



MetroClusterおよび**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
June 19, 2024

目次

MetroClusterおよびSnapMirrorのアクティブな同期用のメディエーターサービス	1
ONTAP メディエーターの概要	1
ONTAP メディ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	5
ONTAP メディエーターサービスを管理します	49
ONTAP メディエーター用のOSホストを維持します	63

MetroClusterおよびSnapMirrorのアクティブな同期用のメディエーターサービス

ONTAP メディエーターの概要

ONTAP メディエーターは、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**

ネットアップ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 メディエーターの最新情報

各リリースでは、ONTAP メディ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。

拡張機能

ONTAP メディエーターのバージョン	拡張機能
1.8	<ul style="list-style-type: none">• RHEL 8.6、8.7、8.8、8.9、8.10、9.2、9.3、9.4• Rocky Linux 8および9のサポート
1.7	<ul style="list-style-type: none">• RHEL 8.5、8.6、8.7、8.8、8.9のサポート 9.0、9.1、9.2、9.3• Rocky Linux 8および9のサポート
1.6	<ul style="list-style-type: none">• Python 3.9のアップデート。• RHEL 8.4-8.8、9.0-9.2、Rocky Linux 8および9のサポート。• 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。
1.5	<ul style="list-style-type: none">• 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。• インストーラに暗号化コード署名が追加されました。• RHEL 7.x / CentOS 7.xの廃止に関する警告が追加されました

1.4	<ul style="list-style-type: none"> • RHEL 8.4および8.5のサポート。 • SCSTバージョン3.6.0が含まれています。 • UEFIベースのファームウェアのセキュアブート（SB）のサポートが追加されました。
1.3	<ul style="list-style-type: none"> • RHEL / CentOS 8.2および8.3のサポート • SCSTバージョン3.5.0が含まれています。
1/2	<ul style="list-style-type: none"> • HTTPSメールボックスのサポート。 • 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。 • SCSTバージョン3.4.0が含まれています。
1.1	<ul style="list-style-type: none"> • RHEL / CentOS 7.6、7.7、8.0、8.1のサポート。 • Perlの依存関係が解消されます。 • SCSTバージョン3.4.0が含まれています。
1.0	<ul style="list-style-type: none"> • iSCSIメールボックスのサポート。 • ONTAP 9.7以降MCC-IP AUSOで使用します。 • RHEL / CentOS 7.6のサポート。

OSサポートマトリックス

ONTAP メディアエ ーター用 のOS	1.8	1.7	1.6	1.5	1.4	1.3	1/2	1.1	1.0
7.6	廃止され た	廃止され た	廃止され た	はい。	はい。	はい。	はい。	はい。	○ (RHEL のみ)
7.7	廃止され た	廃止され た	廃止され た	はい。	はい。	はい。	はい。	いいえ	いいえ
7、8	廃止され た	廃止され た	廃止され た	はい。	はい。	はい。	はい。	いいえ	いいえ
7.9	廃止され た	廃止され た	廃止され た	はい。	はい。	はい。	黙示的	いいえ	いいえ
RHEL 8.0	廃止され た	廃止され た	廃止され た	はい。	はい。	はい。	はい。	はい。	いいえ

RHEL 8.1	廃止された	廃止された	廃止された	はい。	はい。	はい。	はい。	いいえ	いいえ
RHEL 8.2	廃止された	廃止された	廃止された	はい。	はい。	はい。	いいえ	いいえ	いいえ
RHEL 8.3	廃止された	廃止された	廃止された	はい。	はい。	はい。	いいえ	いいえ	いいえ
RHEL 8.4	はい。	はい。	はい。	はい。	はい。	いいえ	いいえ	いいえ	いいえ
RHEL 8.5	はい。	はい。	はい。	はい。	はい。	いいえ	いいえ	いいえ	いいえ
RHEL 8.6	はい。	はい。	はい。	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 8.7	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 8.8	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 8.9	はい。	未定	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 8.10	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
RHEL 9.0	はい。	はい。	はい。	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.1	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 9.2	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 9.3	はい。	未定	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RHEL 9.4	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
CentOS 8およびSTREAM	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	はい。	はい。	はい。	N/A	N/A	N/A	N/A	N/A	N/A
Rocky Linux 9	はい。	はい。	はい。	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。
- 「いいえ」は、OSとONTAP メディエーターに互換性がないことを示します。
- 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最後にサポートされたリリースです。
- 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。

解決済みの問題

IDを変更します	説明
6995122	カーネルの不一致が検出されると、警告メッセージが表示され、ONTAPメディエーターのインストールプロセスが中断することなく続行されます。
7062227	OpenSSL検証エラーが発生したときにONTAPメディエーターのインストールプロセスが停止するように変更が実装されました。
6912810	ONTAPメディエーターの健全性チェックイベントとONTAPのサポート処理のサポートが追加されました。
7028815	アップグレードされた scst 不要なパッチファイルを削除するには、バージョン3.8.0にパッケージ化します。
7097014	ONTAPメディエーター1.8で使用される証明書を検証する新しいスクリプトが導入されました。

インストールまたはアップグレード

ONTAP メディ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を準備します

ONTAPメディ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、既存の環境のインストールまたはアップグレードに使用します。

このタスクについて

- ONTAP 9.7以降では、任意のバージョンのONTAP メディエーターを使用してMetroCluster IP構成を監視できます。
- 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。

作業を開始する前に

次の前提条件を満たしている必要があります。

ONTAP メディエーターのバージョン	サポートされている Linux バージョン
1.8	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 8.6、8.7、8.8、8.9、8.10、9.2、9.3、9.4 • Rocky Linux 8および9
1.7	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 8.5、8.6、8.7、8.8、8.9、9.0、9.1、9.2、9.3 • Rocky Linux 8および9
1.6	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 8.4、8.5、8.6、8.7、8.8、9.0、9.1、9.2 • Rocky Linux 8および9
1.5	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 7.6、7.7、7.8、7.9、8.1、8.2、8.3、8.4、8.5 • CentOS : 7.6、7.7、7.8、7.9
1.4	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 7.6、7.7、7.8、7.9、8.1、8.2、8.3、8.4、8.5 • CentOS : 7.6、7.7、7.8、7.9
1.3	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 7.6、7.7、7.8、7.9、8.1、8.2、8.3 • CentOS : 7.6、7.7、7.8、7.9
1/2	<ul style="list-style-type: none"> • Red Hat Enterprise Linux : 7.6、7.7、7.8、8.1 • CentOS : 7.6、7.7、7.8



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。

- 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

カーネル以外のライブラリパッケージは安全に更新できますが、ONTAP メディエーターアプリケーション内で有効に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。

をインストールした場合 yum-utils パッケージの場合は、を使用できます needs-restarting コマンドを実行します

カーネルコアは、ONTAP メディエーターのバージョンマトリックスでサポートされているバージョンに更新することができます。再起動は必須であるため、サービスウィンドウが必要です。

リブートの前にSCSTカーネルモジュールをアンインストールし、リポート後に再インストールする必要があります。



特定のONTAPメディア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セキュアブートが有効になっている場合、ONTAPメディアーターをインストールするには、ONTAPメディアーター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。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. mokutil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
```

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するには、次のコマンドを実行します。

```
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。



ONTAPメディアーター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。

UEFIセキュアブートを無効にする

ONTAPメディアーター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メディアーターをアップグレードします

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

作業を開始する前に

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。別の方法でインストールまたは設定されたシステムでは、追加の手順が必要になる

- 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バージョンはサポートされないため、互換性があるバージョンの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 リポジトリから依存するソフトウェアを検索するには、Red Hat Enterprise Linux の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ておく必要があります。

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

- 次のポートをメディアーター用に空けておく必要があります。
 - 31784
 - 3260
- サードパーティ製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、を確認する必要があります `sysctl net.ipv4.conf.all.arp_ignore` は「2」に設定されています。

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。

すべての RHEL または CentOS 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">• openssl• openssl-devel• kernel-devel-\$(uname -r)• GCC• メーカー• libselinux-utils• パッチ• Bzip2.• Perl - データダンパー• PERLA-ExtUtils-MakeMaker• efibootmgr• モクティル	<ul style="list-style-type: none">• python3-pip の略• elfutils-libelf-devel• policycoreutils -python-utils• RedHat LSB コアです• ピートン39• Python39 -デベル	<ul style="list-style-type: none">• python3-pip の略• elfutils-libelf-devel• policycoreutils -python-utils• ピートン3• Python3 -デベル

メディアーターのインストールパッケージは自己解凍形式の圧縮 tar ファイルで、次のものが含まれます。

- 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。

このタスクについて

leapp-upgradeツールを使用してONTAP メディエーター用のホスト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、次の手順を実行します。ONTAP メディエーターは、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を検出しようとするため、アンインストールする必要があります。

rpmファイルはONTAP メディエ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、登録されたり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、ONTAP メディエーター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 メディエーターを再インストールするため、接続されているONTAP クラスタは新規インストール後に再接続する必要があります。



次の手順を順番に実行する必要があります。ONTAP メディエーターを再インストールしたらすぐに、ontap_mediatorサービスを停止してログファイルを交換し、サービスを再起動する必要があります。これにより、ログが失われなくなります。

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 メディエーター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 メディエーターを再インストールした直後に残りの手順を実行します。

```
[rootmediator-host ~]# ontap-mediator-1.6.0/ontap-mediator-1.6.0

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 メディエーターにすべてのONTAP クラスタを再接続します

```

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 Active Sync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、再インストールする必要はありません。生成されたデフォルトの自己署名証明書を使用していた場合、またはカスタム証明書を/opt/netappディレクトリに配置していた場合は、その証明書をバックアップおよび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 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

....

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します

インストールプロセス中にONTAP メディ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	リポジトリへのアクセスを指定する必要があります ...
RHEL 7.x	<ul style="list-style-type: none">• rhel-7-server-optional-rpms のいずれかです
RHEL 8.x の場合	<ul style="list-style-type: none">• RHEL-8-For x86_64-baseos-RPMs• RHEL-8-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-RPM
CentOS 7.x	<ul style="list-style-type: none">• C7.6.1810 - ベースリポジトリ
Rocky Linux 8	<ul style="list-style-type: none">• AppStreamの略• ベースオス
Rocky Linux 9	<ul style="list-style-type: none">• AppStreamの略• ベースオス

2. インストールプロセス中にONTAP メディ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メディエーターが、「extras」リポジトリと「optional」リポジトリにあるPythonモジュールとの依存関係を持っている場合は、`rhel-x-for-x86_64-extras-rpms` および `rhel-x-for-x86_64-optional-rpms` ファイル。

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登録します。

```
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. を実行します 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ている必要があります。

```
[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* 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. 必要なリポジトリに登録します。

```
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. を実行します 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登録します。

```
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. を実行します `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* 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のバージョンにはベースリポジトリを使用してください。

手順

1. C7.6.1810 ベースリポジトリを追加します。C7.6.1810 - Baseヴォー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. 次の行を /etc/yum.repos_d/Center-Vault.repo に追加します。

```
[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. を実行します 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 ベース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用の手順

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. を実行します clean 操作：

```
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 ~]#
```

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

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-1.8.0.tgz
```

```
ontap-mediator-1.8.0.tgz
```



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

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、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ディレクトリから Linux メディエーター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-1.8.0.tgz
```

```
ontap-mediator-1.8.0/  
ontap-mediator-1.8.0/csc-prod-ONTAP-Mediator.pem  
ontap-mediator-1.8.0/csc-prod-chain-ONTAP-Mediator.pem  
ontap-mediator-1.8.0/tsa-prod-ONTAP-Mediator.pem  
ontap-mediator-1.8.0/tsa-prod-chain-ONTAP-Mediator.pem  
ontap-mediator-1.8.0/ONTAP-Mediator-production.pub  
ontap-mediator-1.8.0/ontap-mediator-1.8.0  
ontap-mediator-1.8.0/ontap-mediator-1.8.0.sig.tsr  
ontap-mediator-1.8.0/ontap-mediator-1.8.0.tsr  
ontap-mediator-1.8.0/ontap-mediator-1.8.0.sig
```

ONTAP メディエーターコードの署名を確認します

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。

作業を開始する前に

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。

- 基本的な検証のためのOpenSSLバージョン1.0.2~3.0
- Time Stamping Authority (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	署名の検証に使用する公開鍵
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-1.8.0	バージョン1.8.0の製品インストール実行可能ファイル
ontap-mediator-1.8.0.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-1.8.0.sig.tsr	OCSCPがインストーラの署名に使用する失効要求
ontap-mediator-1.8.0.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. 失効チェックをオンにします csc-prod-ONTAP-Mediator.pem Online Certificate Status Protocol (OCSP) を使用します。
 - 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 ocsp -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、ローカルホストに対する有効期限を確認します。

```
openssl verify
```



。openssl パスのバージョンは有効である必要があります cert.pem（自己署名ではありません）。

```
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. を確認します ontap-mediator-1.8.0.sig.tsr および ontap-mediator-1.8.0.tsr 関連する証明書を使用しているファイル：

```
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。TSAサーバへのアクセスは不要です。

```
openssl ts -verify -data ontap-mediator-1.8.0.sig -in ontap-mediator-1.8.0.sig.tsr -CAfile tsa-prod-chain-ONTAP-Mediator.pem -untrusted tsa-prod-ONTAP-Mediator.pem  
openssl ts -verify -data ontap-mediator-1.8.0 -in ontap-mediator-1.8.0.tsr -CAfile tsa-prod-chain-ONTAP-Mediator.pem -untrusted tsa-prod-ONTAP-Mediator.pem
```

4. キーに対して署名を確認します。

```
openssl -dgst -verify
```

```
openssl dgst -sha256 -verify ONTAP-Mediator-production.pub -signature  
ontap-mediator-1.8.0.sig ontap-mediator-1.8.0
```

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-1.8.0]# pwd
/root/ontap-mediator-1.8.0
[root@scspa2695423001 ontap-mediator-1.8.0]# 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-1.8.0
-rw-r--r-- 1 root root      384 Feb 20 15:17 ontap-mediator-1.8.0.sig
-rw-r--r-- 1 root root      5437 Feb 20 15:17 ontap-mediator-
1.8.0.sig.tsr
-rw-r--r-- 1 root root      5436 Feb 20 15:17 ontap-mediator-1.8.0.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-1.8.0]#
[root@scspa2695423001 ontap-mediator-1.8.0]#
/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-1.8.0.sig -in ontap-mediator-
1.8.0.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-1.8.0 -in ontap-mediator-
1.8.0.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-1.8.0.sig ontap-mediator-1.8.0
Verified OK
[root@scspa2695423001 ontap-mediator-1.8.0]#

```

ONTAP メディエーターのインストールパッケージをインストールします

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-1.8.0/ontap-mediator-1.8.0 -y
```

```
[root@scs000099753 ~]# ./ontap-mediator-1.8.0/ontap-mediator-1.8.0 -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。以前のバージョンのメディエーターがホストにインストールされている場合は、アップグレードを確認するプロンプトが表示されます。

2. ONTAP メディエーター1.4以降では、セキュアブートメカニズムはUEFIシステムで有効になっています。セキュアブートが有効になっている場合は、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

- 必要なキーを探します。

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



インストール後は、READMEファイルとキーの場所もシステム出力に含まれています。

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@sdot-r730-0003a-d6 ~]# ontap-mediator-1.8.0/ontap-mediator-1.8.0
-y

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
  WARNING: The OCSP check failed while attempting to test the Code-
  Signature-Check certificate
  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
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
success
success
success

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

+ Installing required packages.
```

Updating Subscription Management repositories.

Unable to read consumer identity

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

Last metadata expiration check: 272 days, 23:59:05 ago on Thu 07 Sep 2023 11:37:05 AM EDT.

Package openssl-1:1.1.1k-9.el8_7.x86_64 is already installed.

Package libselinux-utils-2.9-8.el8.x86_64 is already installed.

Package perl-Data-Dumper-2.167-399.el8.x86_64 is already installed.

Package bzip2-1.0.6-26.el8.x86_64 is already installed.

Package efibootmgr-16-1.el8.x86_64 is already installed.

Package mokutil-1:0.3.0-12.el8.x86_64 is already installed.

Package python3-pip-9.0.3-23.el8.noarch is already installed.

Package policycoreutils-python-utils-2.9-24.el8.noarch is already installed.

Dependencies resolved.

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

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

Installing:

elfutils-libelf-devel		
x86_64	0.189-3.el8	
Local-BaseOS		62 k
gcc		
x86_64	8.5.0-20.el8	
Local-AppStream		23 M
kernel-devel		
x86_64	4.18.0-513.el8	
Local-BaseOS		24 M
make		
x86_64	1:4.2.1-11.el8	
Local-BaseOS		498 k
openssl-devel		
x86_64	1:1.1.1k-9.el8_7	
Local-BaseOS		2.3 M
patch		

```

x86_64 2.7.6-11.e18
Local-BaseOS 138 k
  perl-ExtUtils-MakeMaker
noarch 1:7.34-1.e18
Local-AppStream 301 k
  python39
x86_64 3.9.17-
2.module+e18.9.0+19644+d68f775d
Local-AppStream 34 k
  python39-devel
x86_64 3.9.17-
2.module+e18.9.0+19644+d68f775d
Local-AppStream 229 k
  redhat-lsb-core
x86_64 4.1-47.e18
Local-AppStream 45 k
Installing dependencies:
  annobin
x86_64 11.13-2.e18
Local-AppStream 972 k
  cpp
x86_64 8.5.0-20.e18
Local-AppStream 10 M
  dwz
x86_64 0.12-10.e18
Local-AppStream 109 k
  efi-srpm-macros
noarch 3-3.e18
Local-AppStream 22 k
  gcc-plugin-annobin
x86_64 8.5.0-20.e18
Local-AppStream 36 k
  glibc-devel
noarch 1.4.2-7.e18
Local-AppStream 9.4 k
  glibc-devel
x86_64 2.28-236.e18
Local-BaseOS 84 k
  glibc-headers
x86_64 2.28-236.e18
Local-BaseOS 489 k
  go-srpm-macros
noarch 2-17.e18
Local-AppStream 13 k
  isl
x86_64 0.16.1-6.e18

```

Local-AppStream	841 k
kernel-headers	
x86_64	4.18.0-513.el8
Local-BaseOS	11 M
keyutils-libs-devel	
x86_64	1.5.10-9.el8
Local-BaseOS	48 k
krb5-devel	
x86_64	1.18.2-25.el8_8
Local-BaseOS	562 k
libcom_err-devel	
x86_64	1.45.6-5.el8
Local-BaseOS	39 k
libkadm5	
x86_64	1.18.2-25.el8_8
Local-BaseOS	188 k
libselinux-devel	
x86_64	2.9-8.el8
Local-BaseOS	200 k
libsepol-devel	
x86_64	2.9-3.el8
Local-BaseOS	87 k
libverto-devel	
x86_64	0.3.2-2.el8
Local-BaseOS	18 k
libxcrypt-devel	
x86_64	4.1.1-6.el8
Local-BaseOS	25 k
libzstd-devel	
x86_64	1.4.4-1.el8
Local-BaseOS	44 k
m4	
x86_64	1.4.18-7.el8
Local-BaseOS	223 k
mailx	
x86_64	12.5-29.el8
Local-BaseOS	257 k
ncurses-compat-libs	
x86_64	6.1-10.20180224.el8
Local-BaseOS	329 k
ocaml-srpm-macros	
noarch	5-4.el8
Local-AppStream	9.5 k
openblas-srpm-macros	
noarch	2-2.el8
Local-AppStream	8.0 k

pcr2-devel	
x86_64	10.32-3.e18_6
Local-BaseOS	605 k
pcr2-utf16	
x86_64	10.32-3.e18_6
Local-BaseOS	229 k
pcr2-utf32	
x86_64	10.32-3.e18_6
Local-BaseOS	220 k
perl-CPAN-Meta-YAML	
noarch	0.018-397.e18
Local-AppStream	34 k
perl-ExtUtils-Command	
noarch	1:7.34-1.e18
Local-AppStream	19 k
perl-ExtUtils-Install	
noarch	2.14-4.e18
Local-AppStream	46 k
perl-ExtUtils-Manifest	
noarch	1.70-395.e18
Local-AppStream	37 k
perl-ExtUtils-ParseXS	
noarch	1:3.35-2.e18
Local-AppStream	83 k
perl-JSON-PP	
noarch	1:2.97.001-3.e18
Local-AppStream	68 k
perl-Test-Harness	
noarch	1:3.42-1.e18
Local-AppStream	279 k
perl-devel	
x86_64	4:5.26.3-422.e18
Local-AppStream	600 k
perl-srpm-macros	
noarch	1-25.e18
Local-AppStream	11 k
perl-version	
x86_64	6:0.99.24-1.e18
Local-AppStream	67 k
postfix	
x86_64	2:3.5.8-7.e18
Local-BaseOS	1.5 M
python-rpm-macros	
noarch	3-45.e18
Local-AppStream	16 k
python-srpm-macros	

```

noarch                               3-45.e18
Local-AppStream                       16 k
  python3-pyparsing
noarch                               2.1.10-7.e18
Local-BaseOS                           142 k
  python3-rpm-macros
noarch                               3-45.e18
Local-AppStream                       15 k
  python39-libs
x86_64                                3.9.17-
2.module+el8.9.0+19644+d68f775d
Local-AppStream                       8.2 M
  python39-pip-wheel
noarch                               20.2.4-
8.module+el8.9.0+19644+d68f775d
Local-AppStream                       1.1 M
  python39-setuptools-wheel
noarch                               50.3.2-
4.module+el8.9.0+19644+d68f775d
Local-AppStream                       497 k
  qt5-srpm-macros
noarch                               5.15.3-1.e18
Local-AppStream                       11 k
  redhat-lsb-submod-security
x86_64                                4.1-47.e18
Local-AppStream                       22 k
  redhat-rpm-config
noarch                               131-1.e18
Local-AppStream                       91 k
  rust-srpm-macros
noarch                               5-2.e18
Local-AppStream                       9.3 k
  spax
x86_64                                1.5.3-13.e18
Local-BaseOS                           217 k
  systemtap-sdt-devel
x86_64                                4.9-3.e18
Local-AppStream                       88 k
  zlib-devel
x86_64                                1.2.11-25.e18
Local-BaseOS                           59 k
Installing weak dependencies:
  bison
x86_64                                3.0.4-10.e18
Local-AppStream                       688 k
  flex

```

```

x86_64 2.6.1-9.el8
Local-AppStream 320 k
  perl-CPAN-Meta
noarch 2.150010-396.el8
Local-AppStream 191 k
  perl-CPAN-Meta-Requirements
noarch 2.140-396.el8
Local-AppStream 37 k
  perl-Encode-Locale
noarch 1.05-
10.module+el8.3.0+6498+9eecfe51
Local-AppStream 22 k
  perl-Time-HiRes
x86_64 4:1.9758-2.el8
Local-AppStream 61 k
  python39-pip
noarch 20.2.4-
8.module+el8.9.0+19644+d68f775d
Local-AppStream 1.9 M
  python39-setuptools
noarch 50.3.2-
4.module+el8.9.0+19644+d68f775d
Local-AppStream 871 k
Enabling module streams:
  python39
3.9

Transaction Summary
=====
=====
=====
=====
Install 71 Packages

Total size: 95 M
Installed size: 224 M
Is this ok [y/N]: y
Downloading Packages:
Red Hat Enterprise Linux 9 - BaseOS
45 kB/s | 5.0 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
Is this ok [y/N]: y
Key imported successfully

```

```
Importing GPG key 0xD4082792:
  Userid      : "Red Hat, Inc. (auxiliary key) <security@redhat.com>"
  Fingerprint: 6A6A A7C9 7C88 90AE C6AE BFE2 F76F 66C3 D408 2792
  From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Is this ok [y/N]: y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
1/1
  Installing     : python-srpm-macros-3-45.el8.noarch
1/71
  Installing     : perl-version-6:0.99.24-1.el8.x86_64
2/71
  Installing     : m4-1.4.18-7.el8.x86_64
3/71
  Running scriptlet: m4-1.4.18-7.el8.x86_64
3/71
  Installing     : perl-CPAN-Meta-Requirements-2.140-396.el8.noarch
4/71
  Installing     : python-rpm-macros-3-45.el8.noarch
5/71
  Installing     : python3-rpm-macros-3-45.el8.noarch
6/71
  Installing     : perl-Time-HiRes-4:1.9758-2.el8.x86_64
7/71
  Installing     : perl-JSON-PP-1:2.97.001-3.el8.noarch
8/71
  Installing     : perl-ExtUtils-ParseXS-1:3.35-2.el8.noarch
9/71
  Installing     : zlib-devel-1.2.11-25.el8.x86_64
10/71
  Installing     : make-1:4.2.1-11.el8.x86_64
11/71
  Running scriptlet: make-1:4.2.1-11.el8.x86_64
11/71
  Installing     : perl-Test-Harness-1:3.42-1.el8.noarch
12/71
  Installing     : bison-3.0.4-10.el8.x86_64
13/71
  Running scriptlet: bison-3.0.4-10.el8.x86_64
13/71
  Installing     : flex-2.6.1-9.el8.x86_64
```

```
14/71
  Running scriptlet: flex-2.6.1-9.el8.x86_64
14/71
  Installing      : rust-srpm-macros-5-2.el8.noarch
15/71
  Installing      : redhat-lsb-submod-security-4.1-47.el8.x86_64
16/71
  Installing      : qt5-srpm-macros-5.15.3-1.el8.noarch
17/71
  Installing      : python39-setuptools-wheel-50.3.2-
4.module+el8.9.0+19644+d68f775d.noarch
18/71
  Installing      : python39-pip-wheel-20.2.4-
8.module+el8.9.0+19644+d68f775d.noarch
19/71
  Installing      : python39-libs-3.9.17-
2.module+el8.9.0+19644+d68f775d.x86_64
20/71
  Installing      : python39-3.9.17-
2.module+el8.9.0+19644+d68f775d.x86_64
21/71
  Running scriptlet: python39-3.9.17-
2.module+el8.9.0+19644+d68f775d.x86_64
21/71
  Installing      : python39-setuptools-50.3.2-
4.module+el8.9.0+19644+d68f775d.noarch
22/71
  Running scriptlet: python39-setuptools-50.3.2-
4.module+el8.9.0+19644+d68f775d.noarch
22/71
  Installing      : python39-pip-20.2.4-
8.module+el8.9.0+19644+d68f775d.noarch
23/71
  Running scriptlet: python39-pip-20.2.4-
8.module+el8.9.0+19644+d68f775d.noarch
23/71
  Installing      : perl-srpm-macros-1-25.el8.noarch
24/71
  Installing      : perl-ExtUtils-Manifest-1.70-395.el8.noarch
25/71
  Installing      : perl-ExtUtils-Command-1:7.34-1.el8.noarch
26/71
  Installing      : perl-Encode-Locale-1.05-
10.module+el8.3.0+6498+9eecfe51.noarch
27/71
  Installing      : perl-CPAN-Meta-YAML-0.018-397.el8.noarch
```

```
28/71
  Installing      : perl-CPAN-Meta-2.150010-396.el8.noarch
29/71
  Installing      : openblas-srpm-macros-2-2.el8.noarch
30/71
  Installing      : ocaml-srpm-macros-5-4.el8.noarch
31/71
  Installing      : isl-0.16.1-6.el8.x86_64
32/71
  Running scriptlet: isl-0.16.1-6.el8.x86_64
32/71
  Installing      : go-srpm-macros-2-17.el8.noarch
33/71
  Installing      : ghc-srpm-macros-1.4.2-7.el8.noarch
34/71
  Installing      : efi-srpm-macros-3-3.el8.noarch
35/71
  Installing      : dwz-0.12-10.el8.x86_64
36/71
  Installing      : cpp-8.5.0-20.el8.x86_64
37/71
  Running scriptlet: cpp-8.5.0-20.el8.x86_64
37/71
  Installing      : spax-1.5.3-13.el8.x86_64
38/71
  Running scriptlet: spax-1.5.3-13.el8.x86_64
38/71
  Installing      : python3-pyparsing-2.1.10-7.el8.noarch
39/71
  Installing      : systemtap-sdt-devel-4.9-3.el8.x86_64
40/71
  Running scriptlet: postfix-2:3.5.8-7.el8.x86_64
41/71
  Installing      : postfix-2:3.5.8-7.el8.x86_64
41/71
  Running scriptlet: postfix-2:3.5.8-7.el8.x86_64
41/71
  Installing      : pcre2-utf32-10.32-3.el8_6.x86_64
42/71
  Installing      : pcre2-utf16-10.32-3.el8_6.x86_64
43/71
  Installing      : pcre2-devel-10.32-3.el8_6.x86_64
44/71
  Installing      : patch-2.7.6-11.el8.x86_64
45/71
  Installing      : ncurses-compat-libs-6.1-10.20180224.el8.x86_64
```

```
46/71
  Installing      : mailx-12.5-29.el8.x86_64
47/71
  Installing      : libzstd-devel-1.4.4-1.el8.x86_64
48/71
  Installing      : elfutils-libelf-devel-0.189-3.el8.x86_64
49/71
  Installing      : libverto-devel-0.3.2-2.el8.x86_64
50/71
  Installing      : libsepol-devel-2.9-3.el8.x86_64
51/71
  Installing      : libselinux-devel-2.9-8.el8.x86_64
52/71
  Installing      : libkadm5-1.18.2-25.el8_8.x86_64
53/71
  Installing      : libcom_err-devel-1.45.6-5.el8.x86_64
54/71
  Installing      : keyutils-libs-devel-1.5.10-9.el8.x86_64
55/71
  Installing      : krb5-devel-1.18.2-25.el8_8.x86_64
56/71
  Installing      : openssl-devel-1:1.1.1k-9.el8_7.x86_64
57/71
  Installing      : kernel-headers-4.18.0-513.el8.x86_64
58/71
  Running scriptlet: glibc-headers-2.28-236.el8.x86_64
59/71
  Installing      : glibc-headers-2.28-236.el8.x86_64
59/71
  Installing      : libxcrypt-devel-4.1.1-6.el8.x86_64
60/71
  Installing      : glibc-devel-2.28-236.el8.x86_64
61/71
  Running scriptlet: glibc-devel-2.28-236.el8.x86_64
61/71
  Installing      : gcc-8.5.0-20.el8.x86_64
62/71
  Running scriptlet: gcc-8.5.0-20.el8.x86_64
62/71
  Installing      : annobin-11.13-2.el8.x86_64
63/71
  Installing      : gcc-plugin-annobin-8.5.0-20.el8.x86_64
64/71
  Installing      : redhat-rpm-config-131-1.el8.noarch
65/71
  Running scriptlet: redhat-rpm-config-131-1.el8.noarch
```

```
65/71
  Installing      : perl-ExtUtils-Install-2.14-4.el8.noarch
66/71
  Installing      : perl-devel-4:5.26.3-422.el8.x86_64
67/71
  Installing      : perl-ExtUtils-MakeMaker-1:7.34-1.el8.noarch
68/71
  Installing      : kernel-devel-4.18.0-513.el8.x86_64
69/71
  Running scriptlet: kernel-devel-4.18.0-513.el8.x86_64
69/71
  Installing      : redhat-lsb-core-4.1-47.el8.x86_64
70/71
  Installing      : python39-devel-3.9.17-
2.module+el8.9.0+19644+d68f775d.x86_64
71/71
  Running scriptlet: python39-devel-3.9.17-
2.module+el8.9.0+19644+d68f775d.x86_64
71/71
  Verifying      : elfutils-libelf-devel-0.189-3.el8.x86_64
1/71
  Verifying      : glibc-devel-2.28-236.el8.x86_64
2/71
  Verifying      : glibc-headers-2.28-236.el8.x86_64
3/71
  Verifying      : kernel-devel-4.18.0-513.el8.x86_64
4/71
  Verifying      : kernel-headers-4.18.0-513.el8.x86_64
5/71
  Verifying      : keyutils-libs-devel-1.5.10-9.el8.x86_64
6/71
  Verifying      : krb5-devel-1.18.2-25.el8_8.x86_64
7/71
  Verifying      : libcom_err-devel-1.45.6-5.el8.x86_64
8/71
  Verifying      : libkadm5-1.18.2-25.el8_8.x86_64
9/71
  Verifying      : libselinux-devel-2.9-8.el8.x86_64
10/71
  Verifying      : libsepol-devel-2.9-3.el8.x86_64
11/71
  Verifying      : libverto-devel-0.3.2-2.el8.x86_64
12/71
  Verifying      : libxcrypt-devel-4.1.1-6.el8.x86_64
13/71
  Verifying      : libzstd-devel-1.4.4-1.el8.x86_64
```

```
14/71
  Verifying      : m4-1.4.18-7.el8.x86_64
15/71
  Verifying      : mailx-12.5-29.el8.x86_64
16/71
  Verifying      : make-1:4.2.1-11.el8.x86_64
17/71
  Verifying      : ncurses-compat-libs-6.1-10.20180224.el8.x86_64
18/71
  Verifying      : openssl-devel-1:1.1.1k-9.el8_7.x86_64
19/71
  Verifying      : patch-2.7.6-11.el8.x86_64
20/71
  Verifying      : pcre2-devel-10.32-3.el8_6.x86_64
21/71
  Verifying      : pcre2-utf16-10.32-3.el8_6.x86_64
22/71
  Verifying      : pcre2-utf32-10.32-3.el8_6.x86_64
23/71
  Verifying      : postfix-2:3.5.8-7.el8.x86_64
24/71
  Verifying      : python3-pyparsing-2.1.10-7.el8.noarch
25/71
  Verifying      : spax-1.5.3-13.el8.x86_64
26/71
  Verifying      : zlib-devel-1.2.11-25.el8.x86_64
27/71
  Verifying      : annobin-11.13-2.el8.x86_64
28/71
  Verifying      : bison-3.0.4-10.el8.x86_64
29/71
  Verifying      : cpp-8.5.0-20.el8.x86_64
30/71
  Verifying      : dwz-0.12-10.el8.x86_64
31/71
  Verifying      : efi-srpm-macros-3-3.el8.noarch
32/71
  Verifying      : flex-2.6.1-9.el8.x86_64
33/71
  Verifying      : gcc-8.5.0-20.el8.x86_64
34/71
  Verifying      : gcc-plugin-annobin-8.5.0-20.el8.x86_64
35/71
  Verifying      : ghc-srpm-macros-1.4.2-7.el8.noarch
36/71
  Verifying      : go-srpm-macros-2-17.el8.noarch
```

```
37/71
  Verifying      : isl-0.16.1-6.el8.x86_64
38/71
  Verifying      : ocaml-srpm-macros-5-4.el8.noarch
39/71
  Verifying      : openblas-srpm-macros-2-2.el8.noarch
40/71
  Verifying      : perl-CPAN-Meta-2.150010-396.el8.noarch
41/71
  Verifying      : perl-CPAN-Meta-Requirements-2.140-396.el8.noarch
42/71
  Verifying      : perl-CPAN-Meta-YAML-0.018-397.el8.noarch
43/71
  Verifying      : perl-Encode-Locale-1.05-
10.module+el8.3.0+6498+9eecfe51.noarch
44/71
  Verifying      : perl-ExtUtils-Command-1:7.34-1.el8.noarch
45/71
  Verifying      : perl-ExtUtils-Install-2.14-4.el8.noarch
46/71
  Verifying      : perl-ExtUtils-MakeMaker-1:7.34-1.el8.noarch
47/71
  Verifying      : perl-ExtUtils-Manifest-1.70-395.el8.noarch
48/71
  Verifying      : perl-ExtUtils-ParseXS-1:3.35-2.el8.noarch
49/71
  Verifying      : perl-JSON-PP-1:2.97.001-3.el8.noarch
50/71
  Verifying      : perl-Test-Harness-1:3.42-1.el8.noarch
51/71
  Verifying      : perl-Time-HiRes-4:1.9758-2.el8.x86_64
52/71
  Verifying      : perl-devel-4:5.26.3-422.el8.x86_64
53/71
  Verifying      : perl-srpm-macros-1-25.el8.noarch
54/71
  Verifying      : perl-version-6:0.99.24-1.el8.x86_64
55/71
  Verifying      : python-rpm-macros-3-45.el8.noarch
56/71
  Verifying      : python-srpm-macros-3-45.el8.noarch
57/71
  Verifying      : python3-rpm-macros-3-45.el8.noarch
58/71
  Verifying      : python39-3.9.17-
2.module+el8.9.0+19644+d68f775d.x86_64
```

```
59/71
  Verifying      : python39-devel-3.9.17-
2.module+el8.9.0+19644+d68f775d.x86_64
60/71
  Verifying      : python39-libs-3.9.17-
2.module+el8.9.0+19644+d68f775d.x86_64
61/71
  Verifying      : python39-pip-20.2.4-
8.module+el8.9.0+19644+d68f775d.noarch
62/71
  Verifying      : python39-pip-wheel-20.2.4-
8.module+el8.9.0+19644+d68f775d.noarch
63/71
  Verifying      : python39-setuptools-50.3.2-
4.module+el8.9.0+19644+d68f775d.noarch
64/71
  Verifying      : python39-setuptools-wheel-50.3.2-
4.module+el8.9.0+19644+d68f775d.noarch
65/71
  Verifying      : qt5-srpm-macros-5.15.3-1.el8.noarch
66/71
  Verifying      : redhat-lsb-core-4.1-47.el8.x86_64
67/71
  Verifying      : redhat-lsb-submod-security-4.1-47.el8.x86_64
68/71
  Verifying      : redhat-rpm-config-131-1.el8.noarch
69/71
  Verifying      : rust-srpm-macros-5-2.el8.noarch
70/71
  Verifying      : systemtap-sdt-devel-4.9-3.el8.x86_64
71/71
Installed products updated.
```

Installed:

```
annobin-11.13-2.el8.x86_64
bison-3.0.4-10.el8.x86_64
cpp-8.5.0-20.el8.x86_64
dwz-0.12-10.el8.x86_64
efi-srpm-macros-3-3.el8.noarch
elfutils-libelf-devel-0.189-3.el8.x86_64
flex-2.6.1-9.el8.x86_64
gcc-8.5.0-20.el8.x86_64
gcc-plugin-annobin-8.5.0-20.el8.x86_64
ghc-srpm-macros-1.4.2-7.el8.noarch
glibc-devel-2.28-236.el8.x86_64
glibc-headers-2.28-236.el8.x86_64
```

```
go-srpm-macros-2-17.el8.noarch
isl-0.16.1-6.el8.x86_64
kernel-devel-4.18.0-513.el8.x86_64
  kernel-headers-4.18.0-513.el8.x86_64
keyutils-libs-devel-1.5.10-9.el8.x86_64
krb5-devel-1.18.2-25.el8_8.x86_64
  libcom_err-devel-1.45.6-5.el8.x86_64
libkadm5-1.18.2-25.el8_8.x86_64
libselinux-devel-2.9-8.el8.x86_64
  libsepol-devel-2.9-3.el8.x86_64
libverto-devel-0.3.2-2.el8.x86_64
libxcrypt-devel-4.1.1-6.el8.x86_64
  libzstd-devel-1.4.4-1.el8.x86_64
m4-1.4.18-7.el8.x86_64
mailx-12.5-29.el8.x86_64
  make-1:4.2.1-11.el8.x86_64
ncurses-compat-libs-6.1-10.20180224.el8.x86_64
ocaml-srpm-macros-5-4.el8.noarch
  openblas-srpm-macros-2-2.el8.noarch
openssl-devel-1:1.1.1k-9.el8_7.x86_64
patch-2.7.6-11.el8.x86_64
  pcre2-devel-10.32-3.el8_6.x86_64
pcre2-utf16-10.32-3.el8_6.x86_64
pcre2-utf32-10.32-3.el8_6.x86_64
  perl-CPAN-Meta-2.150010-396.el8.noarch
perl-CPAN-Meta-Requirements-2.140-396.el8.noarch
perl-CPAN-Meta-YAML-0.018-397.el8.noarch
  perl-Encode-Locale-1.05-10.module+el8.3.0+6498+9eecfe51.noarch
perl-ExtUtils-Command-1:7.34-1.el8.noarch
perl-ExtUtils-Install-2.14-4.el8.noarch
  perl-ExtUtils-MakeMaker-1:7.34-1.el8.noarch
perl-ExtUtils-Manifest-1.70-395.el8.noarch
perl-ExtUtils-ParseXS-1:3.35-2.el8.noarch
  perl-JSON-PP-1:2.97.001-3.el8.noarch
perl-Test-Harness-1:3.42-1.el8.noarch
perl-Time-HiRes-4:1.9758-2.el8.x86_64
  perl-devel-4:5.26.3-422.el8.x86_64
perl-srpm-macros-1-25.el8.noarch
perl-version-6:0.99.24-1.el8.x86_64
  postfix-2:3.5.8-7.el8.x86_64
python-rpm-macros-3-45.el8.noarch
python-srpm-macros-3-45.el8.noarch
  python3-pyparsing-2.1.10-7.el8.noarch
python3-rpm-macros-3-45.el8.noarch
python39-3.9.17-2.module+el8.9.0+19644+d68f775d.x86_64
  python39-devel-3.9.17-2.module+el8.9.0+19644+d68f775d.x86_64
```

```
python39-libs-3.9.17-2.module+el8.9.0+19644+d68f775d.x86_64
python39-pip-20.2.4-8.module+el8.9.0+19644+d68f775d.noarch
  python39-pip-wheel-20.2.4-8.module+el8.9.0+19644+d68f775d.noarch
python39-setuptools-50.3.2-4.module+el8.9.0+19644+d68f775d.noarch
python39-setuptools-wheel-50.3.2-4.module+el8.9.0+19644+d68f775d.noarch
  qt5-srpm-macros-5.15.3-1.el8.noarch
redhat-lsb-core-4.1-47.el8.x86_64
redhat-lsb-submod-security-4.1-47.el8.x86_64
  redhat-rpm-config-131-1.el8.noarch
rust-srpm-macros-5-2.el8.noarch
spax-1.5.3-13.el8.x86_64
  systemtap-sdt-devel-4.9-3.el8.x86_64
zlib-devel-1.2.11-25.el8.x86_64
```

Complete!

OS package installations finished

+ Installing ONTAP Mediator. (Log: /root/ontap_mediator.MRjxkr/ontap-mediator-1.8.0/ontap-mediator-1.8.0/install_20240606113556.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_20240606113556.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.module-signing

 to sign the SCST kernel module. Note that reboot will be needed.

 SCST will 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`

インストールを確認します。

ONTAP メディエーターをインストールしたら、ONTAP メディ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
```

インストール後の設定

ONTAP メディエーターサービスをインストールして実行したら、メディエーター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。

- MetroCluster IP 構成で ONTAP メディエーターサービスを使用する場合は、[を参照してください "MetroCluster IP 構成での ONTAP メディエ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 メディエーターサービスをインストールし、ONTAP クラスタの設定を確認します"](#)。

ONTAP メディエーターのセキュリティポリシーを設定します

ONTAP メディエーターサーバでは、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、`low_space_threshold_mib` : 10読み取り専用ファイルで提供されます。

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

に配置されているすべての値 `ontap_mediator.user_config.yaml` デフォルト値は上書きされ、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。

を変更した後 `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。

次の属性を設定できます。



その他のデフォルト値 `ontap_mediator.config.yaml` 変更しないでください。

- デフォルトの自己署名証明書の代わりにサードパーティの**SSL**証明書をインストールするための設定

```
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_valid_days: '1095' # Used to set the expiration
on client certs to 3 years
x509_passin_pwd: 'pass:ontap' # passphrase for the signed
client cert
```

- 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する設定

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

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
```

- * ONTAP メディエーター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するフィールド*

```
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)
```

- で必要な空き容量を制御する設定 `/opt/netapp/lib/ontap_mediator` ディスク。

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問題します。

```
low_space_threshold_mib: 10
```

- * RESERVE_LOG_SPACEを制御する設定。*

ONTAPメディエーターサーバのデフォルトのインストールでは、ログ用に独立したディスクスペースが作成されます。Mediatorのロギン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。

- a. 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
```

- b. Mediatorを再起動します。

```
i. cat /opt/netapp/lib/ontap_mediator/tools/mediator_env | grep
```

```
"RESERVE_LOG_SPACE"
```

```
RESERVE_LOG_SPACE=0
```

ii. `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 メディエ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メディエ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。

このタスクについて

このタスクは、ONTAPメディ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_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)** : コマンドを実行します。 `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 ~]#
```

パスワードを変更します

パスワードは次の手順を使用して変更できます。

このタスクについて

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) : 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 メディエーターサービスを停止します

ONTAP メディエ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メディエ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 メディエーターサービスを再度有効にします

ONTAP メディエ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メディエーターと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メディエーターを起動します。

```
systemctl start ontap_mediator
```

ONTAP メディエーターが正常であることを確認します

ONTAP メディエーターをインストールしたら、ONTAP メディ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

ホストのメンテナンスを実行するには、**SCST**を手動でアンインストールします

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 メディエーター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メディエーター1.8	scst-3.8.0.tar.bz2
ONTAPメディエーター1.7	scst-3.7.0.tar.bz2
ONTAPメディエーター1.6	scst-3.7.0.tar.bz2
ONTAPメディエーター1.5	scst-3.6.0.tar.bz2
ONTAPメディエーター1.4	scst-3.6.0.tar.bz2
ONTAP Mediator 1.3.	scst-3.5.0.tar.bz2
ONTAP メディエーター1.1	scst-3.4.0.tar.bz2
ONTAP Mediator 1.0の略	scst-3.3.0.tar.bz2

2. 「scst」ディレクトリにある次のコマンドを問題 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には、**SCST**を手動でインストールしてください

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 メディエーター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」ディレクトリにある次のコマンドを問題 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. cp scst/src/certs/scst_module_key.der
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/.

i. 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_modu
le_key.priv \
/opt/netapp/lib/ontap_mediator/ontap_mediator/SCST_mod_keys/scst_modu
le_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 メディエーターサービスを削除できます。

作業を開始する前に

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

このタスクについて

このタスクは、ONTAPメディ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

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。

このタスクについて

- 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、ONTAPメディア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、原因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

自己署名証明書を信頼できるサードパーティ証明書に置き換えることができます。

このタスクについて

- 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。

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、次のような自己署名証明書アクターの交換を示しています。ca.key、ca.csr、ca.srl`および `ca.crt` 場所: /opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ サードパーティの証明書アクターを使用します。



次の例は、ONTAPメディア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. 秘密鍵の作成 ca.key 構成ファイル openssl_ca.cnf PKI認証局が証明書を生成するために使用します。

a. 秘密鍵の生成 ca.key :

- 例 *

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

a. 構成ファイル openssl_ca.cnf (場所 :

/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する 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-----
MIIE6TCCAtECAQAwgaMxCzAJBgNVBAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlh
...
erARKhY9z0e8BHPl3g==
-----END CERTIFICATE REQUEST-----
```

3. 証明書署名要求を送信する ca.csr 署名のためにPKI機関に送られます

PKI機関は要求を検証し、.csr、証明書の生成 ca.crt。



SnapMirrorビジネス継続性 (SM-BC) クラスタの場合は、証明書を追加する必要があります。ca.crt ONTAPクラスタに追加します。を参照してください ["SM-BC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で署名されている必要があります。ca.key およびサードパーティの証明書 ca.crt。さらに、構成ファイル

/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/openssl_server.cnf
OpenSSLによって発行されるサーバ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。

サーバ証明書を生成するには、次のコマンドを実行します。

ステップ

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ます。

```
/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
```

```
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 オプションを使用してファイルを生成します。ca.srl。

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_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_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. を交換します ca.* サードパーティの証明書を含むファイル。
2. 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。

以外の名前のサードパーティ証明書を使用できます。 ca.* または、サードパーティの証明書を別の場所に保存します。

手順

1. ファイルの設定

/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config/ontap_mediator.user_config.yaml のデフォルト変数値を上書きするには ontap_mediator.config.yaml ファイル。

例えば、intermediate.crt PKI権限から秘密鍵を保存します。intermediate.key ショウメイシヨメイヨウキユウ intermediate.csr ある場所で

`/opt/netapp/lib/ontap_mediator/ontap_mediator/server_config`この場合、user_configファイルは次のようになります。

```
[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'
```

2. 構成ファイルで証明書が更新されたら、ONTAPメディ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。

で定義されたプロパティ **v3_ca** インチ `openssl_ca.cnf` と表示されます。X509v3 extensions インチ `ca.crt`。

```
[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 で定義されたプロパティ openssl_server.cnf 構成ファイルは次のように表示されます
X509v3 extensions 証明書で。

次の例では、alt_names コマンドの実行によるセクションの作成 hostname -A および hostname -I
ONTAPメディアエーターがインストールされているLinux VM。

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
```

状況に応じて `-modulus` 両方の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、コマンドを使用します。"[OpenSSL-検証](#)"。

ONTAP メディエーター用のOSホストを維持します

最適なパフォーマンスを得るには、ONTAP メディ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。

ホストをリブートします

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。

ホストパッケージの更新

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。

をインストールした場合 yum-utils パッケージでは、を使用します needs-restarting パッケージの変更
にレポートが必要かどうかを検出するコマンド。

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のマイナーカーネルアップグレード

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 メディエ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。（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

このタスクについて

- このタスクは、ONTAPメディ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、原因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。