



# **MetroClusterとSnapMirrorのアクティブな同期用のメディエーターサービス** **ONTAP 9**

NetApp  
December 20, 2024

# 目次

MetroClusterとSnapMirrorのアクティブな同期用のメディアエーターサービス .....	1
ONTAPメディアエーターの概要 .....	1
ONTAP メディアエーターの最新情報 .....	2
インストールまたはアップグレード .....	6
ONTAPメディアエーターサービスを管理します。 .....	49
ONTAPメディアエーター用のOSホストのメンテナンス .....	66

# MetroClusterとSnapMirrorのアクティブな同期用のメディアエーターサービス

## ONTAPメディアエーターの概要

ONTAPメディアエーターは、ONTAPの機能に次のような機能を提供します。

- HAメタデータ用の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維持するためのpingプロキシとして機能します。
- クォーラムの決定に役立つ同期ノード健全性クエリ機能を提供します。

ONTAPメディアエーターは、さらに2つの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。

## システム管理者に提供されるツール

システム管理者に提供されるツール：

- **/usr/local/bin/mediator\_change\_password**

現在のAPIユーザ名とパスワードを指定したときに、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を指定した場合に、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通信に必要なすべてのサポート情報を含むローカルの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で生成され、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メディアエーターパッケージとSCSTカーネルモジュールを削除します。これには、すべての設定、ログ、およびメールボックスデータが含まれます。

- **/usr/local/bin/mediator\_unlock\_user**

認証の再試行の上限に達した場合、APIユーザアカウントのロックアウトが解除されます。この機能は、

ブルートフォースパスワードの派生を防止するために使用されます。ユーザに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ユーザを追加する場合に使用します。

## 特記事項

ONTAPメディアエーターは、SCSTを使用してiSCSIを提供します (を参照 <http://scst.sourceforge.net/index.html>)。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場合は、SCSTの再インストールが必要になることがあります。または、ONTAPメディアエーターをアンインストールしてから再インストールし、ONTAP関係を再設定します。



サーバOSカーネルの更新は、ONTAPのメンテナンス時間に合わせて行う必要があります。

## ONTAP メディアエーターの最新情報

各リリースでは、ONTAPメディアエーターの機能が新たに拡張されています。最新情報をご紹介します。

### キノウカクチョウ

SCSTのバージョン情報については、を参照して[SCSTのサポートマトリックス](#)ください。

ONTAP メディアエーターのバージョン	キノウカクチョウ
1.9	<ul style="list-style-type: none"><li>• RHELのサポート：<ul style="list-style-type: none"><li>◦ 互換性がある：8.4、8.5、8.6、8.7、8.9、9.1、9.3。</li><li>◦ 推奨：8.8、8.10、9.0、9.2、9.4、9.5。</li></ul></li><li>• Rocky Linux 8および9のサポート。</li><li>• RHELおよびRocky LinuxのFIPSサポート。</li><li>• パフォーマンスが強化され、スケーラビリティが向上しました。</li><li>• PKI署名付き証明書のセットアップを簡素化するためのファイル名の改善。</li></ul>
1.8	<ul style="list-style-type: none"><li>• RHEL 8.4、8.5、8.6、8.7、8.8、8.9、8.10、9.0、9.1、9.2、9.3、9.4がサポートされます。</li><li>• Rocky Linux 8および9のサポート。</li></ul>

1.7	<ul style="list-style-type: none"> <li>• RHEL 8.4、8.5、8.6、8.7、8.8、8.9、9.0、9.1、9.2、9.3がサポートされます。</li> <li>• Rocky Linux 8および9のサポート。</li> <li>• 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 (Subject Alternative Name) データのサポート。</li> </ul>
1.6	<ul style="list-style-type: none"> <li>• Python 3.9のアップデート。</li> <li>• RHEL 8.4、8.5、8.6、8.7、8.8、9.0、9.1、9.2がサポートされます。</li> <li>• Rocky Linux 8および9のサポート。</li> <li>• RHEL 7.x / CentOSのすべてのリリースのサポートを廃止しました。</li> </ul>
1.5	<ul style="list-style-type: none"> <li>• RHEL 7.6、7.7、7.8、7.9、8.0、8.1、8.2、8.3、8.4、8.5のサポート。</li> <li>• CentOS 7.6、7.7、7.8、7.9のサポート。</li> <li>• RHEL 7.x / CentOS 7.xの廃止に関する警告が追加されました。</li> <li>• 大規模なSnapMirrorアクティブ同期システムの速度を最適化します。</li> <li>• インストーラに暗号化コード署名が追加されました。</li> </ul>
1.4	<ul style="list-style-type: none"> <li>• RHEL 7.6、7.7、7.8、7.9、8.0、8.1、8.2、8.3、8.4、8.5のサポート。</li> <li>• CentOS 7.6、7.7、7.8、7.9のサポート。</li> <li>• 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。</li> </ul>
1.3	<ul style="list-style-type: none"> <li>• RHEL 7.6、7.7、7.8、7.9、8.0、8.1、8.2、8.3のサポート。</li> <li>• CentOS 7.6、7.7、7.8、7.9のサポート。</li> </ul>
1.2	<ul style="list-style-type: none"> <li>• RHEL 7.6、7.7、7.8、7.9、8.0、8.1のサポート。</li> <li>• CentOS 7.6、7.7、7.8、7.9のサポート。</li> <li>• HTTPSメールボックスのサポート。</li> <li>• ONTAP 9.8以降のMCC-IP AUSOおよびSnapMirrorアクティブ同期ZRTOで使 用します。</li> </ul>
1.1	<ul style="list-style-type: none"> <li>• RHEL 7.6および8.0のサポート。</li> <li>• CentOS 7.6のサポート</li> <li>• Perlの依存関係を削除。</li> </ul>
1.0	<ul style="list-style-type: none"> <li>• iSCSIメールボックスのサポート。</li> <li>• ONTAP 9.7以降MCC-IP AUSOで使用します。</li> <li>• RHEL / CentOS 7.6のサポート。</li> </ul>

## OSのサポート対応表

ONTAP メディア エータ ー用 のOS	1.9	1.8	1.7	1.6	1.5	1.4	1.3	1.2	1.1	1.0
7.6	廃止	廃止	廃止	廃止	○	○	○	○	○	○ (RHEL のみ)
7.7	廃止	廃止	廃止	廃止	○	○	○	○	いいえ	いいえ
7.8	廃止	廃止	廃止	廃止	○	○	○	○	いいえ	いいえ
7.9	廃止	廃止	廃止	廃止	○	○	○	互換性	いいえ	いいえ
RHEL 8.0	廃止	廃止	廃止	廃止	○	○	○	○	○	いいえ
RHEL 8.1	廃止	廃止	廃止	廃止	○	○	○	○	いいえ	いいえ
RHEL 8.2	廃止	廃止	廃止	廃止	○	○	○	いいえ	いいえ	いいえ
RHEL 8.3	廃止	廃止	廃止	廃止	○	○	○	いいえ	いいえ	いいえ
RHEL 8.4	互換性	○	○	○	○	○	いいえ	いいえ	いいえ	いいえ
RHEL 8.5	互換性	○	○	○	○	○	いいえ	いいえ	いいえ	いいえ
RHEL 8.6	互換性	○	○	○	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 8.7	互換性	○	○	○	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 8.8	○	○	○	○	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 8.9	互換性	○	○	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ

RHEL 8.10	○	○	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 9.0	○	○	○	○	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.1	互換性	○	○	○	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.2	○	○	○	○	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.3	互換性	○	○	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.4	○	○	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	○	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
CentOS 8およびSTREAM	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	N/A	N/A	N/A
Rocky Linux 8	○	○	○	○	N/A	N/A	N/A	N/A	N/A	N/A
Rocky Linux 9	○	○	○	○	N/A	N/A	N/A	N/A	N/A	N/A

- 特に指定がないかぎり、「OS」とはRedHatとCentOSの両方のリリースを指します。
- 「はい」は、ONTAPメディアーター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ていることを示します。
- 「いいえ」は、OSとONTAPメディアーターに互換性がないことを示します。
- 「compatible」は、RHELではこのバージョンがサポートされなくなりましたが、ONTAPメディアーターは引き続きインストールできます。
- CentOS 8は再分岐のため全てのリリースで削除された。CentOS Streamは本番用のターゲットOSとしては適切ではないと考えられていた。サポートは予定されていません。
- 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。
- ONTAP Mediator 1.6では、Rocky Linux 8および9のサポートが追加されています。

## SCSTのサポートマトリックス

次の表に、ONTAPメディアエーターのバージョンごとにサポートされるSCSTのバージョンを示します。

ONTAP メディアエーターのバージョン	サポートされる SCST のバージョン
ONTAPメディアエーター1.9	scst-3.8.0.tar.bz2
ONTAPメディアエーター1.8	scst-3.8.0.tar.bz2
ONTAPメディアエーター1.7	scst-3.7.0.tar.bz2
ONTAPメディアエーター1.6	scst-3.7.0.tar.bz2
ONTAPメディアエーター1.5	scst-3.6.0.tar.bz2
ONTAPメディアエーター1.4	scst-3.6.0.tar.bz2
ONTAPメディアエーター1.3	scst-3.5.0.tar.bz2
ONTAPメディアエーター1.2	scst-3.4.0.tar.bz2
ONTAPメディアエーター1.1	scst-3.4.0.tar.bz2
ONTAPメディアエーター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メディアエーターサービスのインストールまたはアップグレードの準備

ONTAPメディアエーターサービスをインストールするには、すべての前提条件を満たしていることを確認し、インストールパッケージを読み込んでホストでインストーラを実行する必要があります。この手順は、既存のインストールのインストールまたはアップグレードに使用します。

- ONTAP 9 .7以降では、任意のバージョンのONTAPメディアエーターを使用してMetroCluster IP設定を監視できます。
- ONTAP 9 .8以降では、任意のバージョンのONTAPメディアエーターを使用してSnapMirrorのアクティブな同期関係を監視できます。

### インストールとアップグレードに関する考慮事項

ONTAPメディアエーターをアップグレードまたはインストールする前に、次の考慮事項を確認してください。



ONTAPメディアエーター1.8以前はRed Hat Enterprise LinuxのFIPSモードと互換性がないため、インストールを実行できません。FIPSモードが有効になっているかどうかは、コマンドを使用して確認できません `fips-mode-setup --check`。FIPSモードを無効にするには、コマンドを使用し `fips-modesetup --disable` ます。FIPSモードを無効にしてからリブートし、ONTAPメディアエーター1.8以前をインストールしてください。

- ONTAPメディアエーター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。以前のバージョンのONTAP MediatorはすべてのONTAPバージョンと下位互換性がありますが、最近のバージョンにはすべてのサードパーティ要素に対するセキュリティパッチが含まれています。



- 新しいバージョンのONTAPメディエーターにアップグレードすると、新しいバージョンが利用可能でないかぎり、推奨されるバージョンに自動的にアップグレードされます。上位バージョンのSCSTを手動でインストールする手順については、を参照してください"[Mediatorサービスを管理します。](#)"。サポートされているバージョンについては、を参照してください"[SCSTのサポートマトリックス](#)"。



インストールに失敗した場合は、ONTAPメディエーターの新しいバージョンへのアップグレードが必要になることがあります。

- パッケージをインストールする場合は yum-utils、コマンドを使用できます needs-restarting。

## OSノヨウケン

オペレーティングシステムが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
- 1 GB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：ルートアクセス

次の表に、ONTAPメディエーターのバージョンごとにサポートされるオペレーティングシステムを示します。

ONTAP メディエーターのバージョン	サポートされるLinuxバージョン
1.9	<ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux               <ul style="list-style-type: none"> <li>◦ 互換性：8.4、8.5、8.6、8.7、8.9、9.1、および9.3<sup>1</sup></li> <li>◦ 推奨：8.8、8.10、9.0、9.2、9.4、9.5</li> </ul> </li> <li>• Rocky Linux 8および9</li> </ul>
1.8	<ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux：8.4、8.5、8.6、8.7、8.8、8.9、8.10、9.0、9.1、9.2、9.3、9.4</li> <li>• Rocky Linux 8および9</li> </ul>
1.7	<ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux：8.4、8.5、8.6、8.7、8.8、8.9、9.0、9.1、9.2、9.3</li> <li>• Rocky Linux 8および9</li> </ul>
1.6	<ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux：8.4、8.5、8.6、8.7、8.8、9.0、9.1、9.2</li> <li>• Rocky Linux 8および9</li> </ul>
1.5	<ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux：7.6、7.7、7.8、7.9、8.0、8.1、8.2、8.3、8.4、8.5</li> <li>• CentOS：7.6、7.7、7.8、7.9</li> </ul>

1.4	<ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux : 7.6、7.7、7.8、7.9、8.0、8.1、8.2、8.3、8.4、8.5</li> <li>• CentOS : 7.6、7.7、7.8、7.9</li> </ul>
1.3	<ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux : 7.6、7.7、7.8、7.9、8.0、8.1、8.2、8.3</li> <li>• CentOS : 7.6、7.7、7.8、7.9</li> </ul>
1.2	<ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux : 7.6、7.7、7.8、7.9、8.0、8.1.</li> <li>• CentOS : 7.6、7.7、7.8、7.9</li> </ul>

1. compatibleと指定すると、RHELではこのバージョンはサポートされなくなりますが、ONTAPメディアエーターは引き続きインストールできます。

#### 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は安全に更新できますが、ONTAPメディアエーターアプリケーション内で変更を適用するにはリブートが必要になる場合があります。再起動が必要な場合は、サービスウィンドウを使用することをお勧めします。
- OSカーネルを最新の状態に保つ必要があります。カーネルコアは、でサポートされているバージョンにアップグレードできます"[ONTAPメディアエーターのバージョンマトリックス](#)"。リブートは必須であるため、停止に備えてメンテナンス時間を計画する必要があります。
  - リブートする前に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する必要があります。
  - カーネルOSのアップグレードを開始する前に、サポートされているバージョンのSCSTを再インストールできる状態にしておく必要があります。



- カーネルのバージョンは、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。
- 特定のONTAPメディアエーターリリースでは、サポートされているOSリリース以降のカーネルにアップグレードすることはできません。(これは、テストしたSCSTモジュールがコンパイルされないことを示している可能性があります)。

#### UEFIセキュアブートが有効になっている場合のセキュリティキーの登録

UEFIセキュアブートを有効にしてONTAPメディアエーター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のインストール時に生成され、秘密鍵と公開鍵のペアとしてマシンに保存されます。ユーティリティを使用し`mokutil`で公開鍵をマシン所有者キー (Mok) としてUEFIファームウェアに追加し、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します。パスフレーズは、システムを再起動してMokをアクティブにするときに必要になるため、安全な場所に保存して`mokutil`ください。

システムがUEFI対応で、セキュアブートがオンになっているかどうかを確認するには、次の手順に従います。

#### 手順

1. がインストールされていない場合は `mokutil`、次のコマンドを実行します。

```
yum install mokutil
```

2. 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。



- 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存する必要があります。UEFIブートマネージャでキーを有効にするには、このパスフレーズが必要です。
- ONTAPメディエーター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。

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のままにすることも、安全な場所に移動することもできます。ただし、公開キーは、Boot Managerで使用するために既存の場所に保持する必要があります。詳細については、次のREADME.module-signingファイルを参照してください。

```
[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ブートマネージャを使用して新しいMokを承認します。手順2で指定したのパスフレーズが必要になり `mokutil` ます。

## UEFIセキュア ブートの無効化

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もできます。

手順

1. 物理マシンのBIOS設定で、「UEFIセキュアブート」オプションを無効にします。
2. VMのVMware設定で、vSphere 6.xの場合は[Safe Start]オプション、vSphere 7.xの場合は[Secure Boot]オプションを無効にします。

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**メディエーターをアップグレードする

ONTAPメディエーター用のホスト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メディエーターをアンインストールする必要があります。

開始する前に

Red Hat Enterprise LinuxまたはRocky Linuxとその関連リポジトリをシステムにインストールする際のベストプラクティスを次に示します。システムのインストールや構成が異なる場合は、追加の手順が必要になること

があります

- Red Hatのベストプラクティスに従ってRed Hat Enterprise LinuxまたはRocky Linuxをインストールする必要があります。CentOS 8.xのサポートが終了したため、互換性のあるバージョンのCentOS 8.xは推奨されません。
- Red Hat Enterprise LinuxまたはRocky LinuxへのONTAPメディアーターサービスのインストール中に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にアクセスしてインストールできるように、システムには適切なリポジトリへのアクセスが必要です。
- yumインストーラでRed Hat Enterprise Linuxリポジトリ内の依存ソフトウェアを検索するには、Red Hat Enterprise Linuxのインストール時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ておく必要があります。

Red Hat Subscription Managerの詳細については、Red Hatのマニュアルを参照してください。

- 次のポートが未使用でメディアーター用に使用可能である必要があります。
  - 31784
  - 3260
- サードパーティ製のファイアウォールを使用している場合は、[を参照してください。"ONTAPメディアーターのファイアウォール要件"](#)
- Linuxホストがインターネットにアクセスできない場所にある場合は、必要なパッケージがローカルリポジトリにあることを確認する必要があります。

Linux環境でLink Aggregation Control Protocol (LACP) を使用している場合は、カーネルを正しく設定し、が「2」に設定されていることを確認する必要があります `sysctl net.ipv4.conf.all.arp_ignore`。

必要なもの

ONTAPメディアーターサービスに必要なパッケージは次のとおりです。

RHEL / CentOSのすべてのバージョン	RHEL 8.x/Rocky Linux 8用の追加パッケージ	RHEL 9.x/Rocky Linux 9用の追加パッケージ
-------------------------	---------------------------------	---------------------------------

<ul style="list-style-type: none"> <li>• openssl</li> <li>• openssl-devel</li> <li>• kernel-devel-\$ (uname -r)</li> <li>• GCC</li> <li>• 作成</li> <li>• libselinux-utils</li> <li>• パッチ</li> <li>• bzip2</li> <li>• Perl -データ-ダンパ</li> <li>• Perl - ExtUtils - MakeMaker</li> <li>• efibootmgr</li> <li>• モクティル</li> </ul>	<ul style="list-style-type: none"> <li>• python3-pip の略</li> <li>• elfutils-libelf-devel</li> <li>• policycoreutils-python-utils</li> <li>• redhat-lsb-core</li> <li>• ピートン39</li> <li>• Python39 -デベル</li> </ul>	<ul style="list-style-type: none"> <li>• python3-pip の略</li> <li>• elfutils-libelf-devel</li> <li>• policycoreutils-python-utils</li> <li>• ピートン3</li> <li>• Python3 -デベル</li> </ul>
--	---	--

Mediatorインストールパッケージは自己解凍型の圧縮tarファイルで、次の内容が含まれます。

- 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証明書を使用することを推奨します。

#### タスクの内容

leapp-upgradeツールを使用してONTAPメディアーターのホストOSを新しいメジャーバージョン（7.xから8.xなど）にアップグレードする場合は、ONTAPメディアーターをアンインストールする必要があります。これは、システムに登録されているリポジトリにインストールされているRPMの新しいバージョンが検出されるためです。

**rpm**ファイルはONTAPメディアーターのインストーラの一部としてインストールされているため、その検索に含まれます。ただし、その**rpm**ファイルはインストーラの一部として展開され、登録されたリポジトリからダウンロードされなかったため、アップグレードが見つかりません。この場合、**leapp-upgrade**ツールはパッケージをアンインストールします。

サポート ケースの優先順位付けに使用されるログ ファイルを保持するには、OSのアップグレード前にファイルをバックアップして、ONTAP Mediatorパッケージの再インストール後にリストアする必要があります。ONTAP Mediatorが再インストールされるため、Mediatorに接続されているONTAPクラスタをインストール後に再接続する必要があります。



以下の手順は記載された順番で実行してください。ONTAP Mediatorの再インストール後、すぐにontap\_mediatorサービスを停止してログ ファイルを置き換え、サービスを再起動する必要があります。これは、ログが失われないようにするためです。

#### 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-1.9.0/ontap-mediator-1.9.0

ONTAP Mediator: Self Extracting Installer

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

## 4. ontap\_mediatorサービスを停止します。

```
[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サービスを開始します。

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

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

## MetroCluster over IPの手順

```
siteA::> metrocluster configuration-settings mediator show
Mediator IP      Port      Node      Configuration
Connection
Status          Status
-----
-----
172.31.40.122
                31784    siteA-node2    true          false
                siteA-node1    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-node1    true          true
                siteB-node2    true          true
                siteB-node2    true          true

siteA::>
```



## SnapMirrorアクティブ同期の手順

SnapMirrorアクティブ同期では、`/opt/netapp`ディレクトリ以外にTLS証明書をインストールした場合、証明書を再インストールする必要はありません。デフォルトの自己署名証明書を使用していた場合、または独自の証明書を`/opt/netapp`ディレクトリに配置していた場合は、証明書をバックアップしてリストアする必要があります。

```
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-nod1    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 2013

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

....

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::>
```

## リポジトリへのアクセスを有効にする

インストールプロセス中にONTAPメディアエーターが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. 次の表に示すように、アクセスする必要があるリポジトリを決定します。

オペレーティングシステム	アクセスを提供するリポジトリ
RHEL 7.x	<ul style="list-style-type: none"><li>• rhel-7-server-optional-rpms</li></ul>
RHEL 8.x	<ul style="list-style-type: none"><li>• rhel-8-for-x86_64-baseos-rpms</li><li>• RHEL-8-FOR-x86_64-AppStream-RPM</li></ul>
RHEL 9.x	<ul style="list-style-type: none"><li>• rhel-9-for-x86_64-baseos-rpms</li><li>• RHEL-9-FOR-X86_64-AppStream-RPM</li></ul>
CentOS 7.x	<ul style="list-style-type: none"><li>• C7.6.1810 - Baseリポジトリ</li></ul>
Rocky Linux 8	<ul style="list-style-type: none"><li>• AppStream</li><li>• ベースオス</li></ul>
Rocky Linux 9	<ul style="list-style-type: none"><li>• AppStream</li><li>• ベースオス</li></ul>

2. インストールプロセス中にONTAPメディアエーターが必要なパッケージにアクセスできるように、上記のリポジトリへのアクセスを有効にするには、次のいずれかの手順を実行します。



ONTAPメディアエーターが「extras」リポジトリと「optional」リポジトリに存在するPythonモジュールと依存関係を持っている場合は、ファイルと `rhel-X-for-x86_64-optional-rpms`` ファイルにアクセスする必要があります。`rhel-X-for-x86\_64-extras-rpms。

オペレーティングシステムが\* RHEL 7.x \*の場合は、次の手順を使用してリポジトリへのアクセスを有効にします。

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. コマンドを実行します yum repolist。

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 ~]#
```

## RHEL 8.xオペレーティングシステムでの手順

オペレーティングシステムが\* RHEL 8.x \*の場合は、次の手順を使用してリポジトリへのアクセスを有効にします。

### 手順

1. 必要なり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. コマンドを実行します `yum repolist`。

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 \*の場合は、次の手順を使用してリポジトリへのアクセスを有効にします。

手順

1. 必要なり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. コマンドを実行します `yum repolist`。

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\* CentOS 7.x \*の場合、次の手順 を使用してリポジトリへのアクセスを有効にします。



以下の例はCentOS 7.6のリポジトリを示していますが、他のバージョンのCentOSでは機能しない可能性があります。使用しているCentOSのバージョンに対応したベースリポジトリを使用します。

### 手順

1. C7.6.1810-Baseリポジトリを追加します。C7.6.1810 - Baseデフォルトリポジトリには、ONTAPメディアーターに必要な"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. コマンドを実行します `yum repolist`。

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 ~]#
```

## Rocky Linux 8または9オペレーティングシステムの手順

この手順は、オペレーティング・システムが\* Rocky Linux 8\*または\* Rocky Linux 9\*の場合に使用して、リポジトリへのアクセスを有効にします。

### 手順

1. 必要なリポジトリにサブスクライブします。

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

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

2. 操作を実行し `clean` 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 8 - AppStream
baseos                 Rocky Linux 8 - 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 ~]#
```

## Mediatorインストールパッケージをダウンロードする

インストールプロセスの一環として、Mediatorのインストールパッケージをダウンロードします。

### 手順



1. ONTAPメディエーターのページから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 メディエーターのダウンロードページ"](#)

2. メディエーターのインストールパッケージが現在の作業ディレクトリにあることを確認します。

```
[root@sdot-r730-0003a-d6 ~]# ls ontap-mediator-1.9.0.tgz
```

```
ontap-mediator-1.9.0.tgz
```



ONTAPメディエーターバージョン1.4以前の場合、インストーラの名前は `ontap-mediator`。

インターネットにアクセスできない場所にいる場合は、インストーラが必要なパッケージにアクセスできることを確認する必要があります。

3. 必要に応じて、MediatorのインストールパッケージをダウンロードディレクトリからLinux Mediatorホストのインストールディレクトリに移動します。
4. インストーラパッケージを解凍します。

```
tar xvfz ontap-mediator-1.9.0.tgz
```

```
ontap-mediator-1.9.0/  
ontap-mediator-1.9.0/csc-prod-ONTAP-Mediator.pem  
ontap-mediator-1.9.0/csc-prod-chain-ONTAP-Mediator.pem  
ontap-mediator-1.9.0/tsa-prod-ONTAP-Mediator.pem  
ontap-mediator-1.9.0/tsa-prod-chain-ONTAP-Mediator.pem  
ontap-mediator-1.9.0/ONTAP-Mediator-production.pub  
ontap-mediator-1.9.0/ontap-mediator-1.9.0  
ontap-mediator-1.9.0/ontap-mediator-1.9.0.sig.tsr  
ontap-mediator-1.9.0/ontap-mediator-1.9.0.tsr  
ontap-mediator-1.9.0/ontap-mediator-1.9.0.sig
```

## ONTAPメディエーターコード署名の確認

ONTAPメディエーターインストールパッケージをインストールする前に、ONTAPメディエーターコードの署名を確認する必要があります。

開始する前に

ONTAPメディエーターコードの署名を確認する前に、システムが次の要件を満たしている必要があります。

- OpenSSLバージョン1.0.2～3.0（基本検証用）
- タイムスタンプ局（TSA）操作のOpenSSLバージョン1.1.0以降

- OCSP検証のためのパブリックインターネットアクセス

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	説明
ONTAP-Mediator-production.pub	署名の検証に使用される公開鍵
csc-prod-chain-ONTAP-Mediator.pem	パブリック証明書CAの信頼チェーン
csc-prod-ONTAP-Mediator.pem	キーの生成に使用する証明書
ontap-mediator-1.9.0	バージョン1.9.0の製品インストール実行ファイル
ontap-mediator-1.9.0.sig	SHA-256はハッシュ化され、CSC-prodキーを使用してRSA署名されます（インストーラの署名）。
ontap-mediator-1.9.0.sig.tsr	OCSCPがインストーラの署名に使用する失効要求
ontap-mediator-1.9.0.tsr	タイムスタンプ署名要求ファイル
tsa-prod-ONTAP-Mediator.pem	TSRのパブリック証明書
tsa-prod-chain-ONTAP-Mediator.pem	TSRのパブリック証明書CAチェーン

## 手順

1. Online Certificate Status Protocol (OCSP) を使用して、で失効チェックを実行します `csc-prod-ONTAP-Mediator.pem`。
  - a. 開発者証明書ではURIが指定されていない可能性があるため、証明書の登録に使用するOCSP URLを検索します。

```
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. CSCの信頼チェーンと有効期限をローカルホストと照合して確認します。

```
openssl verify
```



openssl `パスのバージョンは有効な（自己署名ではない）必要があります`cert.pem。

```
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. 関連付けられた証明書を使用してファイルと `ontap-mediator-1.9.0.tsr` ファイルを確認し `ontap-mediator-1.9.0.sig.tsr` ます。

```
openssl ts -verify
```



`tsr`ファイルには、インストーラとコード署名に関連付けられたタイムスタンプ応答が含まれています。タイムスタンプにTSAから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が確認されます。検証はマシン上でローカルに実行されます。独立して、TSAサーバーにアクセスする必要はありません。

```
openssl ts -verify -data ontap-mediator-1.9.0.sig -in ontap-mediator-  
1.9.0.sig.tsr -CAfile tsa-prod-chain-ONTAP-Mediator.pem -untrusted tsa-  
prod-ONTAP-Mediator.pem  
openssl ts -verify -data ontap-mediator-1.9.0 -in ontap-mediator-  
1.9.0.tsr -CAfile tsa-prod-chain-ONTAP-Mediator.pem -untrusted tsa-prod-  
ONTAP-Mediator.pem
```

## 4. キーに対する署名を確認します。

```
openssl -dgst -verify
```

```
openssl dgst -sha256 -verify ONTAP-Mediator-production.pub -signature  
ontap-mediator-1.9.0.sig ontap-mediator-1.9.0
```

## ONTAPメディアーターコード署名の確認例 (コンソール出力)

```
[root@scspa2695423001 ontap-mediator-1.9.0]# pwd
/root/ontap-mediator-1.9.0
[root@scspa2695423001 ontap-mediator-1.9.0]# ls -l
total 63660
-r--r--r-- 1 root root      8582 Feb 19 15:02 csc-prod-chain-ONTAP-
Mediator.pem
-r--r--r-- 1 root root      2373 Feb 19 15:02 csc-prod-ONTAP-
Mediator.pem
-r-xr-xr-- 1 root root 65132818 Feb 20 15:17 ontap-mediator-1.9.0
-rw-r--r-- 1 root root      384 Feb 20 15:17 ontap-mediator-1.9.0.sig
-rw-r--r-- 1 root root      5437 Feb 20 15:17 ontap-mediator-
1.9.0.sig.tsr
-rw-r--r-- 1 root root      5436 Feb 20 15:17 ontap-mediator-1.9.0.tsr
-r--r--r-- 1 root root      625 Feb 19 15:02 ONTAP-Mediator-
production.pub
-r--r--r-- 1 root root      3323 Feb 19 15:02 tsa-prod-chain-ONTAP-
Mediator.pem
-r--r--r-- 1 root root      1740 Feb 19 15:02 tsa-prod-ONTAP-
Mediator.pem
[root@scspa2695423001 ontap-mediator-1.9.0]#
[root@scspa2695423001 ontap-mediator-1.9.0]#
/root/verify_ontap_mediator_signatures.sh
++ openssl version -d
++ cut -d '"' -f2
+ OPENSSLDIR=/etc/pki/tls
+ openssl version
OpenSSL 1.1.1k  FIPS 25 Mar 2021
++ openssl x509 -noout -ocsp_uri -in csc-prod-chain-ONTAP-Mediator.pem
+ ocsp_uri=http://ocsp.entrust.net
+ echo http://ocsp.entrust.net
http://ocsp.entrust.net
+ openssl ocsp -issuer csc-prod-chain-ONTAP-Mediator.pem -CAfile csc-
prod-chain-ONTAP-Mediator.pem -cert csc-prod-ONTAP-Mediator.pem -reqout
req.der
+ openssl ocsp -issuer csc-prod-chain-ONTAP-Mediator.pem -CAfile csc-
prod-chain-ONTAP-Mediator.pem -cert csc-prod-ONTAP-Mediator.pem -url
http://ocsp.entrust.net -resp_text -respout resp.der -verify_other csc-
prod-chain-ONTAP-Mediator.pem
OCSP Response Data:
  OCSP Response Status: successful (0x0)
  Response Type: Basic OCSP Response
  Version: 1 (0x0)
  Responder Id: C = US, O = "Entrust, Inc.", CN = Entrust Extended
```

Validation Code Signing CA - EVCS2

Produced At: Feb 28 05:01:00 2023 GMT

Responses:

Certificate ID:

Hash Algorithm: sha1

Issuer Name Hash: 69FA640329AB84E27220FE0927647B8194B91F2A

Issuer Key Hash: CE894F8251AA15A28462CA312361D261F8FE78

Serial Number: 511A542B57522AEB7295A640DC6200E5

Cert Status: good

This Update: Feb 28 05:00:00 2023 GMT

Next Update: Mar 4 04:59:59 2023 GMT

Signature Algorithm: sha512WithRSAEncryption

3c:1d:49:b0:93:62:37:3e:c7:38:e3:9f:9f:62:82:73:ed:f4:  
ea:00:6b:f1:01:cd:79:57:92:f1:9d:5d:85:9b:60:59:f8:6c:  
e6:f4:50:51:f3:4c:8a:51:dd:50:68:16:8f:20:24:7e:39:b0:  
44:94:8d:b0:61:da:b9:08:36:74:2d:44:55:62:fb:92:be:4a:  
e7:6c:8c:49:dd:0c:fd:d8:ce:20:08:0d:0f:5a:29:a3:19:03:  
9f:d3:df:41:f4:89:0f:73:18:3f:ac:bb:a7:a3:96:7d:c5:70:  
4c:57:cd:17:17:c6:8a:60:d1:37:c9:2d:81:07:2a:d7:a6:02:  
ee:ce:88:16:22:db:e3:43:64:1e:9b:0d:4d:31:66:fa:ab:a5:  
52:99:94:4a:4a:d0:52:c5:34:f5:18:c7:15:5b:ce:74:c2:fc:  
61:ea:55:aa:f1:2f:82:a3:6a:95:8d:7e:2b:38:49:4f:bf:b1:  
68:7b:1b:24:8b:1f:4d:c5:77:f0:71:af:9c:34:c8:7a:82:50:  
09:a2:19:6e:c6:30:4f:da:a2:79:08:f9:d0:ff:85:d9:2a:84:  
cf:0c:aa:75:8f:72:c9:a7:a2:83:e8:8b:cf:ed:0c:69:75:b6:  
2a:7b:6b:58:99:01:d8:34:ad:e1:89:25:27:1b:fa:d9:6d:32:  
97:3a:0b:0a:8e:a3:9e:e3:f4:e0:d6:1a:c9:b5:14:8c:3e:54:  
3b:37:17:1a:93:44:84:8b:4a:87:97:1e:76:43:3e:d3:ec:8b:  
7e:56:4a:3f:01:31:c0:e5:58:fb:50:ce:6f:b1:e7:35:f9:b7:  
a3:ef:6b:3b:21:95:37:a6:5b:8f:f0:15:18:36:65:89:a1:9c:  
9b:69:00:b4:b1:65:6a:bc:11:2d:d4:9b:b4:97:cc:cb:7a:0c:  
16:11:c1:75:58:7e:13:ab:56:3c:3f:93:5b:95:24:c6:54:52:  
1f:86:a9:16:ce:d9:ea:8b:3a:f3:4f:c4:8f:ad:de:e8:3e:3c:  
d2:51:51:ad:33:7f:d8:c5:33:24:26:f1:2d:9d:0e:9f:55:d0:  
68:bf:af:bd:68:4a:40:08:bc:92:a0:62:54:7d:16:7b:36:29:  
15:b1:cd:58:8e:fb:4a:f2:3e:94:8b:fe:56:95:cc:24:32:af:  
5f:71:99:18:ed:0c:64:94:f7:54:48:87:48:d0:6d:b3:42:04:  
96:03:73:a2:8e:8a:6a:b2:af:ee:56:19:a1:c6:35:12:59:ad:  
19:6a:fe:e0:f1:27:cc:96:4e:f0:4f:fb:6a:bd:ce:05:2c:aa:  
79:7c:df:02:5c:ca:53:7d:60:12:88:7c:ce:15:c7:d4:02:27:  
c1:ab:cf:71:30:1e:14:ba

WARNING: no nonce in response

Response verify OK

csc-prod-ONTAP-Mediator.pem: good

This Update: Feb 28 05:00:00 2023 GMT

```

Next Update: Mar  4 04:59:59 2023 GMT
+ openssl verify -untrusted csc-prod-chain-ONTAP-Mediator.pem -CApath
/etc/pki/tls csc-prod-ONTAP-Mediator.pem
csc-prod-ONTAP-Mediator.pem: OK
+ openssl verify -untrusted tsa-prod-chain-ONTAP-Mediator.pem -CApath
/etc/pki/tls tsa-prod-ONTAP-Mediator.pem
tsa-prod-ONTAP-Mediator.pem: OK
+ openssl ts -verify -data ontap-mediator-1.9.0.sig -in ontap-mediator-
1.9.0.sig.tsr -CAfile tsa-prod-chain-ONTAP-Mediator.pem -untrusted tsa-
prod-ONTAP-Mediator.pem
Using configuration from /etc/pki/tls/openssl.cnf
Verification: OK
+ openssl ts -verify -data ontap-mediator-1.9.0 -in ontap-mediator-
1.9.0.tsr -CAfile tsa-prod-chain-ONTAP-Mediator.pem -untrusted tsa-
prod-ONTAP-Mediator.pem
Using configuration from /etc/pki/tls/openssl.cnf
Verification: OK
+ openssl dgst -sha256 -verify ONTAP-Mediator-production.pub -signature
ontap-mediator-1.9.0.sig ontap-mediator-1.9.0
Verified OK
[root@scspa2695423001 ontap-mediator-1.9.0]#

```

## ONTAPメディエーターインストールパッケージのインストール

ONTAP メディエーターサービス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てホスト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-1.9.0/ontap-mediator-1.9.0 -y
```

```
[root@scs000099753 ~]# ./ontap-mediator-1.9.0/ontap-mediator-1.9.0 -y
```

インストール プロセスが開始され、必要なアカウントの作成と必要なパッケージのインストールが行われます。ホストに以前のバージョンのMediatorがインストールされている場合は、アップグレードするかどうかを確認するメッセージが表示されます。

2. ONTAP Mediator 1.4以降では、UEFIシステムでセキュア ブート メカニズムが有効になります。セキュア ブートが有効な場合、インストール後に追加の手順を実行してセキュリティ キーを登録する必要があります。

- READMEファイルの手順に従って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

- 必要なキーを特定します。

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



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



## ONTAP メディエーターのインストールの例 (コンソール出力)

```
[root@mediator_host ~]# cat /etc/os-release
NAME="Red Hat Enterprise Linux"
VERSION="9.4 (Plow) "
ID="rhel"
ID_LIKE="fedora"
VERSION_ID="9.4"
PLATFORM_ID="platform:el9"
PRETTY_NAME="Red Hat Enterprise Linux 9.4 (Plow) "
ANSI_COLOR="0;31"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:redhat:enterprise_linux:9::baseos"
HOME_URL="https://www.redhat.com/"
DOCUMENTATION_URL="https://access.redhat.com/documentation/en-
us/red_hat_enterprise_linux/9"
BUG_REPORT_URL="https://bugzilla.redhat.com/"

REDHAT_BUGZILLA_PRODUCT="Red Hat Enterprise Linux 9"
REDHAT_BUGZILLA_PRODUCT_VERSION=9.4
REDHAT_SUPPORT_PRODUCT="Red Hat Enterprise Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.4"
[root@mediator_host ~]#

[root@mediator_host ~]# tar -zxvf ontap-mediator-1.9.0.tgz
ontap-mediator-1.9.0/
ontap-mediator-1.9.0/csc-prod-chain-ONTAP-Mediator.pem
ontap-mediator-1.9.0/csc-prod-ONTAP-Mediator.pem
ontap-mediator-1.9.0/tsa-prod-ONTAP-Mediator.pem
ontap-mediator-1.9.0/tsa-prod-chain-ONTAP-Mediator.pem
ontap-mediator-1.9.0/ONTAP-Mediator-production.pub
ontap-mediator-1.9.0/ontap-mediator-1.9.0
ontap-mediator-1.9.0/ontap-mediator-1.9.0.sig.tsr
ontap-mediator-1.9.0/ontap-mediator-1.9.0.tsr
ontap-mediator-1.9.0/ontap-mediator-1.9.0.sig
[root@mediator_host ~]# ontap-mediator-1.9.0/ontap-mediator-1.9.0

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:SSL_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)? y(es)/N(o): 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).
Using default account names: netapp + mediatoradmin
```

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

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

```
+ Checking if SELinux is in enforcing mode
```

```
+ Checking for default Linux firewall
```

```
#####
Preparing for installation of ONTAP Mediator packages.
```

```
+ Installing required packages.
```

```
Last metadata expiration check: 0:15:55 ago on Thu 17 Oct 2024 09:06:29
AM EDT.
```

```
Package openssl-1:3.0.7-27.el9.x86_64 is already installed.
```

```
Package openssl-devel-1:3.0.7-27.el9.x86_64 is already installed.
```

```
Package kernel-devel-5.14.0-427.22.1.el9_4.x86_64 is already installed.
```

```
Package gcc-11.4.1-3.el9.x86_64 is already installed.
```

```
Package make-1:4.3-8.el9.x86_64 is already installed.
```

```
Package libselinux-utils-3.6-1.el9.x86_64 is already installed.
```

```
Package perl-Data-Dumper-2.174-462.el9.x86_64 is already installed.
```

```
Package bzip2-1.0.8-8.el9.x86_64 is already installed.
```

```
Package elfutils-libelf-devel-0.190-2.el9.x86_64 is already installed.
```

```
Package policycoreutils-python-utils-3.6-2.1.el9.noarch is already
```

installed.

Package python3-3.9.18-3.el9.x86\_64 is already installed.

Dependencies resolved.

```
=====
=====
=====
=====
```

```
Package
Architecture                Version
Repository                    Size
```

```
=====
=====
=====
=====
```

Installing:

```
  efibootmgr                x86_64
16-12.el9                   rhel-9-for-x86_64-
baseos-rpms                 48 k
  mokutil                   x86_64
2:0.6.0-4.el9               rhel-9-for-x86_64-
baseos-rpms                 50 k
  patch                     x86_64
2.7.6-16.el9                rhel-9-for-x86_64-
appstream-rpms              130 k
  perl-ExtUtils-MakeMaker   noarch
2:7.60-3.el9                rhel-9-for-x86_64-
appstream-rpms              304 k
  python3-devel             x86_64
3.9.18-3.el9_4.5            rhel-9-for-x86_64-
appstream-rpms              248 k
  python3-pip               noarch
21.2.3-8.el9                rhel-9-for-x86_64-
appstream-rpms              2.0 M
```

Upgrading:

```
  openssl                   x86_64
1:3.0.7-28.el9_4            rhel-9-for-x86_64-
baseos-rpms                 1.2 M
  openssl-devel             x86_64
1:3.0.7-28.el9_4            rhel-9-for-x86_64-
appstream-rpms              4.1 M
  openssl-libs              i686
1:3.0.7-28.el9_4            rhel-9-for-x86_64-
baseos-rpms                 1.9 M
  openssl-libs              x86_64
1:3.0.7-28.el9_4            rhel-9-for-x86_64-
baseos-rpms                 1.9 M
```

```

python-unversioned-command                                noarch
3.9.18-3.el9_4.5                                         rhel-9-for-x86_64-
appstream-rpms                                           10 k
python3                                                  x86_64
3.9.18-3.el9_4.5                                         rhel-9-for-x86_64-
baseos-rpms                                              30 k
python3-libs                                             x86_64
3.9.18-3.el9_4.5                                         rhel-9-for-x86_64-
baseos-rpms                                              7.9 M
Installing dependencies:
efi-filesystem                                           noarch
6-2.el9_0                                                rhel-9-for-x86_64-
baseos-rpms                                              9.5 k
efivar-libs                                             x86_64
38-3.el9                                                 rhel-9-for-x86_64-
baseos-rpms                                              124 k
perl-AutoSplit                                           noarch
5.74-481.el9                                             rhel-9-for-x86_64-
appstream-rpms                                           22 k
perl-Benchmark                                           noarch
1.23-481.el9                                             rhel-9-for-x86_64-
appstream-rpms                                           27 k
perl-CPAN-Meta-YAML                                     noarch
0.018-461.el9                                           rhel-9-for-x86_64-
appstream-rpms                                           29 k
perl-Devel-PPPort                                       x86_64
3.62-4.el9                                               rhel-9-for-x86_64-
appstream-rpms                                           216 k
perl-ExtUtils-Command                                   noarch
2:7.60-3.el9                                             rhel-9-for-x86_64-
appstream-rpms                                           16 k
perl-ExtUtils-Constant                                  noarch
0.25-481.el9                                           rhel-9-for-x86_64-
appstream-rpms                                           49 k
perl-ExtUtils-Install                                   noarch
2.20-4.el9                                               rhel-9-for-x86_64-
appstream-rpms                                           47 k
perl-ExtUtils-Manifest                                  noarch
1:1.73-4.el9                                             rhel-9-for-x86_64-
appstream-rpms                                           37 k
perl-ExtUtils-ParseXS                                   noarch
1:3.40-460.el9                                          rhel-9-for-x86_64-
appstream-rpms                                           190 k
perl-File-Compare                                       noarch
1.100.600-481.el9                                       rhel-9-for-x86_64-
appstream-rpms                                           14 k

```

```

perl-JSON-PP                                noarch
1:4.06-4.el9                                rhel-9-for-x86_64-
appstream-rpms                               69 k
perl-Test-Harness                            noarch
1:3.42-461.el9                               rhel-9-for-x86_64-
appstream-rpms                              299 k
perl-lib                                     x86_64
0.65-481.el9                                 rhel-9-for-x86_64-
appstream-rpms                               15 k
perl-version                                 x86_64
7:0.99.28-4.el9                              rhel-9-for-x86_64-
appstream-rpms                              67 k
systemtap-sdt-devel                          x86_64
5.0-4.el9                                    rhel-9-for-x86_64-
appstream-rpms                               77 k
Installing weak dependencies:
perl-CPAN-Meta                               noarch
2.150010-460.el9                             rhel-9-for-x86_64-
appstream-rpms                              206 k
perl-CPAN-Meta-Requirements                 noarch
2.140-461.el9                                rhel-9-for-x86_64-
appstream-rpms                              34 k
perl-devel                                   x86_64
4:5.32.1-481.el9                            rhel-9-for-x86_64-
appstream-rpms                              680 k
perl-doc                                     noarch
5.32.1-481.el9                              rhel-9-for-x86_64-
appstream-rpms                              4.6 M

```

Transaction Summary

```

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

```

```

Install 27 Packages
Upgrade 7 Packages

```

Total download size: 27 M

Is this ok [y/N]: y

Downloading Packages:

```

(1/34): perl-CPAN-Meta-YAML-0.018-461.el9.noarch.rpm
220 kB/s | 29 kB      00:00
(2/34): perl-CPAN-Meta-Requirements-2.140-461.el9.noarch.rpm
249 kB/s | 34 kB      00:00
(3/34): perl-ExtUtils-Install-2.20-4.el9.noarch.rpm
4.2 MB/s | 47 kB      00:00

```

```
(4/34): perl-CPAN-Meta-2.150010-460.el9.noarch.rpm
1.3 MB/s | 206 kB      00:00
(5/34): perl-version-0.99.28-4.el9.x86_64.rpm
5.5 MB/s | 67 kB      00:00
(6/34): perl-ExtUtils-Manifest-1.73-4.el9.noarch.rpm
3.9 MB/s | 37 kB      00:00
(7/34): perl-ExtUtils-MakeMaker-7.60-3.el9.noarch.rpm
16 MB/s | 304 kB     00:00
(8/34): perl-ExtUtils-ParseXS-3.40-460.el9.noarch.rpm
11 MB/s | 190 kB     00:00
(9/34): patch-2.7.6-16.el9.x86_64.rpm
15 MB/s | 130 kB     00:00
(10/34): perl-Test-Harness-3.42-461.el9.noarch.rpm
15 MB/s | 299 kB     00:00
(11/34): perl-Devel-PPPort-3.62-4.el9.x86_64.rpm
14 MB/s | 216 kB     00:00
(12/34): perl-ExtUtils-Command-7.60-3.el9.noarch.rpm
1.4 MB/s | 16 kB     00:00
(13/34): perl-JSON-PP-4.06-4.el9.noarch.rpm
6.9 MB/s | 69 kB     00:00
(14/34): perl-Benchmark-1.23-481.el9.noarch.rpm
3.9 MB/s | 27 kB     00:00
(15/34): systemtap-sdt-devel-5.0-4.el9.x86_64.rpm
9.4 MB/s | 77 kB     00:00
(16/34): perl-AutoSplit-5.74-481.el9.noarch.rpm
2.8 MB/s | 22 kB     00:00
(17/34): perl-ExtUtils-Constant-0.25-481.el9.noarch.rpm
5.9 MB/s | 49 kB     00:00
(18/34): perl-File-Compare-1.100.600-481.el9.noarch.rpm
1.7 MB/s | 14 kB     00:00
(19/34): perl-devel-5.32.1-481.el9.x86_64.rpm
21 MB/s | 680 kB     00:00
(20/34): perl-lib-0.65-481.el9.x86_64.rpm
2.1 MB/s | 15 kB     00:00
(21/34): python3-pip-21.2.3-8.el9.noarch.rpm
26 MB/s | 2.0 MB     00:00
(22/34): efi-filesystem-6-2.el9_0.noarch.rpm
1.8 MB/s | 9.5 kB     00:00
(23/34): python3-devel-3.9.18-3.el9_4.5.x86_64.rpm
8.6 MB/s | 248 kB     00:00
(24/34): efibootmgr-16-12.el9.x86_64.rpm
5.0 MB/s | 48 kB     00:00
(25/34): efivar-libs-38-3.el9.x86_64.rpm
15 MB/s | 124 kB     00:00
(26/34): mokutil-0.6.0-4.el9.x86_64.rpm
5.2 MB/s | 50 kB     00:00
```

```

(27/34): python-unversioned-command-3.9.18-3.el9_4.5.noarch.rpm
2.2 MB/s | 10 kB      00:00
(28/34): python3-3.9.18-3.el9_4.5.x86_64.rpm
6.9 MB/s | 30 kB      00:00
(29/34): perl-doc-5.32.1-481.el9.noarch.rpm
27 MB/s | 4.6 MB      00:00
(30/34): openssl-3.0.7-28.el9_4.x86_64.rpm
30 MB/s | 1.2 MB      00:00
(31/34): openssl-devel-3.0.7-28.el9_4.x86_64.rpm
25 MB/s | 4.1 MB      00:00
(32/34): openssl-libs-3.0.7-28.el9_4.x86_64.rpm
22 MB/s | 1.9 MB      00:00
(33/34): openssl-libs-3.0.7-28.el9_4.i686.rpm
29 MB/s | 1.9 MB      00:00
(34/34): python3-libs-3.9.18-3.el9_4.5.x86_64.rpm
27 MB/s | 7.9 MB      00:00
-----
-----
-----
-----
Total
44 MB/s | 27 MB      00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
1/1
  Upgrading      : openssl-libs-1:3.0.7-28.el9_4.x86_64
1/41
  Installing     : perl-version-7:0.99.28-4.el9.x86_64
2/41
  Installing     : perl-CPAN-Meta-Requirements-2.140-461.el9.noarch
3/41
  Upgrading      : python3-libs-3.9.18-3.el9_4.5.x86_64
4/41
  Upgrading      : python3-3.9.18-3.el9_4.5.x86_64
5/41
  Upgrading      : python-unversioned-command-3.9.18-3.el9_4.5.noarch
6/41
  Installing     : efivar-libs-38-3.el9.x86_64
7/41
  Installing     : perl-File-Compare-1.100.600-481.el9.noarch
8/41
  Installing     : perl-JSON-PP-1:4.06-4.el9.noarch

```

```
9/41
  Installing      : perl-ExtUtils-ParseXS-1:3.40-460.el9.noarch
10/41
  Installing      : python3-pip-21.2.3-8.el9.noarch
11/41
  Installing      : systemtap-sdt-devel-5.0-4.el9.x86_64
12/41
  Installing      : efi-filessystem-6-2.el9_0.noarch
13/41
  Installing      : perl-lib-0.65-481.el9.x86_64
14/41
  Installing      : perl-doc-5.32.1-481.el9.noarch
15/41
  Installing      : perl-ExtUtils-Constant-0.25-481.el9.noarch
16/41
  Installing      : perl-AutoSplit-5.74-481.el9.noarch
17/41
  Installing      : perl-Benchmark-1.23-481.el9.noarch
18/41
  Installing      : perl-Test-Harness-1:3.42-461.el9.noarch
19/41
  Installing      : perl-ExtUtils-Command-2:7.60-3.el9.noarch
20/41
  Installing      : perl-Devel-PPPport-3.62-4.el9.x86_64
21/41
  Installing      : perl-ExtUtils-Manifest-1:1.73-4.el9.noarch
22/41
  Installing      : perl-CPAN-Meta-YAML-0.018-461.el9.noarch
23/41
  Installing      : perl-CPAN-Meta-2.150010-460.el9.noarch
24/41
  Installing      : perl-devel-4:5.32.1-481.el9.x86_64
25/41
  Installing      : perl-ExtUtils-Install-2.20-4.el9.noarch
26/41
  Installing      : perl-ExtUtils-MakeMaker-2:7.60-3.el9.noarch
27/41
  Installing      : efibootmgr-16-12.el9.x86_64
28/41
  Installing      : python3-devel-3.9.18-3.el9_4.5.x86_64
29/41
  Installing      : mokutil-2:0.6.0-4.el9.x86_64
30/41
  Upgrading       : openssl-devel-1:3.0.7-28.el9_4.x86_64
31/41
  Upgrading       : openssl-1:3.0.7-28.el9_4.x86_64
```



```
32/41
  Installing      : patch-2.7.6-16.el9.x86_64
33/41
  Upgrading      : openssl-libs-1:3.0.7-28.el9_4.i686
34/41
  Cleanup        : openssl-devel-1:3.0.7-27.el9.x86_64
35/41
  Cleanup        : python-unversioned-command-3.9.18-3.el9.noarch
36/41
  Cleanup        : openssl-1:3.0.7-27.el9.x86_64
37/41
  Cleanup        : openssl-libs-1:3.0.7-27.el9.i686
38/41
  Cleanup        : python3-3.9.18-3.el9.x86_64
39/41
  Cleanup        : python3-libs-3.9.18-3.el9.x86_64
40/41
  Cleanup        : openssl-libs-1:3.0.7-27.el9.x86_64
41/41
  Running scriptlet: openssl-libs-1:3.0.7-27.el9.x86_64
41/41
  Verifying      : perl-CPAN-Meta-2.150010-460.el9.noarch
1/41
  Verifying      : perl-CPAN-Meta-Requirements-2.140-461.el9.noarch
2/41
  Verifying      : perl-CPAN-Meta-YAML-0.018-461.el9.noarch
3/41
  Verifying      : perl-ExtUtils-Install-2.20-4.el9.noarch
4/41
  Verifying      : perl-version-7:0.99.28-4.el9.x86_64
5/41
  Verifying      : perl-ExtUtils-MakeMaker-2:7.60-3.el9.noarch
6/41
  Verifying      : perl-ExtUtils-Manifest-1:1.73-4.el9.noarch
7/41
  Verifying      : perl-ExtUtils-ParseXS-1:3.40-460.el9.noarch
8/41
  Verifying      : perl-Test-Harness-1:3.42-461.el9.noarch
9/41
  Verifying      : patch-2.7.6-16.el9.x86_64
10/41
  Verifying      : perl-Devel-PPPport-3.62-4.el9.x86_64
11/41
  Verifying      : perl-ExtUtils-Command-2:7.60-3.el9.noarch
12/41
  Verifying      : perl-JSON-PP-1:4.06-4.el9.noarch
```

```
13/41
  Verifying      : perl-Benchmark-1.23-481.el9.noarch
14/41
  Verifying      : python3-pip-21.2.3-8.el9.noarch
15/41
  Verifying      : systemtap-sdt-devel-5.0-4.el9.x86_64
16/41
  Verifying      : perl-AutoSplit-5.74-481.el9.noarch
17/41
  Verifying      : perl-ExtUtils-Constant-0.25-481.el9.noarch
18/41
  Verifying      : perl-File-Compare-1.100.600-481.el9.noarch
19/41
  Verifying      : perl-devel-4:5.32.1-481.el9.x86_64
20/41
  Verifying      : perl-doc-5.32.1-481.el9.noarch
21/41
  Verifying      : perl-lib-0.65-481.el9.x86_64
22/41
  Verifying      : python3-devel-3.9.18-3.el9_4.5.x86_64
23/41
  Verifying      : efi-filesystem-6-2.el9_0.noarch
24/41
  Verifying      : efibootmgr-16-12.el9.x86_64
25/41
  Verifying      : efivar-libs-38-3.el9.x86_64
26/41
  Verifying      : mokutil-2:0.6.0-4.el9.x86_64
27/41
  Verifying      : python-unversioned-command-3.9.18-3.el9_4.5.noarch
28/41
  Verifying      : python-unversioned-command-3.9.18-3.el9.noarch
29/41
  Verifying      : openssl-devel-1:3.0.7-28.el9_4.x86_64
30/41
  Verifying      : openssl-devel-1:3.0.7-27.el9.x86_64
31/41
  Verifying      : python3-3.9.18-3.el9_4.5.x86_64
32/41
  Verifying      : python3-3.9.18-3.el9.x86_64
33/41
  Verifying      : python3-libs-3.9.18-3.el9_4.5.x86_64
34/41
  Verifying      : python3-libs-3.9.18-3.el9.x86_64
35/41
  Verifying      : openssl-1:3.0.7-28.el9_4.x86_64
```

```

36/41
  Verifying      : openssl-1:3.0.7-27.el9.x86_64
37/41
  Verifying      : openssl-libs-1:3.0.7-28.el9_4.x86_64
38/41
  Verifying      : openssl-libs-1:3.0.7-27.el9.x86_64
39/41
  Verifying      : openssl-libs-1:3.0.7-28.el9_4.i686
40/41
  Verifying      : openssl-libs-1:3.0.7-27.el9.i686
41/41
Installed products updated.

Upgraded:
  openssl-1:3.0.7-28.el9_4.x86_64      openssl-devel-1:3.0.7-
28.el9_4.x86_64      openssl-libs-1:3.0.7-28.el9_4.i686      openssl-
libs-1:3.0.7-28.el9_4.x86_64      python-unversioned-command-3.9.18-
3.el9_4.5.noarch
  python3-3.9.18-3.el9_4.5.x86_64      python3-libs-3.9.18-
3.el9_4.5.x86_64
Installed:
  efi-filesystem-6-2.el9_0.noarch
efibootmgr-16-12.el9.x86_64      efivar-libs-38-
3.el9.x86_64      mokutil-2:0.6.0-4.el9.x86_64
  patch-2.7.6-16.el9.x86_64      perl-
AutoSplit-5.74-481.el9.noarch      perl-Benchmark-1.23-
481.el9.noarch      perl-CPAN-Meta-2.150010-
460.el9.noarch
  perl-CPAN-Meta-Requirements-2.140-461.el9.noarch      perl-
CPAN-Meta-YAML-0.018-461.el9.noarch      perl-Devel-PPPort-
3.62-4.el9.x86_64      perl-ExtUtils-Command-2:7.60-
3.el9.noarch
  perl-ExtUtils-Constant-0.25-481.el9.noarch      perl-
ExtUtils-Install-2.20-4.el9.noarch      perl-ExtUtils-
MakeMaker-2:7.60-3.el9.noarch      perl-ExtUtils-Manifest-1:1.73-
4.el9.noarch
  perl-ExtUtils-ParseXS-1:3.40-460.el9.noarch      perl-
File-Compare-1.100.600-481.el9.noarch      perl-JSON-PP-1:4.06-
4.el9.noarch      perl-Test-Harness-1:3.42-
461.el9.noarch
  perl-devel-4:5.32.1-481.el9.x86_64      perl-doc-
5.32.1-481.el9.noarch      perl-lib-0.65-
481.el9.x86_64      perl-version-7:0.99.28-
4.el9.x86_64
  python3-devel-3.9.18-3.el9_4.5.x86_64      python3-
pip-21.2.3-8.el9.noarch      systemtap-sdt-devel-5.0-

```

4.el9.x86\_64

Complete!

OS package installations finished

+ Installing ONTAP Mediator. (Log: /root/ontap\_mediator.T7uce6/ontap-mediator-1.9.0/ontap-mediator-1.9.0/install\_20241017092214.log)

This step will take several minutes. Use the log file to view progress.

Sudoer config verified

ONTAP Mediator rsyslog and logging rotation enabled

+ Install successful. (Moving log to /opt/netapp/lib/ontap\_mediator/log/install\_20241017092214.log)

+ 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 Thu 2024-10-17 09:27:14 EDT; 1min 12s ago
```

```
Process: 54470
```

```
ExecStartPre=/opt/netapp/lib/ontap_mediator/tools/otm_logs_fs.sh (code=exited, status=0/SUCCESS)
```

```
Main PID: 54489 (uwsgi)
```

```
Status: "uWSGI is ready"
```

```
Tasks: 3 (limit: 11104)
```

```
Memory: 77.1M
```

```
CPU: 2.507s
```

```
CGroup: /system.slice/ontap_mediator.service
```

```
└─54489 /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi
```

```
--ini /opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini
```

```

└─54504 /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi
--ini /opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini
└─54507 /opt/netapp/lib/ontap_mediator/pyenv/bin/uwsgi
--ini /opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini

Oct 17 09:27:10 mediator_host ontap_mediator[54476]: Creating
filesystem with 192000 4k blocks and 48000 inodes
Oct 17 09:27:10 mediator_host ontap_mediator[54476]: Filesystem UUID:
b1fa0a40-0e7d-4c67-bbff-33421f3ec61b
Oct 17 09:27:10 mediator_host ontap_mediator[54476]: Superblock backups
stored on blocks:
Oct 17 09:27:10 mediator_host ontap_mediator[54476]:          32768,
98304, 163840
Oct 17 09:27:10 mediator_host ontap_mediator[54476]: [41B blob data]
Oct 17 09:27:10 mediator_host ontap_mediator[54476]: [38B blob data]
Oct 17 09:27:10 mediator_host ontap_mediator[54476]: Creating journal
(4096 blocks): done
Oct 17 09:27:10 mediator_host ontap_mediator[54476]: [75B blob data]
Oct 17 09:27:10 mediator_host ontap_mediator[54489]: [uWSGI] getting
INI configuration from
/opt/netapp/lib/ontap_mediator/uwsgi/ontap_mediator.ini
Oct 17 09:27:14 mediator_host systemd[1]: Started 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 Thu 2024-10-17 09:27:08 EDT; 1min
   32s ago
     Process: 54384 ExecStart=/etc/init.d/scst start (code=exited,
   status=0/SUCCESS)
     Process: 54467 ExecStartPost=/usr/sbin/modprobe scst_vdisk
   (code=exited, status=0/SUCCESS)
    Main PID: 54425 (iscsi-scstd)
       Tasks: 1 (limit: 11104)
      Memory: 1.2M
         CPU: 494ms
    CGroup: /system.slice/mediator-scst.service
           └─54425 /usr/local/sbin/iscsi-scstd

Oct 17 09:27:07 mediator_host systemd[1]: Starting mediator-
scst.service...
Oct 17 09:27:08 mediator_host iscsi-scstd[54423]: max_data_seg_len
1048576, max_queued_cmds 2048
Oct 17 09:27:08 mediator_host scst[54384]: Loading and configuring SCST
Oct 17 09:27:08 mediator_host systemd[1]: Started mediator-

```

```
scst.service.  
[root@mediator_host ~]#
```

## インストールの確認

ONTAPメディアエーターをインストールしたら、ONTAPメディアエーターサービスが実行されていることを確認する必要があります。

### 手順

1. ONTAPメディアエーターサービス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メディアーターサービスで使用されてい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
```

## インストール後の設定

ONTAP Mediatorサービスをインストールして実行したら、Mediatorの機能を使用するために、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。

- MetroCluster IP構成でONTAPメディアーターサービスを使用するには、を参照してください"[MetroCluster IP構成でのONTAPメディアーターサービスの設定](#)"。
- SnapMirrorアクティブ同期を使用するには、を参照してください"[ONTAPメディアーターサービスをインストールし、ONTAPクラスタ構成を確認](#)"。

### ONTAPメディアーターのセキュリティポリシーの設定

ONTAPメディアーターサーバでは、いくつかの設定可能なセキュリティ設定がサポートされます。すべての設定のデフォルト値は、読み取り専用ファイルで提供され `low\_space\_threshold\_mib: 10` ます。

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

に設定した値はすべて `ontap\_mediator.user\_config.yaml` デフォルト値よりも優先され、ONTAPメディアエーターのすべてのアップグレードで維持されます。

変更したら `ontap_mediator.user_config.yaml`、ONTAPメディアエーターサービスを再起動します。

```
systemctl restart ontap_mediator
```

**ONTAPメディアエーターの属性を変更します。**

このセクションで説明するONTAPメディアエーターの属性は、必要に応じて変更できます。



ONTAPメディアエーターのアップグレード時には変更後の値が維持されないため、のその他のデフォルト値は `ontap\_mediator.config.yaml` 変更しないでください。

ONTAPメディアエーターの属性を変更するには、必要な変数をファイルにコピーして `ontap\_mediator.user\_config.yaml` デフォルトの設定を上書きします。

サードパーティの**SSL**証明書のインストール

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を使用して、ONTAPメディアエーターサービスで使用される証明書ファイルが制御されます。

次の表に示すデフォルトの変数は、ファイルに含まれて

`/opt/netapp/lib/ontap\_mediator/ontap\_mediator/server\_config/ontap\_mediator.config.yaml` います。

変数	パス
<code>cert_path</code>	<code>/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.crt</code>
<code>key_path</code>	<code>/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_server.key</code>
<code>ca_cert_path</code>	<code>/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.crt</code>
<code>ca_key_path</code>	<code>/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.key</code>
<code>ca_serial_path</code>	<code>/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ca.srl</code>
<code>cert_valid_days</code>	1095
<code>x509_passin_pwd</code>	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、署名済みクライアント証明書のパスフレーズです。

次の表に示すデフォルトの変数は、ファイルに含まれて  
`/opt/netapp/lib/ontap\_mediator/uwsgi/ontap\_mediator.ini` 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する場合は、ONTAPメディアエーターサービスを再起動して変更を適用します。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、[を参照してください"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"](#)。

#### パスワード攻撃からの保護

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、およびの `retry\_limit` 値を設定し `window\_seconds` ます。

例：

- 5分間の猶予期間を設けて推測し、失敗回数をゼロにリセットします。

```
authentication_lock_window_seconds: 300
```

- 期間内に5つの障害が発生した場合は、アカウントをロックします。

```
authentication_retry_limit: 5
```

- 各試行を拒否する前に発生する遅延を設定することで、ブルートフォースパスワード推測攻撃の影響を軽減し、攻撃の速度を低下させます。

```
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に関するルール

次のフィールドは、ONTAPメディアエーターAPIユーザ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)
```

## 空きスペースの制御

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

スペースが設定されたしきい値を下回ると、サービスは警告イベントを発行します。

```
low_space_threshold_mib: 10
```

## リザーブ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メディアエーターサーバをインストールすると、ログ用に独立したディスクスペースが作成されます。Mediatorのログに明示的に使用される、合計700MBのディスクスペースを含む新しい固定サイズのファイルがインストーラによって作成されます。

この機能を無効にしてデフォルトのディスク容量を使用するには、次の手順に従います。

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`

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



ディスクスペースを切り替えても、既存のログは消去されません。以前のログはすべてバックアップされ、Mediatorの切り替えと再起動のあとに現在のディスクスペースに移動されます。

## ONTAPメディアーターサービスを管理します。

ユーザクレデンシャルの変更、サービスの停止と再有効化、健全性の確認、ホストメンテナンスのためのSCSTのインストールとアンインストールなど、ONTAPメディアーターサービスの管理を行います。また、自己署名証明書の再生成、信頼された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、証明書の管理も行うことができます。

### ユーザ名の変更

ユーザ名は次の手順を使用して変更できます。

#### タスクの内容

このタスクは、ONTAPメディアーターサービスがインストールされているLinuxホストで実行します。

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) : コマンドを実行し mediator\_change\_user、次の例に示すようにプロンプトに応答します。

```
[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

パスワードは次の手順を使用して変更できます。

### タスクの内容

このタスクは、ONTAPメディアエーターサービスがインストールされているLinuxホストで実行します。

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)** : コマンドを実行し `mediator_change_password`、次の例に示すようにプロンプトに応答します。

```
[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、パスワードが「mediator1」から「mediator2」に変更されています。

```
[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メディアーターサービスを停止します。

ONTAPメディアーターサービスを停止するには、次の手順を実行します。

手順

1. ONTAPメディアーターを停止します。

```
systemctl stop ontap_mediator
```

2. SCSTを停止します。

```
systemctl stop mediator-scst
```

3. ONTAPメディアーターとSCSTを無効にします。

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

## ONTAPメディアーターサービスを再度有効にする

ONTAPメディアーターサービスを再度有効にするには、次の手順を実行します。

手順

1. ONTAPメディアーターとSCSTを有効にします。

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

2. SCSTを開始します。

```
systemctl start mediator-scst
```

3. ONTAPメディアーターを起動します。

```
systemctl start ontap_mediator
```

## ONTAPメディアーターが正常であることの確認

ONTAPメディアーターをインストールしたら、ONTAPメディアーターサービスが実行されていることを確認する必要があります。

手順

1. ONTAPメディアーターサービス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メディアエーターサービスで使用されてい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

## ホストメンテナンスを実行するためのSCSTの手動アンインストール

SCSTをアンインストールするには、インストールされているONTAPメディアエータのバージョンに使用するSCST tarバンドルが必要です。

### 手順

1. 次の表に示すように、適切なSCSTバンドルをダウンロードして解凍します。

バージョン	使用するtarバンドル
ONTAPメディアエーター1.9	scst-3.8.0.tar.bz2
ONTAPメディアエーター1.8	scst-3.8.0.tar.bz2
ONTAPメディアエーター1.7	scst-3.7.0.tar.bz2
ONTAPメディアエーター1.6	scst-3.7.0.tar.bz2
ONTAPメディアエーター1.5	scst-3.6.0.tar.bz2
ONTAPメディアエーター1.4	scst-3.6.0.tar.bz2
ONTAPメディアエーター1.3	scst-3.5.0.tar.bz2
ONTAPメディアエーター1.1	scst-3.4.0.tar.bz2
ONTAPメディアエーター1.0	scst-3.3.0.tar.bz2

2. 「scst」ディレクトリで次の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`

## ホストのメンテナンスを実行するためのSCSTの手動インストール

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ONTAPメディアエータのバージョンに使用するSCST tarバンドルが必要です（を参照上の表）。

1. 「scst」ディレクトリで次の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`
- 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`

2. セキュアブートが有効になっている場合は、リブートする前に、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確認します。

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

- b. カーネルのリリースを確認します。

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

- c. 各ファイルに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_module_key.priv \
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/scst_module_key.der \
_module-filename_
```



d. UEFIファームウェアで正しいキーをインストールします。

UEFIキーのインストール手順は、次の場所にあります。

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

生成されたUEFIキーは次の場所にあります。

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

3. リブートを実行します。

```
reboot
```

## ONTAPメディアエーターサービスをアンインストールします。

必要に応じて、ONTAPメディアエーターサービスを削除できます。

開始する前に

ONTAPメディアエーターサービスを削除する前に、ONTAPメディアエーターをONTAPから切断する必要があります。

タスクの内容

このタスクは、ONTAPメディアエーターサービスがインストールされているLinuxホストで実行する必要があります。

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
```

ステップ

1. ONTAPメディアエーターサービス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メディアエーター1.7以降では、次の手順を使用して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メディアエーター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せん。

### タスクの内容

- このタスクは、ONTAPメディアエーターサービスがインストールされているLinuxホストで実行します。
- このタスクは、ONTAPメディアエーター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廃止され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証明書に置き換えたあと、このタスクを使用して証明書を再生成します。自己署名証明書がないと、この手順は失敗します。

### ステップ

現在のホストの新しい一時的な自己署名証明書を再生成するには、次の手順を実行します。

1. ONTAPメディアエーターサービスを再起動します。

```
./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

```

## 自己署名証明書を信頼できるサードパーティ証明書に置き換える

自己署名証明書がサポートされている場合は、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。

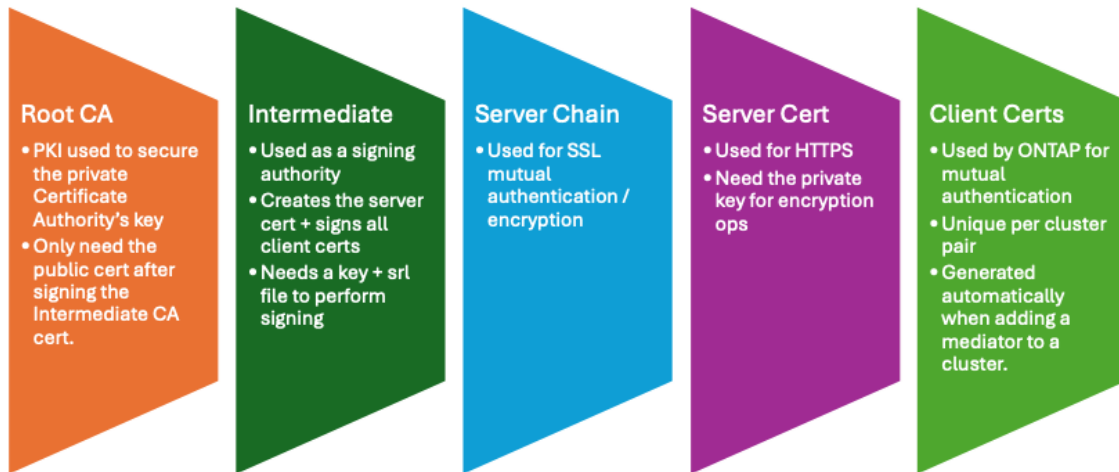


- サードパーティ証明書は、ONTAP 9.16.1以降および一部の以前のONTAPパッチリリースでのみサポートされています。を参照して "[NetApp Bugs OnlineのバグIDCONTAP-243278](#)"
- サードパーティの証明書は、ONTAP Mediator 1.7以降を実行するシステムでのみサポートされます。

### タスクの内容

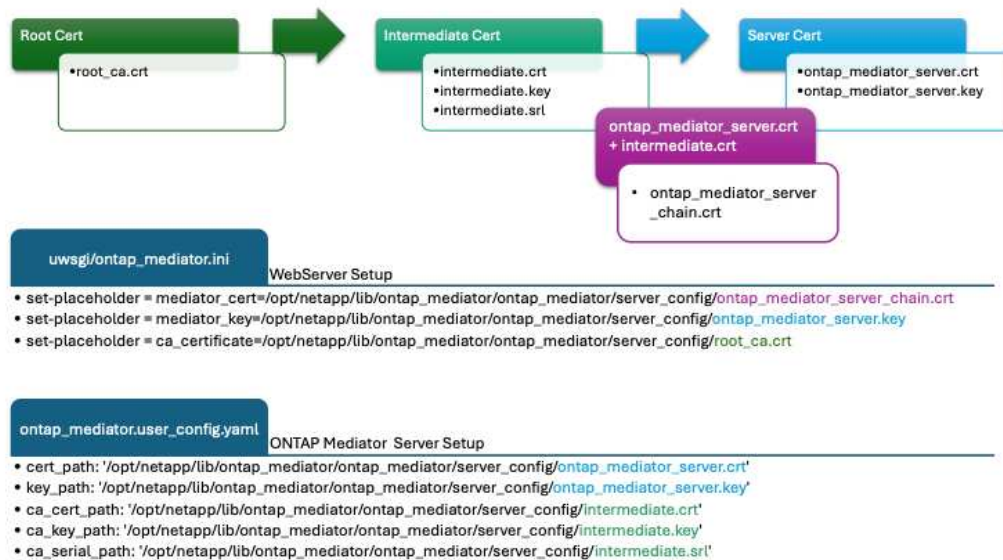
- このタスクは、ONTAPメディアーターサービスがインストールされているLinuxホストで実行します。
- このタスクは、生成された自己署名証明書を、信頼された下位の認証局（CA）から取得した証明書に置き換える必要がある場合に実行できます。これを実現するには、信頼できる公開キーインフラストラクチャ（PKI）権限にアクセスする必要があります。
- 次の図は、各ONTAPメディアーター証明書の目的を示しています。

# ONTAP Mediator Certificate Purposes



• 次の図は、WebサーバとONTAPメディエーターサーバのセットアップの設定を示しています。

## ONTAP Mediator Certificates



手順1：CA証明書を発行しているサードパーティから証明書を取得する

次の手順を使用して、PKI認証局から証明書を取得できます。

次の例は、`ca.csr`にある自己署名証明書アクター、つまり、`ca.srl`、および`ca.crt`を`/opt/netapp/lib/ontap\_mediator/ontap\_mediator/server\_config/`サードパーティの証明書アクターに置き換える

方法を示して `ca.key` ます。



次の例は、ONTAPメディアエーターサービスに必要な証明書に必要な基準を示しています。この手順とは異なる方法で、PKI認証局から証明書を取得できます。ビジネスニーズに合わせて手順を調整します。

手順

1. PKI機関が証明書を生成するために使用する秘密鍵とコンフィギュレーションファイルを `openssl_ca.cnf` 作成します `ca.key`。

- a. 秘密鍵を生成し `ca.key` ます。

- 例 \*

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

- a. 構成ファイル `openssl_ca.cnf` (にあります `/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/openssl_ca.cnf`) は、生成された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 `ca.csr` ます。

- 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-----  
MIIE6TCCAtECAQAwwgMxCzAJBgNVBAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlh  
...  
erARKhY9z0e8BHPl3g==  
-----END CERTIFICATE REQUEST-----
```

3. 証明書署名要求をPKI機関に送信し `ca.csr` で署名を求めます。

PKI認証局は要求を検証し、に署名して `.csr` 証明書を生成し `ca.crt` ます。さらに、PKI認証局から証明書に署名した証明書を取得する必要があります `root\_ca.crt ca.crt` ます。



SnapMirrorビジネス継続性 (SM-BC) クラスタの場合は、証明書と証明書をONTAPクラスタに追加する必要があります `ca.crt root_ca.crt`。を参照して ["SnapMirror Active Sync用のONTAPメディアエーターとクラスタの設定"](#)

手順2：サードパーティのCA証明書で署名してサーバ証明書を生成する

サーバ証明書は、秘密鍵とサードパーティ証明書 `ca.crt` によって署名されている必要があります

`ca.key。また、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。

サーバー証明書を生成するには、次のコマンドを実行します。

手順

1. サーバ証明書署名要求 (CSR) を生成するには、フォルダから次のコマンドを実行し  
/opt/netapp/lib/ontap\_mediator/ontap\_mediator/server\_config

```
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からサーバ証明書を生成するには、フォルダから次のコマンドを実行し  
/opt/netapp/lib/ontap\_mediator/ontap\_mediator/server\_config



ca.crt`ファイルと `ca.key` ファイルはPKI権限から取得されました。別の証明書名 (、  
など) を使用している場合は intermediate.crt intermediate.key、およびをそれぞ  
れとに置き換えます ca.crt ca.key intermediate.crt intermediate.key。

```
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`オプションを使用すると、使用している証明書名に応じてファイルまたはファイルが生成され `ca.srl intermediate.srl` ます。

手順3：ONTAPメディアエーター構成で新しいサードパーティのCA証明書とサーバ証明書を交換する

証明書の設定は、にある構成ファイルでONTAPメディアエーターサービスに提供されます

/opt/netapp/lib/ontap\_mediator/ontap\_mediator/server\_config/ontap\_mediator.config.  
.yaml。このファイルには、次の属性が含まれています。

```
cert_path:  
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediato  
r_server.crt'  
key_path:  
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediato  
r_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. すべてのファイルをサードパーティの証明書に置き換えます ca.\*。
2. および証明書から証明書チェーンを作成し ca.crt ontap\_mediator\_server.crt ます。

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

3. ファイルを更新し /opt/netapp/lib/ontap\_mediator/uwsgi/ontap\_mediator.ini ます。

、、 およびの値を更新し mediator\_cert mediator\_key `ca\_certificate` ます。

```
set-placeholder = mediator_cert =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_ser  
ver_chain.crt
```

```
set-placeholder = mediator_key =  
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_ser  
ver.key
```

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

◦ mediator\_cert `値はファイルのパス` `ontap\_mediator\_server\_chain.crt` です。

◦ mediator\_key value `は、ファイル内のキーパスです` `ontap\_mediator\_server.crt`  
ontap\_mediator\_server.key。

◦ ca\_certificate `値はファイルのパス` `root\_ca.crt` 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グループ所有者： netapp:netapp

◦ Linux権限： 600

5. ONTAPメディアエーターを再起動します。

```
systemctl restart ontap_mediator
```

手順4：必要に応じて、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。

とは別の名前のサードパーティ証明書を使用することも、サードパーティ証明書を別の場所に保存することもできます ca.\*。

#### 手順

1. ファイル内のデフォルトの変数値を上書きするようにファイルを構成し  
/opt/netapp/lib/ontap\_mediator/ontap\_mediator/server\_config/ontap\_mediator.use  
r\_config.yaml ontap\_mediator.config.yaml ます。

PKI権限から取得した秘密鍵をの場所に保存した場合、 intermediate.crt intermediate.key  
/opt/netapp/lib/ontap\_mediator/ontap\_mediator/server\_config  
ontap\_mediator.user\_config.yaml ファイルは次の例のようになります。



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

```
[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_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/intermediat
e.crt'
ca_key_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediat
e.key'
ca_serial_path:
'/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/intermediat
e.srl'
```

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



この手順の前半で、PKI機関から証明書と証明書を入手しておく必要があります  
`intermediate.crt` `ontap_mediator_server.crt`。



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

b. ファイルを更新し /opt/netapp/lib/ontap\_mediator/uwsgi/ontap\_mediator.ini ます。

、およびの値を更新し mediator\_cert mediator\_key `ca\_certificate` ます。

```
set-placeholder = mediator_cert =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_s
erver_chain.crt
```

```
set-placeholder = mediator_key =
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator_s
erver.key
```

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

- mediator\_cert`値はファイルのパス `ontap\_mediator\_server\_chain.crt` です。
- この mediator\_key 値は、ファイル内のキーパス ( ontap\_mediator\_server.crt ) です。 ontap\_mediator\_server.key
- ca\_certificate`値はファイルのパス `root\_ca.crt` です。



SnapMirrorビジネス継続性 (SM-BC) クラスタの場合は、証明書と証明書をONTAPクラスタに追加する必要があります intermediate.crt root\_ca.crt を参照して "[SnapMirror Active Sync用のONTAPメディアエーターとクラスタの設定](#)"

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グループ所有者： netapp:netapp
- Linux権限： 600

2. 構成ファイルで証明書が更新されたら、ONTAPメディアエーターを再起動します。

```
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

証明書の有効範囲を確認するには、次のコマンドを使用します。

```
[root@scs000216982 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証明書でのX509v3拡張の確認

次のコマンドを使用して、CA証明書のX509v3拡張を確認します。

で `openssl_ca.cnf` 定義されたプロパティは `v3_ca`、の `ca.crt` ように表示され `X509v3 extensions` ます。

```
[root@scs000216982 server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@scs000216982 server_config]# cat openssl_ca.cnf
...
[ v3_ca ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, cRLSign, digitalSignature, keyCertSign

[root@scs000216982 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
```

サーバ証明書およびサブジェクトの代替名のX509v3拡張を確認する

```
`v3_req`構成ファイルで定義されたプロパティは  
`openssl_server.cnf`、証明書でと表示され `X509v3 extensions`ます。
```

次の例では、ONTAPメディエーターがインストールされているLinux VMでコマンドと `hostname -I` コマンドを実行することで、セクションの `hostname -A` 変数を取得できます `@alt_names`。

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。

```
[root@scs000216982 server_config]# pwd
/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config

[root@scs000216982 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@scs000216982 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することを確認する

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。

```
[root@scs000216982 server_config]# openssl rsa -noout -modulus -in
intermediate.key | openssl md5
Enter pass phrase for intermediate.key:
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
[root@scs000216982 server_config]# openssl x509 -noout -modulus -in
intermediate.crt | openssl md5
(stdin)= 14c6b98b0c7c59012b1de89eee4a9dbc
```

両方の属性が一致する場合、`-modulus`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する

サーバ証明書が特定のCA証明書から作成されたことを確認するには、次のコマンドを使用します。

```
[root@scs000216982 server_config]# openssl verify -CAfile ca.crt
ontap_mediator_server.crt
ontap_mediator_server.crt: OK
```

Online Certificate Status Protocol (OCSP) 検証を使用している場合は、コマンドを使用し ["OpenSSL -検証"](#) します。

## ONTAPメディアエーター用のOSホストのメンテナンス

最適なパフォーマンスを得るには、ONTAPメディアエーター用のホストOSを定期的に保守する必要があります。

### ホストをリブートする

クラスタが正常な状態になったらホストをリブートします。ONTAPメディアエーターがオフラインの間は、クラスタが障害に適切に対応できなくなるリスクがあります。再起動が必要な場合は、サービスウィンドウを使用することをお勧めします。

ONTAPメディアエーターはリブート中に自動的に再開され、ONTAPクラスタで以前に設定した関係が再入力されます。

### ホストパッケージノコウシン

ライブラリやyumパッケージ（カーネルを除く）は安全に更新できますが、有効に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、サービスウィンドウを使用することをお勧めします。

パッケージをインストールした場合 `yum-utils` は、コマンドを使用し `needs-restarting` で、パッケージの変更にリブートが必要かどうかを確認します。

実行中のプロセスにはすぐには反映されないため、ONTAPメディアエーターの依存関係が更新された場合はリブートする必要があります。

## ホストOSのカーネルのマイナーアップグレード

SCSTは、使用しているカーネル用にコンパイルされている必要があります。OSを更新するには、メンテナンス時間が必要です。

### 手順

ホストOSカーネルをアップグレードするには、次の手順を実行します。

1. ONTAPメディアエーターの停止
2. SCSTパッケージをアンインストールします。（SCSTにはアップグレードメカニズムはありません）。
3. OSをアップグレードし、再起動します。
4. SCSTパッケージを再インストールします。
5. ONTAPメディアエーターサービスを再度有効にします。

## ホストがホスト名またはIPに変更

### タスクの内容

- このタスクは、ONTAPメディアエーターサービスがインストールされているLinuxホストで実行します。
- このタスクは、ONTAPメディアエーター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廃止され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証明書に置き換えたあと、このタスクを使用して証明書を再生成します。自己署名証明書がないと、この手順は失敗します。

### ステップ

現在のホストの新しい一時的な自己署名証明書を再生成するには、次の手順を実行します。

1. ONTAPメディアエーターを再起動します。

```
./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 © 2024 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権を有し、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権については、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。