を参照してください。

ONTAP Mediator は ONTAP 機能に対していくつかの機能を提供します：

- HAメタデータ用にフェンシングされた永続的なストアを提供します。
- コントローラの稼働を確認するpingプロキシとして機能します。
- クォーラムを判定するために、ノードの健全性を同期的に照会します。

ONTAP Mediator は、次の2つの追加のsystemctl サービスを提供します。

- **ontap_mediator.service**

ONTAP 関係を管理するための REST API サーバーを管理します。

- **mediator-scst.service**

iSCSIモジュール（SCST）の起動とシャットダウンを制御します。

システム管理者向けのツール

システム管理者向けのツールは以下のとおりです。

- **/usr/local/bin/mediator_change_password**

現在のAPIユーザ名とパスワードが提供された場合に、新しいAPIパスワードを設定します。

- **/usr/local/bin/mediator_change_user**

現在のAPIユーザ名とパスワードが提供された場合に、新しいAPIユーザ名を設定します。

- **/usr/local/bin/mediator_generate_support_bundle**

NetAppカスタマーサポートとのやり取りに必要なすべての有用なサポート情報を含むローカルtgzファイルを生成します。これには、アプリケーション構成、ログ、および一部のシステム情報が含まれます。バンドルはローカルディスク上に生成され、必要に応じて手動で転送できます。保存場所
： /opt/netapp/data/support_bundles/

- **/usr/local/bin/uninstall_ontap_mediator**

ONTAP MediatorパッケージとSCSTカーネル モジュールを削除します。すべての設定、ログ、およびメールボックス データが削除されます。

- `/usr/local/bin/mediator_unlock_user`

認証の最大試行回数を超えたAPIユーザ アカウントに対し、ロックアウトを解除します。この機能は、総当たり攻撃によるパスワードの漏えいを防ぐために使用されます。ユーザには、正しいユーザ名とパスワードの入力を求めるプロンプトが表示されます。

- `/usr/local/bin/mediator_add_user`

(サポートのみ) インストール時にAPIユーザを追加します。

注意事項

ONTAP Mediatorは、iSCSIを提供するためにSCSTに依存しています (<http://scst.sourceforge.net/index.html>を参照)。このパッケージは、インストール時にカーネル専用コンパイルされるカーネルモジュールです。カーネルを更新すると、SCSTの再インストールが必要になる場合があります。または、ONTAP Mediatorをアンインストールしてから再インストールし、ONTAP関係を再設定してください。



サーバOSカーネルの更新には、ONTAPのメンテナンス時間を設ける必要があります。

ONTAP Mediatorの新機能

ONTAP Mediatorはリリースごとに新しい機能強化が提供されます。新機能は以下のとおりです。

機能強化

SCST のバージョン情報については、[SCSTのサポート対応表](#)を参照してください。

ONTAP Mediatorのバージョン	機能強化
1.11	<ul style="list-style-type: none">• RHELのサポート：<ul style="list-style-type: none">◦ 互換性：9.5。◦ 推奨：10.1、10.0、9.7、9.6、9.4、8.10。• Rocky Linux 10.1、9.7、8.10 のサポート。• Oracle Linux 10.0 および 9.6 のサポート。• MetroCluster IP 構成に IPv6 のサポートを追加します。• fapolicyd のサポートを追加します。

1.10	<ul style="list-style-type: none"> • RHELのサポート： <ul style="list-style-type: none"> ◦ 互換性：9.5。 ◦ 推奨：10.0、9.6、9.4、8.10。 • Rocky Linux 10.0、9.6、8.10 のサポート。 • 基本 Python バージョンを Python 3.9 から Python 3.12 にアップグレードします。
1.9.1	<ul style="list-style-type: none"> • RHELのサポート： <ul style="list-style-type: none"> ◦ 互換性：9.3、9.1、8.9、8.7、8.6、8.5、8.4。 ◦ 推奨：9.5、9.4、9.2、9.0、8.10、8.8。 • Rocky Linux 9.5および8.10のサポート。 • コード署名検証用の新しい証明書を追加します。 • <code>-skip-code-signature-check</code> フラグを使用してコード署名チェックをスキップするためのサポートが追加されました。 • インストーラーは、期限切れのコード署名証明書を検出すると警告を表示します。
1.9	<ul style="list-style-type: none"> • RHELのサポート： <ul style="list-style-type: none"> ◦ 互換性：9.3、9.1、8.9、8.7、8.6、8.5、8.4。 ◦ 推奨：9.5、9.4、9.2、9.0、8.10、8.8。 • Rocky Linux 9.5および8.10のサポート。 • RHELおよびRocky LinuxでのFIPSのサポート。 • パフォーマンスの強化によって拡張性が向上。 • ファイル名の改善によってPKI署名証明書のセットアップを簡易化。
1.8	<ul style="list-style-type: none"> • RHELのサポート： <ul style="list-style-type: none"> ◦ 互換性：8.7、8.6、8.5、8.4。 ◦ 推奨：9.4、9.3、9.2、9.1、9.0、8.10、8.9、8.8。 • Rocky Linux 9.4および8.10のサポート。
1.7	<ul style="list-style-type: none"> • RHELのサポート： <ul style="list-style-type: none"> ◦ 互換性：8.7、8.6、8.5、8.4。 ◦ 推奨：9.3、9.2、9.1、9.0、8.9、8.8。 • Rocky Linux 9.3および8.9のサポート。 • 自己署名証明書およびサードパーティ署名証明書でのSAN（サブジェクトの別名）データのサポート。

1.6	<ul style="list-style-type: none"> • Python 3.9の更新。 • RHELのサポート： <ul style="list-style-type: none"> ◦ 互換性：8.7、8.6、8.5、8.4。 ◦ 推奨：9.2、9.1、9.0、8.8。 • Rocky Linux 9.2および8.8のサポート。 • RHEL 7.x / CentOSの全リリースのサポートを終了。
1.5	<ul style="list-style-type: none"> • RHEL 8.5、8.4、8.3、8.2、8.1、8.0、7.9、7.8、7.7、7.6のサポート。 • CentOS 7.9、7.8、7.7、7.6のサポート。 • RHEL 7.x / CentOS 7.xの廃止に関する警告を追加。 • 大規模なSnapMirrorアクティブ同期システム向けに速度を最適化。 • インストーラーに暗号コード署名が追加されました。
1.4	<ul style="list-style-type: none"> • RHEL 8.5、8.4、8.3、8.2、8.1、8.0、7.9、7.8、7.7、7.6のサポート。 • CentOS 7.9、7.8、7.7、7.6のサポート。 • UEFIベースのファームウェアのセキュアブート（SB）をサポート。
1.3	<ul style="list-style-type: none"> • RHEL 8.3、8.2、8.1、8.0、7.9、7.8、7.7、7.6のサポート。 • CentOS 7.9、7.8、7.7、7.6のサポート。
1.2	<ul style="list-style-type: none"> • RHEL 8.1、8.0、7.9、7.8、7.7、7.6のサポート。 • CentOS 7.9、7.8、7.7、7.6のサポート。 • HTTPSメールボックスのサポート。 • ONTAP 9.8+ MCC-IP AUSO およびSnapMirror Active Sync ZRTO で使用します。
1.1	<ul style="list-style-type: none"> • RHEL 8.0および7.6のサポート。 • CentOS 7.6のサポート。 • Perlの依存関係を削除。
1.0	<ul style="list-style-type: none"> • iSCSIメールボックスのサポート。 • ONTAP 9.7+ MCC-IP AUSO で使用します。 • RHEL / CentOS 7.6のサポート。

OSのサポート対応表

ONTAP Mediator or OS	1.11	1.10	1.9.1	1.9	1.8	1.7	1.6	1.5	1.4	1.3	1.2	1.1	1.0
RHEL 10.1	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 10.0	はい	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.7	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.6	はい	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.5	互換性がある	互換性がある	はい	はい	いいえ								
RHEL 9.4	はい	はい	はい	はい	はい	いいえ							
RHEL 9.3	いいえ	いいえ	互換性がある	互換性がある	はい	はい	いいえ						
RHEL 9.2	いいえ	いいえ	はい	はい	はい	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.1	いいえ	いいえ	互換性がある	互換性がある	はい	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.0	いいえ	いいえ	はい	はい	はい	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 8.10	はい	はい	はい	はい	はい	いいえ							
RHEL 8.9	いいえ	いいえ	互換性がある	互換性がある	はい	はい	いいえ						
RHEL 8.8	いいえ	いいえ	はい	はい	はい	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ

RHEL 8.7	いいえ	いいえ	互換性がある	互換性がある	はい	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 8.6	いいえ	いいえ	互換性がある	互換性がある	はい	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 8.5	いいえ	いいえ	互換性がある	互換性がある	はい	はい	はい	はい	はい	いいえ	いいえ	いいえ	いいえ
RHEL 8.4	いいえ	いいえ	互換性がある	互換性がある	はい	はい	はい	はい	はい	いいえ	いいえ	いいえ	いいえ
RHEL 8.3	廃止	廃止	廃止	廃止	廃止	廃止	廃止	はい	はい	はい	いいえ	いいえ	いいえ
RHEL 8.2	廃止	廃止	廃止	廃止	廃止	廃止	廃止	はい	はい	はい	いいえ	いいえ	いいえ
RHEL 8.1	廃止	廃止	廃止	廃止	廃止	廃止	廃止	はい	はい	はい	はい	いいえ	いいえ
RHEL 8.0	廃止	廃止	廃止	廃止	廃止	廃止	廃止	はい	はい	はい	はい	はい	いいえ
RHEL および CentOS 7.9	廃止	廃止	廃止	廃止	廃止	廃止	廃止	はい	はい	はい	互換性がある	いいえ	いいえ
RHEL および CentOS 7.8	廃止	廃止	廃止	廃止	廃止	廃止	廃止	はい	はい	はい	はい	いいえ	いいえ
RHEL および CentOS 7.7	廃止	廃止	廃止	廃止	廃止	廃止	廃止	はい	はい	はい	はい	いいえ	いいえ

RHEL および CentOS 7.6	廃止	はい	はい	はい	はい	はい	はい (RHELのみ)						
CentOS 8 とstream	いいえ	いいえ	いいえ	該当なし	該当なし	該当なし							
Rocky Linux 10.0	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
Rocky Linux 9	はい	該当なし	該当なし	該当なし	該当なし	該当なし	該当なし						
Rocky Linux 8	はい	該当なし	該当なし	該当なし	該当なし	該当なし	該当なし						
Oracle Linux 10.0	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ						
Oracle Linux 9	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ						

- 「○」は、ONTAP Mediatorのインストールに推奨されるOSであり、完全な互換性があり、サポートされることを意味します。
- 「×」は、OSとONTAP Mediatorに互換性がないことを意味します。
- 「互換性あり」とは、Red Hat がこれらの RHEL バージョンをサポートしなくなったものの、ONTAP Mediator をこれらのバージョンにインストールできることを意味します。
- ONTAP Mediator 1.6では、Rocky Linux 9および8のサポートが追加されています。
- ONTAP Mediator 1.5は、RHEL 7.xブランチ オペレーティング システムでサポートされる最後のリリースです。
- CentOS 8は再ブランチのためすべてのリリースで削除されました。CentOS Streamは、本番環境のターゲットOSとしては適切でないとみなされていました。サポートは予定されていません。

SCSTのサポート対応表

次の表に、ONTAP Mediatorの各バージョンでサポートされるSCSTのバージョンを示します。

ONTAP Mediatorのバージョン	サポートされるSCSTのバージョン
ONTAP Mediator 1.11	scst-3.9.tar.gz
ONTAP Mediator 1.10	scst-3.9.tar.gz
ONTAP Mediator 1.9.1	scst-3.8.0.tar.bz2
ONTAP Mediator 1.9	scst-3.8.0.tar.bz2
ONTAP Mediator 1.8	scst-3.8.0.tar.bz2
ONTAP Mediator 1.7	scst-3.7.0.tar.bz2
ONTAP Mediator 1.6	scst-3.7.0.tar.bz2
ONTAP Mediator 1.5	scst-3.6.0.tar.bz2
ONTAP Mediator 1.4	scst-3.6.0.tar.bz2
ONTAP Mediator 1.3	scst-3.5.0.tar.bz2
ONTAP Mediator 1.2	scst-3.4.0.tar.bz2
ONTAP Mediator 1.1	scst-3.4.0.tar.bz2
ONTAP Mediator 1.0	scst-3.3.0.tar.bz2

インストールまたはアップグレード

ONTAP Mediatorのインストールワークフローの概要

ONTAP Mediator のインストールには、インストールの準備、リポジトリへのアクセスの提供、インストールパッケージのダウンロード、コード署名の検証、ONTAP Mediator パッケージのインストール、およびインストール後の設定タスクの実行が含まれます。

1

"ONTAP Mediatorのインストールまたはアップグレードの準備"

ONTAP Mediator をインストールまたはアップグレードするには、すべての前提条件が満たされていることを確認する必要があります。

2

"ホストOSとMediatorのアップグレード"

既存のONTAP Mediatorをアップグレードする場合は、まず以前のバージョンをアンインストールしてから、新しいバージョンをインストールする必要があります。ONTAP Mediatorを初めてインストールする場合は、この手順をスキップできます。

3

"リポジトリアクセスを提供する"

ONTAP Mediator がインストール プロセス中に必要なパッケージにアクセスできるように、リポジトリへのアクセスを有効にする必要があります。

4

"ONTAPメディエーターのインストールパッケージをダウンロードします"

ONTAP Mediator ダウンロード ページから ONTAP Mediator インストール パッケージをダウンロードします。

5

"ONTAPメディエーターインストールパッケージのコード署名を確認します"

NetApp は、ONTAP Mediator インストールパッケージをインストールする前に、ONTAP Mediator のコード署名を検証することを推奨しています。

6

"ONTAPメディエーターをインストールする"

ONTAP Mediator をインストールするには、インストールパッケージを取得し、ホスト上でインストーラを実行する必要があります。

7

"ONTAPメディエーターのインストールを確認する"

ONTAP Mediator をインストールした後、正常に実行されていることを確認します。

8

"インストール後の構成タスクを実行する"

ONTAP Mediator をインストールして実行した後、ONTAP Mediator 機能を使用するには追加の設定タスクを実行する必要があります。

ONTAPメディエーターをインストールまたはアップグレードする

ONTAP Mediator をインストールまたはアップグレードするには、すべての前提条件を満たし、インストールパッケージをダウンロードして、ホスト上でインストーラを実行する必要があります。

- ONTAP 9.8以降では、任意のバージョンのONTAP Mediatorを使用してSnapMirrorアクティブ同期関係を監視できます。
- 任意のバージョンのONTAP Mediatorを使用してMetroCluster IP構成を監視できます。

インストールとアップグレードに関する考慮事項

ONTAP Mediator をアップグレードまたはインストールする前に、これらの点を確認してください。

i

ONTAP Mediator 1.8以前はRed Hat Enterprise Linux (RHEL) FIPSモードと互換性がないため、正常にインストールできません。FIPSモードが有効になっているかどうかは `fips-mode-setup --check` コマンドを使用して確認できます。FIPSモードを無効にするには `fips-modesetup --disable` コマンドを使用してください。FIPSモードを無効にした後、再起動すると、ONTAP Mediator 1.8以前を正常にインストールできます。

- ONTAP Mediatorを最新バージョンにアップグレードする必要があります。古いバージョンもすべてのONTAPリリースで動作しますが、新しいバージョンにはサードパーティ製コンポーネントのセキュリティパッチが含まれています。

- ONTAP Mediatorの新しいバージョンにアップグレードすると、より上位のバージョンが利用可能でない限り、インストーラは推奨されるSCSTバージョンに自動的にアップグレードします。より上位のSCSTバージョンを手動でインストールする手順については、"[ONTAP Mediatorの管理](#)"を参照してください。サポートされているバージョンについては、"[SCSTのサポート対応表](#)"を参照してください。



- インストールが失敗した場合は、ONTAP Mediatorの新しいバージョンにアップグレードする必要がある場合があります。
- 2025年6月15日以降、コード署名証明書の有効期限が切れているため、ONTAP Mediator 1.9および1.8のインストールまたはアップグレードは行えません。インストールまたはアップグレードに失敗した場合は、代わりにONTAP Mediator 1.9.1パッチバージョンをご使用ください。

- `yum-utils`パッケージをインストールすると、`needs-restarting`コマンドを使用できます。
- ONTAP Mediator 1.11 以降では、MetroCluster IP 構成で IPv6 がサポートされます。

ホストの要件

RHEL または Rocky Linux をインストールし、関連するリポジトリを構成するときは、次の要件に従ってください。



インストールまたは構成プロセスを変更する場合は、追加の手順を実行する必要がある場合があります。

Linuxディストリビューションの要件

- Red Hatのベストプラクティスに従って、RHELまたはRocky Linuxをインストールしてください。CentOS 8.xはサポート終了となったため、CentOS 8.xの互換バージョンは推奨されません。
- ONTAP Mediator をインストールするときは、インストール プログラムが必要なすべてのソフトウェア依存関係を取得してインストールできるように、システムが必要なリポジトリにアクセスできることを確認してください。
- yum インストーラーが RHEL リポジトリ内の依存ソフトウェアを見つけられるようにするには、インストール中またはインストール後に有効な Red Hat サブスクリプションを使用してシステムを登録します。



詳細については、Red Hat Subscription Manager のドキュメントを参照してください。

ネットワーク要件

次のポートが ONTAP Mediator で使用可能であり、未使用であることを確認します。

ポート/サービス	ソース	送受信方向	デスティネーション	目的
22 / TCP	管理ホスト	インバウンド	ONTAP Mediator	(オプション) SSH / ONTAP Mediator管理
31784 / TCP	cluster-mgmt および node-mgmt LIF	インバウンド	ONTAP Mediator のWebサーバ	(必須) REST API (HTTPS)

3260 / TCP	ノード管理LIF	インバウンド	ONTAP Mediator のiSCSIターゲット	(MetroCluster IP構 成に必須) メールボ ックスのiSCSIデー タ接続
------------	----------	--------	-------------------------------	---

SMBC のお客様の場合、ONTAP ではポート 3260 を有効にしたり接続したりする必要はありません。

- サードパーティのファイアウォールを使用している場合は、"[ONTAP Mediatorのファイアウォール要件](#)"を参照してください。
- インターネットにアクセスできない Linux ホストの場合は、必要なパッケージがローカルリポジトリで使用可能であることを確認してください。

Linux 環境で Link Aggregation Control Protocol (LACP) を使用している場合は、カーネルを構成して `sysctl net.ipv4.conf.all.arp_ignore` を `2` に設定します。

OSの要件

OS は次の要件を満たしている必要があります：

- 64ビットの物理インストールまたは仮想マシン
- 8GBのRAM
- 1GBのディスク スペース (アプリケーションのインストール、サーバ ログ、およびデータベースに使用)
- ユーザー：Root アクセス

次の表は ONTAP Mediator の各バージョンでサポートされている OS を示しています。

ONTAP Mediatorのバージョン	サポートされるLinuxバージョン
1.11	<ul style="list-style-type: none"> • Red Hat Enterprise Linux <ul style="list-style-type: none"> ◦ 互換性：9.5¹ ◦ 推奨：10.1、10.0、9.7、9.6、9.4、8.10 • Rocky Linux 10.1、9.7、および 8.10 • Oracle Linux 10.0 および 9.6
1.10	<ul style="list-style-type: none"> • Red Hat Enterprise Linux <ul style="list-style-type: none"> ◦ 互換性：9.5¹ ◦ 推奨：10.0、9.6、9.4、8.10 • Rocky Linux 10.0、9.6、および 8.10

1.9.1	<ul style="list-style-type: none"> • Red Hat Enterprise Linux <ul style="list-style-type: none"> ◦ 互換性：9.3、9.1、8.9、8.7、8.6、8.5、8.4¹ ◦ 推奨：9.5、9.4、9.2、9.0、8.10、8.8 • Rocky Linux 9.5、8.10
1.9	<ul style="list-style-type: none"> • Red Hat Enterprise Linux <ul style="list-style-type: none"> ◦ 互換性：9.3、9.1、8.9、8.7、8.6、8.5、8.4¹ ◦ 推奨：9.5、9.4、9.2、9.0、8.10、8.8 • Rocky Linux 9.5、8.10
1.8	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : <ul style="list-style-type: none"> ◦ 互換性：8.7、8.6、8.5、8.4¹ ◦ 推奨：9.4、9.3、9.2、9.1、9.0、8.10、8.9、8.8 • Rocky Linux 9.4、8.10
1.7	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : <ul style="list-style-type: none"> ◦ 互換性：8.7、8.6、8.5、8.4¹ ◦ 推奨：9.3、9.2、9.1、9.0、8.9、8.8 • Rocky Linux 9.3、8.9
1.6	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : <ul style="list-style-type: none"> ◦ 互換性：8.7、8.6、8.5、8.4¹ ◦ 推奨：9.2、9.1、9.0、8.8 • Rocky Linux 9.2、8.8
1.5	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 8.5、8.4、8.3、8.2、8.1、8.0、7.9、7.8、7.7、および 7.6 • CentOS : 7.9、7.8、7.7、7.6
1.4	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 8.5、8.4、8.3、8.2、8.1、8.0、7.9、7.8、7.7、および 7.6 • CentOS : 7.9、7.8、7.7、7.6
1.3	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 8.3、8.2、8.1、8.0、7.9、7.8、7.7、および 7.6 • CentOS : 7.9、7.8、7.7、7.6

1.2	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 8.1、8.0、7.9、7.8、7.7、および 7.6 • CentOS : 7.9、7.8、7.7、7.6
-----	--

1. 互換性があるとは、Red Hat がこれらの RHEL バージョンをサポートしなくなったものの、ONTAP Mediator をこれらのバージョンにインストールできることを意味します。

OS必須パッケージ

ONTAP Mediator には次のパッケージが必要です。



パッケージは事前にインストールされているか、ONTAP Mediator インストーラによって自動的にインストールされます。

RHEL / CentOSのすべてのバージョン	RHEL 10.x / Rocky Linux 10 用の追加パッケージ	RHEL 9.x / Rocky Linux 9 用の追加パッケージ	RHEL 8.x / Rocky Linux 8 用の追加パッケージ
<ul style="list-style-type: none"> • openssl • openssl-devel • kernel-devel-\$ (uname -r) • gcc • make • libselinux-utils • パッチ • bzip2 • perl-Data-Dumper • perl-ExtUtils-MakeMaker • efibootmgr • mokutil 	<ul style="list-style-type: none"> • python3.12 • python3.12-devel 	<ul style="list-style-type: none"> • elfutils-libelf-devel • policycoreutils-python-utils • python3 • python3-devel 	<ul style="list-style-type: none"> • elfutils-libelf-devel • policycoreutils-python-utils • redhat-lsb-core • python39 • python39-devel

Mediator インストールパッケージは、次の内容を含む自己解凍型の圧縮 tar ファイルです：

- サポートされているリリースのリポジトリから取得できないすべての依存関係を含むRPMファイル。
- インストール スクリプト。

有効な SSL 証明書を推奨します。

OSのアップグレードに関する考慮事項とカーネルの互換性

- カーネルを除くすべてのライブラリパッケージを更新できますが、ONTAP Mediatorに変更を適用するには再起動が必要になる場合があります。再起動が必要な場合は、ダウンタイムをスケジュールしてください。
- OSカーネルを最新の状態に維持してください。カーネルコアを、"[ONTAP Mediatorのバージョンマトリックス](#)"に記載されているサポート対象バージョンにアップグレードしてください。システムを再起動する必

必要があるため、停止期間に備えてメンテナンス時間を設けてください。

- 再起動する前に SCST カーネルモジュールをアンインストールし、その後再インストールします。
- カーネル OS のアップグレードを開始する前に、再インストールするサポート対象バージョンの SCST を準備します。



- カーネルのバージョンがオペレーティング システムのバージョンと一致している必要があります。
- テスト済みの SCST モジュールが動作しない可能性があるため、ONTAP Mediator リリースでサポートされている OS バージョンを超えてカーネルをアップグレードしないでください。

UEFIセキュアブートが有効になっている場合は**ONTAP Mediator**をインストールします

ONTAP Mediator は、UEFI セキュア ブートが有効になっているかどうかに関係なくシステムにインストールできます。

タスク概要

ONTAP Mediatorが不要な場合、またはONTAP Mediatorのインストールに関する問題をトラブルシューティングする場合は、ONTAP Mediatorをインストールする前にUEFIセキュアブートを無効にすることができます。マシンの設定からUEFIセキュアブートオプションを無効にしてください。



UEFI セキュアブートを無効にする詳細な手順については、ホスト OS のドキュメントを参照してください。

UEFI セキュアブートを有効にした ONTAP Mediator をインストールするには、サービスを開始する前にセキュリティキーを登録する必要があります。このキーは SCST インストールのコンパイルステップで生成され、秘密鍵と公開鍵のペアとしてマシンに保存されます。`mokutil`ユーティリティを使用して、公開鍵をマシンオーナーキー (MOK) として UEFI ファームウェアに追加することで、システムが署名済みモジュールを信頼してロードできるようになります。`mokutil`パスフレーズは安全な場所に保存してください。システムを再起動して MOK をアクティブ化する際に必要になります。

手順

1. システムでUEFIセキュアブートが有効になっているかどうかを確認します：

```
mokutil --sb-state
```

結果は、このシステムでUEFIセキュア ブートが有効になっているかどうかを示します。

状況	移動先...
UEFIセキュアブートが有効になっています	
UEFIセキュアブートが無効になっています	"ホストオペレーティングシステムをアップグレードしてから、ONTAP Mediatorをアップグレードします。"



- パスフレーズを作成するように求められます。パスフレーズは安全な場所に保管しておく必要があります。このパスフレーズは、UEFI Boot Managerでキーを有効にする際に必要になります。
- ONTAP Mediator 1.2.0以前のバージョンでは、このモードはサポートされていません。

2. `mokutil`ユーティリティがインストールされていない場合は、次のコマンドを実行します：

```
yum install mokutil
```

ホストOSとONTAP Mediatorをアップグレードする

ONTAP MediatorのホストOSを新しいバージョンにアップグレードするには、最初にONTAP Mediatorをアンインストールする必要があります。

タスク概要

leapp-upgrade ツールを使用して ONTAP Mediator のホスト OS をアップグレードする前に、ONTAP Mediator をアンインストールしてください。このツールは、登録済みリポジトリ内の新しい RPM バージョンをチェックします。

ONTAP Mediatorインストーラは.rpmファイルをインストールしますが、leapp-upgradeツールはこのファイルを検索対象に含めません。インストーラは登録済みリポジトリからファイルをダウンロードするのではなく、解凍するため、ツールはアップグレードを見つけることができません。パッケージをアンインストールするには、leapp-upgradeツールを使用する必要があります。

手順

1. ログ ファイルをバックアップします：

```
[rootmediator-host ~]# tar -czf ontap_mediator_file_backup.tgz -C /opt/netapp/lib/ontap_mediator ./log
./ontap_mediator/server_config/ontap_mediator.user_config.yaml
[rootmediator-host ~]# tar -tf ontap_mediator_file_backup.tgz
./log/
./log/ontap_mediator.log
./log/scstadmin.log
./log/ontap_mediator_stdout.log
./log/ontap_mediator_requests.log
./log/install_20230419134611.log
./log/scst.log
./log/ontap_mediator_syslog.log
./ontap_mediator/server_config/ontap_mediator.user_config.yaml
[rootmediator-host ~]#
```

2. leapp-upgrade ツールを使用してアップグレードを実行します：

```
[rootmediator-host ~]# leapp preupgrade --target 8.4
  ..<snip upgrade checks>..
  ..<fix issues found>..
[rootmediator-host ~]# leapp upgrade --target 8.4
  ..<snip upgrade>..
[rootmediator-host ~]# cat /etc/os-release | head -2
NAME="Red Hat Enterprise Linux"
VERSION="8.4 (Ootpa)"
[rootmediator-host ~]#
```

3. ONTAP Mediatorを再インストールします：



ログ ファイルが失われないように、ONTAP Mediatorの再インストール後すぐに残りの手順を実行してください。

```
[rootmediator-host ~]# ontap-mediator-1.11.0/ontap-mediator-1.11.0

ONTAP Mediator: Self Extracting Installer

  ..<snip installation>..
[rootmediator-host ~]#
```

4. ontap_mediatorを停止します：

```
[rootmediator-host ~]# systemctl stop ontap_mediator
[rootmediator-host ~]#
```

5. ログ ファイルを置き換えます：

```
[rootmediator-host ~]# tar -xf ontap_mediator_log_backup.tgz -C
/opt/netapp/lib/ontap_mediator
[rootmediator-host ~]#
```

6. ontap_mediatorを起動します：

```
[rootmediator-host ~]# systemctl start ontap_mediator
[rootmediator-host ~]#
```

7. すべてのONTAPクラスタをアップグレードしたONTAP Mediatorに再接続します：

IP経由のMetroCluster

```
siteA::> metrocluster configuration-settings mediator show
Mediator IP      Port      Node      Configuration
Connection
Status          Status
-----
-----
172.31.40.122
          31784  siteA-node2  true      false
          siteA-nod1  true      false
          siteB-node2  true      false
          siteB-node2  true      false

siteA::> metrocluster configuration-settings mediator remove
Removing the mediator and disabling Automatic Unplanned Switchover. It
may take a few minutes to complete.
Please enter the username for the mediator: mediatoradmin
Please enter the password for the mediator:
Confirm the mediator password:
Automatic Unplanned Switchover is disabled for all nodes...
Removing mediator mailboxes...
Successfully removed the mediator.

siteA::> metrocluster configuration-settings mediator add -mediator
-address 172.31.40.122
Adding the mediator and enabling Automatic Unplanned Switchover. It may
take a few minutes to complete.
Please enter the username for the mediator: mediatoradmin
Please enter the password for the mediator:
Confirm the mediator password:
Successfully added the mediator.

siteA::> metrocluster configuration-settings mediator show
Mediator IP      Port      Node      Configuration
Connection
Status          Status
-----
-----
172.31.40.122
          31784  siteA-node2  true      true
          siteA-nod1  true      true
          siteB-node2  true      true
          siteB-node2  true      true

siteA::>
```

SnapMirrorアクティブ同期

SnapMirror Active Sync の場合、/opt/netapp の外部に保存されている TLS 証明書を再インストールする必要はありません。/opt/netapp に保存されている証明書をバックアップして復元してください。

```
peer1::> snapmirror mediator show
Mediator Address Peer Cluster      Connection Status Quorum Status
-----
172.31.49.237    peer2                unreachable      true

peer1::> snapmirror mediator remove -mediator-address 172.31.49.237
-peer-cluster peer2

Info: [Job 39] 'mediator remove' job queued

peer1::> job show -id 39

Job ID Name                Owing
Vserver      Node                State
-----
39    mediator remove    peer1    peer1-node1    Success
Description: Removing entry in mediator

peer1::> security certificate show -common-name ONTAPMediatorCA
Vserver      Serial Number  Certificate Name                Type
-----
peer1
4A790360081F41145E14C5D7CE721DC6C210007F
ONTAPMediatorCA                server-
ca
Certificate Authority: ONTAP Mediator CA
Expiration Date: Mon Apr 17 10:27:54 2073

peer1::> security certificate delete -common-name ONTAPMediatorCA *
1 entry was deleted.

peer1::> security certificate install -type server-ca -vserver peer1

Please enter Certificate: Press <Enter> when done
..<snip ONTAP Mediator CA public key>..

You should keep a copy of the CA-signed digital certificate for future
reference.

The installed certificate's CA and serial number for reference:
CA: ONTAP Mediator CA
serial: 44786524464C5113D5EC966779D3002135EA4254
```

The certificate's generated name for reference: ONTAPMediatorCA

```
peer2::> security certificate delete -common-name ONTAPMediatorCA *
1 entry was deleted.
```

```
peer2::> security certificate install -type server-ca -vserver peer2
```

```
Please enter Certificate: Press <Enter> when done
..<snip ONTAP Mediator CA public key>..
```

You should keep a copy of the CA-signed digital certificate for future reference.

The installed certificate's CA and serial number for reference:

CA: ONTAP Mediator CA

serial: 44786524464C5113D5EC966779D3002135EA4254

The certificate's generated name for reference: ONTAPMediatorCA

```
peer1::> snapmirror mediator add -mediator-address 172.31.49.237 -peer
-cluster peer2 -username mediatoradmin
```

Notice: Enter the mediator password.

Enter the password:

Enter the password again:

Info: [Job: 43] 'mediator add' job queued

```
peer1::> job show -id 43
```

Job ID	Name	Owning Vserver	Node	State
43	mediator add	peer1	peer1-node2	Success

Description: Creating a mediator entry

```
peer1::> snapmirror mediator show
```

Mediator Address	Peer	Cluster	Connection Status	Quorum Status
172.31.49.237	peer2		connected	true

```
peer1::>
```

関連情報

- ["security certificate delete"](#)
- ["security certificate install"](#)
- ["セキュリティ証明書の表示"](#)
- ["SnapMirror Mediator add"](#)
- ["SnapMirror Mediator の削除"](#)
- ["storage iscsi-initiator show"](#)

ONTAP Mediatorのインストールにリポジトリアクセスを提供する

ONTAP Mediator がインストール プロセス中に必要なパッケージにアクセスできるように、リポジトリへのアクセスを有効にする必要があります。

手順

1. 次の表を参照して、アクセスする必要があるリポジトリを特定します。

オペレーティング システム	アクセスが必要なリポジトリ
RHEL 10.x	<ul style="list-style-type: none">• rhel-10-for-x86_64-baseos-rpms• rhel-10-for-x86_64-appstream-rpms
RHEL 9.x	<ul style="list-style-type: none">• rhel-9-for-x86_64-baseos-rpms• rhel-9-for-x86_64-appstream-rpms
RHEL 8.x	<ul style="list-style-type: none">• rhel-8-for-x86_64-baseos-rpms• rhel-8-for-x86_64-appstream-rpms
RHEL 7.x	<ul style="list-style-type: none">• rhel-7-server-optional-rpms
CentOS 7.x	<ul style="list-style-type: none">• C7.6.1810 - Baseリポジトリ
Rocky Linux 10	<ul style="list-style-type: none">• appstream• baseos
Rocky Linux 9	<ul style="list-style-type: none">• appstream• baseos
Rocky Linux 8	<ul style="list-style-type: none">• appstream• baseos

2. インストール中にONTAP Mediatorが必要なパッケージにアクセスできるように、次のいずれかの手順を実行して上記のリポジトリへのアクセスを有効にします。



ONTAP Mediator が「extras」リポジトリと「optional」リポジトリにある Python モジュールに依存している場合は、`rhel-X-for-x86_64-extras-rpms` および `rhel-X-for-x86_64-optional-rpms` ファイルにアクセスする必要がある場合があります。

RHEL 10.x オペレーティング システムの手順

オペレーティング システムが **RHEL 10.x** の場合、リポジトリへのアクセスを有効にするには、次の手順を使用します：

手順

1. 必要なリポジトリにサブスクライブします。

```
subscription-manager repos --enable rhel-10-for-x86_64-baseos-rpms
```

```
subscription-manager repos --enable rhel-10-for-x86_64-appstream-rpms
```

次の例は、このコマンドの実行例を示しています。

```
[root@localhost ~]# subscription-manager repos --enable rhel-10-for-x86_64-baseos-rpms
Repository 'rhel-10-for-x86_64-baseos-rpms' is enabled for this system.
[root@localhost ~]# subscription-manager repos --enable rhel-10-for-x86_64-appstream-rpms
Repository 'rhel-10-for-x86_64-appstream-rpms' is enabled for this system.
```

2. `yum repolist` コマンドを実行します。

新しくサブスクライブしたリポジトリがリストに表示されます。

オペレーティング システムが **RHEL 9.x** の場合、リポジトリへのアクセスを有効にするには、次の手順を使用します：

手順

1. 必要なリポジトリにサブスクライブします。

```
subscription-manager repos --enable rhel-9-for-x86_64-baseos-rpms
```

```
subscription-manager repos --enable rhel-9-for-x86_64-appstream-rpms
```

次の例は、このコマンドの実行例を示しています。

```
[root@localhost ~]# subscription-manager repos --enable rhel-9-for-x86_64-baseos-rpms
Repository 'rhel-9-for-x86_64-baseos-rpms' is enabled for this system.
[root@localhost ~]# subscription-manager repos --enable rhel-9-for-x86_64-appstream-rpms
Repository 'rhel-9-for-x86_64-appstream-rpms' is enabled for this system.
```

2. `yum repolist` コマンドを実行します。

新しくサブスクライブしたリポジトリがリストに表示されます。

オペレーティング システムが **RHEL 8.x** の場合、リポジトリへのアクセスを有効にするには、次の手順を使用します：

手順

1. 必要なリポジトリにサブスクライブします。

```
subscription-manager repos --enable rhel-8-for-x86_64-baseos-rpms
```

```
subscription-manager repos --enable rhel-8-for-x86_64-appstream-rpms
```

次の例は、このコマンドの実行例を示しています。

```
[root@localhost ~]# subscription-manager repos --enable rhel-8-for-x86_64-baseos-rpms
Repository 'rhel-8-for-x86_64-baseos-rpms' is enabled for this system.
[root@localhost ~]# subscription-manager repos --enable rhel-8-for-x86_64-appstream-rpms
Repository 'rhel-8-for-x86_64-appstream-rpms' is enabled for this system.
```

2. `yum repolist` コマンドを実行します。

新しくサブスクライブしたリポジトリがリストに表示されます。

オペレーティング システムが **RHEL 7.x** の場合、リポジトリへのアクセスを有効にするには、次の手順を使用します：

手順

1. 必要なリポジトリにサブスクライブします。

```
subscription-manager repos --enable rhel-7-server-optional-rpms
```

次の例は、このコマンドの実行例を示しています。

```
[root@localhost ~]# subscription-manager repos --enable rhel-7-
server-optional-rpms
Repository 'rhel-7-server-optional-rpms' is enabled for this system.
```

2. `yum repolist` コマンドを実行します。

次の例は、このコマンドの実行例を示しています。「rhel-7-server-optional-rpms」リポジトリがリストに表示されます。

```
[root@localhost ~]# yum repolist
Loaded plugins: product-id, search-disabled-repos, subscription-
manager
rhel-7-server-optional-rpms | 3.2 kB  00:00:00
rhel-7-server-rpms | 3.5 kB  00:00:00
(1/3): rhel-7-server-optional-rpms/7Server/x86_64/group
| 26 kB  00:00:00
(2/3): rhel-7-server-optional-rpms/7Server/x86_64/updateinfo
| 2.5 MB  00:00:00
(3/3): rhel-7-server-optional-rpms/7Server/x86_64/primary_db
| 8.3 MB  00:00:01
repo id                                repo name
status
rhel-7-server-optional-rpms/7Server/x86_64  Red Hat Enterprise
Linux 7 Server - Optional (RPMs)  19,447
rhel-7-server-rpms/7Server/x86_64          Red Hat Enterprise
Linux 7 Server (RPMs)                26,758
repolist: 46,205
[root@localhost ~]#
```

オペレーティング システムが **CentOS 7.x** の場合、リポジトリへのアクセスを有効にするには、次の手順に従います：



次の例は、CentOS 7.6用のリポジトリを示しており、その他のCentOSバージョンでは使用できない可能性があります。お使いのバージョンのCentOS用のBaseリポジトリを使用してください。

手順

1. C7.6.1810 - Baseリポジトリを追加します。C7.6.1810 - Base保管用リポジトリには、ONTAP Mediatorに必要な「kernel-devel」パッケージが含まれています。
2. `/etc/yum.repos.d/CentOS-Vault.repo`に次の行を追加します。

```
[C7.6.1810-base]
name=CentOS-7.6.1810 - Base
baseurl=http://vault.centos.org/7.6.1810/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
enabled=1
```

3. `yum repolist` コマンドを実行します。

次の例は、このコマンドの実行例を示しています。CentOS-7.6.1810 - Baseリポジトリがリストに表示されます。

```
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: distro.ibiblio.org
* extras: distro.ibiblio.org
* updates: ewr.edge.kernel.org
C7.6.1810-base | 3.6 kB 00:00:00
(1/2): C7.6.1810-base/x86_64/group_gz | 166 kB 00:00:00
(2/2): C7.6.1810-base/x86_64/primary_db | 6.0 MB 00:00:04
repo id repo name status
C7.6.1810-base/x86_64 CentOS-7.6.1810 - Base 10,019
base/7/x86_64 CentOS-7 - Base 10,097
extras/7/x86_64 CentOS-7 - Extras 307
updates/7/x86_64 CentOS-7 - Updates 1,010
repolist: 21,433
[root@localhost ~]#
```

オペレーティングシステムが **Rocky Linux 10**、**Rocky Linux 9**、または **Rocky Linux 8** の場合、リポジトリへのアクセスを有効にするには、次の手順に従います：

手順

1. 必要なリポジトリにサブスクライブします。

```
dnf config-manager --set-enabled baseos
```

```
dnf config-manager --set-enabled appstream
```

2. `clean`操作を実行します：

```
dnf clean all
```

3. リポジトリのリストを確認します。

```
dnf repolist
```

```
[root@localhost ~]# dnf config-manager --set-enabled baseos
[root@localhost ~]# dnf config-manager --set-enabled appstream
[root@localhost ~]# dnf clean all
[root@localhost ~]# dnf repolist
repo id                repo name
appstream              Rocky Linux 10 - AppStream
baseos                 Rocky Linux 10 - BaseOS
[root@localhost ~]#
```

```
[root@localhost ~]# dnf config-manager --set-enabled baseos
[root@localhost ~]# dnf config-manager --set-enabled appstream
[root@localhost ~]# dnf clean all
[root@localhost ~]# dnf repolist
repo id                repo name
appstream              Rocky Linux 9 - AppStream
baseos                 Rocky Linux 9 - BaseOS
[root@localhost ~]#
```

```
[root@localhost ~]# dnf config-manager --set-enabled baseos
[root@localhost ~]# dnf config-manager --set-enabled appstream
[root@localhost ~]# dnf clean all
[root@localhost ~]# dnf repolist
repo id                repo name
appstream              Rocky Linux 8 - AppStream
baseos                 Rocky Linux 8 - BaseOS
[root@localhost ~]#
```

ONTAPメディアエーターのインストールパッケージをダウンロードします

ONTAP Mediator インストール パッケージをダウンロードしてインストールします。

手順

1. ONTAP Mediator ダウンロード ページから ONTAP Mediator インストール パッケージをダウンロードします。

["ONTAP Mediatorのダウンロードページ"](#)

2. Mediator インストールパッケージが現在の作業ディレクトリに配置されていることを確認してください：

```
[root@sdot-r730-0003a-d6 ~]# ls ontap-mediator-1.11.0.tgz
```

```
ontap-mediator-1.11.0.tgz
```



ONTAP Mediator バージョン 1.4 以前では、インストーラの名前は `ontap-mediator` です。

システムがインターネットにアクセスできない場合は、インストーラーが必要なパッケージにアクセスできることを確認してください。

3. 必要に応じて、Mediator インストール パッケージをインストール ディレクトリに移動します。
4. インストール パッケージを解凍します。

```
tar xvfz ontap-mediator-1.11.0.tgz
```

```
ontap-mediator-1.11.0/  
ontap-mediator-1.11.0/csc-prod-ONTAP-Mediator.pem  
ontap-mediator-1.11.0/csc-prod-chain-ONTAP-Mediator.pem  
ontap-mediator-1.11.0/tsa-prod-ONTAP-Mediator.pem  
ontap-mediator-1.11.0/tsa-prod-chain-ONTAP-Mediator.pem  
ontap-mediator-1.11.0/ONTAP-Mediator-production.pub  
ontap-mediator-1.11.0/ontap-mediator-1.11.0  
ontap-mediator-1.11.0/ontap-mediator-1.11.0.sig.tsr  
ontap-mediator-1.11.0/ontap-mediator-1.11.0.tsr  
ontap-mediator-1.11.0/ontap-mediator-1.11.0.sig
```

ONTAP Mediatorのコード署名の検証

NetAppでは、インストール前にONTAP Mediatorのコード署名を検証することを推奨しています。この手順はオプションです。

開始する前に

ONTAP Mediator コード署名を検証する前に、システムがこれらの要件を満たしていることを確認してください。



- 2025年6月15日以降、コード署名検証証明書の有効期限が切れているため、ONTAP Mediator 1.9および1.8のインストールまたはアップグレードはできません。代わりに、ONTAP Mediator 1.11または1.10をインストールまたはアップグレードしてください。
- システムが以下の要件を満たしていない場合、検証プロセスは必要なく、直接"[ONTAP Mediatorのインストール パッケージのインストール](#)"に進むことができます。

- OpenSSLバージョン1.0.2~3.0：基本的な検証の場合
- OpenSSLバージョン1.1.0以降：時刻認証局（TSA）の処理の場合
- パブリック インターネットへのアクセス：OCSP検証の場合

ダウンロード パッケージには次のファイルが含まれています：

ファイル	概要
ONTAP-Mediator-production.pub	署名の検証に使用された公開鍵
csc-prod-chain-ONTAP-Mediator.pem	公開証明書CAの信頼チェーン
csc-prod-ONTAP-Mediator.pem	キーの生成に使用された証明書
ontap-mediator-1.11.0	バージョン1.11の製品インストール実行ファイル

ontap-mediator-1.11.0.sig	SHA-256でハッシュ化され、csc-prodキーを使用してRSA署名された、インストーラの署名
ontap-mediator-1.11.0.sig.tsr	OCSPがインストーラの署名に使用する失効要求
ontap-mediator-1.11.0.tsr	タイムスタンプ署名要求ファイル
tsa-prod-ONTAP-Mediator.pem	TSRの公開証明書
tsa-prod-chain-ONTAP-Mediator.pem	TSRの公開証明書のCAチェーン

手順

1. `csc-prod-ONTAP-Mediator.pem`でオンライン証明書ステータス プロトコル (OCSP) を使用して失効チェックを実行します。

- a. 証明書のOCSP URLを見つけます。開発者証明書ではURIが提供されない場合があります：

```
openssl x509 -noout -ocsp_uri -in csc-prod-chain-ONTAP-Mediator.pem
```

- b. 証明書のOCSP要求を生成します。

```
openssl ocsf -issuer csc-prod-chain-ONTAP-Mediator.pem -CAfile csc-prod-chain-ONTAP-Mediator.pem -cert csc-prod-ONTAP-Mediator.pem -reqout req.der
```

- c. OCSP Managerに接続してOCSP要求を送信します。

```
openssl ocsf -issuer csc-prod-chain-ONTAP-Mediator.pem -CAfile csc-prod-chain-ONTAP-Mediator.pem -cert csc-prod-ONTAP-Mediator.pem -url ${ocsp_uri} -resp_text -respout resp.der -verify_other csc-prod-chain-ONTAP-Mediator.pem
```

2. ローカル ホストに照らしてCSCの信頼チェーンと有効期限を検証します。

```
openssl verify
```



PATHからのopensslバージョンには、有効な`cert.pem` (自己署名ではない) が必要です。

```
openssl verify -untrusted csc-prod-chain-ONTAP-Mediator.pem -CApath
${OPENSSLDIR} csc-prod-ONTAP-Mediator.pem # Failure action: The Code-
Signature-Check certificate has expired or is invalid. Download a newer
version of the ONTAP Mediator.
openssl verify -untrusted tsa-prod-chain-ONTAP-Mediator.pem -CApath
${OPENSSLDIR} tsa-prod-ONTAP-Mediator.pem # Failure action: The Time-
Stamp certificate has expired or is invalid. Download a newer version of
the ONTAP Mediator.
```

3. 関連する証明書を使用して、`ontap-mediator-1.11.0.sig.tsr`および`ontap-mediator-1.11.0.tsr`ファイルを検証します：

OpenSSL 3.x

```
openssl ts -verify -data ontap-mediator-1.11.0.sig -in ontap-mediator-
1.11.0.sig.tsr -CAfile tsa-prod-chain-ONTAP-Mediator.pem -untrusted tsa-
prod-ONTAP-Mediator.pem
```

OpenSSL 1.x

```
openssl ts -verify -data ontap-mediator-1.11.0 -in ontap-mediator-
1.11.0.tsr -CAfile tsa-prod-chain-ONTAP-Mediator.pem -partial_chain
```



`tsr`ファイルには、インストーラに関連付けられたタイムスタンプレスポンスとコード署名が含まれています。処理では、タイムスタンプにTSAの有効な署名があり、入力ファイルが変更されていないことを確認します。検証はマシン上でローカルに実行されます。TSAサーバーにアクセスする必要はありません。

4. キーに照らして署名を検証します。

```
openssl -dgst -verify
```

```
openssl dgst -sha256 -verify ONTAP-Mediator-production.pub -signature
ontap-mediator-1.11.0.sig ontap-mediator-1.11.0
```

ONTAP Mediatorのインストール パッケージのインストール

ONTAP Mediator をインストールするには、インストールパッケージを取得し、ホスト上でインストーラを実行する必要があります。

手順

1. インストーラを実行し、必要に応じてプロンプトに応答します。

```
./ontap-mediator-1.11.0/ontap-mediator-1.11.0 -y
```

```
[root@scs000099753 ~]# ./ontap-mediator-1.11.0/ontap-mediator-1.11.0 -y
```



インストール中に署名のチェックをスキップするには、次のコマンドを使用します：

```
./ontap-mediator-1.11.0/ontap-mediator-1.11.0 -y --skip-code  
-signature-check
```

インストーラーは必要なアカウントを作成し、必要なパッケージをインストールします。Mediatorが既にインストールされている場合は、アップグレードを促すメッセージが表示されます。

ONTAP Mediatorのインストール例 (コンソール出力)

```
[root@mediator_host ~]# tar -zxvf ontap-mediator-1.11.0.tgz
ontap-mediator-1.11.0/
ontap-mediator-1.11.0/csc-prod-chain-ONTAP-Mediator.pem
ontap-mediator-1.11.0/csc-prod-ONTAP-Mediator.pem
ontap-mediator-1.11.0/tsa-prod-ONTAP-Mediator.pem
ontap-mediator-1.11.0/tsa-prod-chain-ONTAP-Mediator.pem
ontap-mediator-1.11.0/ONTAP-Mediator-production.pub
ontap-mediator-1.11.0/ontap-mediator-1.11.0
ontap-mediator-1.11.0/ontap-mediator-1.11.0.sig.tsr
ontap-mediator-1.11.0/ontap-mediator-1.11.0.tsr
ontap-mediator-1.11.0/ontap-mediator-1.11.0.sig
[root@mediator_host ~]# ./ontap-mediator-1.11.0/ontap-mediator-1.11.0

ONTAP Mediator: Self Extracting Installer

+ Extracting the ONTAP Mediator installation/upgrade archive
+ Performing the ONTAP Mediator run-time code signature check
  Using openssl from the path: /usr/bin/openssl configured for
CApath:/etc/pki/tls
Error querying OCSP responder
80BBA032607F0000:error:1E800080:HTTP
routines:OSSL_HTTP_REQ_CTX_nbio:failed reading
data:crypto/http/http_client.c:549:
80BBA032607F0000:error:1E800067:HTTP
routines:OSSL_HTTP_REQ_CTX_exchange:error
receiving:crypto/http/http_client.c:901:server=http://ocsp.entrust.net:
80
  WARNING: The OCSP check failed while attempting to test the Code-
Signature-Check certificate
  Continue without code signature checking (only recommended if
integrity has been established manually)? yes/no: yes
  SKIPPING: Code signature check, manual override due to lack of OCSP
response
+ Unpacking the ONTAP Mediator installer

ONTAP Mediator requires two user accounts. One for the service
(netapp), and one for use by ONTAP to the mediator API (mediatoradmin).
Would you like to use the default account names: netapp +
mediatoradmin? (Y(es)/n(o)): yes

Enter ONTAP Mediator user account (mediatoradmin) password:
```

Re-Enter ONTAP Mediator user account (mediatoradmin) password:

+ Checking if SELinux is in enforcing mode

The installer will change the SELinux context type of /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi from type 'lib_t' to 'bin_t'.

+ Checking for default Linux firewall

+ Installing required packages.

Updating Subscription Management repositories.

Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Last metadata expiration check: 5 days, 14:34:13 ago on Thu 10 Jul 2025 01:28:32 AM EDT.

Package openssl-1:3.2.2-16.el10.x86_64 is already installed.

Package libselinux-utils-3.8-1.el10.x86_64 is already installed.

Package perl-Data-Dumper-2.189-512.el10.x86_64 is already installed.

Package bzip2-1.0.8-25.el10.x86_64 is already installed.

Package efibootmgr-18-8.el10.x86_64 is already installed.

Package mokutil-2:0.6.0-11.el10.x86_64 is already installed.

Package polycycoreutils-python-utils-3.8-1.el10.noarch is already installed.

Package python3-3.12.9-1.el10.x86_64 is already installed.

Dependencies resolved.

=====
=====
=====
=====

Package	Version
Architecture	Size
Repository	
=====	
=====	
=====	
=====	

Installing:

elfutils-libelf-devel	
x86_64	0.192-5.el10

```

AppStream 50 k
gcc
x86_64 14.2.1-7.e110
AppStream 37 M
kernel-devel
x86_64 6.12.0-55.9.1.e110_0
AppStream 22 M
make
x86_64 1:4.4.1-9.e110
BaseOS 591 k
openssl-devel
x86_64 1:3.2.2-16.e110
AppStream 3.9 M
patch
x86_64 2.7.6-26.e110
AppStream 134 k
perl-ExtUtils-MakeMaker
noarch 2:7.70-513.e110
AppStream 297 k
python3-devel
x86_64 3.12.9-1.e110
AppStream
334 k
python3-pip
noarch 23.3.2-7.e110
AppStream 3.2 M
Installing dependencies:
annobin-docs
noarch 12.92-1.e110
AppStream 94 k
annobin-plugin-gcc
x86_64 12.92-1.e110
AppStream 985 k
bison
x86_64 3.8.2-9.e110
AppStream 1.0 M
cmake-filesystem
x86_64 3.30.5-2.e110
AppStream 29 k
cpp
x86_64 14.2.1-7.e110
AppStream 12 M
dwz
x86_64 0.15-7.e110
AppStream 139 k
efi-srpm-macros

```

noarch	6-6.e110
AppStream	25 k
flex	
x86_64	2.6.4-19.e110
AppStream	303 k
fonts-srpm-macros	
noarch	1:2.0.5-18.e110
AppStream	29 k
forge-srpm-macros	
noarch	0.4.0-6.e110
AppStream	23 k
gcc-plugin-annobin	
x86_64	14.2.1-7.e110
AppStream	62 k
glibc-devel	
x86_64	2.39-37.e110
AppStream	641 k
go-srpm-macros	
noarch	3.6.0-4.e110
AppStream	29 k
kernel-headers	
x86_64	6.12.0-55.9.1.e110_0
AppStream	2.3 M
kernel-srpm-macros	
noarch	1.0-25.e110
AppStream	11 k
libxcrypt-devel	
x86_64	4.4.36-10.e110
AppStream	33 k
libzstd-devel	
x86_64	1.5.5-9.e110
AppStream	53 k
lua-srpm-macros	
noarch	1-15.e110
AppStream	10 k
m4	
x86_64	1.4.19-11.e110
AppStream	309 k
ocaml-srpm-macros	
noarch	10-4.e110
AppStream	10 k
openblas-srpm-macros	
noarch	2-19.e110
AppStream	9.0 k
package-notes-srpm-macros	

noarch	0.5-13.e110
AppStream	11 k
perl-AutoSplit	
noarch	5.74-512.e110
AppStream	23 k
perl-Benchmark	
noarch	1.25-512.e110
AppStream	28 k
perl-CPAN-Meta-Requirements	
noarch	2.143-11.e110
AppStream	39 k
perl-CPAN-Meta-YAML	
noarch	0.018-512.e110
AppStream	29 k
perl-Devel-PPPort	
x86_64	3.72-512.e110
AppStream	223 k
perl-ExtUtils-Command	
noarch	2:7.70-513.e110
AppStream	16 k
perl-ExtUtils-Constant	
noarch	0.25-512.e110
AppStream	47 k
perl-ExtUtils-Install	
noarch	2.22-511.e110
AppStream	47 k
perl-ExtUtils-Manifest	
noarch	1:1.75-511.e110
AppStream	37 k
perl-ExtUtils-ParseXS	
noarch	1:3.51-512.e110
AppStream	190 k
perl-File-Compare	
noarch	1.100.800-512.e110
AppStream	15 k
perl-File-Copy	
noarch	2.41-512.e110
AppStream	22 k
perl-I18N-Langinfo	
x86_64	0.24-512.e110
AppStream	28 k
perl-JSON-PP	
noarch	1:4.16-512.e110
AppStream	69 k
perl-Test-Harness	
noarch	1:3.48-512.e110

```

AppStream 288 k
  perl-lib
x86_64 0.65-512.e110
AppStream 16 k
  perl-srpm-macros
noarch 1-57.e110
AppStream 9.7 k
  perl-version
x86_64 8:0.99.32-4.e110
AppStream 68 k
  pyproject-srpm-macros
noarch 1.16.2-1.e110
AppStream 16 k
  python-srpm-macros
noarch 3.12-9.1.e110
AppStream 26 k
  python3-pyparsing
noarch 3.1.1-7.e110
BaseOS 273 k
  qt6-srpm-macros
noarch 6.8.1-3.e110
AppStream
  11 k
  redhat-rpm-config
noarch 288-1.e110
AppStream 83 k
  rust-toolset-srpm-macros
noarch 1.84.1-1.e110
AppStream 13 k
  systemtap-sdt-devel
x86_64 5.2-2.e110
AppStream 78 k
  systemtap-sdt-dtrace
x86_64 5.2-2.e110
AppStream 72 k
  zlib-ng-compat-devel
x86_64 2.2.3-1.e110
AppStream 41 k
Installing weak dependencies:
  perl-CPAN-Meta
noarch 2.150010-511.e110
AppStream 202 k
  perl-Encode-Locale
noarch 1.05-31.e110
AppStream 21 k
  perl-Time-HiRes

```

```

x86_64                                4:1.9777-511.el10
AppStream                               62 k
  perl-devel
x86_64                                4:5.40.1-512.el10
AppStream                               772 k
  perl-doc
noarch                                  5.40.1-512.el10
AppStream                               4.9 M

```

Transaction Summary

```

=====
=====
=====
=====

```

Install 63 Packages

Total size: 94 M

Installed size: 282 M

Downloading Packages:

BaseOS Packages Red Hat Enterprise Linux 10

439 kB/s | 3.7 kB 00:00

Importing GPG key 0xFD431D51:

Userid : "Red Hat, Inc. (release key 2) <security@redhat.com>"

Fingerprint: 567E 347A D004 4ADE 55BA 8A5F 199E 2F91 FD43 1D51

From : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

Key imported successfully

Importing GPG key 0x5A6340B3:

Userid : "Red Hat, Inc. (auxiliary key 3) <security@redhat.com>"

Fingerprint: 7E46 2425 8C40 6535 D56D 6F13 5054 E4A4 5A63 40B3

From : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

Key imported successfully

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

Preparing :

1/1

Installing : perl-version-8:0.99.32-4.el10.x86_64

1/63

Installing : perl-File-Copy-2.41-512.el10.noarch

2/63

Installing : perl-CPAN-Meta-Requirements-2.143-11.el10.noarch

3/63

Installing : perl-Time-HiRes-4:1.9777-511.el10.x86_64

4/63

```
Installing      : perl-JSON-PP-1:4.16-512.el10.noarch
5/63
Installing      : perl-File-Compare-1.100.800-512.el10.noarch
6/63
Installing      : perl-ExtUtils-ParseXS-1:3.51-512.el10.noarch
7/63
Installing      : m4-1.4.19-11.el10.x86_64
8/63
Installing      : make-1:4.4.1-9.el10.x86_64
9/63
Installing      : bison-3.8.2-9.el10.x86_64
10/63
Installing      : flex-2.6.4-19.el10.x86_64
11/63
Installing      : perl-ExtUtils-Command-2:7.70-513.el10.noarch
12/63
Installing      : perl-ExtUtils-Manifest-1:1.75-511.el10.noarch
13/63
Installing      : systemtap-sdt-devel-5.2-2.el10.x86_64
14/63
Installing      : rust-toolset-srpm-macros-1.84.1-1.el10.noarch
15/63
Installing      : qt6-srpm-macros-6.8.1-3.el10.noarch
16/63
Installing      : python3-pip-23.3.2-7.el10.noarch
17/63
Installing      : pyproject-srpm-macros-1.16.2-1.el10.noarch
18/63
Installing      : perl-srpm-macros-1-57.el10.noarch
19/63
Installing      : perl-lib-0.65-512.el10.x86_64
20/63
Installing      : perl-doc-5.40.1-512.el10.noarch
21/63
Installing      : perl-I18N-Langinfo-0.24-512.el10.x86_64
22/63
Installing      : perl-Encode-Locale-1.05-31.el10.noarch
23/63
Installing      : perl-ExtUtils-Constant-0.25-512.el10.noarch
24/63
Installing      : perl-Devel-PPPort-3.72-512.el10.x86_64
25/63
Installing      : perl-CPAN-Meta-YAML-0.018-512.el10.noarch
26/63
Installing      : perl-CPAN-Meta-2.150010-511.el10.noarch
27/63
```

```
Installing      : perl-Benchmark-1.25-512.el10.noarch
28/63
Installing      : perl-Test-Harness-1:3.48-512.el10.noarch
29/63
Installing      : perl-AutoSplit-5.74-512.el10.noarch
30/63
Installing      : package-notes-srpm-macros-0.5-13.el10.noarch
31/63
Installing      : openssl-devel-1:3.2.2-16.el10.x86_64
32/63
Installing      : openblas-srpm-macros-2-19.el10.noarch
33/63
Installing      : ocaml-srpm-macros-10-4.el10.noarch
34/63
Installing      : lua-srpm-macros-1-15.el10.noarch
35/63
Installing      : libzstd-devel-1.5.5-9.el10.x86_64
36/63
Installing      : kernel-srpm-macros-1.0-25.el10.noarch
37/63
Installing      : kernel-headers-6.12.0-55.9.1.el10_0.x86_64
38/63
Installing      : libxcrypt-devel-4.4.36-10.el10.x86_64
39/63
Installing      : glibc-devel-2.39-37.el10.x86_64
40/63
Installing      : efi-srpm-macros-6-6.el10.noarch
41/63
Installing      : dwz-0.15-7.el10.x86_64
42/63
Installing      : cpp-14.2.1-7.el10.x86_64
43/63
Installing      : gcc-14.2.1-7.el10.x86_64
44/63
Installing      : gcc-plugin-annobin-14.2.1-7.el10.x86_64
45/63
Installing      : cmake-filesystem-3.30.5-2.el10.x86_64
46/63
Installing      : zlib-ng-compat-devel-2.2.3-1.el10.x86_64
47/63
Installing      : elfutils-libelf-devel-0.192-5.el10.x86_64
48/63
Installing      : annobin-docs-12.92-1.el10.noarch
49/63
Installing      : annobin-plugin-gcc-12.92-1.el10.x86_64
50/63
```

```

Installing      : fonts-srpm-macros-1:2.0.5-18.el10.noarch
51/63
Installing      : forge-srpm-macros-0.4.0-6.el10.noarch
52/63
Installing      : go-srpm-macros-3.6.0-4.el10.noarch
53/63
Installing      : python-srpm-macros-3.12-9.1.el10.noarch
54/63
Installing      : redhat-rpm-config-288-1.el10.noarch
55/63
Running scriptlet: redhat-rpm-config-288-1.el10.noarch
55/63
Installing      : python3-pyparsing-3.1.1-7.el10.noarch
56/63
Installing      : systemtap-sdt-dtrace-5.2-2.el10.x86_64
57/63
Installing      : perl-devel-4:5.40.1-512.el10.x86_64
58/63
Installing      : perl-ExtUtils-Install-2.22-511.el10.noarch
59/63
Installing      : perl-ExtUtils-MakeMaker-2:7.70-513.el10.noarch
60/63
Installing      : kernel-devel-6.12.0-55.9.1.el10_0.x86_64
61/63
Running scriptlet: kernel-devel-6.12.0-55.9.1.el10_0.x86_64
61/63
Installing      : python3-devel-3.12.9-1.el10.x86_64
62/63
Installing      : patch-2.7.6-26.el10.x86_64
63/63
Running scriptlet: patch-2.7.6-26.el10.x86_64
63/63
Installed products updated.

```

Installed:

```

annobin-docs-12.92-1.el10.noarch          annobin-plugin-gcc-
12.92-1.el10.x86_64                      bison-3.8.2-9.el10.x86_64
cmake-filesystem-3.30.5-2.el10.x86_64    cpp-14.2.1-
7.el10.x86_64
dwz-0.15-7.el10.x86_64                   efi-srpm-macros-6-
6.el10.noarch                            elfutils-libelf-devel-0.192-
5.el10.x86_64  flex-2.6.4-19.el10.x86_64          fonts-
srpm-macros-1:2.0.5-18.el10.noarch
forge-srpm-macros-0.4.0-6.el10.noarch     gcc-14.2.1-
7.el10.x86_64                            gcc-plugin-annobin-14.2.1-
7.el10.x86_64  glibc-devel-2.39-37.el10.x86_64      go-

```

```

srpm-macros-3.6.0-4.el10.noarch
  kernel-devel-6.12.0-55.9.1.el10_0.x86_64      kernel-headers-6.12.0-
55.9.1.el10_0.x86_64      kernel-srpm-macros-1.0-25.el10.noarch
libxcrypt-devel-4.4.36-10.el10.x86_64      libzstd-devel-1.5.5-
9.el10.x86_64
  lua-srpm-macros-1-15.el10.noarch      m4-1.4.19-
11.el10.x86_64      make-1:4.4.1-9.el10.x86_64
ocaml-srpm-macros-10-4.el10.noarch      openblas-srpm-macros-2-
19.el10.noarch
  openssl-devel-1:3.2.2-16.el10.x86_64      package-notes-srpm-
macros-0.5-13.el10.noarch      patch-2.7.6-26.el10.x86_64
perl-AutoSplit-5.74-512.el10.noarch      perl-Benchmark-1.25-
512.el10.noarch
  perl-CPAN-Meta-2.150010-511.el10.noarch      perl-CPAN-Meta-
Requirements-2.143-11.el10.noarch      perl-CPAN-Meta-YAML-0.018-
512.el10.noarch      perl-Devel-PPPort-3.72-512.el10.x86_64      perl-
Encode-Locale-1.05-31.el10.noarch
  perl-ExtUtils-Command-2:7.70-513.el10.noarch      perl-ExtUtils-Constant-
0.25-512.el10.noarch      perl-ExtUtils-Install-2.22-511.el10.noarch
perl-ExtUtils-MakeMaker-2:7.70-513.el10.noarch      perl-ExtUtils-Manifest-
1:1.75-511.el10.noarch
  perl-ExtUtils-ParseXS-1:3.51-512.el10.noarch      perl-File-Compare-
1.100.800-512.el10.noarch      perl-File-Copy-2.41-512.el10.noarch
perl-I18N-Langinfo-0.24-512.el10.x86_64      perl-JSON-PP-1:4.16-
512.el10.noarch
  perl-Test-Harness-1:3.48-512.el10.noarch      perl-Time-HiRes-
4:1.9777-511.el10.x86_64      perl-devel-4:5.40.1-512.el10.x86_64
perl-doc-5.40.1-512.el10.noarch      perl-lib-0.65-
512.el10.x86_64
  perl-srpm-macros-1-57.el10.noarch      perl-version-8:0.99.32-
4.el10.x86_64      pyproject-srpm-macros-1.16.2-1.el10.noarch
python-srpm-macros-3.12-9.1.el10.noarch      python3-devel-3.12.9-
1.el10.x86_64
  python3-pip-23.3.2-7.el10.noarch      python3-pyparsing-
3.1.1-7.el10.noarch      qt6-srpm-macros-6.8.1-3.el10.noarch
redhat-rpm-config-288-1.el10.noarch      rust-toolset-srpm-
macros-1.84.1-1.el10.noarch
  systemtap-sdt-devel-5.2-2.el10.x86_64      systemtap-sdt-dtrace-
5.2-2.el10.x86_64      zlib-ng-compat-devel-2.2.3-1.el10.x86_64

```

Complete!

OS package installations finished

+ Installing ONTAP Mediator. (Log: /root/ontap_mediator.vdizgQ/ontap-mediator-1.11.0/ontap-mediator-1.11.0/install_20250715160240.log)

This step takes several minutes. View progress in the log file.

Sudoer config verified

```
ONTAP Mediator rsyslog and logging rotation enabled
+ Install successful. (Moving log to
/opt/netapp/lib/ontap_mediator/log/install_20250715160240.log)
+ WARNING: This system supports UEFI
      Secure Boot (SB) is currently disabled on this system.
      If SB is enabled in the future, SCST will not work unless
the following action is taken:
      Using the keys in
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys follow
      instructions in
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/README.modu
le-signing
      to sign the SCST kernel module. Note that a reboot is
needed.
      SCST does not start automatically when Secure Boot is enabled and
not configured properly.

+ Note: ONTAP Mediator generated a self-signed server certificate for
temporary use on
      this host. If the DNS name or IP address for the host is changed,
the certificate
      will no longer be valid. The default certificates should be
replaced with secure
      trusted certificates signed by a known certificate authority prior
to use for production.
      For more information, see /opt/netapp/lib/ontap_mediator/README

+ Note: ONTAP Mediator uses a kernel module compiled specifically for
the current
      OS. Using 'yum update' to upgrade the kernel might cause
service interruption.
      For more information, see /opt/netapp/lib/ontap_mediator/README
root@mediator_host:~# systemctl status ontap_mediator
● ontap_mediator.service - ONTAP Mediator
   Loaded: loaded (/etc/systemd/system/ontap_mediator.service;
enabled; preset: disabled)
   Active: active (running) since Tue 2025-07-15 16:07:29 EDT; 4min
9s ago
     Invocation: 395e9479487e4e308be2ae030c800c7f
       Process: 28745
ExecStartPre=/opt/netapp/lib/ontap_mediator/tools/otm_logs_fs.sh
(code=exited, status=0/SUCCESS)
   Main PID: 28759 (python)
      Tasks: 1 (limit: 22990)
     Memory: 66.8M (peak: 68.8M)
        CPU: 2.865s
```

```

CGroup: /system.slice/ontap_mediator.service
└─28759 /opt/netapp/lib/ontap_mediator/pyenv/bin/python
/opt/netapp/lib/ontap_mediator/ontap_mediator/server

Jul 15 16:07:29 mediator_host systemd[1]: Starting
ontap_mediator.service - ONTAP Mediator...
Jul 15 16:07:29 mediator_host systemd[1]: Started
ontap_mediator.service - ONTAP Mediator.
root@mediator_host:~# systemctl status mediator-scst
● mediator-scst.service
   Loaded: loaded (/etc/systemd/system/mediator-scst.service;
enabled; preset: disabled)
   Active: active (running) since Tue 2025-07-15 16:07:29 EDT; 4min
15s ago
     Invocation: f1d3be6ca1f9492b943e61872676f384
      Process: 28653 ExecStart=/etc/init.d/scst start (code=exited,
status=0/SUCCESS)
      Process: 28738 ExecStartPost=/usr/sbin/modprobe scst_vdisk
(code=exited, status=0/SUCCESS)
    Main PID: 28696 (iscsi-scstd)
       Tasks: 1 (limit: 22990)
      Memory: 5.2M (peak: 35.2M)
         CPU: 547ms
    CGroup: /system.slice/mediator-scst.service
           └─28696 /usr/local/sbin/iscsi-scstd

Jul 15 16:07:28 mediator_host systemd[1]: Starting mediator-
scst.service...
Jul 15 16:07:29 mediator_host iscsi-scstd[28694]: max_data_seg_len
1048576, max_queued_cmds 2048
Jul 15 16:07:29 mediator_host scst[28653]: Loading and configuring SCST
Jul 15 16:07:29 mediator_host systemd[1]: Started mediator-
scst.service.
root@mediator_host:~#

```

UEFIセキュアブートのセキュリティキーを登録する

ONTAP Mediator 1.4以降、UEFIシステムでSecure Bootメカニズムが有効になっています。Secure Bootを有効にすると、インストール後にセキュリティキーを登録するための追加手順が必要になります。

手順

1. README ファイルの指示に従って、SCST カーネルモジュールに署名します：

```

/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/README.module-
signing

```

2. 必要なキーを特定します。

```
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys
```



インストール後、システム出力に README ファイルとキーの場所が表示されます。

3. 公開鍵をMOKリストに追加します。

```
mokutil --import
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/scst_module_key.de
r
```



秘密鍵はデフォルトの場所に置いておくことも、安全な場所に移動することもできます。公開鍵はBoot Managerが使用できるように、既存の場所に保管する必要があります。詳細については、README.module-signing ファイルをご覧ください：

```
[root@hostname ~]# ls
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/
README.module-signing scst_module_key.der scst_module_key.priv
```

4. ホストを再起動し、デバイスのUEFI Boot Managerを使用して新しいMOKを承認します。["UEFIセキュアブートが有効になっている場合はONTAP Mediatorをインストールしません"](#)の `mokutil` ユーティリティで提供されたパスフレーズが必要になります。

SCSTカーネルモジュールの署名

ONTAP Mediator をインストールした後、`systemctl status `mediator-scst`` が failed (inactive) と表示される場合は、次の手順に従って SCST カーネル モジュールに署名します。

手順

1. ビルド プロセス中に、次のコマンドを使用して `/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/`` ディレクトリに公開キーと秘密キーのペアが生成されます：

```
[root@mediator-host ~]# ls
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/ README.module-
signing scst_module_key.der scst_module_key.priv [root@mediator-host ~]#
```

2. 次のコマンドを実行して、公開キーを UEFI キー リポジトリにインポートするプロセスを開始します：

```
[root@mediator-host ~]# mokutil --import
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/scst_module_key.de
r
input password: input password again:
```

```
[root@mediator-host ~]#
```

3. `mokutil` ソフトウェアは、インポート プロセス中にこのキーに使用する一時パスワードを要求します。
4. インポートプロセスが ``mokutil --list-new`` で開始されたかどうかを確認してから、システムを再起動します。ブートローダーがEFI MOKマネージャーを起動します。
5. 画面上のメニューを使用して、SCSTカーネルモジュールキーをオンにします。起動後、``systemctl status``

mediator-scst'を実行します。サービスが起動すると、SCSTカーネルモジュールが署名されます。

ONTAP Mediatorのインストールステータスを確認する

ONTAP Mediator をインストールした後、正常に実行されていることを確認します。

手順

1. ONTAP Mediator のステータスを表示します：

a. `systemctl status ontap_mediator`

```
[root@scspr1915530002 ~]# systemctl status ontap_mediator

ontap_mediator.service - ONTAP Mediator
Loaded: loaded (/etc/systemd/system/ontap_mediator.service; enabled;
vendor preset: disabled)
Active: active (running) since Mon 2022-04-18 10:41:49 EDT; 1 weeks 0
days ago
Process: 286710 ExecStop=/bin/kill -s INT $MAINPID (code=exited,
status=0/SUCCESS)
Main PID: 286712 (uwsgi)
Status: "uWSGI is ready"
Tasks: 3 (limit: 49473)
Memory: 139.2M
CGroup: /system.slice/ontap_mediator.service
├─286712 /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi --ini
/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini
├─286716 /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi --ini
/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini
└─286717 /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi --ini
/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini

[root@scspr1915530002 ~]#
```

b. `systemctl status mediator-scst`

```
[root@scspr1915530002 ~]# systemctl status mediator-scst
Loaded: loaded (/etc/systemd/system/mediator-scst.service;
enabled; vendor preset: disabled)
Active: active (running) since Mon 2022-04-18 10:41:47 EDT; 1
weeks 0 days ago
Process: 286595 ExecStart=/etc/init.d/scst start (code=exited,
status=0/SUCCESS)
Main PID: 286662 (iscsi-scstd)
Tasks: 1 (limit: 49473)
Memory: 1.2M
CGroup: /system.slice/mediator-scst.service
└─286662 /usr/local/sbin/iscsi-scstd

[root@scspr1915530002 ~]#
```

2. ONTAP Mediator で使用されるポートを確認します。

```
netstat
```

```
[root@scspr1905507001 ~]# netstat -anlt | grep -E '3260|31784'

tcp        0      0 0.0.0.0:31784      0.0.0.0:*        LISTEN
tcp        0      0 0.0.0.0:3260      0.0.0.0:*        LISTEN
tcp6       0      0 :::3260           :::*              LISTEN
```

インストール後のONTAP Mediator の設定

ONTAP Mediator をインストールして実行した後、ONTAP Mediator の機能を使用するには、ONTAP ストレージ システムで追加の設定タスクを実行する必要があります。

- ONTAP Mediator をMetroCluster IP 構成で使用するには、"[MetroCluster IP設定からONTAP Mediatorを設定する](#)"を参照してください。
- SnapMirrorアクティブ同期を使用するには、"[ONTAP Mediatorをインストールし、ONTAPクラスタ構成を確認します](#)"を参照してください。

ONTAP Mediatorのセキュリティ ポリシーの設定

ONTAP Mediatorは、設定可能な複数のセキュリティ設定をサポートしています。すべての設定のデフォルト値は、`low_space_threshold_mib: 10`読み取り専用ファイルで提供されています：

```
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.user_conf.yaml
```

`ontap_mediator.user_config.yaml`に配置されたすべての値は、デフォルト値を上書きし、すべての ONTAP Mediator アップグレードにわたって維持されます。

変更後 `ontap_mediator.user_config.yaml`、ONTAP Mediator を再起動します。

```
systemctl restart ontap_mediator
```

ONTAP Mediatorの属性の変更

このセクションで説明する ONTAP Mediator 属性は、必要に応じて変更できます。



`ontap_mediator.config.yaml`内のその他のデフォルト値は、変更された値がONTAP Mediatorのアップグレード中に維持されないため、変更しないでください。

必要な変数を `ontap_mediator.user_config.yaml` ファイルにコピーしてデフォルト設定を上書きすることで、ONTAP Mediator 属性を変更します。

サードパーティの**SSL**証明書をインストールする

デフォルトの自己署名証明書をサードパーティの SSL 証明書に置き換える必要がある場合は、次のファイル内の特定の属性を変更します：

- `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.config.yaml`
- `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini`

これらのファイル内の変数は、ONTAP Mediator で使用される証明書ファイルを制御するために使用されません。

ONTAP Mediator 1.9以降

次の表にリストされているデフォルトの変数が

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.config.yaml`ファイルに含まれています。

変数	パス
cert_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.crt
key_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key
ca_cert_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.crt
ca_key_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.key
ca_serial_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.srl
cert_valid_days	1095
x509_passin_pwd	pass:ontap

- `cert_valid_days`は、クライアント証明書の有効期限を設定するために使用されます。最大値は3年（1095日）です。
- x509_passin_pwd は署名されたクライアント証明書のパスフレーズです。

次の表にリストされているデフォルトの変数が `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` ファイルに含まれています。

変数	パス
mediator_cert	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.crt
mediator_key	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key
ca_cert_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.crt

ONTAP Mediator 1.8 以前

次の表にリストされているデフォルトの変数が

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.config.yaml`ファイルに含まれています。

変数	パス
cert_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.crt
key_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key
ca_cert_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.crt
ca_key_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.key
ca_serial_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.srl
cert_valid_days	1095
x509_passin_pwd	pass:ontap

- `cert_valid_days`は、クライアント証明書の有効期限を設定するために使用されます。最大値は3年（1095日）です。
- x509_passin_pwd は署名されたクライアント証明書のパスフレーズです。

次の表にリストされているデフォルトの変数が `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` ファイルに含まれています。

変数	パス
mediator_cert	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.crt
mediator_key	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key
ca_cert_path	/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.crt

これらの属性を変更した場合は、変更を適用するためにONTAP Mediatorを再起動してください。デフォルトの証明書をサードパーティの証明書に置き換える方法の詳細については、["自己署名証明書を信頼できるサードパーティの証明書に置き換える"](#)を参照してください。

パスワード攻撃保護

次の設定は、ブルートフォースパスワード推測攻撃に対する保護を提供します。

この機能を有効にするには、`window_seconds`と`retry_limit`の値を設定します。

例：

- 類推に使用できる時間を5分に設定し、5分後に失敗回数をゼロにリセットします。

```
authentication_lock_window_seconds: 300
```

- 時間内に5回失敗した場合にアカウントをロックします。

```
authentication_retry_limit: 5
```

- 各試行の拒否を遅延させることで、攻撃をスローダウンさせ、総当たりによるパスワード類推攻撃の影響を軽減します。

```
authentication_failure_delay_seconds: 5
```

```
authentication_failure_delay_seconds: 0    # seconds (float) to delay
failed auth attempts prior to response, 0 = no delay
authentication_lock_window_seconds: null   # seconds (int) since the
oldest failure before resetting the retry counter, null = no window
authentication_retry_limit: null          # number of retries to allow
before locking API access, null = unlimited
```

パスワードの複雑さのルール

次のフィールドは ONTAP Mediator API ユーザーアカウントのパスワードの複雑さのルールを制御します。

```
password_min_length: 8

password_max_length: 64

password_uppercase_chars: 0    # min. uppercase characters
password_lowercase_chars: 1    # min. lowercase character
password_special_chars: 1      # min. non-letter, non-digit
password_nonletter_chars: 2    # min. non-letter characters (digits,
specials, anything)
```

空き容量の制御

```
`/opt/netapp/lib/ontap_mediator` ディスクに必要な空き容量を制御する設定があります。
```

スペースが設定されたしきい値を下回ると、警告イベントが発行されます。

```
low_space_threshold_mib: 10
```

予備ログスペースの制御

RESERVE_LOG_SPACE は特定の設定によって制御されます。デフォルトでは、ONTAP Mediator のインストール時にログ用に別のディスクスペースが作成されます。インストーラは ONTAP Mediator のログ記録専用として、合計 700MB のディスクスペースを持つ新しい固定サイズのファイルを作成します。

この機能を無効にしてデフォルトのディスク スペースを使用するには、以下の手順に従います。

1. 次のファイルで、RESERVE_LOG_SPACE の値を 1 から 0 に変更します：

```
/opt/netapp/lib/ontap_mediator/tools/mediator_env
```

2. Mediatorを再起動します。

- a. `cat /opt/netapp/lib/ontap_mediator/tools/mediator_env | grep "RESERVE_LOG_SPACE"`

```
RESERVE_LOG_SPACE=0
```

- b. `systemctl restart ontap_mediator`

機能を再度有効にするには、値を 0 から 1 に変更し、Mediator を再起動します。



ディスク スペースを切り替えても、既存のログはパージされません。以前のログはすべてバックアップされ、切り替えとMediatorの再起動を行ったあとに現在のディスク スペースに移動されます。

ONTAP Mediatorの管理

ONTAPメディアーターの管理（ユーザークレデンシャルの変更、サービスの停止と再有効化、ヘルスチェック、ホストメンテナンスのためのSCSTのインストールまたはアンインストールなど）を行います。また、自己署名証明書の再生成、信頼できるサードパーティ証明書への置き換え、証明書関連の問題のトラブルシューティングなど、証明書の管理も可能です。

ユーザー名を変更する

以下の手順でユーザー名を変更できます。

タスク概要

この作業は、ONTAP Mediator をインストールした Linux ホストで実行してください。

このコマンドを実行できない場合は、次の例のように完全パスを使用してコマンドを実行する必要があります。

```
/usr/local/bin/mediator_username
```

手順

次のいずれかを実行してユーザ名を変更します。

- オプション **(a)** : `mediator_change_user` コマンドを実行し、次の例に示すようにプロンプトに回答します :

```
[root@mediator-host ~]# mediator_change_user
Modify the Mediator API username by entering the following values:
  Mediator API User Name: mediatoradmin
  Password:
New Mediator API User Name: mediator
The account username has been modified successfully.
[root@mediator-host ~]#
```

- オプション **(b)**: 次のコマンドを実行します :

```
MEDIATOR_USERNAME=mediator MEDIATOR_PASSWORD=mediator2
MEDIATOR_NEW_USERNAME=mediatoradmin mediator_change_user
```

```
[root@mediator-host ~]# MEDIATOR_USERNAME=mediator
MEDIATOR_PASSWORD='mediator2' MEDIATOR_NEW_USERNAME=mediatoradmin
mediator_change_user
The account username has been modified successfully.
[root@mediator-host ~]#
```

パスワードの変更

以下の手順でパスワードを変更できます。

タスク概要

この作業は、ONTAP Mediator をインストールした Linux ホストで実行してください。

このコマンドを実行できない場合は、次の例のように完全パスを使用してコマンドを実行する必要があります。

```
/usr/local/bin/mediator_change_password
```

手順

次のいずれかを実行してパスワードを変更します。

- オプション **(a)** : `mediator_change_password` コマンドを実行し、次の例に示すようにプロンプトに回答します :

```
[root@mediator-host ~]# mediator_change_password
Change the Mediator API password by entering the following values:
  Mediator API User Name: mediatoradmin
    Old Password:
    New Password:
    Confirm Password:
The password has been updated successfully.
[root@mediator-host ~]#
```

- オプション (b): 次のコマンドを実行します：

```
MEDIATOR_USERNAME=mediatoradmin MEDIATOR_PASSWORD=mediator1
MEDIATOR_NEW_PASSWORD=mediator2 mediator_change_password
```

この例では、パスワードを「mediator1」から「mediator2」に変更しています。

```
[root@mediator-host ~]# MEDIATOR_USERNAME=mediatoradmin
MEDIATOR_PASSWORD=mediator1 MEDIATOR_NEW_PASSWORD=mediator2
mediator_change_password
The password has been updated successfully.
[root@mediator-host ~]#
```

ONTAP Mediatorを停止する

ONTAP Mediator を停止するには、次の手順を実行します：

手順

1. ONTAP Mediatorを停止します：

```
systemctl stop ontap_mediator
```

2. SCSTを停止します。

```
systemctl stop mediator-scst
```

3. ONTAP メディエーターと SCST を無効にします：

```
systemctl disable ontap_mediator mediator-scst
```

ONTAP メディエーターを再度有効にする

ONTAP Mediator を再度有効にするには、次の手順を実行します：

手順

1. ONTAP MediatorとSCSTを有効にします：

```
systemctl enable ontap_mediator mediator-scst
```

2. SCST を開始：

```
systemctl start mediator-scst
```

3. ONTAP Mediatorを起動します：

```
systemctl start ontap_mediator
```

ONTAPメディアーターが正常であることを確認する

ONTAP Mediator をインストールした後、正常に実行されていることを確認します。

手順

1. ONTAP Mediator のステータスを表示します：

a. systemctl status ontap_mediator

```
[root@scspr1915530002 ~]# systemctl status ontap_mediator

ontap_mediator.service - ONTAP Mediator
Loaded: loaded (/etc/systemd/system/ontap_mediator.service; enabled;
vendor preset: disabled)
Active: active (running) since Mon 2022-04-18 10:41:49 EDT; 1 weeks 0
days ago
Process: 286710 ExecStop=/bin/kill -s INT $MAINPID (code=exited,
status=0/SUCCESS)
Main PID: 286712 (uwsgi)
Status: "uWSGI is ready"
Tasks: 3 (limit: 49473)
Memory: 139.2M
CGroup: /system.slice/ontap_mediator.service
├─286712 /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi --ini
/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini
├─286716 /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi --ini
/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini
└─286717 /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi --ini
/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini

[root@scspr1915530002 ~]#
```

b. systemctl status mediator-scst

```
[root@scspr1915530002 ~]# systemctl status mediator-scst
Loaded: loaded (/etc/systemd/system/mediator-scst.service;
enabled; vendor preset: disabled)
Active: active (running) since Mon 2022-04-18 10:41:47 EDT; 1
weeks 0 days ago
Process: 286595 ExecStart=/etc/init.d/scst start (code=exited,
status=0/SUCCESS)
Main PID: 286662 (iscsi-scstd)
Tasks: 1 (limit: 49473)
Memory: 1.2M
CGroup: /system.slice/mediator-scst.service
└─286662 /usr/local/sbin/iscsi-scstd

[root@scspr1915530002 ~]#
```

2. ONTAP Mediator で使用されるポートを確認します。

```
netstat
```

```
[root@scspr1905507001 ~]# netstat -anlt | grep -E '3260|31784'

tcp        0      0 0.0.0.0:31784      0.0.0.0:*        LISTEN
tcp        0      0 0.0.0.0:3260      0.0.0.0:*        LISTEN
tcp6       0      0 :::3260           :::*              LISTEN
```

ONTAP Mediatorのアンインストール

必要に応じて、ONTAP Mediatorを削除できます。

開始する前に

ONTAP から ONTAP Mediator を削除する前に、切断する必要があります。

タスク概要

この作業は、ONTAP Mediator をインストールした Linux ホストで実行してください。

このコマンドを実行できない場合は、次の例のように完全パスを使用してコマンドを実行する必要があります。

```
/usr/local/bin/uninstall_ontap_mediator
```

手順

1. ONTAP Mediator をアンインストールします：

```
uninstall_ontap_mediator
```

```
[root@mediator-host ~]# uninstall_ontap_mediator

ONTAP Mediator: Self Extracting Uninstaller

+ Removing ONTAP Mediator. (Log:
/tmp/ontap_mediator.GmRGdA/uninstall_ontap_mediator/remove.log)
+ Remove successful.
[root@mediator-host ~]#
```

一時的な自己署名証明書の再生成

ONTAP Mediator 1.7 以降では、次の手順で一時的な自己署名証明書を再生成できます。



この手順は、ONTAP Mediator 1.7以降を実行しているシステムでのみサポートされます。

タスク概要

- この作業は、ONTAP Mediator をインストールした Linux ホストで実行してください。
- このタスクは、ONTAP Mediator のインストール後にホストのホスト名または IP アドレスが変更されたために、生成された自己署名証明書が古くなった場合にのみ実行できます。
- 一時的な自己署名証明書を信頼できるサードパーティの証明書に置き換えた後は、このタスクを使用して証明書を再生成し_ない_でください。自己署名証明書がない場合、この手順は失敗します。

手順

現在のホストの一時的な自己署名証明書を新たに再生成するには、以下の手順を実行します。

1. ONTAP Mediatorを再起動します：

```
./make_self_signed_certs.sh overwrite
```

```

[root@xyz000123456 ~]# cd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config
[root@xyz000123456 server_config]# ./make_self_signed_certs.sh overwrite

Adding Subject Alternative Names to the self-signed server certificate
#
# OpenSSL example configuration file.
Generating self-signed certificates
Generating RSA private key, 4096 bit long modulus (2 primes)
.....
.....
.....++++
.....++++
e is 65537 (0x010001)
Generating a RSA private key
.....++++
.....
.....+++
+
writing new private key to 'ontap_mediator_server.key'
-----
Signature ok
subject=C = US, ST = California, L = San Jose, O = "NetApp, Inc.", OU =
ONTAP Core Software, CN = ONTAP Mediator, emailAddress =
support@netapp.com
Getting CA Private Key

```

信頼できるサードパーティの証明書への自己署名証明書の置き換え

サポートされている場合は、自己署名証明書を信頼できるサードパーティの証明書に置き換えることができます。

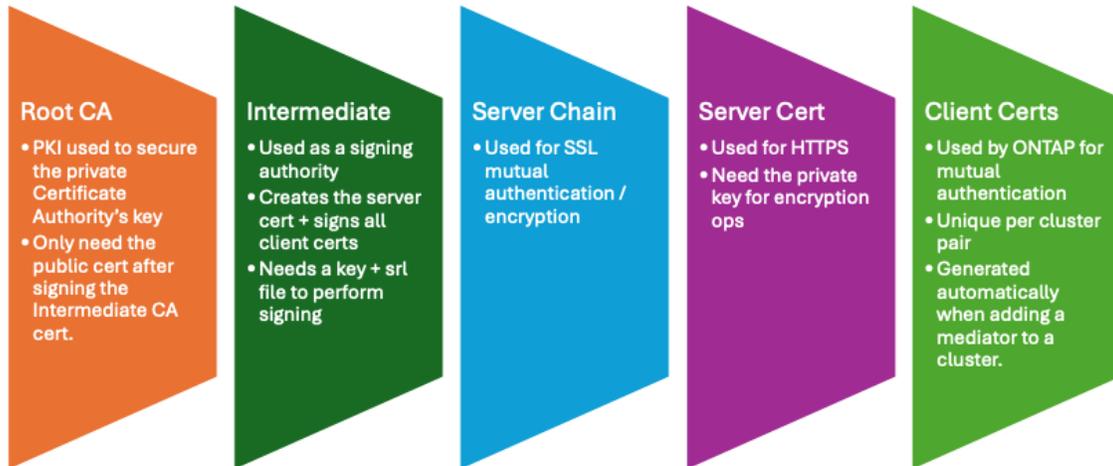


- サードパーティ証明書はONTAP 9.16.1以降およびそれ以前の一部のONTAPパッチリリースでのみサポートされます。"[NetApp Bugs Online バグID CONTAP-243278](#)"を参照してください。
- サードパーティの証明書は、ONTAP Mediator 1.7以降を実行しているシステムでのみサポートされます。

タスク概要

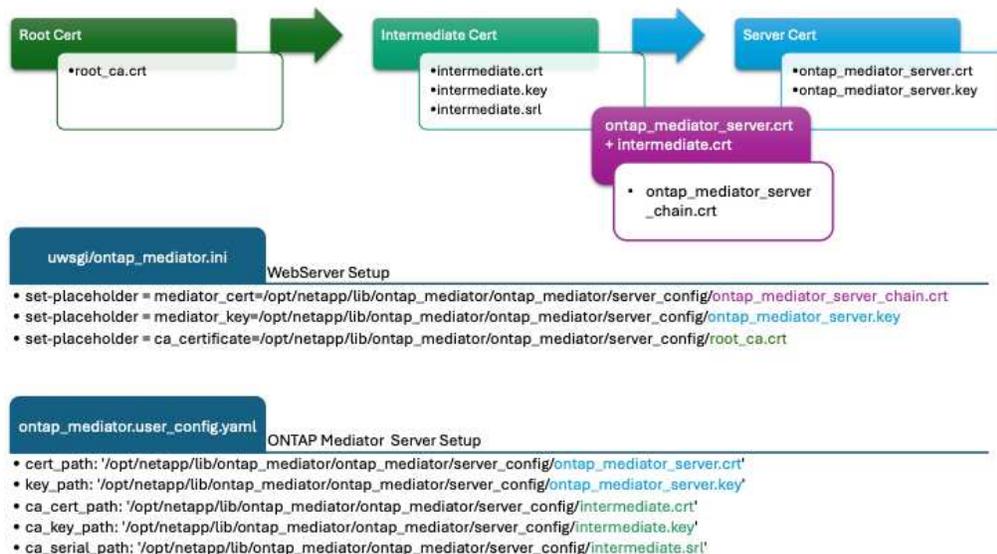
- この作業は、ONTAP Mediator をインストールした Linux ホストで実行してください。
- 生成された自己署名証明書を、信頼できる従属証明機関 (CA) から取得した証明書に置き換える必要がある場合に、このタスクを実行できます。これを行うには、信頼できる公開鍵基盤 (PKI) 機関にアクセスする必要があります。
- 次の図は、各ONTAP Mediator証明書の目的を示しています。

ONTAP Mediator Certificate Purposes



• 次の図は、Web サーバのセットアップと ONTAP Mediator のセットアップの構成を示しています。

ONTAP Mediator Certificates



ステップ1：CA証明書を発行するサードパーティから証明書を取得する

以下の手順に従って、PKI 機関から証明書を取得できます。

次の例は、自己署名証明書アクターを `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/` にあるサードパーティ証明書アクターに置き換える方法を示しています。



この例では、ONTAP Mediatorに必要な証明書の必要な基準を示しています。PKI認証局から証明書を取得する方法は、この手順とは異なる場合があります。ビジネスニーズに応じて手順を調整してください。

ONTAP Mediator 1.9以降

1. PKI機関が証明書を生成するために使用する秘密キー `intermediate.key` と構成ファイル `openssl_ca.cnf` を作成します。

- a. 秘密鍵を生成します `intermediate.key` :

例

```
openssl genrsa -aes256 -out intermediate.key 4096
```

- a. 構成ファイル `openssl_ca.cnf` (`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/openssl_ca.cnf` にあります) は、生成される証明書に必要なプロパティを定義します。

2. 秘密鍵と構成ファイルを使用して証明書署名要求を作成します `intermediate.csr` :

例 :

```
openssl req -key <private_key_name>.key -new -out  
<certificate_csr_name>.csr -config <config_file_name>.cnf
```

```
[root@scs000216655 server_config]# openssl req -key intermediate.key  
-new -config openssl_ca.cnf -out intermediate.csr  
Enter pass phrase for intermediate.key:  
[root@scs000216655 server_config]# cat intermediate.csr  
-----BEGIN CERTIFICATE REQUEST-----  
<certificate_value>  
-----END CERTIFICATE REQUEST-----
```

3. 証明書署名要求 `intermediate.csr` をPKI機関に送信して署名をもらいます。

PKI機関はリクエストを検証し、`.csr`に署名して証明書 `intermediate.crt` を生成します。さらに、PKI機関から `intermediate.crt` 証明書に署名した `root_ca.crt` 証明書を取得する必要があります。



SnapMirrorビジネス継続性 (SM-BC) クラスタの場合は、`intermediate.crt` および `root_ca.crt` 証明書をONTAPクラスタに追加する必要があります。["SnapMirror アクティブ同期のために ONTAP Mediator とクラスタを設定する"](#)を参照してください。

ONTAP Mediator 1.8 以前

1. PKI 機関が証明書を生成するために使用する秘密キー `ca.key` と構成ファイル `openssl_ca.cnf` を作成します。

- a. 秘密鍵を生成します `ca.key` :

例

```
openssl genrsa -aes256 -out ca.key 4096
```

a. 構成ファイル `openssl_ca.cnf` (`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/openssl_ca.cnf` にあります) は、生成される証明書に必要なプロパティを定義します。

2. 秘密鍵と構成ファイルを使用して証明書署名要求を作成します `ca.csr` :

例 :

```
openssl req -key <private_key_name>.key -new -out  
<certificate_csr_name>.csr -config <config_file_name>.cnf
```

```
[root@scs000216655 server_config]# openssl req -key ca.key -new  
-config openssl_ca.cnf -out ca.csr  
Enter pass phrase for ca.key:  
[root@scs000216655 server_config]# cat ca.csr  
-----BEGIN CERTIFICATE REQUEST-----  
<certificate_value>  
-----END CERTIFICATE REQUEST-----
```

3. 証明書署名要求 `ca.csr` をPKI機関に送信して署名をもらいます。

PKI機関はリクエストを検証し、`.csr` に署名して証明書を生成します `ca.crt`。さらに、PKI機関から `root_ca.crt` that signed the `ca.crt` 証明書を取得する必要があります。



SnapMirrorビジネス継続性 (SM-BC) クラスタの場合は、`ca.crt` および `root_ca.crt` 証明書をONTAPクラスタに追加する必要があります。["SnapMirror アクティブ同期のために ONTAP Mediator とクラスタを設定する"](#)を参照してください。

ステップ2：サードパーティのCA証明書で署名してサーバー証明書を生成する

ONTAP Mediator 1.9以降

サーバー証明書は、秘密鍵 `intermediate.key` とサードパーティの証明書 `intermediate.crt` によって署名されている必要があります。さらに、構成ファイル `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/openssl_server.cnf` には、OpenSSLによって発行されるサーバー証明書に必要なプロパティを指定する特定の属性が含まれています。

次のコマンドでサーバー証明書を生成できます。

手順

1. サーバー証明書署名要求 (CSR) を生成するには、
`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config` フォルダーから次のコマンドを実行します：

```
openssl req -config openssl_server.cnf -extensions v3_req -nodes -newkey  
rsa:4096 -sha512 -keyout ontap_mediator_server.key -out  
ontap_mediator_server.csr
```

2. CSR からサーバー証明書を生成するには、
`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config` フォルダーから次のコマンドを実行します：



これらのファイルはPKI認証局から取得されました。異なる証明書名を使用している場合は、`intermediate.crt` と `intermediate.key` を該当するファイル名に置き換えてください。

```
openssl x509 -extfile openssl_server.cnf -extensions v3_req -CA  
intermediate.crt -CAkey intermediate.key -CAcreateserial -sha512 -days 1095  
-req -in ontap_mediator_server.csr -out ontap_mediator_server.crt
```

- `-CAcreateserial` オプションは `intermediate.srl` ファイルを生成するために使用されます。

ONTAP Mediator 1.8 以前

サーバー証明書は、秘密鍵 `ca.key` とサードパーティの証明書 `ca.crt` によって署名されている必要があります。さらに、構成ファイル `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/openssl_server.cnf` には、OpenSSLによって発行されるサーバー証明書に必要なプロパティを指定する特定の属性が含まれています。

次のコマンドでサーバー証明書を生成できます。

手順

1. サーバー証明書署名要求 (CSR) を生成するには、
`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config` フォルダーから次のコマンドを実行します：

```
openssl req -config openssl_server.cnf -extensions v3_req -nodes -newkey  
rsa:4096 -sha512 -keyout ontap_mediator_server.key -out  
ontap_mediator_server.csr
```

2. CSR からサーバー証明書を生成するには、
`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config` フォルダーから次の

コマンドを実行します：



これらのファイルはPKI認証局から取得されました。異なる証明書名を使用している場合は、`ca.crt`と`ca.key`を該当するファイル名に置き換えてください。

```
openssl x509 -extfile openssl_server.cnf -extensions v3_req -CA ca.crt  
-CAkey ca.key -CAcreateserial -sha512 -days 1095 -req -in  
ontap_mediator_server.csr -out ontap_mediator_server.crt
```

° `-CAcreateserial` オプションは `ca.srl` ファイルを生成するために使用されます。

ステップ3：ONTAP Mediator構成で新しいサードパーティCA証明書とサーバ証明書を置き換える

ONTAP Mediator 1.10以降

証明書の設定は、

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.config.yaml`にある設定ファイルでONTAP Mediatorに提供されます。このファイルには、以下の属性が含まれます：

```
cert_path:
  '/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.crt'
key_path:
  '/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key'
ca_cert_path:
  '/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.crt'
ca_key_path:
  '/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.key'
ca_serial_path:
  '/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.srl'
```

- `cert_path`および`key_path`はサーバー証明書変数です。
- `ca_cert_path`、`ca_key_path`、および`ca_serial_path`はCA証明書変数です。

手順

1. すべての`intermediate.*`ファイルをサードパーティの証明書に置き換えます。
2. `intermediate.crt`および`ontap_mediator_server.crt`証明書から証明書チェーンを作成します：

```
cat ontap_mediator_server.crt intermediate.crt >
ontap_mediator_server_chain.crt
```

3. `/opt/netapp/lib/ontap_mediator/unicorn/config.json`ファイルを更新します。

`ssl_keyfile`、`ssl_certfile`、`ssl_ca_certs`の値を更新します：

```
ssl_keyfile:
  /opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key

ssl_certfile:
  /opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server_chain.crt

ssl_ca_certs:
```

```
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- `ssl_keyfile` 値は `ontap_mediator_server.crt` ファイル内のキーパスで、`ontap_mediator_server.key` です。
- `ssl_certfile` 値は `ontap_mediator_server_chain.crt` ファイルのパスです。
- `ssl_ca_certs` 値は `root_ca.crt` ファイルのパスです。

4. 新しく生成された証明書の次の属性が正しく設定されていることを確認します：

- Linux グループのオーナー: netapp:netapp
- Linux の権限: 600

5. ONTAP Mediatorを再起動します：

```
systemctl restart ontap_mediator
```

ONTAP Mediator 1.9.1 および 1.9

証明書の設定は、

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.config.yaml`にある設定ファイルでONTAP Mediatorに提供されます。このファイルには、以下の属性が含まれます：

```
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediate.srl'
```

- `cert_path` および `key_path` はサーバー証明書変数です。
- `ca_cert_path`、`ca_key_path`、および `ca_serial_path` は CA 証明書変数です。

手順

1. すべての `intermediate.*` ファイルをサードパーティの証明書に置き換えます。
2. `intermediate.crt` および `ontap_mediator_server.crt` 証明書から証明書チェーンを作成します：

```
cat ontap_mediator_server.crt intermediate.crt >
ontap_mediator_server_chain.crt
```

3. `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` ファイルを更新します。

```
`mediator_cert`、`mediator_key`、および  
`ca_certificate` の値を更新します：
```

```
set-placeholder = mediator_cert =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_  
server_chain.crt
```

```
set-placeholder = mediator_key =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_  
server.key
```

```
set-placeholder = ca_certificate =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- ``mediator_cert`` 値は ``ontap_mediator_server_chain.crt`` ファイルのパスです。
 - `mediator_key value`` は ``ontap_mediator_server.crt`` ファイル内のキーパスです
``ontap_mediator_server.key``。
 - ``ca_certificate`` 値は ``root_ca.crt`` ファイルのパスです。
4. 新しく生成された証明書の次の属性が正しく設定されていることを確認します：
- Linux グループのオーナー: `netapp:netapp`
 - Linux の権限: `600`

5. ONTAP Mediatorを再起動します：

```
systemctl restart ontap_mediator
```

ONTAP Mediator 1.8 以前

証明書の設定は、

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.config.yaml` にある設定ファイルでONTAP Mediatorに提供されます。このファイルには、以下の属性が含まれます：

```
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi
ator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_medi
ator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.srl'
```

- `cert_path` および `key_path` はサーバー証明書変数です。
- `ca_cert_path`、`ca_key_path`、および `ca_serial_path` は CA 証明書変数です。

手順

1. すべての `ca.*` ファイルをサードパーティの証明書に置き換えます。
2. `ca.crt` および `ontap_mediator_server.crt` 証明書から証明書チェーンを作成します：

```
cat ontap_mediator_server.crt ca.crt > ontap_mediator_server_chain.crt
```

3. `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` ファイルを更新します。

```
`mediator_cert`、`mediator_key`、および  
`ca_certificate` の値を更新します：
```

```
set-placeholder = mediator_cert =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_  
server_chain.crt
```

```
set-placeholder = mediator_key =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_  
server.key
```

```
set-placeholder = ca_certificate =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- `mediator_cert` 値は `ontap_mediator_server_chain.crt` ファイルのパスです。
 - `mediator_key value` は `ontap_mediator_server.crt` ファイル内のキーパスです
`ontap_mediator_server.key`。
 - `ca_certificate` 値は `root_ca.crt` ファイルのパスです。
4. 新しく生成された証明書の次の属性が正しく設定されていることを確認します：
 - Linux グループのオーナー: netapp:netapp

◦ Linux の権限： 600

5. ONTAP Mediatorを再起動します：

```
systemctl restart ontap_mediator
```

ステップ4：オプションで、サードパーティの証明書に別のパスまたは名前を使用します

ONTAP Mediator 1.10以降

``intermediate.*``以外の別の名前でサードパーティの証明書を使用したり、別の場所に保存したりすることもできます。

手順

1. ``/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.user_config.yaml`` ファイルを設定して、``ontap_mediator.config.yaml`` ファイル内のデフォルトの変数値を上書きします。

``intermediate.crt`` をPKI機関から取得し、その秘密鍵 ``intermediate.key`` を ``/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config`` の場所に保存する場合、
``ontap_mediator.user_config.yaml`` ファイルは次の例のようになります：



``intermediate.crt`` を使用して ``ontap_mediator_server.crt`` 証明書に署名した場合、
``intermediate.srl`` ファイルが生成されます。詳細については、[ステップ2：サードパーティのCA証明書で署名してサーバー証明書を生成する](#)を参照してください。

```
[root@scs000216655 server_config]# cat
ontap_mediator.user_config.yaml

# This config file can be used to override the default settings in
ontap_mediator.config.yaml
# To override a setting, copy the property key from
ontap_mediator.config.yaml to this file and
# set the property to the desired value. e.g.,
#
# The default value for 'default_mailboxes_per_target' is 4 in
ontap_mediator.config.yaml
#
# To override this value with 6 mailboxes per target, add the
following key/value pair
# below this comment:
#
# 'default_mailboxes_per_target': 6
#
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/interme
diate.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/interme
diate.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/interme
diate.srl'
```

- a. ``root_ca.crt`` 証明書が ``ontap_mediator_server.crt`` 証明書に署名する ``intermediate.crt`` 証明書を提
供する証明書構造を使用している場合は、``intermediate.crt`` 証明書と
``ontap_mediator_server.crt`` 証明書から証明書チェーンを作成します。



手順の早い段階で、PKI 機関から ``intermediate.crt`` 証明書と
``ontap_mediator_server.crt`` 証明書を取得しておく必要があります。

```
cat ontap_mediator_server.crt intermediate.crt >
ontap_mediator_server_chain.crt
```

- b. ``/opt/netapp/lib/ontap_mediator/unicorn/config.json`` ファイルを更新します。

``ssl_keyfile``、``ssl_certfile``、``ssl_ca_certs``の値を更新します：

```
ssl_keyfile:  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key
```

```
ssl_certfile:  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server_chain.crt
```

```
ssl_ca_certs:  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- ``ssl_keyfile``値は ``ontap_mediator_server.crt`` ファイル内のキーパスで、``ontap_mediator_server.key`` です。
- ``ssl_certfile``値は ``ontap_mediator_server_chain.crt`` ファイルのパスです。
- ``ssl_ca_certs``値は ``root_ca.crt`` ファイルのパスです。



SnapMirrorビジネス継続性 (SM-BC) クラスタの場合は、``intermediate.crt`` および ``root_ca.crt`` 証明書をONTAPクラスタに追加する必要があります。["SnapMirror アクティブ同期のために ONTAP Mediator とクラスタを設定する"](#)を参照してください。

c. 新しく生成された証明書の次の属性が正しく設定されていることを確認します：

- Linux グループのオーナー: `netapp:netapp`
- Linux の権限: `600`

2. 設定ファイルで証明書が更新されたら、ONTAP Mediator を再起動します。

```
systemctl restart ontap_mediator
```

ONTAP Mediator 1.9.1 および 1.9

``intermediate.*`` 以外の別の名前ですードパーティの証明書を使用したり、別の場所に保存したりすることもできます。

手順

1. ``/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.user_config.yaml`` ファイルを設定して、``ontap_mediator.config.yaml`` ファイル内のデフォルトの変数値を上書きします。

``intermediate.crt`` をPKI機関から取得し、その秘密鍵 ``intermediate.key`` を ``/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config`` の場所に保存する場合、``ontap_mediator.user_config.yaml`` ファイルは次の例のようになります：



`intermediate.crt`を使用して`ontap_mediator_server.crt`証明書に署名した場合、`intermediate.srl`ファイルが生成されます。詳細については、[ステップ2：サードパーティのCA証明書で署名してサーバー証明書を生成する](#)を参照してください。

```
[root@scs000216655 server_config]# cat
ontap_mediator.user_config.yaml

# This config file can be used to override the default settings in
ontap_mediator.config.yaml
# To override a setting, copy the property key from
ontap_mediator.config.yaml to this file and
# set the property to the desired value. e.g.,
#
# The default value for 'default_mailboxes_per_target' is 4 in
ontap_mediator.config.yaml
#
# To override this value with 6 mailboxes per target, add the
following key/value pair
# below this comment:
#
# 'default_mailboxes_per_target': 6
#
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/interme
diate.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/interme
diate.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/interme
diate.srl'
```

- a. `root_ca.crt`証明書が`ontap_mediator_server.crt`証明書に署名する`intermediate.crt`証明書を提供している証明書構造を使用している場合は、`intermediate.crt`証明書と`ontap_mediator_server.crt`証明書から証明書チェーンを作成します。



手順の早い段階で、PKI 機関から`intermediate.crt`証明書と`ontap_mediator_server.crt`証明書を取得しておく必要があります。

```
cat ontap_mediator_server.crt intermediate.crt >
ontap_mediator_server_chain.crt
```

- b. `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` ファイルを更新します。

``mediator_cert``、``mediator_key``、および
``ca_certificate`` の値を更新します：

```
set-placeholder = mediator_cert =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediato
r_server_chain.crt
```

```
set-placeholder = mediator_key =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediato
r_server.key
```

```
set-placeholder = ca_certificate =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- ``mediator_cert`` 値は ``ontap_mediator_server_chain.crt`` ファイルのパスです。
- ``mediator_key`` 値は ``ontap_mediator_server.crt`` ファイル内のキーパスで、``ontap_mediator_server.key`` です。
- ``ca_certificate`` 値は ``root_ca.crt`` ファイルのパスです。



SnapMirrorビジネス継続性 (SM-BC) クラスタの場合は、``intermediate.crt`` および ``root_ca.crt`` 証明書をONTAPクラスタに追加する必要があります。"[SnapMirror アクティブ同期のために ONTAP Mediator とクラスタを設定する](#)"を参照してください。

- c. 新しく生成された証明書の次の属性が正しく設定されていることを確認します：

- Linux グループのオーナー: `netapp:netapp`
- Linux の権限: `600`

2. 設定ファイルで証明書が更新されたら、ONTAP Mediator を再起動します。

```
systemctl restart ontap_mediator
```

ONTAP Mediator 1.8 以前

``ca.*`` 以外の別の名前ですードパーティの証明書を使用したり、別の場所に保存したりすることもできます。

手順

1. `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.user_config.yaml`` ファイルを設定して、``ontap_mediator.config.yaml`` ファイル内のデフォルトの変数値を上書きします。

`ca.crt`をPKI機関から取得し、その秘密鍵 `ca.key`を
`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config`の場所に
保存する場合、
`ontap_mediator.user_config.yaml`ファイルは次の例のようになります：



`ca.crt`を使用して `ontap_mediator_server.crt` 証明書に署名した場合、`ca.srl` ファイルが生成されます。詳細については、[ステップ2：サードパーティのCA証明書で署名してサーバー証明書を生成する](#)を参照してください。

```
[root@scs000216655 server_config]# cat
ontap_mediator.user_config.yaml

# This config file can be used to override the default settings in
ontap_mediator.config.yaml
# To override a setting, copy the property key from
ontap_mediator.config.yaml to this file and
# set the property to the desired value. e.g.,
#
# The default value for 'default_mailboxes_per_target' is 4 in
ontap_mediator.config.yaml
#
# To override this value with 6 mailboxes per target, add the
following key/value pair
# below this comment:
#
# 'default_mailboxes_per_target': 6
#
cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.crt'
key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_m
ediator_server.key'
ca_cert_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.srl'
```

- a. `root_ca.crt` 証明書が `ontap_mediator_server.crt` 証明書に署名する `ca.crt` 証明書を提供する証明書構造を使用している場合は、`ca.crt` 証明書と `ontap_mediator_server.crt` 証明書から証明書チェーンを作成します：



手順の早い段階で、PKI 機関から `ca.crt` 証明書と `ontap_mediator_server.crt` 証明書を取得しておく必要があります。

```
cat ontap_mediator_server.crt ca.crt > ontap_mediator_server_chain.crt
```

b. `/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini` ファイルを更新します。

```
`mediator_cert`、`mediator_key`、および  
`ca_certificate` の値を更新します：
```

```
set-placeholder = mediator_cert =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediato  
r_server_chain.crt
```

```
set-placeholder = mediator_key =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediato  
r_server.key
```

```
set-placeholder = ca_certificate =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/root_ca.crt
```

- `mediator_cert` 値は `ontap_mediator_server_chain.crt` ファイルのパスです。
- `mediator_key` 値は `ontap_mediator_server.crt` ファイル内のキーパスで、`ontap_mediator_server.key` です。
- `ca_certificate` 値は `root_ca.crt` ファイルのパスです。



SnapMirrorビジネス継続性 (SM-BC) クラスタの場合は、`ca.crt` および `root_ca.crt` 証明書を ONTAP クラスタに追加する必要があります。["SnapMirror アクティブ同期のために ONTAP Mediator とクラスタを設定する"](#)を参照してください。

c. 新しく生成された証明書の次の属性が正しく設定されていることを確認します：

- Linux グループのオーナー: netapp:netapp
- Linux の権限: 600

2. 設定ファイルで証明書が更新されたら、ONTAP Mediator を再起動します。

```
systemctl restart ontap_mediator
```

証明書関連の問題のトラブルシューティング

証明書の特定のプロパティを確認できます。

証明書の有効期限を確認する

証明書の有効範囲を識別するには、次のコマンドを使用します。

ONTAP Mediator 1.9以降

```
[root@mediator_host server_config]# openssl x509 -in intermediate.crt
-text -noout
Certificate:
  Data:
  ...
    Validity
      Not Before: Feb 22 19:57:25 2024 GMT
      Not After  : Feb 15 19:57:25 2029 GMT
```

ONTAP Mediator 1.8 以前

```
[root@mediator_host server_config]# openssl x509 -in ca.crt -text
-noout
Certificate:
  Data:
  ...
    Validity
      Not Before: Feb 22 19:57:25 2024 GMT
      Not After  : Feb 15 19:57:25 2029 GMT
```

CA 証明書における X509v3 拡張の検証

次のコマンドを使用して、CA 証明書の X509v3 拡張機能を検証します。

ONTAP Mediator 1.9以降

`openssl_ca.cnf`内の`*v3_ca*`で定義されたプロパティは、
`intermediate.crt`では`X509v3 extensions`として表示されます。

```
[root@mediator_host server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@mediator_host server_config]# cat openssl_ca.cnf
...
[ v3_ca ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, cRLSign, digitalSignature, keyCertSign

[root@mediator_host server_config]# openssl x509 -in intermediate.crt
-text -noout
Certificate:
    Data:
    ...
        X509v3 extensions:
            X509v3 Subject Key Identifier:

9F:06:FA:47:00:67:BA:B2:D4:82:70:38:B8:48:55:B5:24:DB:FC:27
            X509v3 Authority Key Identifier:

keyid:9F:06:FA:47:00:67:BA:B2:D4:82:70:38:B8:48:55:B5:24:DB:FC:27

            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
```

ONTAP Mediator 1.8 以前

`openssl_ca.cnf`内の`*v3_ca*`で定義されたプロパティは、`ca.crt`では
`X509v3 extensions`として表示されず。

```

[root@mediator_host server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@mediator_host server_config]# cat openssl_ca.cnf
...
[ v3_ca ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, cRLSign, digitalSignature, keyCertSign

[root@mediator_host server_config]# openssl x509 -in ca.crt -text
-noout
Certificate:
    Data:
    ...
        X509v3 extensions:
            X509v3 Subject Key Identifier:

9F:06:FA:47:00:67:BA:B2:D4:82:70:38:B8:48:55:B5:24:DB:FC:27
            X509v3 Authority Key Identifier:

keyid:9F:06:FA:47:00:67:BA:B2:D4:82:70:38:B8:48:55:B5:24:DB:FC:27

            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign

```

サーバー証明書とサブジェクトの代替名における **X509v3** 拡張機能を検証する

`openssl_server.cnf`構成ファイルで定義された `v3_req` プロパティは、証明書内で `X509v3 extensions` として表示されます。

次の例では、ONTAP Mediator がインストールされている Linux VM で `hostname -A` および `hostname -l` コマンドを実行することで、`alt_names` セクション内の変数を取得できます。

変数の正しい値については、ネットワーク管理者に確認してください。

ONTAP Mediator 1.9以降

```
[root@mediator_host server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@mediator_host server_config]# cat openssl_server.cnf
...
[ v3_req ]
basicConstraints          = CA:false
extendedKeyUsage         = serverAuth
keyUsage                  = keyEncipherment, dataEncipherment
subjectAltName            = @alt_names

[ alt_names ]
DNS.1 = abc.company.com
DNS.2 = abc-v6.company.com
IP.1 = 1.2.3.4
IP.2 = abcd:abcd:abcd:abcd:abcd:abcd

[root@mediator_host server_config]# openssl x509 -in intermediate.crt
-text -noout
Certificate:
    Data:
    ...

        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Key Usage:
                Key Encipherment, Data Encipherment
            X509v3 Subject Alternative Name:
                DNS:abc.company.com, DNS:abc-v6.company.com, IP
Address:1.2.3.4, IP Address:abcd:abcd:abcd:abcd:abcd:abcd
```

ONTAP Mediator 1.8 以前

```
[root@mediator_host server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@mediator_host server_config]# cat openssl_server.cnf
...
[ v3_req ]
basicConstraints          = CA:false
extendedKeyUsage         = serverAuth
keyUsage                 = keyEncipherment, dataEncipherment
subjectAltName           = @alt_names

[ alt_names ]
DNS.1 = abc.company.com
DNS.2 = abc-v6.company.com
IP.1 = 1.2.3.4
IP.2 = abcd:abcd:abcd:abcd:abcd:abcd

[root@mediator_host server_config]# openssl x509 -in ca.crt -text
-noout
Certificate:
    Data:
    ...

        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Key Usage:
                Key Encipherment, Data Encipherment
            X509v3 Subject Alternative Name:
                DNS:abc.company.com, DNS:abc-v6.company.com, IP
Address:1.2.3.4, IP Address:abcd:abcd:abcd:abcd:abcd:abcd
```

秘密鍵が証明書と一致していることを確認する

特定の秘密鍵が証明書と一致するかどうかを確認できます。

キーと証明書に対してそれぞれ次の OpenSSL コマンドを使用します。

ONTAP Mediator 1.9以降

```
[root@mediator_host server_config]# openssl rsa -noout -modulus -in
intermediate.key | openssl md5
Enter pass phrase for intermediate.key:
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
[root@mediator_host server_config]# openssl x509 -noout -modulus -in
intermediate.crt | openssl md5
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
```

ONTAP Mediator 1.8 以前

```
[root@mediator_host server_config]# openssl rsa -noout -modulus -in
ca.key | openssl md5
Enter pass phrase for ca.key:
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
[root@mediator_host server_config]# openssl x509 -noout -modulus -in
ca.crt | openssl md5
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
```

両方の`-modulus`属性が一致する場合、秘密鍵と証明書のペアに互換性があり、相互に動作できることを示します。

サーバー証明書が特定のCA証明書から作成されたことを確認する

次のコマンドを使用して、サーバー証明書が特定のCA証明書から作成されていることを確認できます。

ONTAP Mediator 1.9以降

```
[root@mediator_host server_config]# openssl verify -CAfile root_ca.crt
--untrusted intermediate.crt ontap_mediator_server.crt
ontap_mediator_server.crt: OK
[root@mediator_host server_config]#
```

ONTAP Mediator 1.8 以前

```
[root@mediator_host server_config]# openssl verify -CAfile ca.crt
ontap_mediator_server.crt
ontap_mediator_server.crt: OK
```

オンライン証明書ステータス プロトコル (OCSP) 検証を使用している場合は、コマンド"[openssl-verify](#)"を使用します。

ONTAP MediatorのホストOSを保守する

最適なパフォーマンスを実現するために、ONTAP Mediator のホスト OS が定期的にメンテナンスされていることを確認してください。

ホストのリブート

クラスタが正常な状態になった場合にのみホストを再起動してください。ONTAP Mediatorがオフラインの間は、クラスタは障害に対応できません。再起動する前に、メンテナンスの時間を設定してください。

ONTAP Mediator は再起動時に自動的に再開し、ONTAP クラスタとの以前に設定された関係を再入力します。

ホスト パッケージの更新

カーネルを除くすべてのライブラリまたはyumパッケージを更新します。変更を有効にするために必要な場合は、ホストを再起動します。ホストを再起動する前に、サービスウィンドウをスケジュールします。

```
`yum-utils`パッケージをインストールする場合は、`needs-restarting`コマンドを使用して、パッケージの変更に再起動が必要かどうかを検出します。
```

変更はすぐには有効にならないため、ONTAP Mediator の依存関係を更新した後は再起動してください。

ホストOSカーネルをアップグレードする

SCSTは、使用しているカーネルに合わせてコンパイルする必要があります。OSをアップデートするには、メンテナンスのための時間をスケジュールする必要があります。

手順

ホスト OS カーネルをアップグレードするには、次の手順に従います。



カーネルをアップグレードする前に、OSとONTAP Mediatorのバージョンに互換性があることを確認してください。サポートされているバージョンについては、"[OSのサポート対応表](#)"を参照してください。

1. ONTAPメディアーターを停止します。
2. SCST パッケージをアンインストールします。[\[ホストのメンテナンスを実行する\]](#)を参照してください。(SCST ではアップグレード メカニズムは提供されません。)
3. OSをアップグレードし、リブートします。
4. SCSTパッケージを再インストールします。
5. ONTAP Mediator を再度有効にします。

ホストのメンテナンスを実行する

VMカーネルをアップグレードすると、SCSTモジュールとの互換性の問題が発生する可能性があります。SCSTを手動でアンインストールし、再インストールしてください。

ステップ1：SCSTをアンインストールする

SCST をアンインストールするには、ONTAP Mediator バージョンの tar バンドルを使用します。

手順

1. 適切な SCST バンドル（次の表を参照）をダウンロードして解凍します。

このバージョンでは...	この tar バンドルを使用してください...
ONTAP Mediator 1.11	scst-3.9.tar.gz
ONTAP Mediator 1.10	scst-3.9.tar.gz
ONTAP Mediator 1.9.1	scst-3.8.0.tar.bz2
ONTAP Mediator 1.9	scst-3.8.0.tar.bz2
ONTAP Mediator 1.8	scst-3.8.0.tar.bz2
ONTAP Mediator 1.7	scst-3.7.0.tar.bz2
ONTAP Mediator 1.6	scst-3.7.0.tar.bz2
ONTAP Mediator 1.5	scst-3.6.0.tar.bz2
ONTAP Mediator 1.4	scst-3.6.0.tar.bz2
ONTAP Mediator 1.3	scst-3.5.0.tar.bz2
ONTAP Mediator 1.1	scst-3.4.0.tar.bz2
ONTAP Mediator 1.0	scst-3.3.0.tar.bz2

- a. ["SCST SourceForge ダウンロード"](#)からオープンソース パッケージにアクセスします。
 - b. *Download released versions*を選択します。
 - c. バンドルを VM に抽出します。
2. `scst` ディレクトリ内で次のアンインストール コマンドを実行します：
 - a. `systemctl stop mediator-scst`
 - b. `make scstadm_uninstall`
 - c. `make iscsi_uninstall`
 - d. `make usr_uninstall`
 - e. `make scst_uninstall`
 - f. `depmod`

ステップ2：SCSTをインストールする

SCSTを手動でインストールするには、インストールされているバージョンのONTAP Mediatorで使用されるSCST tarバンドルが必要です（[SCSTテーブルARPテーブル](#)を参照）。



ONTAP Mediatorをインストールする前にこの手順を実行してください。使用しているSCSTのバージョンがONTAP Mediatorインストーラにバンドルされているバージョンよりも新しい場合、インストーラはこの手順をスキップします。

1. `scst`ディレクトリで次のインストールコマンドを実行します：

- a. `make 2release`
- b. `make scst_install`
- c. `make usr_install`
- d. `make iscsi_install`
- e. `make scstadm_install`
- f. `depmod`



初めてインストールを実行し、ONTAP Mediatorを事前にインストールする場合は、次の手順に進む前に次のコマンドを実行します：

```
mkdir -p  
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys
```

- g. `cp scst/src/certs/scst_module_key.der /opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/`
- h. `patch /etc/init.d/scst < /opt/netapp/lib/ontap_mediator/systemd/scst.patch`



初回インストール時にONTAP Mediatorの前にSCSTを事前にインストールする場合は、この手順をスキップしてください。インストーラによって関連するSCSTパッチが適用されます。

2. オプションとして、セキュアブートが有効になっている場合は、再起動する前に次の手順を実行します：

- a. `scst_vdisk`、`scst`、および `iscsi_scst` モジュールの各ファイル名を決定します：

```
[root@localhost ~]# modinfo -n scst_vdisk  
[root@localhost ~]# modinfo -n scst  
[root@localhost ~]# modinfo -n iscsi_scst
```

- b. カーネルリリースを確認します：

```
[root@localhost ~]# uname -r
```

c. 各モジュール ファイルをカーネルで署名します：

```
[root@localhost ~]# /usr/src/kernels/<KERNEL-RELEASE>/scripts/sign-  
file \sha256 \  
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/scst_modu-  
le_key.priv \  
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/scst_modu-  
le_key.der \  
_module-filename_
```

d. ファームウェアとともに UEFI キーをインストールします。

UEFIキーのインストール手順については、以下を参照してください。

```
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/README.module-  
signing
```

生成されたUEFIキーは次の場所に保存されます。

```
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/scst_module_key.de-  
r
```

3. システムを再起動します：

```
reboot
```

ホストのホスト名またはIPの変更

タスク概要

- この作業は、ONTAP Mediator をインストールした Linux ホストで実行してください。
- ONTAP Mediator をインストールした後にホスト名または IP アドレスが変更されたために自己署名証明書が有効でなくなった場合にのみ、これを実行してください。
- 一時的な自己署名証明書を信頼できるサードパーティの証明書に置き換えた後は、このタスクを使用して証明書を再生成しないでください。自己署名証明書をお持ちでない場合は、この手順は使用できません。

手順

現在のホストの一時的な自己署名証明書を作成します：

1. ONTAP Mediatorを再起動します：

```
./make_self_signed_certs.sh overwrite
```

```
[root@xyz000123456 ~]# cd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config
[root@xyz000123456 server_config]# ./make_self_signed_certs.sh overwrite

Adding Subject Alternative Names to the self-signed server certificate
#
# OpenSSL example configuration file.
Generating self-signed certificates
Generating RSA private key, 4096 bit long modulus (2 primes)
.....
.....
.....++++
.....++++
e is 65537 (0x010001)
Generating a RSA private key
.....++++
.....++++
+
writing new private key to 'ontap_mediator_server.key'
-----
Signature ok
subject=C = US, ST = California, L = San Jose, O = "NetApp, Inc.", OU =
ONTAP Core Software, CN = ONTAP Mediator, emailAddress =
support@netapp.com
Getting CA Private Key

[root@xyz000123456 server_config]# systemctl restart ontap_mediator
```

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。