gRPC エージェント

gRPC エージェントについて

gRPC は、最新のオープンソースの高性能なリモートプロシージャコール（Remote Procedure Call、RPC）フレームワークです。Cisco NX-OS は、gNMI や gNOI などの gRPC 関連サービスをサポートする gRPC エージェントを提供します。

更新履歴

リリース	説明（Description）
9.3(3)	次のサポートの追加： • grpc ポート • grpc 証明書
10.1(1)	クライアント証明書ベースの認証のサポートを追加 • client root certificate を grep 処理
10.3(3)	GRPC プロキシとして機能する NGINX のサポート

gRPC エージェントに関するガイドラインと制限事項

以下は、gRPC エージェントに関するガイドラインと制限事項です。

- 管理 VRF とデフォルト VRF の両方で gRPC を有効にし、後でデフォルト VRF で無効にすると、管理 VRF の gNMI 操作は機能しなくなります。

  回避策として、**no feature grpc** コマンドを入力して gRPC を完全に無効にし、**feature grpc** コマンド、または **grpc certificate** や **grpc port** のような任意の既存 gRPC 構成コマンドを入力して、再プロビジョニングします。また、管理 VRF の既存の通知に再登録する必要もあります。

- gRPC 証明書が明示的に設定されている場合、保存されたスタートアップ コンフィギュレーションを使用して以前の Cisco NX-OS 9.3(x) イメージにリロードした後、gRPC 機能は接続を受け入れません。

  show grpc gnmi service statistics コマンドを入力して確認します。次のステータスエラーメッセージが表示されます。

  `Status: Not running - Initializing...Port not available or certificate invalid.`（ステータス:実行していません-初期化中...ポートが使用できないか、証明書が無効です。）

  サービスを復元するには、適切な証明書コマンドを設定解除して設定します。

- カスタム gRPC 証明書を構成している場合、**reload ascii** コマンドを入力すると構成が失われます。デフォルトの day-1 証明書に戻ります。**reload ascii** コマンドを入力した後には、スイッチをリロードします。スイッチが再び起動したら、gRPC カスタム証明書を再設定する必要があります。

（注） これは、grpc 証明書コマンドを入力した場合に適用されます。

- gRPC のデフォルト以外の VRF の到達可能性は、L3VNI/EVPN および IP 経由でのみサポートされます。ただし、デフォルト以外の VRF および VXLAN フラッドおよびラーニングでの MPLS を介した到達可能性はサポートされていません。

- 9.3(x) より前の Cisco NX-OS リリースにおいてサポートされるプラットフォームの詳細については、そのリリース向けガイドの「プログラマビリティ機能のプラットフォーム サポート」を参照してください。Cisco NX-OS リリース 9.3(x) 以降でサポートされているプラットフォームについては、『Nexus Switch Platform Matrix』を参照してください。

- gRPC プロセスは、CPU 使用率を CPU の 75% に、メモリを 4 GB に制限する HIGH_PRIO 制御グループを使用します。

- gRPC エージェントは、各スイッチ上で、合計で 2 台の gRPC サーバに対し、管理 VRF と 1 台のユーザー指定 VRF をサポートします。ユーザー指定 VRF（たとえばデフォルト

VRF）で gRPC をサポートすれば、大量のトラフィック負荷が望ましくない管理 VRF からの gRPC 呼び出しの処理を、柔軟にオフロードできます。

- 2 つの gRPC サーバーを構成する場合は、次の点に注意してください。
  - VRF 境界は厳密に適用されるため、各 gRPC サーバーは相互に独立して要求を処理します。要求は VRF 間を通過しません。
  - 2 台のサーバーは HA またはフォールト トレラントではありません。一方の gRPC サーバーは他方をバックアップせず、それらの間でスイッチオーバーまたはスイッチバックはありません。
  - gRPC サーバーの制限は VRF 単位です。
- Cisco NX-OS リリース 10.4(3)F 以降、gRPC は 92348GC-X でサポートされます。

gRPC エージェントの構成

gRPC の構成

gNMI 機能は、grpc コマンドを使用して構成します。

grpc certificateコマンドで使用される証明書をスイッチにインポートするには、『Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド』の「アイデンティティ証明書のインストール」のセクションを参照してください。

（注） インストールされている ID 証明書または と の値を変更すると、gRPC サーバーが再起動して変更が適用される場合があります。**grpc portgrpc certificate** gRPC サーバが再起動すると、アクティブなサブスクリプションはすべてドロップされるため、再サブスクライブする必要があります。

始める前に

サーバ認証に必要な証明書ファイルを準備し、署名します。

これは gRPC に固有ではないため、既存のトラストポイントファイルを再利用できます。

手順の概要

1. **configure terminal**
2. （任意） **crypto ca trustpoint** <*server-trustpoint*>
3. **crypto ca import** <*server-trustpoint*> **pkcs12 bootflash:** *:*<*server-ca-file*> <*pkcs-password*>
4. **feature grpc**
5. （任意） **grpc port** *port-id*
6. **grpc certificate** *certificate-id*
7. （任意） **use-vrf default**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		構成モードに入ります。

ステップ 2	（任意） **crypto ca trustpoint** <*server-trustpoint*> 例： `switch# crypto ca trustpoint tls_server_trustpoint`	サーバ認証用のトラストポイントを作成します。 使用可能なサーバトラストポイントがすでに存在する場合、ステップ 2 ～ 3 はオプションです。
ステップ 3	**crypto ca import** <*server-trustpoint*> **pkcs12 bootflash:** *:<server-ca-file>* <*pkcs-password*> 例： `switch# crypto ca import tls_server_trustpoint pkcs12 bootflash:server.pfx test`	サーバの pkcs12 ファイルをトラストポイントにインポートします。

ステップ 4	**feature grpc**
	例：
	`switch# feature grpc`
	`switch(config)#`
		ダイヤルイン用の gNMI インターフェイスをサポートする gRPC エージェントを有効にします。

ステップ 5	（任意） **grpc port** *port-id* 例： `switch(config)# grpc port 50051`	ポート番号を構成します。port-id の範囲は 1024 ～ 65535 です。50051 がデフォルトです。
ステップ 6	**grpc certificate** *certificate-id* 例： `switch(config)# grpc certificate cert-1`	証明書トラストポイント ID を指定します。詳細については、『Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド』の「アイデンティティ証明書のインストール」セクションで、証明書のインポートについて確認してください。
ステップ 7	（任意） **use-vrf default** 例： `switch(config)# grpc use-vrf default`	gRPC エージェントがデフォルト VRF からの着信（ダイヤルイン）RPC 要求を受け入れられるようにします。この手順により、デフォルト VRF が着信 RPC 要求を処理できるようになります。デフォルトでは、gRPC機能が有効になっている場合、管理VRFが着信 RPC 要求を処理します。

キー/証明書の生成

次に、スイッチの bash シェルで自己署名キー/証明書を生成する例を示します。これは実験のみを目的としています。アイデンティティ証明書の生成の詳細については、『Cisco Nexus 9000

シリーズ NX-OS セキュリティ構成ガイド』の「アイデンティティ証明書のインストール」のセクションを参照してください。

（注） このタスクは、スイッチで証明書を生成する方法の例です。任意の Linux 環境で証明書を生成することもできます。実稼働環境では、CA 署名付き証明書の使用を検討する必要があります。

手順の概要

1. 自己署名キーと pem ファイルを生成します。
2. キー ファイルと pem ファイルを生成した後、トラストポイント CA アソシエーションで使用するためにキー ファイルと pem ファイルをバンドルする必要があります。
3. pkcs12 バンドルをトラストポイントに入力して、トラストポイント CA アソシエーションを設定します。
4. セットアップを確認します。

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	自己署名キーと pem ファイルを生成します。	`switch# run bash sudo su` `bash-4.3# openssl req -x509 -newkey rsa:2048 -keyout self_sign2048.key -out self_sign2048.pem -days 365 -nodes`

ステップ 2	キー ファイルと pem ファイルを生成した後、トラストポイント CA アソシエーションで使用するためにキー ファイルと pem ファイルをバンドルする必要があります。	`After generating the key and pem files, you must bundle the key and pem files for use in the trustpoint CA Association.`
		`switch# run bash sudo su bash-4.3# cd /bootflash/`
		`bash-4.3# openssl pkcs12 -export -out self_sign2048.pfx -inkey self_sign2048.key -in self_sign2048.pem -certfile self_sign2048.pem -password pass:Ciscolab123!`
		`bash-4.3# exit`

ステップ 3	pkcs12 バンドルをトラストポイントに入力して、トラストポイント CA アソシエーションを設定します。	`switch(config)# crypto ca trustpoint mytrustpoint` `switch(config-trustpoint)# crypto ca import mytrustpoint pkcs12 self_sign2048.pfx Ciscolab123!`

ステップ 4	セットアップを確認します。	`switch(config)# show crypto ca certificates`
		`Trustpoint: mytrustpoint certificate:`
		`subject= /C=US/O=Cisco Systems, Inc./OU=CSG/L=San Jose/ST=CA/street=3700 Cisco Way/postalCode=95134/CN=ems.cisco.com/serialNumber=FGE18420K0R`
		`issuer= /C=US/O=Cisco Systems, Inc./OU=CSG/L=San Jose/ST=CA/street=3700 Cisco Way/postalCode=95134/CN=ems.cisco.com/serialNumber=FGE18420K0R`
		`serial=0413`
		`notBefore=Nov 5 16:48:58 2015 GMT`
		`notAfter=Nov 5 16:48:58 2035 GMT`
		`SHA1 Fingerprint=2E:99:2C:CE:2F:C3:B4:EC:C7:E2:52:3A:19:A2:10:D0:54:CA:79:3E`

	コマンドまたはアクション	目的
		purposes: sslserver sslclient CA certificate 0: subject= /C=US/O=Cisco Systems, Inc./OU=CSG/L=San Jose/ST=CA/street=3700 Cisco Way/postalCode=95134/CN=ems.cisco.com/serialNumber=FGE18420K0R issuer= /C=US/O=Cisco Systems, Inc./OU=CSG/L=San Jose/ST=CA/street=3700 Cisco Way/postalCode=95134/CN=ems.cisco.com/serialNumber=FGE18420K0R serial=0413 notBefore=Nov 5 16:48:58 2015 GMT notAfter=Nov 5 16:48:58 2035 GMT SHA1 Fingerprint=2E:99:2C:CE:2F:C3:B4:EC:C7:E2:52:3A:19:A2:10:D0:54:CA:79:3E purposes: sslserver sslclient

gRPC クライアント証明書認証の構成

gRPC では、証明書ファイル（公開キー）に基づいてクライアントを認証することもできます。これにより、パスワードベースの認証よりも安全であると考えられるパスワードレス認証が提供されます。

始める前に

サーバ認証に必要な証明書ファイルを準備し、署名します。

これは gRPC に固有ではないため、既存のトラストポイントファイルを再利用できます。

手順の概要

1. **configure terminal**
2. （任意） **crypto ca trustpoint** <*server-trustpoint*>
3. **rsakeypair** <*client-key*>
4. （任意） **crypto ca authenticate** <*client-root-trustpoint*>
5. **grpc client root certificate** <*client-root-trustpoint*>

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		構成モードに入ります。

ステップ 2	（任意） **crypto ca trustpoint** <*server-trustpoint*> 例： `switch# crypto ca trustpoint tls_server_trustpoint`	サーバ認証用のトラストポイントを作成します。 使用可能なサーバトラストポイントがすでに存在する場合、ステップ 2 ～ 3 はオプションです。

	コマンドまたはアクション	目的
ステップ 3	**rsakeypair** <*client-key*> 例： `switch# rsakeypar client-key`	クライアント トラストポイントの rsa キーペアを生成します。
ステップ 4	（任意） **crypto ca authenticate** <*client-root-trustpoint*> 例： `switch# crypto ca authenticate client_trustpoint`	クライアント証明書をインポートします。この手順では、手動でコピーして貼り付ける必要があります。指示に従ってください。
ステップ 5	**grpc client root certificate** <*client-root-trustpoint*> 例： `switch(config)# grpc client root certificate client_trustpoint`	クライアント CA ルート証明書をホストするトラストポイントを入力します。

例

構成例

このセクションでは、説明のために構成シーケンスの例を示します。

1. クライアント ルートCA証明書を準備します。
2. 証明書のインポート

   クライアント root に対する新しい証明書が正常に生成されたときの、スイッチで証明書を構成するためのコマンド例とその出力を次に示します。

```
switch(config)# crypto ca trustpoint my_client_trustpoint
switch(config-trustpoint)# crypto ca authenticate my_client_trustpoint
input (cut & paste) CA certificate (chain) in PEM format; end the input with a line
 containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
MIIDUDCCAjigAwIBAgIJAJLisBKCGjQOMA0GCSqGSIb3DQEBCwUAMD0xCzAJBgNV
BAYTAlVTMQswCQYDVQQIDAJDQTERMA8GA1UEBwwIU2FuIEpvc2UxDjAMBgNVBAoM
BUNpc2NvMB4XDTIwMTAxNDIwNTYyN1oXDTQwMTAwOTIwNTYyN1owPTELMAkGA1UE
BhMCVVMxCzAJBgNVBAgMAkNBMREwDwYDVQQHDAhTYW4gSm9zZTEOMAwGA1UECgwF
Q2lzY28wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDEX7qZ2EdogZU4
EW0NSpB3EjY0nSlFLOw/iLKSXfIiQJD0Qhaw16fDnnYZj6vzWEa0ls8canqHCXQl
gUyxFOdGDXa6neQFTqLowSA6UCSQA+eenN2PIpMOjfdFpaPiHu3mmcTI1xP39Ti3
/y548NNORSepApBNkZ1rJSB6Cu9AIFMZgrZXFqDKBGSUOf/CPnvIDZeLcun+zpUu
CxJLA76Et4buPMysuRqMGHIX8CYw8MtjmuCuCTHXNN31ghhgpFxfrW/69pykjU3R
YOrwlSUkvYQhtefHuTHBmqym7MFoBEchwrlC5YTduDzmOvtkhsmpogRe3BiIBx45
AnZdtdi1AgMBAAGjUzBRMB0GA1UdDgQWBBSh3IqRrm+mtB5GNsoLXFb3bAVg5TAf
BgNVHSMEGDAWgBSh3IqRrm+mtB5GNsoLXFb3bAVg5TAPBgNVHRMBAf8EBTADAQH/
MA0GCSqGSIb3DQEBCwUAA4IBAQAZ4Fpc6lRKzBGJQ/7oK1FNcTX/YXkneXDk7Zrj
8W0RS0Khxgke97d2Cwl5P5reXO27kvXsnsz/VZn7JYGUvGSlxTlcCb6x6wNBr4Qr
t9qDBu+LykwqNOFe4VCAv6e4cMXNbH2wHBVS/NSoWnM2FGZ10VppjEGFm6OM+N6z
8n4/rWslfWFbn7T7xHH+Nl0Ffc+8q8h37opyCnb0ILj+a4rnyus8xXJPQb05DfJe
ahPNfdEsXKDOWkrSDtmKwtWDqdtjSQC4xioKHoshnNgWBJbovPlMQ64UrajBycwV
z9snWBm6p9SdTsV92YwF1tRGUqpcI9olsBgH7FUVU1hmHDWE
-----END CERTIFICATE----END OF INPUT
Fingerprint(s): SHA1
Fingerprint=0A:61:F8:40:A0:1A:C7:AF:F2:F7:D9:C7:12:AE:29:15:52:9D:D2:AE
Do you accept this certificate? [yes/no]:yes switch(config)#
```

```
NOTE: Use the CA Certificate from the .pem file content.
switch# show crypto ca certificates Trustpoint: my_client_trustpoint CA certificate
 0:
subject=C = US, ST = CA, L = San Jose, O = Cisco
issuer=C = US, ST = CA, L = San Jose, O = Cisco
serial=B7E30B8F4168FB87 notBefore=Oct 1 17:29:47 2020 GMT notAfter=Sep 26 17:29:47
2040 GMT
SHA1 Fingerprint=E4:91:4E:D4:41:D2:7D:C0:5A:E8:F7:2D:32:81:B3:37:94:68:89:10 purposes:
 sslserver sslclient
```

3. gRPC へのトラストポイントの関連付け

   クライアントルートに新しい証明書を正常に構成した後、スイッチ上でトラストポイントを gRPC サーバに関連付ける出力例を次に示します。

```
switch(config)# feature grpc
switch(config)# grpc client root certificate my_client_trustpoint switch(config)#
show run grpc
!Command: show running-config grpc
!Running configuration last done at: Wed Dec 16 20:18:35 2020
!Time: Wed Dec 16 20:18:40 2020
version 10.1(1) Bios:version N/A feature grpc
grpc gnmi max-concurrent-calls 14 grpc use-vrf default grpc certificate my_trustpoint
grpc client root certificate my_client_trustpoint grpc port 50003
```

4. 証明書の詳細の検証

   スイッチの gRPC にトラストポイントを正常に関連付けられた場合の、証明書の詳細を検証するための出力例を次に示します。

```
switch# show grpc gnmi service statistics
============= gRPC Endpoint =============
Vrf : management
Server address : [::]:50003
Cert notBefore : Mar 13 19:05:24 2020 GMT
Cert notAfter : Nov 20 19:05:24 2033 GMT
Client Root Cert notBefore : Oct 1 17:29:47 2020 GMT
Client Root Cert notAfter : Sep 26 17:29:47 2040 GMT
...
```

5. 任意の gNMI クライアントのクライアント証明書認証を使用した接続の確認。

   クライアント証明書は、秘密キー（pkey）と CA チェーン（cchain）を使用して要求を行います。現在では、パスワードはオプションです。クライアントがルート CAからクライアント証明書への完全なチェーンを提供する必要があることを確認してください。

6. gRPC からトラストポイント参照を削除するには（no コマンド）、次のコマンドを使用します。

```
switch(config)# no grpc client root certificate my_client_trustpoint
```

   コマンドは、gRPCエージェントのトラストポイント参照だけを削除します。トラストポイントCA 証明書は削除されません。スイッチ上の gRPC サーバーへのクライアント証明書認証を使用する接続は確立されませんが、ユーザー名とパスワードによる基本認証は通過します。

GRPC 向けの NGINX プロキシの構成

NetconfやRestconfと同様に、gRPCエージェントは専用のサーバ/ポートで実行されます。gRPCクライアントは、gRPC エージェント/サーバに直接接続する必要があります。

リリース10.3(3)F 以降、NX-OS NGINX は gRPCトラフィックをリレーすることで GRPCプロキシとしても機能できます。これは特定のユースケースに役立ちます。

- GRPC ポートがブロックされました：GRPC エージェントはポート 50051 でリッスンします。このポートがファイアウォールによってブロックされている場合、GRPC クライアントは NGINX HTTPS ポート 443 を介して gRPC サービスに間接的にアクセスできます。
- VRF サポートの強化：現在、GRPC サービスには、管理 VRF または 1 つのユーザー指定 VRF を経由してのみアクセスできます。NGINXプロキシは、任意の VRF からの gRPC 要求を転送できます。

この新しいサポートは、既存の動作には影響しません。GRPCクライアントは、引き続きGRPCエージェントに直接接続できます。代わりにNGINXサーバに接続することもできます。NGINXサーバは、プロキシとして、GRPC 要求を GRPCエージェントに送ります。このようなリダイレクトを行うと、追加の要求応答遅延が発生したと見なされることに注意してください。

すべてのサーバーとクライアントの認証は、NGINX によって処理されます。GRPC を有効にして、NGINX サーバー証明書やクライアント証明書を構成するだけで十分です。

始める前に

grpc 機能を有効にします。

NX-API 証明書を準備します。詳細については、「NX-API CLI の使用」を参照してください。

手順の概要

1. **configure terminal**
2. **feature nxapi**
3. **nxapi certificate httpscrt certfile** *cert-file*
4. **nxapi certificate httpscrt keyfile** *key-file* **password** <*password*>
5. **nxapi certificate enable**
6. （任意） **crypto ca trustpoint** <*trustpoint*>
7. （任意） **crypto ca authenticate** <*trustpoint*>
8. （任意） **nxapi client certificate authentication**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	構成モードに入ります。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	

ステップ 2	**feature nxapi**
	例：
	`switch# feature nxapi`
	`switch(config)#`
		nxapi 機能を有効にします。

ステップ 3	**nxapi certificate httpscrt certfile** *cert-file* 例： `switch# nxapi certificate httpscrt certfile bootflash:nxapi.crt`	証明書ファイルを構成します。
ステップ 4	**nxapi certificate httpscrt keyfile** *key-file* **password** *<password>* 例： `switch# nxapi certificate httpskey keyfile bootflash:nxapi.key password cisco123`	キーファイルを構成します。
ステップ 5	**nxapi certificate enable** 例： `switch# nxapi certificate enable`	証明書認証を有効にします。
ステップ 6	（任意） **crypto ca trustpoint** *<trustpoint>* 例： `switch# crypto ca trustpoint grpcClientCA`	サーバ認証用のトラストポイントを作成します。
ステップ 7	（任意） **crypto ca authenticate** *<trustpoint>* 例： `switch# crypto ca authenticate grpcClientCA`	クライアントルート証明書をトラストポイントにインポートします。
ステップ 8	（任意） **nxapi client certificate authentication** 例： `switch# nxapi client certificate authentication`	クライアント証明書認証を有効にします。

トラブルシューティング

機能ステータスの確認

- Cisco NX-OS デバイスで、**show feature grpc** コマンドを入力してエージェントの構成を確認します。
- gRPCエージェントのステータスを表示するには、 **show feature** コマンドを使用します。

```
switch-1# show feature | grep grpc
restconf 1 enabled
switch-1#
```

接続性の確認

クライアント システムから、スイッチの管理ポートに ping を実行して、スイッチが到達可能であることを確認します。

gRPC エージェントログの収集

/volatile ディレクトリには、grpc エージェントログが格納されます。

```
bash-4.3# cd /volatile/ bash-4.3# ls /volatile -al
…
-rw-rw-rw- 1 root root 103412 Jun 21 16:14 grpc-internal-log
…
```

TM トレース ログの収集

```
tmtrace.bin -f gnmi-logs gnmi-events gnmi-errors following are available 2.
Usage:
bash-4.3# tmtrace.bin -d gnmi-events | tail -30 Gives the last 30
…
[06/21/19 15:58:38.969 PDT f8f 3133] [3981658944][tm_transport_internal.c:43] dn:
Cisco-NX-OS-device:System/cdp-items, sub_id: 0,
sub_id_str: 2329, dc_start_time: 0, length: 124, sync_response:1
[06/21/19 15:58:43.210 PDT f90 3133] [3621780288][tm_ec_yang_data_processor.c:93] TM_EC:
 [Y] Data received for 2799743488: 49
{
  "cdp-items": {
    "inst-items": {
      "if-items": {
        "If-list": [
          {
            "id": "mgmt0",
            "ifstats-items": {
              "v2Sent": "74",
              "validV2Rcvd": "79"
            }
          }
        ]
      }
    }
  }
} [06/21/19 15:58:43.210 PDT f91 3133] [3981658944][tm_transport_internal.c:43] dn:
Cisco-NX-OS-device:System/cdp-items, sub_id: 0,
sub_id_str: 2329, dc_start_time: 0, length: 141, sync_response:1
[06/21/19 15:59:01.341 PDT f92 3133] [3981658944][tm_transport_internal.c:43] dn:
Cisco-NX-OS-device:System/intf-items, sub_id:
4091, sub_id_str: , dc_start_time: 1561157935518, length: 3063619, sync_response:0
[06/21/19 15:59:03.933 PDT f93 3133] [3981658944][tm_transport_internal.c:43] dn:
Cisco-NX-OS-device:System/cdp-items, sub_id:
4091, sub_id_str: , dc_start_time: 1561157940881, length: 6756, sync_response:0 [06/21/19
 15:59:03.940 PDT f94 3133] [3981658944][tm_transport_internal.c:43] dn:
Cisco-NX-OS-device:System/lldp-items, sub_id:
```

MTX 内部ログの収集

1. /opt/mtx/conf/mtxlogger.cfg を修正して構成を変更します。

   優先フィルタを切り替えるには、「診断と有用性」のセクションを参照してください。

2. **feature grpc**を無効にし、それから有効にします。

3. /volatileディレクトリには、mtx-internal.logがあります。ログは時間の経過とともにロールオーバーされるため、ロールオーバーしてしまう前にログをダウンロードしてください。

```
bash-4.3# cd /volatile/ bash-4.3# ls /volatile -al
…
-rw-r--r-- 1 root root 24 Jun 21 14:44 mtx-internal-19-06-21-14-46-21.log
-rw-r--r-- 1 root root 24 Jun 21 14:46 mtx-internal-19-06-21-14-46-46.log
-rw-r--r-- 1 root root 175 Jun 21 15:11 mtx-internal-19-06-21-15-11-57.log
-rw-r--r-- 1 root root 175 Jun 21 15:12 mtx-internal-19-06-21-15-12-28.log
-rw-r--r-- 1 root root 175 Jun 21 15:13 mtx-internal-19-06-21-15-13-17.log
-rw-r--r-- 1 root root 175 Jun 21 15:13 mtx-internal-19-06-21-15-13-42.log
-rw-r--r-- 1 root root 24 Jun 21 15:13 mtx-internal-19-06-21-15-14-22.log
-rw-r--r-- 1 root root 24 Jun 21 15:14 mtx-internal-19-06-21-15-19-05.log
-rw-r--r-- 1 root root 24 Jun 21 15:19 mtx-internal-19-06-21-15-47-09.log
-rw-r--r-- 1 root root 24 Jun 21 15:47 mtx-internal.log
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。