ハイ アベイラビリティのインストールの管理

ここでは、ハイ アベイラビリティ用に IoT FND を設定する方法について説明します。具体的な内容は次のとおりです。

IoT FND ハイ アベイラビリティの概要

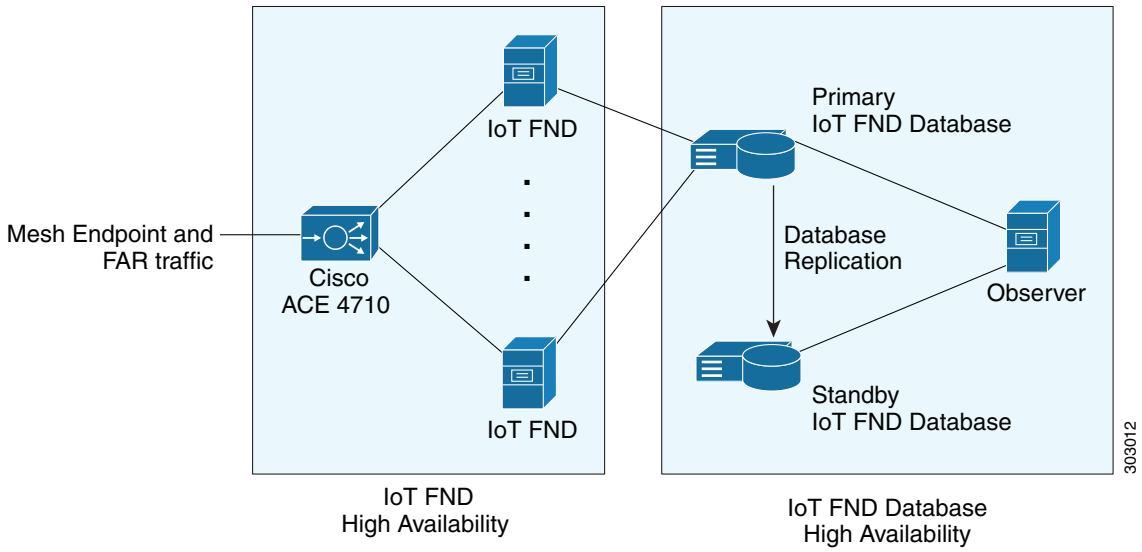
ここでは、IoT FND ハイ アベイラビリティのインストールの概要を提供します。具体的な内容は次のとおりです。

IoT FNDは、Connected Grid のモニタおよび管理にとって重要なアプリケーションです。IoT FND ハイ アベイラビリティ（IoT FND HA）ソリューションは、ソフトウェア、ネットワーク、またはハードウェアの障害発生時に、IoT FND の全体的な可用性に対応します。

図 1 に示すように、IoT FND は 2 つの主要なレベルの HA を提供します。

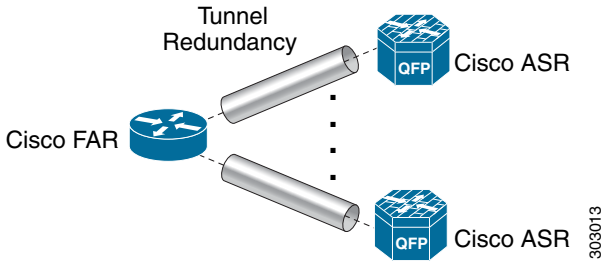
- IoT FND サーバ HA：これは複数の IoT FND サーバを Cisco ACE 4710 ロード バランサに接続することで実現されます。ME、FAR、ASR で発生するトラフィックは、ロード バランサに送られます。ロード バランサは、ラウンドロビン プロトコルを使用して IoT FND クラスタ サーバ間で負荷を分散します。
- IoT FND データベース HA：これは 2 つの IoT FND データベース サーバ（プライマリ サーバとスタンバイ（セカンダリ）サーバ）を設定することで実現されます。プライマリ データベースは新しいデータを受信すると、コピーをスタンバイ データベースに送信します。別のシステムがオブザーバを実行します。オブザーバは IoT FND データベース サーバをモニタするプログラムで、スタンバイ サーバでも実行できます。プライマリ データベースに障害が発生すると、オブザーバはスタンバイ サーバを新しいプライマリ データベースとして設定します。IoT FND データベース HA は、シングルおよびクラスタ IoT FND サーバ展開でも機能します。

図 1　IoT FND サーバおよびデータベース HA



IoT FND サーバとデータベース HA に加え、トンネルの冗長性を加えることで IoT FND の信頼性が向上します。これは 1 つの FAR と複数の ASR 間で複数のトンネルを定義することで実現されます。1 つのトンネルで障害が発生すると、FAR は別のトンネル経由でトラフィックをルーティングします。

図 2　IoT FND トンネルの冗長性



IoT FND HA は、以下の障害シナリオに対応します。

障害のタイプ	説明
IoT FND サーバの障害	IoT FND サーバ クラスタ内の 1 台のサーバに障害が発生すると、ロード バランサがクラスタ内の他のサーバにトラフィックをルーティングします。
IoT FND データベースの障害	プライマリ データベースに障害が発生すると、関連付けられたスタンバイ データベースがプライマリ データベースになります。これは IoT FND サーバに対してトランスペアレントです。クラスタ内のすべての IoT FND サーバが新しいプライマリ データベースに接続します。
トンネルの障害	トンネルに障害が発生すると、トラフィック フローは別のトンネルを経由します。

ロード バランサ

ロード バランサ（LB）は以下のタスクを実行するため、IoT FND HA において重要な役割を担います。

- IoT FND へのトラフィックを負荷分散します。
- クラスタ内のサーバとのハートビートを維持し、障害を検出します。IoT FND サーバに障害が発生すると、LB は他のクラスタ メンバーにトラフィックを向けます。

この展開では、ロード バランサとして Cisco ACE 4710（Cisco ACE）を使用することを推奨します。Cisco ACE 4710 の詳細については、http://www.cisco.com/en/US/partner/products/ps7027/tsd_products_support_series_home.html を参照してください。

サーバのハートビート

LB は、クラスタ内の各 IoT FND サーバとのハートビートを維持します。IoT FND ソリューション（代替ソリューションあり）で採用されているヘルス モニタリング メカニズムでは、ハートビートはポート 80 での IoT FND への 通常の GET メッセージです。IoT FND はアクティブな IoT FND サーバからの「HTTP 200 OK」の応答を求めます。

LB で次のハートビート パラメータを設定できます。

- Periodicity of probes：これはハートビート間の秒数です。Cisco ACE でのデフォルト値は 15 秒です。
- Number of retries：これは LB が応答しない IoT FND サーバにダウンを宣言する前に、ハートビートの送信を試行する回数です。デフォルトの再試行回数は 3、
- Regular checks after failure detection：LB はこの時間間隔でサーバがオンラインに戻ったかどうかを確認します。障害検出チェックのデフォルト値は 60 秒です。

データベース ハイ アベイラビリティ

IoT FND データベース HA は、IoT FND シングル サーバーとクラスタ展開で機能します。IoT FND HA は Oracle Active Dataguard を使用して、Oracle HA を展開します。IoT FND データベース用に HA を設定するには、Oracle Recovery Manager（RMAN）と Dataguard Management CLI（DGMGRL）を使用します。

IoT FND データベース HA 設定プロセスには以下が含まれます。

- 別の物理サーバでプライマリ データベースとセカンダリ データベースを同じように設定します。

  (注) セカンダリ データベース サーバは、スタンバイ データベースとも呼ばれます。

  (注) データベースのフェールオーバー時に、データが失われる可能性があります。

- Oracle *ウォレット*を使用して、データ レプリケーションが SSL を介して実行されるように設定します。このウォレットには、迅速な展開を促進する自己署名証明書が含まれています。

  (注) IoT FND RPM にバンドルされている Oracle ウォレットは、自己署名証明書を使用します。カスタム証明書とウォレットを設定して、レプリケーションを円滑に行うことができます。

  (注) SSL を介してデータ レプリケーションを実行しても、パフォーマンスへの影響はありません。

- レプリケーションには、cgms_dev *ではなく*、sys ユーザを使用します。
- パフォーマンスのボトルネックを防止するため、レプリケーションを非同期に設定します。

デフォルトでは、IoT FND は TCP を使用し、ポート 1522 を介してデータベースに接続します。レプリケーションはポート 1622 で TCPS（TCP over SSL）を使用します。

IoT FND データベース HA を設定するためのスクリプトは、IoT FND Oracle Database RPM パッケージ（cgms-oracle-*version_number*.x86_64.rpm）に含まれています。IoT FND データベースをインストールすると、HA スクリプトは $ORACLE_HOME/cgms/scripts/ha に配置されます。

トンネルの冗長性

IoT FND の展開にさらなる冗長性を追加するには、FAR トンネル プロビジョニング グループ内のすべての FAR を複数の ASR に接続する複数のトンネルを設定します。たとえば、すべての FAR に 2 つのトンネルをプロビジョニングするように IoT FND を設定することができます。1 つのトンネルがセルラー インターフェイスを介してアクティブになっている間、冗長トンネルは WiMAX インターフェイスを介して 2 番目の ASR と通信するように設定します。

トンネルの冗長性を設定するには、以下を実行する必要があります。

1. トンネル プロビジョニング グループに ASR を追加します。
2. トンネル プロビジョニング テンプレートを変更して、追加のトンネルを作成するコマンドを含めます。
3. FAR と ASR のインターフェイスで、インターフェイス間のマッピングを決定するポリシーを定義します。
   - トンネル プロビジョニング ポリシーの設定
   - トンネル冗長性のためのトンネル プロビジョニング テンプレートの変更

HA の注意事項および制限事項

IoT FND HA の設定に関して、次の点に注意してください。

- IoT FND HA には、FAR、ASR、ロード バランサなどの他のネットワーク コンポーネントの HA サポートは含まれていません。
- IoT FND HA ではゼロ サービス ダウンタイムを目指していますが、これを保証してはいません。
- IoT FND ノードはすべて同じサブネット上にある必要があります。
- IoT FND ノードはすべて、同じようなハードウェアで実行する必要があります。
- すべての IoT FND ノードが同じソフトウェア バージョンを実行する必要があります。
- すべてのノードで IoT FND セットアップ スクリプト（/opt/cgms/bin/setupCgms.sh）を実行します。
- DB の移行のスクリプト（/opt/cgms/bin/db-migrate）は、1 つのノードでのみ実行します。
- /opt/cgms/bin/print_cluster_view.sh スクリプトは、IoT FND クラスタ メンバーに関する情報を表示します。

HA 用の IoT FND インストールの設定

ここでは、IoT FND HA インストールのさまざまな設定について説明します。具体的な内容は次のとおりです。

- HA 用の IoT FND データベースの設定
- IoT FND データベース HA の無効化
- ロード バランシング ポリシー
- LB の実行コンフィギュレーションの例
- トンネル プロビジョニング ポリシーの設定
- トンネル冗長性のためのトンネル プロビジョニング テンプレートの変更

HA 用の IoT FND データベースの設定

IoT FND HA データベースを設定するには、次の手順を実行します。

1. スタンバイ データベースを設定します(「スタンバイ データベースの設定」を参照)。

   (注) 必ず最初にスタンバイ データベースを設定します。

   - スタンバイ サーバのデフォルト SID は **cgms_s** で、cgms *ではありません*。
   - HA 用のスタンバイ サーバを設定する前に、スタンバイ サーバの環境変数 $ORACLE_SID が **cgms_s** に設定されていることを確認します。
   - ポートは常に 1522 です。

2. プライマリ データベースを設定します(「プライマリ データベースの設定」を参照)。

   - プライマリ サーバのデフォルト SID は **cgms** です。
   - HA 用のプライマリ サーバを設定する前に、プライマリ サーバの環境変数 $ORACLE_SID が **cgms** に設定されていることを確認します。

3. データベース HA 用に IoT FND を設定します(「データベース HA 用の IoT FND の設定」を参照)。
4. データベース オブザーバを設定します(「オブザーバの設定」を参照)。

スタンバイ データベースの設定

HA 用のスタンバイ データベース サーバを設定するには、setupStandbyDb.sh スクリプトを実行します。このスクリプトでは、プライマリ データベースの IP アドレスなど、スタンバイ データベースに必要な設定情報を入力するように求められます。

```
$ ./setupStandbyDb.sh
$ Are you sure you want to setup a standby database ? (y/n)? y

09-20-2012 13:59:18 PDT: INFO: User response: y
09-20-2012 13:59:18 PDT: INFO: CGMS_S database does not exist.
Enter the SYS DBA password.NOTE: This password should be same as the one set on the primary server:
Re-enter password for SYS DBA:
09-20-2012 13:59:58 PDT: INFO: User entered SYS DBA password.

Enter new password for CG-NMS database:
Re-enter new password CG-NMS database:
09-20-2012 14:00:09 PDT: INFO: User entered CG-NMS DB password.
Enter primary database server IP address: 192.168.1.12
09-20-2012 14:00:27 PDT: INFO: Cleaning up instance - cgms_s
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production
…
Total System Global Area  329895936 bytes
Fixed Size    2228024 bytes
Variable Size  255852744 bytes
Database Buffers   67108864 bytes
Redo Buffers    4706304 bytes
...
09-20-2012 14:00:29 PDT: INFO: ========== CGMS_S Database Setup Completed Successfully ==========
```

プライマリ データベースの設定

HA 用のプライマリ データベース サーバを設定するには、setupHaForPrimary.sh スクリプトを実行します。このスクリプトでは、スタンバイ データベースの IP アドレスなど、プライマリ データベースに必要な設定情報を入力するように求められます。

```
$ ./setupHaForPrimary.sh
[oracle@pdb ha]$ ./setupHaForPrimary.sh
09-20-2012 13:58:39 PDT: INFO: ORACLE_BASE: /home/oracle/app/oracle
09-20-2012 13:58:39 PDT: INFO: ORACLE_HOME: /home/oracle/app/oracle/product/11.2.0/dbhome_1
09-20-2012 13:58:39 PDT: INFO: ORACLE_SID : cgms
09-20-2012 13:58:39 PDT: INFO: Make sure the above environment variables are what you expect

Are you sure you wish to configure high availability for this database server ? (y/n)? y

09-20-2012 13:58:45 PDT: INFO: User response: y
Enter standby database server IP address: 192.168.1.10
09-20-2012 13:58:56 PDT: INFO: Secondary listener reachable.Moving on with configuration
mkdir: cannot create directory `/home/oracle/app/oracle/oradata/cgms': File exists
09-20-2012 13:58:58 PDT: INFO: Reloading the listener to pick the new settings

LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 20-SEP-2012 13:58:58

…
DGMGRL> 09-20-2012 14:14:54 PDT: INFO: Please start the 'Observer' on appropriate server for ha
monitoring
Total time taken to perform the operation: 975 seconds
09-20-2012 14:14:54 PDT: INFO: ==========  Completed Successfully ==========
```

オブザーバの設定

オブザーバは個別のサーバで実行する必要がありますが、スタンバイ データベースをホストしているサーバで設定できます。

(注) オブザーバの実行に必要なパスワードは、SYS DBA パスワードと同じです。IoT FND Oracle データベースの作成を参照してください。

オブザーバを設定するには、次の手順を実行します。

1. 個別のサーバでオブザーバ スクリプトを実行します。

   ```
   $ ./manageObserver.sh start cgms_s password
   $ DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production
   ...
   Welcome to DGMGRL, type "help" for information.
   DGMGRL> Connected.
   DGMGRL> Observer started
   ```

2. getHaStatus.sh スクリプトを実行して、データベースが HA 用に設定されていることを確認します。

   ```
   $ ./getHaStatus.sh
   ...
   Configuration - cgms_dgconfig

     Protection Mode: MaxPerformance
     Databases:
       cgms   - Primary database
       cgms_s - (*) Physical standby database

   Fast-Start Failover: ENABLED

   Configuration Status:
   SUCCESS

   DGMGRL>
   ```

```
Database - cgms

  Role:          PRIMARY
  Intended State:  TRANSPORT-ON
  Instance(s):
    cgms

Database Status:
SUCCESS

DGMGRL>
Database - cgms_s

  Role:          PHYSICAL STANDBY
  Intended State:  APPLY-ON
  Transport Lag:   0 seconds
  Apply Lag:      0 seconds
  Real Time Query: OFF
  Instance(s):
    cgms_s

Database Status:
SUCCESS
```

データベース HA 用の IoT FND の設定

データベース HA 用に IoT FND を設定するには、次の手順を実行します。

1. IoT FND を停止します。
2. setupCgms.sh スクリプトを実行します。

   このスクリプトでは、データベース設定の変更を求められます。**y** を入力します。次に、スクリプトによって、プライマリ データベース サーバの情報（IP アドレス、ポート、データベース SID）を入力するように求められます。この後、スクリプトによって他のデータベース サーバを追加するように求められます。**y** を入力します。次に、スクリプトによって、次のようにスタンバイ データベース サーバの情報（IP アドレス、ポート、データベース SID）を入力するように求められます。

   (注) IoT FND は常にポート 1522 を使用してデータベースと通信します。ポート 1622 は、データベースがレプリケーションのためだけに使用します。

```
# cd /opt/cgms/bin
# ./setupCgms.sh
09-13-2012 17:10:00 PDT: INFO: ========== CG-NMS Setup Started - 2012-09-13-17-10-00 ==========
09-13-2012 17:10:00 PDT: INFO: Log file: /opt/cgms/bin/../server/cgms/log/cgms_setup.log

Are you sure you want to setup CG-NMS (y/n)? y

09-13-2012 17:10:02 PDT: INFO: User response: y

Do you want to change the database settings (y/n)? y

09-13-2012 17:10:05 PDT: INFO: User response: y

Enter database server IP address [128.107.154.246]: 128.107.154.246
09-13-2012 17:11:02 PDT: INFO: Database server IP: 128.107.154.246

Enter database server port [1522]:
09-13-2012 17:11:07 PDT: INFO: Database server port: 1522

Enter database SID [cgms]:
09-13-2012 17:11:12 PDT: INFO: Database SID: cgms
```

```
Do you wish to configure another database server for this CG-NMS ? (y/n)? y

09-13-2012 17:11:18 PDT: INFO: User response: y
Enter database server IP address []: 128.107.154.20
09-13-2012 17:11:02 PDT: INFO: Database server IP: 128.107.154.20
Enter database server port []: 1522
09-13-2012 17:11:07 PDT: INFO: Database server port: 1522
Enter database SID []: cgms_s
09-13-2012 17:11:12 PDT: INFO: Database SID: cgms_s
09-13-2012 17:11:18 PDT: INFO: Configuring database settings.This may take a while.Please wait ...
09-13-2012 17:11:19 PDT: INFO: Database settings configured.

Do you want to change the database password (y/n)? y

09-13-2012 17:15:07 PDT: INFO: User response: y

Enter database password:
Re-enter database password:

09-13-2012 17:15:31 PDT: INFO: Configuring database password.This may take a while.Please wait ...
09-13-2012 17:15:34 PDT: INFO: Database password configured.

Do you want to change the keystore password (y/n)? n

09-13-2012 17:16:18 PDT: INFO: User response: n

Do you want to change the web application 'root' user password (y/n)? n

09-13-2012 17:16:34 PDT: INFO: User response: n

Do you want to change the FTP settings (y/n)? n

09-13-2012 17:16:45 PDT: INFO: User response: n
09-13-2012 17:16:45 PDT: INFO: ========== CG-NMS Setup Completed Successfully ==========
```

IoT FND データベース HA の無効化

IoT FND データベース HA を無効化するには、次の手順を実行します。

1. オブザーバ プログラムを実行しているサーバで、オブザーバを停止します。

```
$ ./manageObserver.sh stop cgms_s password
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production

Copyright (c) 2000, 2009, Oracle.All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> Connected.
DGMGRL> Done.
$ Observer stopped
```

2. スタンバイ IoT FND データベース サーバで、スタンバイ データベースを削除します。

```
$ ./deleteStandbyDb.sh

Are you sure you want to delete the standby database ? All replicated data will be lost (y/n)? y

09-20-2012 14:27:02 PDT: INFO: User response: y
09-20-2012 14:27:02 PDT: INFO: Cleaning up instance - cgms_s
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production

Copyright (c) 2000, 2009, Oracle.All rights reserved.
```

```
Welcome to DGMGRL, type "help" for information.
DGMGRL> Connected.
DGMGRL> Done.
DGMGRL> DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production

Copyright (c) 2000, 2009, Oracle.All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> Connected.
DGMGRL> Disabled.
DGMGRL> 09-20-2012 14:27:06 PDT: INFO: Removing dataguard configuration
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production

Copyright (c) 2000, 2009, Oracle.All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> Connected.
DGMGRL> Removed configuration
DGMGRL> 09-20-2012 14:27:07 PDT: INFO: Stopping the database

SQL*Plus: Release 11.2.0.3.0 Production on Thu Sep 20 14:27:07 2012

Copyright (c) 1982, 2011, Oracle.All rights reserved.


接続先
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> ORA-01109: database not open


Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 20-SEP-2012 14:27:19

Copyright (c) 1991, 2011, Oracle.All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=test-scale-15krpm)(PORT=1522))(CONNECT_DATA=(SERVER=DEDIC
ATED)(SERVICE_NAME=cgms_s)))
The command completed successfully
Cleaning up instance - cgms_s
09-20-2012 14:27:29 PDT: INFO: ==========  Completed Successfully ==========
```

3. プライマリ IoT FND データベース サーバで、HA 設定を削除します。

```
$ ./deletePrimaryDbHa.sh
Are you sure you want to delete the high availability configuration ? All replicated data will be
lost (y/n)? y

09-20-2012 14:25:25 PDT: INFO: User response: y
09-20-2012 14:25:25 PDT: INFO: Removing secondary configuration from primary

SQL*Plus: Release 11.2.0.3.0 Production on Thu Sep 20 14:25:25 2012

Copyright (c) 1982, 2011, Oracle.All rights reserved.


接続先
```

```
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
System altered.
...
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
09-20-2012 14:25:28 PDT: INFO: Removing data guard config files
09-20-2012 14:25:28 PDT: INFO: Removing standby redo logs
09-20-2012 14:25:29 PDT: INFO: Creating listener file
09-20-2012 14:25:29 PDT: INFO: Listener successfully configured.
09-20-2012 14:25:29 PDT: INFO: Recreating tnsnames ora file
09-20-2012 14:25:29 PDT: INFO: reloading the listener

LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 20-SEP-2012 14:25:29

Copyright (c) 1991, 2011, Oracle.All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=test-scale-15krpm-db2)(PORT=1522)))
The command completed successfully

LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 20-SEP-2012 14:25:30

Copyright (c) 1991, 2011, Oracle.All rights reserved.

Starting /home/oracle/app/oracle/product/11.2.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 11.2.0.3.0 - Production
System parameter file is /home/oracle/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Log messages written to
/home/oracle/app/oracle/diag/tnslsnr/test-scale-15krpm-db2/cgmstns/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=test-scale-15krpm-db2)(PORT=1522)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=test-scale-15krpm-db2)(PORT=1522)))
STATUS of the LISTENER
------------------------
Alias                     cgmstns
Version                   TNSLSNR for Linux: Version 11.2.0.3.0 - Production
Start Date                20-SEP-2012 14:25:30
Uptime                    0 days 0 hr.0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/home/oracle/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Listener Log File
/home/oracle/app/oracle/diag/tnslsnr/test-scale-15krpm-db2/cgmstns/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=test-scale-15krpm-db2)(PORT=1522)))
Services Summary...
Service "cgms" has 1 instance(s).
  Instance "cgms", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
09-20-2012 14:25:30 PDT: INFO: ==========  Completed Successfully ==========
```

ロード バランシング ポリシー

次の表に、LB がサポートするトラフィック タイプごとのロード バランシング ポリシーを示します。

Traffic	ロード バランシング ポリシー
ブラウザおよび IoT FND API クライアント（IPv4：ポート 80 および 443）間の HTTPS トラフィック	LB は Web ブラウザおよび IoT FND API クライアントからのすべてのトラフィックにレイヤ 7 の ロード バランシングを使用します。 LB は一般的な HTTPS トラフィックにでスティッキ性を使用します。
ポート 9121 および 9120 に向かう FAR IPv4 トラフィックの場合： ■ HTTPS を介したポート 9120 でのトンネル プロビジョニング ■ HTTPS を介したポート 9121 での通常の定期的な登録	LB はすべての FAR トラフィックにレイヤ 3 のロード バランシングを使用します。これが FAR から IoT FND へのトラフィックです。
メッシュ エンドポイント（ME）との IPv6 CSMP トラフィックの場合： ■ ポート 61624 を介した UDP トラフィック – 登録 – メトリックの定期的な送信 – ファームウェア プッシュ – 設定転送 ■ ポート 61625 を介した UDP トラフィック ME によって送信される停止通知用。	LB はポート 61624 へのすべての ME トラフィックとポート 61625 への停止メッセージに、レイヤ 3 のロード バランシングを使用します。

LB の実行コンフィギュレーションの例

以下に、適切に設定された IoT FND LB の実行コンフィギュレーションの例を示します。

```
# show running-config
Generating configuration....

ssh maxsessions 10

boot system image:c4710ace-t1k9-mz.A5_1_1.bin

hostname cgnmslb2
interface gigabitEthernet 1/1
  switchport access vlan 10
  no shutdown
interface gigabitEthernet 1/2
  description server-side
  switchport access vlan 11
  no shutdown
interface gigabitEthernet 1/3
  description client-side
  switchport access vlan 8
  no shutdown
interface gigabitEthernet 1/4
```

```
  switchport access vlan 55
  no shutdown



access-list ALL line 8 extended permit ip any any
access-list everyone line 8 extended permit ip any any
access-list everyone line 16 extended permit icmp any any
access-list ipv6_acl line 8 extended permit ip anyv6 anyv6
access-list ipv6_acl2 line 8 extended permit icmpv6 anyv6 anyv6



ip domain-lookup
ip domain-name cisco.com
ip name-server 171.68.226.120
ip name-server 171.70.168.183

probe http probe_cgnms-http
  port 80
  interval 15
  passdetect interval 60
  expect status 200 200
  open 1

rserver host 12-12-1-31
  ip address 12.12.1.31
  conn-limit max 4000000 min 4000000
  probe probe_cgnms-http
  inservice
rserver host 12-12-1-32
  ip address 12.12.1.32
  conn-limit max 4000000 min 4000000
  probe probe_cgnms-http
  inservice
rserver host 2002-cafe-server-202
  description realserver 2002:cafe:server::202
  ip address 2002::202
  conn-limit max 4000000 min 4000000
  probe probe_cgnms-http
  inservice
rserver host 2002-cafe-server-211
  ip address 2002:cafe:server::211
  conn-limit max 4000000 min 4000000
  probe probe_cgnms-http
  inservice

serverfarm host cgnms_2
  description cgnms-serverfarm
  probe probe_cgnms-http
  rserver 2002-cafe-server-202 61624
    conn-limit max 4000000 min 4000000
    inservice
  rserver 2002-cafe-server-211 61624
    conn-limit max 4000000 min 4000000
    inservice
serverfarm host cgnms_2_ipv4
  probe probe_cgnms-http
  rserver 12-12-1-31
    conn-limit max 4000000 min 4000000
    inservice
  rserver 12-12-1-32
    conn-limit max 4000000 min 4000000
    inservice
```

```
sticky ip-netmask 255.255.255.255 address source CGNMS_SRC_STICKY
  serverfarm cgnms_2_ipv4


class-map type management match-any remote_access
  2 match protocol xml-https any
  3 match protocol icmp any
  4 match protocol telnet any
  5 match protocol ssh any
  6 match protocol http any
  7 match protocol https any
  8 match protocol snmp any
class-map type management match-all ssh_allow_access
  2 match protocol ssh any
class-map match-any virtual-server-cgnms
  2 match virtual-address 2002:server:cafe::210 udp eq 61624
class-map match-any vs_cgnms_ipv4
  3 match virtual-address 12.12.1.101 tcp eq https
  4 match virtual-address 12.12.1.101 tcp eq 9120
  5 match virtual-address 12.12.1.101 tcp eq 9121
  6 match virtual-address 12.12.1.101 tcp eq 8443
  7 match virtual-address 12.12.1.101 tcp any

policy-map type management first-match remote_mgmt_allow_policy
  class remote_access
    permit

policy-map type loadbalance first-match virtual_cgnms_l7
  class class-default
    serverfarm cgnms_2
policy-map type loadbalance first-match vs_cgnms_l7_v4
  class class-default
    sticky-serverfarm CGNMS_SRC_STICKY

policy-map multi-match cgnms_policy_ipv6
  class virtual-server-cgnms
    loadbalance vip inservice
    loadbalance policy virtual_cgnms_l7
    loadbalance vip icmp-reply active
policy-map multi-match int1000
  class vs_cgnms_ipv4
    loadbalance vip inservice
    loadbalance policy vs_cgnms_l7_v4
    loadbalance vip icmp-reply active

interface vlan 8
  bridge-group 1
  access-group input everyone
  access-group input ipv6_acl
  no shutdown
interface vlan 10
  bridge-group 2
  access-group input everyone
  access-group input ipv6_acl
  service-policy input int1000
  no shutdown
interface vlan 11
  bridge-group 2
  access-group input everyone
  access-group input ipv6_acl
  no shutdown
interface vlan 55
  bridge-group 1
```

```
  access-group input everyone
  access-group input ipv6_acl
  service-policy input cgnms_policy_ipv6
  no shutdown

interface bvi 1
  ipv6 enable
  ip address 2002:server:cafe::206/64
  no shutdown
interface bvi 2
  ip address 12.12.1.100 255.255.255.0
  no shutdown

domain cisco.com

ip route 2011::/16 2002:server:cafe::101
ip route 2001:server:cafe::/64 2002:cafe::101
ip route 11.1.0.0 255.255.0.0 12.12.1.33
ip route 15.1.0.0 255.255.0.0 12.12.1.33
ip route 13.211.0.0 255.255.0.0 12.12.1.33

context VC_Setup1
  allocate-interface vlan 40
  allocate-interface vlan 50
  allocate-interface vlan 1000

username admin password 5 $1$CB34uAB9$BW8a3ijjxvBGttuGtTcST/  role Admin domain
default-domain
username www password 5 $1$q/YDKDp4$9PkZl1SBMQW7yZ7E.sOZA/  role Admin domain de
fault-domain

ssh key rsa 1024 force
```

トンネル プロビジョニング ポリシーの設定

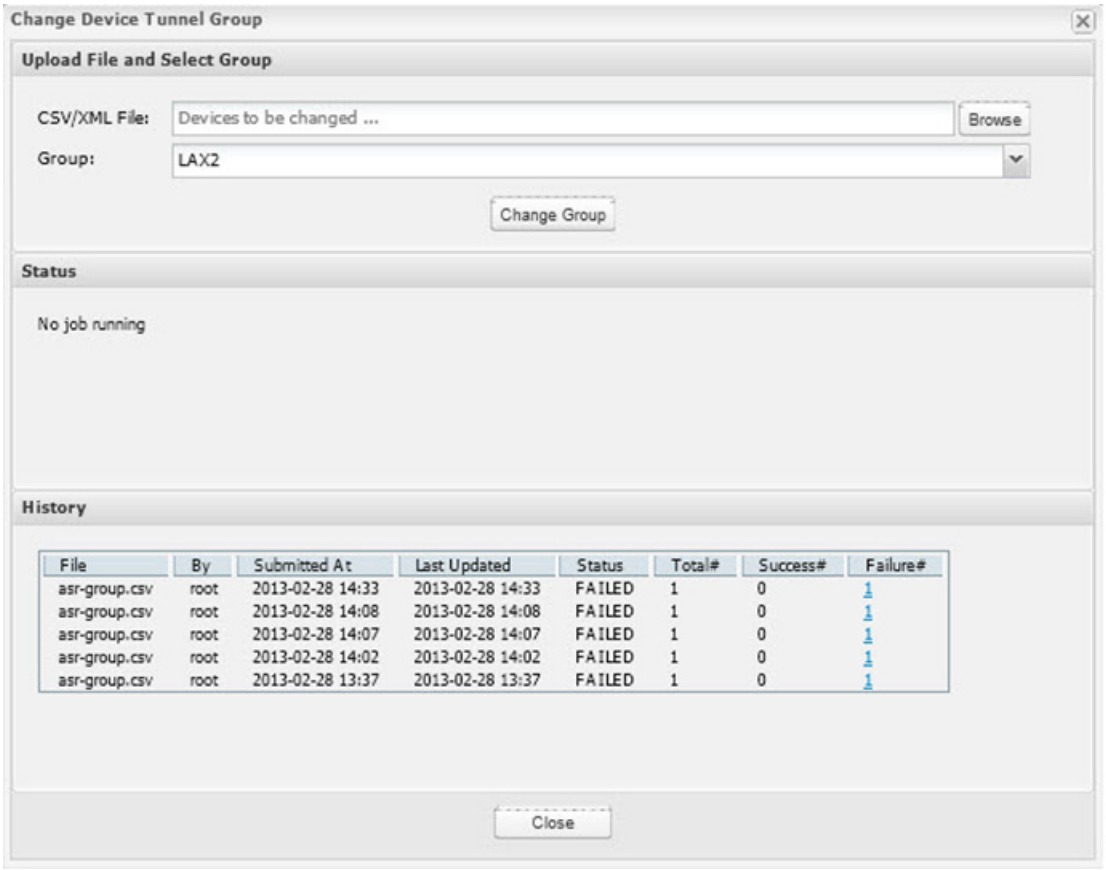
トンネル ポリシーを使用して、FAR に複数のトンネルを設定します。各トンネルは FAR および HER のインターフェイスに関連付けられています。トンネル プロビジョニング グループに 1 つ以上の HER がある場合、IoT FND は [Tunnel Provisioning Policies] タブ（[Config] > [Tunnel Provisioning]）にポリシーを表示します。このポリシーを使用して、FAR と HER 間にインターフェイス マッピングを設定します。

IoT FND で FAR と HER インターフェイスをマッピングするには、次の手順を実行します。

1. [Config] > [Tunnel Provisioning] の順に選択します。
2. [TUNNEL GROUPS] ペインで、トンネルの冗長性を設定するグループを選択します。
3. HER をリストした CSV ファイルまたは XML ファイルを作成して、次のように *EID, device type* の形式でグループに追加します。

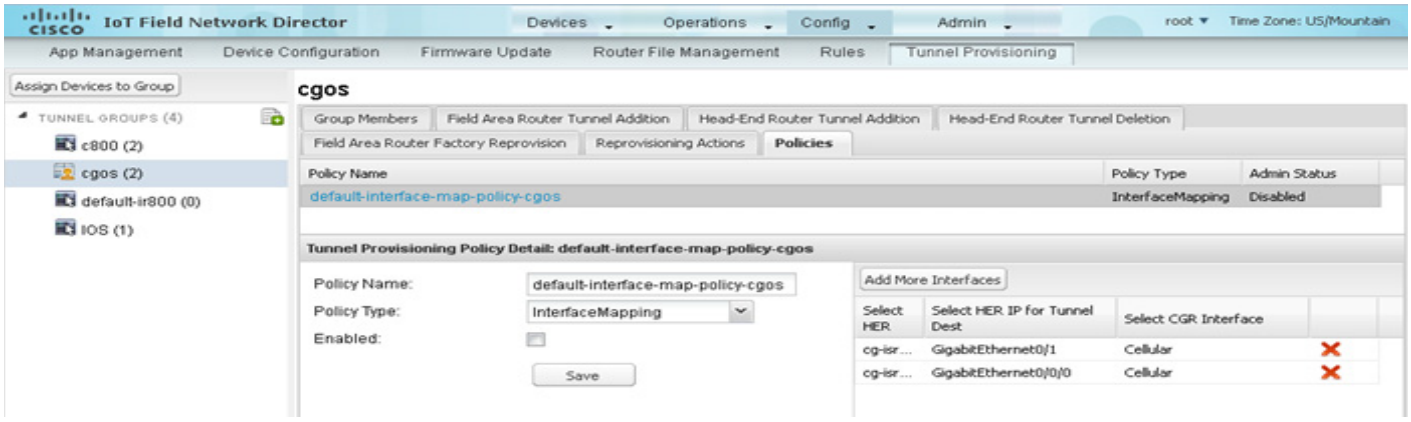
   ```
   eid,deviceType
   asr-0, asr1000
   asr-1, asr1000
   asr-2, asr1000
   ```

4. [Assign Devices to Group] をクリックして、ファイルをインポートして HER をグループに追加します。



(注)HER は複数のトンネル プロビジョニング グループのメンバーになることができます。

5. トンネル プロビジョニング グループを選択し、[Policies] タブをクリックします。

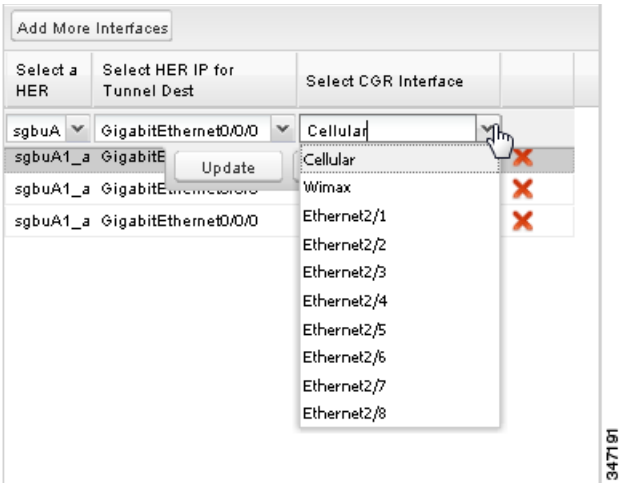


デフォルトでは、IoT FND は InterfaceMapping ポリシーを表示します。

(注)InterfaceMapping は、現在 IoT FND でサポートされている唯一のポリシー タイプです。

IoT FND はグループ内のすべての HER に対して 1 つのインターフェイス マッピング エントリを表示します。インターフェイス マッピング エントリは、必要に応じて追加または削除することができます。

6. [Policy Name] フィールドに、ポリシーの名前を入力します。
7. ポリシーにインターフェイス マッピング エントリを追加するには、[Add More Interfaces] をクリックします。



エントリを削除するには、そのエントリの [Delete]( )をクリックします。

8. インターフェイス マッピング エントリを設定するには、ポリシー名のリンクをクリックし、必要に応じて以下を実行します。
   a. 別の HER を選択するには、現在選択されている HER をクリックして、[Select a HER] ドロップダウン メニューから別の HER を選択します。
   b. HER でトンネル先の HER IP を選択するには、選択されているインターフェイスをクリックして、[Select HER IP] ドロップダウン メニューから別の HER IP を選択します。
   c. 選択した HER インターフェイスにマップする FAR インターフェイスを選択するには、[Select CGR Interface] ドロップダウン メニューからインターフェイスを選択します。
   d. [Update] をクリックします。
9. ポリシーを有効にするには、[Enabled] チェック ボックスをオンにします。
10. [Save(保存)] をクリックします。

トンネル冗長性のためのトンネル プロビジョニング テンプレートの変更

トンネル プロビジョニング グループにトンネル プロビジョニング ポリシーを設定したら、フィールド エリア ルータ トンネル追加テンプレートとヘッドエンド ルータ トンネル追加テンプレートを変更して、ポリシーで定義された複数のトンネルを確立するためのコマンドを含めます。

フィールド エリア ルータ トンネル追加テンプレートの例

この例では、太字は、複数のトンネルを作成するために行ったデフォルトのフィールド エリア ルータ トンネル追加テンプレートへの変更を示しています。

```
<#--
  Configure a Loopback0 interface for the FAR.This is done first as features
  look for this interface and use it as a source.

  This is independent of policies
-->
interface Loopback0
  <#--
```

```
    Now obtain an IPv4 address that can be used to for this FAR's Loopback
    interface.The template API provides methods for requesting a lease from
    a DHCP server.The IPv4 address method requires a DHCP client ID and a link
    address to send in the DHCP request.The 3rd parameter is optional and
    defaults to "CG-NMS".This value is sent in the DHCP user class option.
    The API also provides the method "dhcpClientId".This method takes a DHCPv6
    Identity association identifier (IAID) and a DHCP Unique  IDentifier (DUID)
    and generates a DHCPv4 client identifier as specified in RFC 4361.This
    provides some consistency in how network elements are identified by the
    DHCP server.
  -->
  ip address ${far.ipv4Address(dhcpClientId(far.enDuid, 0), far.dhcpV4LoopbackLink).address}/32
  <#--
    Now obtain an IPv6 address that can be used to for this FAR's loopback
    interface.The method is similar to the one used for IPv4, except clients
    in DHCPv6 are directly identified by their DUID and IAID.IAIDs used for
    IPv4 are separate from IAIDs used for IPv6, so we can use zero for both
    requests.
  -->
  ipv6 address ${far.ipv6Address(far.enDuid, 0, far.dhcpV6LoopbackLink).address}/128
exit

<#-- Make certain the required features are enabled on the FAR.-->
feature crypto ike
feature ospf
feature ospfv3
feature tunnel
<#-- Features ike and tunnel must be enabled before ipsec.-->
feature crypto ipsec virtual-tunnel

<#--
  Toggle on/off the c1222r feature to be certain it uses the Loopback0
  interface as its source IP.
-->
no feature c1222r
feature c1222r

<#-- Configure Open Shortest Path First routing processes for IPv4 and IPv6.-->
router ospf 1
exit
router ospfv3 2
exit

<#--
  Now that OSPF has been configured complete the configuration of Loopback0.
-->
interface Loopback0
  ip router ospf 1 area ${far.ospfArea1!"1"}
  ipv6 router ospfv3 2 area ${far.ospfV3Area1!"0"}
exit

<#-- Configure Internet Key Exchange for use by the IPsec tunnel(s).-->
crypto ike domain ipsec
  identity hostname
  policy 1
    <#-- Use RSA signatures for the authentication method.-->
    authentication rsa-sig
    <#-- Use the 1536-bit modular exponential group.-->
    group 5
  exit
exit
crypto ipsec transform-set IPSecTransformSet esp-aes 128 esp-sha1-hmac
crypto ipsec profile IPSecProfile
```

```
  set transform-set IPSecTransformSet
exit

<#--
  Define template variables to keep track of the next available IAID (IPv4)
  and the next available tunnel interface number.We used zero when leasing
  addresses for Loopback0, so start the IAID at one.
-->
<#assign iaId = 1>
<#assign interfaceNumber = 0>

<#--
  The same logic is needed for each of the IPsec tunnels, so a macro is used
  to avoid duplicating configuration.The first parameter is the prefix to
  use when looking for the WAN interface on the FAR to use for the source of
  the tunnel.The second parameter is the OSPF cost to assign to the tunnel.
-->
<#macro configureTunnel interfaceNamePrefix destinationInterface her tunnelIndex ospfCost>
  <#--
    If an interface exists on the FAR whose name starts with the given prefix
    and an IPv4 address as been assigned to that interface then the IPsec
    tunnel can be configured, otherwise no tunnel will be configured.The
    template API interfaces method will return all interfaces whose name
    starts with the given prefix.
  -->
  <#assign wanInterface = far.interfaces(interfaceNamePrefix)>
  <#-- Check if an interface was found and it has an IPv4 address.-->
  <#if (wanInterface[0].v4.addresses[0].address)??>
    <#--
      Determine the HER destination address to use when configuring the tunnel.
      If the optional property "ipsecTunnelDestAddr1" has been set on this FAR
      then use the value of that property.Otherwise look for that same property
      on the HER.If the property is not set on the FAR or the HER, then fallback
      to using an address on the HER GigabitEthernet0/0/0 interface.
    -->
    <#assign destinationAddress = her.interfaces(destinationInterface)[0].v4.addresses[0].address>

    <#if !(destinationAddress??)>
      ${provisioningFailed("Unable to determine the destination address for IPsec tunnels")}
    </#if>
    interface Tunnel${interfaceNumber}
      <#assign interfaceNumber = interfaceNumber + 1>
      description IPsec tunnel to ${her.eid}
      <#--
        For a tunnel interface two addresses in their own tiny subnet are
        needed.The template API provides an ipv4Subnet method for leasing an
        IPv4 from a DHCP server.The parameters match those of ipv4Address,
        with a fourth optional parameter that can be used to specify the
        prefix length of the subnet to request.If not specified the prefix
        length requested will default to 31, which provides the two addresses
        needed for a point to point link.

        NOTE: If the DHCP server being used does not support leasing an IPv4
        subnet, then this call will have to be changed to use the ipv4Address
        method and the DHCP server will have to be configured to respond
        appropriately to the request made here and the second request that
        will have to be made when configuring the HER side of the tunnel.
        That may require configuring the DHCP server with reserved addresses
        for the client identifiers used in the calls.
      -->
      <#assign lease = far.ipv4Subnet(dhcpClientId(far.enDuid, tunnelIndex), far.dhcpV4TunnelLink)>
      <#assign iaId = iaId + 1>
      <#-- Use the second address in the subnet for this side of the tunnel.-->
      ip address ${lease.secondAddress}/${lease.prefixLength}
      ip ospf cost ${ospfCost}
```

```
      ip ospf mtu-ignore
      ip router ospf 1 area ${far.ospfArea1!"1"}
      tunnel destination ${destinationAddress}
      tunnel mode ipsec ipv4
      tunnel protection ipsec profile IPSecProfile
      tunnel source ${wanInterface[0].name}
      no shutdown
    exit
  </#if>
</#macro>

<#--
  Since we are doing policies for each tunnel here, the list of policies passed to this template can be
  iterated over to get the tunnel configuration viz interface mapping

  tunnelObject.ipSecTunnelDestInterface is the "interface on CGR"
  tunnelObject.ipSecTunnelSrcInterface is the "interface on HER"
  tunnelObject.her is the HER of interest
 -->

<#list far.tunnels("ipSec") as tunnelObject>
    <@configureTunnel tunnelObject.ipSecTunnelDestInterface tunnelObject.ipSecTunnelSrcInterface
tunnelObject.her tunnelObject.tunnelIndex 100/>  <----- Loop through policies (aka Tunnels)
</#list>

<#--
  Make certain provisioning fails if we were unable to configure any IPsec
  tunnels.For example this could happen if the interface properties are
  set incorrectly.
-->
<#if iaId = 1>
  ${provisioningFailed("Did not find any WAN interfaces to use as the source for IPsec tunnels")}
</#if>

<#--
  Configure an IPv6-in-IPv4 GRE tunnel to allow IPv6 traffic to reach the data
  center.
-->
<#macro configureGreTunnel destinationInterface her tunnelIndex>

<#assign destinationAddress = her.interfaces(destinationInterface)[0].v4.addresses[0].address>

<#if !(destinationAddress??)>
     ${provisioningFailed("Unable to determine the destination address for GRE tunnels")}
</#if>

interface Tunnel${interfaceNumber}
  <#assign interfaceNumber = interfaceNumber + 1>
  description GRE IPv6 tunnel to ${her.eid}
  <#--
    The ipv6Subnet method is similar to the ipv4Subnet method except instead
    of obtaining an IPv4 subnet it uses DHCPv6 prefix delegation to obtain an
    IPv6 prefix.The prefix length will default to 127, providing the two
    addresses needed for the point to point link.For the IAID, zero was used
    when requesting an IPv6 address for loopback0, so use one in this request.
  -->
  <#assign lease = far.ipv6Subnet(far.enDuid, tunnelIndex, far.dhcpV6TunnelLink)>
  ipv6 address ${lease.secondAddress}/${lease.prefixLength}
  ipv6 router ospfv3 2 area ${far.ospfV3Area1!"0"}
  ospfv3 mtu-ignore
  tunnel destination ${destinationAddress}
  tunnel mode gre ip
  tunnel source Loopback0
```

```
  no shutdown
exit

</#macro>

<#-- Loop through the policies for GRE tunnels -->
<#list far.tunnels("gre") as greTunnelObj>
    <@configureGreTunnel greTunnelObj.greDestInterface greTunnelObj.her greTunnelObj.tunnelIndex/>
</#list>
```

ヘッドエンド ルータ トンネル追加テンプレート

この例では、太字は、複数のトンネルを作成するために行ったデフォルトのヘッドエンド ルータ トンネル追加テンプレートへの変更を示しています。

```
<#--
  Define template variables to keep track of the IAID (IPv4) that was used by
  the FAR template when configuring the other end of the tunnel.This template
  must use the same IAID in order to locate the same subnet that was leased by
  the FAR template so both endpoints are in the matching subnet.
-->
<#assign iaId = 1>

<#--
  The same logic is needed for each of the IPsec tunnels, so a macro is used.
-->
<#macro configureTunnel ipSecTunnelSrcInterface ipSecTunnelDestInterface her tunnelIndex ospfCost>
  <#--
    Only configure the HER tunnel end point if the FAR tunnel end point was
    configured.This must match the corresponding logic in the FAR tunnel
    template.The tunnel will not have been configured if the WAN interface
    does not exist on the FAR or does not have an address assigned to it.
  -->
  <#assign wanInterface = far.interfaces(ipSecTunnelDestInterface)>
  <#if (wanInterface[0].v4.addresses[0].address)??>
    <#-- Obtain the full interface name based on the prefix.-->
    <#assign interfaceName = wanInterface[0].name>
    <#--
      Locate a tunnel interface on the HER that is not in use.The template
      API provides an unusedInterfaceNumber method for this purpose.All of
      the parameters are optional.The first parameter is a name prefix
      identifying the type of interfaces, it defaults to "tunnel".The second
      parameter is a lower bound on the range the unused interface number must
      be in, it defaults to zero.The third parameter is the upper bound on
      the range, it defaults to max integer (signed).The method remembers
      the unused interface numbers it has returned while the template is
      being processed and excludes previously returned numbers.If no unused
      interface number meets the constraints an exception will be thrown.
    -->
    interface Tunnel${her.unusedInterfaceNumber()}
      description IPsec tunnel to ${far.eid}
      <#assign lease = far.ipv4Subnet(dhcpClientId(far.enDuid, tunnelIndex), far.dhcpV4TunnelLink)>
      <#assign iaId = iaId + 1>
      ip address ${lease.firstAddress} ${lease.subnetMask}
      ip ospf cost ${ospfCost}
      ip ospf mtu-ignore
      tunnel destination ${wanInterface[0].v4.addresses[0].address}
      tunnel mode ipsec ipv4
      tunnel protection ipsec profile IPSecProfile
      tunnel source ${ipSecTunnelSrcInterface}
      no shutdown
    exit
    router ospf 1
```

```
      network ${lease.prefix} ${lease.wildcardMask} area ${far.ospfArea1!"1"}
    exit
  </#if>
</#macro>

<#list far.tunnels("ipSec") as tunnelObject>
        <@configureTunnel tunnelObject.ipSecTunnelSrcInterface tunnelObject.ipSecTunnelDestInterface
tunnelObject.her tunnelObject.tunnelIndex 100/>
</#list>

<#--
  Configure an IPv6-in-IPv4 GRE tunnel to allow IPv6 traffic to reach the data
  center.
-->
<#macro configureGreTunnel greSrcInterface her tunnelIndex>
interface Tunnel${her.unusedInterfaceNumber()}
  description GRE IPv6 tunnel to ${far.eid}
  <#assign lease = far.ipv6Subnet(far.enDuid, tunnelIndex, far.dhcpV6TunnelLink)>
  ipv6 address ${lease.firstAddress}/${lease.prefixLength}
  ipv6 enable
  ipv6 ospf 2 area ${far.ospfV3Area1!"0"}
  ipv6 ospf mtu-ignore
  tunnel destination ${far.interfaces("Loopback0")[0].v4.addresses[0].address}
  tunnel mode gre ip
  tunnel source ${greSrcInterface}
exit
</#macro>

<#-- Loop through the policies for GRE tunnels -->
<#list far.tunnels("gre") as greTunnelObj>
        <@configureGreTunnel greTunnelObj.greSrcInterface greTunnelObj.her greTunnelObj.tunnelIndex/>
</#list>
```