



インストール

- [OpenStack Platform 16.2 Director を使用した OpenStack 対応 Cisco ACI \(1 ページ\)](#)
- [OSP Director を使用した OpenStack での Cisco ACI の要件および前提条件 \(2 ページ\)](#)
- [関連資料 \(3 ページ\)](#)
- [OpFlex の展開 \(3 ページ\)](#)
- [OpenStack 向け Cisco ACI のインストール準備 \(4 ページ\)](#)
- [オーバークラウドのインストール \(9 ページ\)](#)

OpenStack Platform 16.2 Director を使用した OpenStack 対応 Cisco ACI

Cisco Application Centric Infrastructure (ACI) は、インテリジェントなコントローラベースのネットワークスイッチングファブリックを提供する包括的なポリシーベースのアーキテクチャです。このファブリックは、OpenStack を含む複数のオーケストレーション ツール、自動化 ツール、および管理ツールに直接統合可能な API インターフェイスを通じてプログラムすることにより管理する設計となっています。Cisco ACI を OpenStack と統合することによって、ネットワーク構造体の動的な作成を OpenStack 要件に従って直接駆動するだけでなく、Cisco Application Policy Infrastructure Controller (APIC) 内のさらなる可視性を個別の仮想マシン (VM) インスタンスのレベルに至るまで実現できます。

OpenStack は、クラウドコンピューティング環境を構築するための柔軟なソフトウェアアーキテクチャを明確にします。OpenStack のリファレンスソフトウェアベースの実装では、VLAN、GRE、VXLAN を含む複数のレイヤ 2 トランスポートが考慮されています。OpenStack 内の Neutron プロジェクトは、ソフトウェアベースのレイヤ 3 転送も提供できます。Cisco ACI と ACI OpenStack Unified ML2 プラグインを組み合わせると、統合されたレイヤ 2 およびレイヤ 3 VXLAN ベースのオーバーレイ ネットワーキング機能が提供されます。このアーキテクチャは、ハードウェアベースのネットワークのパフォーマンスと運用上の利点に加えて、ソフトウェア オーバーレイ ネットワーキングの柔軟性を提供します。

Cisco ACI OpenStack プラグインは、ML2 モードまたは GBP モードで使用できます。モジュラ レイヤ 2 (ML2) モードでは、ネットワークの作成に標準 Neutron API が使用されます。これは OpenStack に VM およびサービスを導入するための従来の方法です。グループ ベース ポリ

シー（GBP）モードでは、アプリケーションをポリシーグループとして説明、作成、展開する新しいAPIが提供されます。これにより、ネットワーク固有の詳細を気にする必要がなくなります。単一の OpenStack プロジェクトで GBP と Neutron API を混在させることはサポートされていないことに注意してください。詳細については、次の URL にある『*OpenStack Group-Based Policy User Guide*』を参照してください。

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/openstack/b_OpenStack_Group-Based_Policy_User_Guide.html

OSP Director を使用した OpenStack での Cisco ACI の要件および前提条件

- 対象者：Linux、Red Hat OpenStack ディストリビューション、Cisco Application Centric Infrastructure（ACI）ポリシーモデル、および Cisco Application Policy Infrastructure Controller（APIC）構成に関する実践的な知識が必要です。また、OpenStack のアーキテクチャと展開に精通している必要があります。
- Cisco ACI ファブリック：『[Cisco ACI Virtualization Compatibility Matrix](#)』に記載されている最小サポートバージョンでインストールおよび初期化された Cisco ACI ファブリックが必要です。



(注) 複数のリーフペア間の通信の場合、ファブリックには、OpenStack 外部ネットワークを使用できるように有効になっている BGP ルートリフレクタが必要です。

- 仮想ポートチャネル（vPC）でボンディングされたファブリックインターフェイスを使用する場合、ファブリックインターフェイスの ovs_bond の追加はサポートされていません。これは、Open vSwitch（OVS）ブリッジに単一のインターフェイスとして追加する必要があるためです。ファブリックインターフェイスを集約するには、タイプを linux_bond に設定する必要があります。nic-config テンプレートでファブリックインターフェイスを作成する方法の簡単な例を示します。

```
type: ovs_bridge
  name: {get_input: bridge_name}
  mtu: 1500
  members:
    -
      type: linux_bond
      name: bond1
      ovs_options: {get_param: BondInterfaceOvsOptions}
      mtu: 1600
      members:
        -
          type: interface
          name: nic1
          primary: true
          mtu: 1600
        -
```

```
type: interface
name: nic2
mtu: 1600
```

- ボンディングを使用する際にサポートされるのは 802.3ad のみです。
- UCS-B シリーズを使用して展開する場合、冗長性を確保するために、ボンディングを備えたデュアル vNIC のみがファブリック インターフェイスでサポートされます。



(注) ハードウェア フェールオーバーで単一の vNIC を使用しないでください。

- Cisco APIC GUI で、ファブリックの OpFlex 認証を無効にします。[システム (System)] > [システム設定 (System Settings)] > [ファブリック幅設定 (Fabric Wide Setting)] > [ファブリック幅設定ポリシー (Fabric Wide Setting Policy)] ウィンドウで、[GOLF と Linux に対して Opflex クライアント証明書の認証を施行する (To enforce Opflex client certificate authentication for GOLF and Linux)] がオフになっていることを確認します。
- オーバークラウド Heat スタックを削除すると、オーバークラウドノードは解放されますが、仮想マシンマネージャ (VMM) ドメインは Cisco APICに残ります。VMM ドメインを手動で削除しない限り、VMMはテナントとともに古いVMM ドメインとしてCisco APICに表示されます。

VMM ドメインを削除する前に、スタックがアンダークラウドから削除されていることを確認し、VMM ドメインの下に表示されるハイパーバイザが接続状態でなくなっていることを確認します。これらの条件の両方が満たされたら、Cisco APIC から VMM ドメインを安全に削除できます。

関連資料

詳細については、Red Hat Web サイトにある『*Director Installation and Usage Red Hat OpenStack Platform 16.2*』ドキュメントを参照してください。

OpFlex の展開

このセクションでは、Red Hat OpenStack ディストリビューションに Cisco Application Centric Infrastructure (ACI) OpenStack プラグインをインストールして構成する方法について説明します。

これらの手順の例は、Red Hat OpenStack の OpenStack Platform 16.2 リリースで検証されています。OpenStack システムは、インストール方法によって大きく異なる可能性があります。したがって、ここで示した例は、特定のインストール状態に適応するための基本として使用してください。

Red Hat OpenStack Platform Director のインストール マニュアルに従って、OpenStack Platform Director を用意し、適切な導入ファイルとリソース ファイルを作成してください。

詳細については、このガイドの「[関連資料 \(3 ページ\)](#)」を参照してください。

OpenStack 向け Cisco ACI のインストール準備

Cisco APIC とネットワークの設定

ここでは、Cisco Application Policy Infrastructure Controller (APIC) およびネットワークのセットアップ方法について説明します。

次の図に示すようなネットワーク レイアウトについては、OpenStack Platform Director ドキュメントの「Network Planning」の項を参照してください。詳細については、Red Hat Web サイトにある『*Director Installation and Usage Red Hat OpenStack Platform*』ドキュメントを参照してください。

図 1: 一般的な OpenStack Platform トポロジ

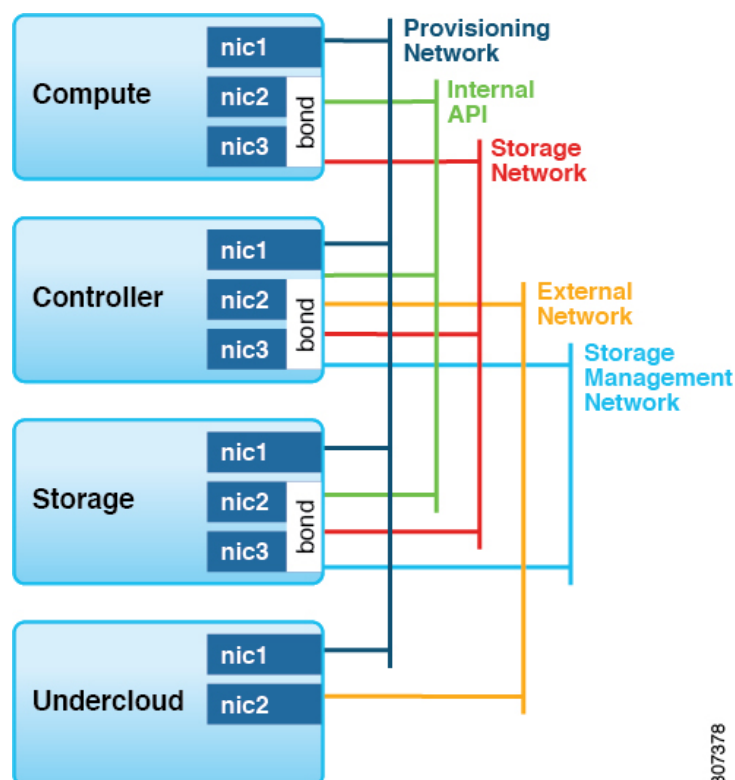
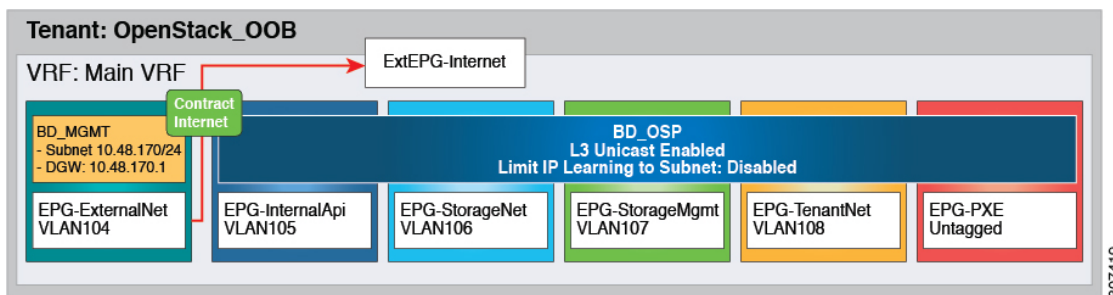


図 2: Cisco ACI プラグインを使用した Red Hat OpenStack Platform 16.2 のインストールの一般的なトポロジ



- PXE ネットワークはネイティブ VLAN を使用する必要があります。ネイティブ VLAN は通常、OpenStack ノードの専用 NIC として定義されるため、PXE ネットワーク インターフェイスを Cisco Application Centric Infrastructure (ACI) ファブリックまたは別のスイッチング ファブリックに接続できます。
- PXE を除くすべての OpenStack Platform (OSP) ネットワークは、Cisco ACI を介するインバンド (IB) です。次の VLAN はその例です。
 - API : VLAN 10
 - ストレージ : VLAN 11
 - StorageMgmt : VLAN 12
 - テナント : VLAN 13
 - 外部 : VLAN 14
 - Cisco ACI インフラ : VLAN 4093
- この例で使用される ExtEPG-Internet は、OpenStack 外部ネットワークのインターネットへの接続を可能にする L3Out 外部 EPG です。また、要件に応じて、内部 API OpenStack ネットワークに外部接続を提供する必要がある場合もあります。

インバンド構成の Cisco ACI を準備するには、物理ドメインと、これらのネットワーク用に作成された EPG への静的バインドを使用できます。これにより、必要な物理ドメインと接続可能アクセス エンティティ プロファイル (AEP) が作成されます。AEP でインフラ VLAN を有効にする必要があることに注意してください。詳細については、ナレッジベースの記事「[特定のポートに EPG を導入するためのドメイン、接続エンティティ プロファイル、および VLAN の作成](#)」を参照してください。

手順

ステップ 1 Cisco APIC GUI にログインし、OpenStack プラットフォームのインストールに必要な VLAN の VLAN プールを作成します。

- メニュー バーで **[ファブリック (Fabric)]** > **[アクセス ポリシー (Access Policies)]** > **[プール (Pools)]** を選択し、**[VLAN]** を右クリックして VLAN プールを作成します。

- b) **[名前 (Name)]** フィールドに、ポリシー名を入力します。（たとえば、OSP16.2-infra）。
- c) （任意）**[Description]** フィールドに、VLAN 範囲の名前空間ポリシーの説明を入力します。
- d) **[Encap Blocks]** セクションで **[+]** アイコンをクリックし、**encap** ブロックの範囲を入力します。
- e) **[送信 (Submit)]** をクリックします。

ステップ 2 接続可能エンティティ プロファイルを作成して上記の物理ドメインを割り当てます。また、**[Enable Infra VLAN]** が選択されていることを確認します。

- a) メニュー バーで、**[ファブリック (Fabric)]** > **[アクセス ポリシー (Access Policies)]** > **[グローバル ポリシー (Global Policyies)]** を選択し、**[接続可能なアクセス エンティティ プロファイル (Attachable Access Entity Profiles)]** を右クリックして、接続可能なアクセス エンティティ プロファイルを作成します。
- b) **[名前 (Name)]** フィールドに、接続可能アクセス エンティティ プロファイルの名前を入力します。（たとえば、OSP16.2-AEP）。
- c) （任意）**[Description]** フィールドに、接続可能アクセス エンティティ プロファイルの説明を入力します。
- d) **[Enable Infrastructure VLAN]** チェックボックスをオンにしてインフラストラクチャ VLAN を有効にします。
- e) **[Domains (VMM, Physical or External) To Be Associated To Interfaces:]** セクションで **[+]** アイコンをクリックし、ドロップダウンリストからドメインプロファイルを選択して **[Update]** をクリックします。
- f) **[次へ (Next)]** をクリックします。
- g) **[完了 (Finish)]** をクリックします。

ステップ 3 物理ドメインを作成して VLAN プールを割り当てます。

- a) メニュー バーで、**[ファブリック (Fabric)]** > **[アクセスポリシー (Access Policies)]** > **[物理ドメインと外部ドメイン (Physical and External Domains)]** を選択し、**[物理ドメイン (Physical Domains)]** を右クリックして、物理ドメインを作成します。
- b) **[名前 (Name)]** フィールドに、物理ドメインの名前を入力します。（OSP16.2-phys）。
- c) **[Associated Attachable Entity Profile]** フィールドで、関連付けられた接続可能エンティティ プロファイルを選択します。
- d) **[VLAN Pool]** フィールドで VLAN プール（**[OSP16.2-infra-dynamic]**）を選択します。

OpenStack ノードと Cisco ACI リーフ スイッチ間のカプセル化方式として VLAN を使用する場合は、OpenStack Neutron ネットワークから使用されるプールに応じて VLAN プール範囲を選択する必要があります。

- e) **[送信 (Submit)]** をクリックします。

ステップ 4 別のテナントで、**[Common]** を使用してアプリケーションプロファイルを作成することもできます。（OSP-16.2 など）。OSP ネットワークの EPG、ブリッジドメイン、および VRF を作成します。PXE ネットワークも Cisco ACI にアクセスする場合は、PXE の EPG と BD も作成します（この例では省略します）。

ステップ 5 必要な VLAN のスタティック バインディング（パス）を追加します。**[Static Binding Paths]** を表示するには、EPG を展開する必要があります。

- a) 作成した物理ドメインがこの EPG に接続されていることを確認します。[アプリケーション プロファイル (Application Profiles)] > [EPG] > [EPG_name] > [Domains] を使用して物理ドメインを追加できます。
- b) メニューバーで、[テナント (Tenants)] > [テナント共通 (Tenant common)] > [アプリケーション プロファイル (Application Profiles)] > [ACI-OSP16.2 > Application EPGs] > [EPG API] > [静的バインド パス (Static Binding Paths)] を選択します。

ステップ 6 EPG に物理ドメインが接続されていることを確認します。

- (注) Cisco ACI を前述のネットワーク用にプロビジョニングする必要があります (テナント、外部、およびフローティング IP ネットワークを除く)。これには、必要な物理ドメインと接続エンティティプロファイルの作成が含まれます。接続エンティティプロファイルでインフラ VLAN を有効にする必要があることに注意してください。

Cisco ACI の OpenStack 導入に対する準備はこれで完了です。

オーバークラウドの設定

『*Director Installation and Usage, Red Hat OpenStack Platform 16.2*』ドキュメントに従って、OpenStack Platform 16.2 Director を準備し、正しい展開ファイルとリソースファイルを作成する必要があります。

詳細については、Red Hat の Web サイトにあるドキュメントを参照してください。第 5 章「コンテナイメージソースの構成」に従う場合は、レジストリ アドレスを書き留めてください。Red Hat のマニュアルに従って、必要に応じてカスタム NIC テンプレートを準備する必要があります。

OpenStack Platform Director を設定したら、展開を続行する前に Cisco Application Centric Infrastructure (ACI) TripleO オークストレーションをインストールする必要があります。

OpFlex オークストレーションを備えた Cisco ACI のアンダークラウドの準備

このセクションでは、Cisco Application Centric Infrastructure (ACI) と OpFlex オークストレーションの統合パッケージをインストールする方法について説明します。

手順

ステップ 1 ユーザ stack としてアンダークラウドにログインします。

ステップ 2 Cisco ACI OSP (tripleo-ciscoaci-16) RPM 5.1.3 以降および対応するプラグイン tarball (openstack-ciscorpms-repo-16) を Cisco.com からダウンロードし、OpenStack Platform Director に配置します。

ステップ 3 RPM をインストールします。これによって依存関係がインストールされます。

rpm コマンドを使用して RPM をインストールする場合、一部の依存関係は手動によるインストールが必要な場合があります。

例：

```
$ sudo yum --nogpgcheck localinstall <rpm file>
```

ステップ 4 次の手順を実行して、Cisco ACI コンテナを作成します。

- 次のコマンドを実行します。 **sudo podman loginregistry.connect.redhat.com**
- プロンプトが表示されたら、Red Hat のログイン情報を使用して、redhat のユーザー名とパスワードを入力します。
- ログインしたら、ルートとして次のスクリプトを実行して Cisco ACI コンテナを作成します。

```
/opt/ciscoaci-tripleo-heat-templates/tools/build_openstack_aci_containers.py
```

- ダウンロードしたプラグイン tarball をスクリプトで指定します。

例：

```
/opt/ciscoaci-tripleo-heat-templates/tools/build_openstack_aci_containers.py -z  
/home/stack/openstack-ciscorpms-repo-16-778.tar.gz
```

コマンドは、アップストリームの Red Hat 認定 ACI コンテナ イメージをプルし、ローカル コンテナ リポジトリにプッシュしま

す。/home/stack/templates/ciscoaci_containers.yaml という名前の環境ファイルを作成します。これは、オーバークラウドの展開時にテンプレートとして含める必要があります。-o オプションを使用すると、出力ファイル名を上書きできます。指定したとおり出力ファイルが作成されたことを確認します。

注

- コンテナをローカルに構築するには、「-p」オプションを省略できます。ただし、これらのコンテナは Red Hat 認定ではない場合があります。コンテナを構築する前に、**registry.redhat.io** にログインする必要があります。
- ローカル コンテナ作成コマンドの実行中に、コマンド **/bin/gbp-db-manage** によって生成されたエラーが表示される場合があります。このエラーは無視しても問題ありません。スクリプトの実行が失敗することはありません。
- OpenStack Director 16.2 の展開では、Docker レジストリの設定がサポートされています。レジストリについては、次の選択肢があります。
 - アップストリーム レジストリ（ローカル サテライト サーバー（現在は Red Hat レジストリ）を使用できます）
 - ダウンストリーム レジストリ アドレス/ポート/URI（現在はアンダーレイ コントローラ、8787、/rhosp16.2）

Docker レジストリは、build_openstack_aci_containers.py スクリプトを使用して構成されます。

```
usage: build_openstack_aci_containers.py [-h] [-u UCLLOUD_IP] [-o OUTPUT_FILE]  
[-c CONTAINERS_TB]  
[-s UPSTREAM_REGISTRY]  
[-d DESTINATION_REGISTRY]  
[-r REGSEPARATOR] [-i RELEASE_TAG]  
[-t TAG]
```

```

...]]]
                                [-a [ADDITIONAL_REPOS [ADDITIONAL_REPOS
                                [--force] [-p] (-f FILE | -z FILE)

Build containers for ACI Plugin

optional arguments:
  -h, --help                show this help message and exit
  -u UCLOUD_IP, --ucloud_ip UCLOUD_IP
                           Undercloud ip address
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                           Environment file to create, default is
                           /home/stack/templates/ciscoaci_containers.yaml
  -c CONTAINERS_TB, --container CONTAINERS_TB
                           Containers to build, comma separated, default is all
  -s UPSTREAM_REGISTRY, --upstream UPSTREAM_REGISTRY
                           Upstream registry to pull base images from, eg.
                           registry.access.redhat.com/rhosp13, defaults to
                           registry.access.redhat.com/rhosp13
  -d DESTINATION_REGISTRY, --destregistry DESTINATION_REGISTRY
                           Destination registry to push to, eg:
                           1.100.1.1:8787/rhosp13
  -r REGSEPARATOR, --regseparator REGSEPARATOR
                           Upstream registry separator for images, eg. '/' for
                           normal upstream registries (default). Will be added
                           between upstream registry name and container name. Use
                           '_' for satellite based registries.
  -i RELEASE_TAG, --image-tag RELEASE_TAG
                           Upstream release tag for images, defaults to 16.2
  -t TAG, --tag TAG          tag for images, defaults to current timestamp
  -a [ADDITIONAL_REPOS [ADDITIONAL_REPOS ...]], --additional-repos
[ADDITIONAL_REPOS [ADDITIONAL_REPOS ...]]
                           Additional repos to use when building containers
                           (defaults to empty list). Use with
                           'rhel-8-for-x86_64-baseos-eus-rpms
                           rhel-8-for-x86_64-appstream-eus-rpms' when using
                           satellite.
  --force                    Override check for md5sum mismatch
  -p, --pull                 Pull upstream containers instead of building locally
  -f FILE, --aci_repo_file FILE
                           Path to yum repository file, which describes the
                           repository which provides ACI plugin rpm files. If you
                           want this script to create a repository on undercloud,
                           please use the -z option to provide path to openstack-
                           aci-rpms-repo tar file downloaded from cisco website
  -z FILE, --aci_rpm_repo_tar_file FILE
                           Path to openstack-aci-rpms-repo tar file. This will be
                           use to create a local yum repository on undercloud

```

オーバークラウドのインストール

ここでは、オーバークラウドのインストール方法について説明します。

手順

ステップ 1 /usr/share/openstack-tripleo-heat-templates/roles_data.yaml ファイルをプライベートな場所にコピーします。

例 :

```
cp /usr/share/openstack-tripleo-heat-templates/roles_data.yaml
/home/stack/templates/aci_roles_data.yaml
```

ステップ 2 roles_data.yaml (aci_roles_data.yaml) のローカルコピーを編集して、CiscoAciAIM と CiscoAciLldp サービスをコントローラ ロールに、CiscoAciLldp サービスをコンピューティング ロールに追加します。

a) コントローラ ロールの下に、次の行を追加します。

```
- OS::TripleO::Services::CiscoAciAIM
- OS::TripleO::Services::CiscoAciLldp
- OS::TripleO::Services::CiscoAciOpflexAgent
```

b) コンピューティング ロールの下に、次の行を追加します。

```
- OS::TripleO::Services::CiscoAciLldp
- OS::TripleO::Services::CiscoAciOpflexAgent
```

アップストリームの /usr/share/openstack-tripleo-heat-templates/roles_data.yaml ファイルを変更してこれらのロールを追加する **ansible** プレイブックが提供されています。上記の手順をスキップし、代わりに `ansible-playbook -i ~/inventory.yaml /opt/ciscoaci-tripleo-heat-templates/tools/generate_ciscoaci_role_data.yaml` コマンドを使用してプレイブックを実行できます。

ステップ 3 Cisco Application Centric Infrastructure (ACI) 環境のリソースを宣言します。

展開に含める .yaml テンプレートファイルで Cisco ACI リソースを定義します。たとえば、/home/stack/templates/aci_cs.yaml などです。この手順では、OpFlex エージェントの使用例のリソース宣言について説明します。

- (注)
- 完全なリソース宣言の例については、このガイドの付録の「リソース宣言の例」の項を参照してください。
 - OpFlex ではないユース ケース (neutron-openvswitch-agent) のリソース宣言の例については、このガイドの付録の「Neutron OVS エージェントを使用する場合のリソース宣言の例」のセクションを参照してください。
 - Cisco ACI 環境に必要なパラメータのリストについては、このガイドの付録の「Cisco ACI 環境のパラメータ」の項を参照してください。

例 :

次に、opflex を使用して OSP を展開するためのリソースの例を示します。

```
resource_registry:
```

```
#controller
OS::TripleO::ControllerExtraConfigPre: /opt/ciscoaci-tripleo-heat-templates//nodepre.yaml

OS::TripleO::Services::NeutronOvsAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/neutron_opflex/neutron-opflex-agent-container-puppet.yaml

OS::TripleO::Docker::NeutronMl2PluginBase:
/opt/ciscoaci-tripleo-heat-templates/deployment/neutron/neutron-ml2-ciscoaci.yaml
OS::TripleO::Services::CiscoAciAIM:
/opt/ciscoaci-tripleo-heat-templates/deployment/aciain/cisco-aciain-container-puppet.yaml

OS::TripleO::Services::NeutronMetadataAgent:
/usr/share/openstack-tripleo-heat-templates/deployment/neutron/neutron-metadata-container-puppet.yaml

OS::TripleO::Services::NeutronDhcpAgent:
/usr/share/openstack-tripleo-heat-templates/deployment/neutron/neutron-dhcp-container-puppet.yaml

#compute
OS::TripleO::ComputeExtraConfigPre: /opt/ciscoaci-tripleo-heat-templates//nodepre.yaml

OS::TripleO::Services::ComputeNeutronOvsAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/neutron_opflex/neutron-opflex-agent-container-puppet.yaml

OS::TripleO::Services::ComputeNeutronMetadataAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/compute_neutron_metadata/compute-neutron-metadata.yaml

OS::TripleO::Services::CiscoAciOpflexAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/opflex/opflex-agent-container-puppet.yaml

OS::TripleO::Services::CiscoAciLldp:
/opt/ciscoaci-tripleo-heat-templates/deployment/lldp/cisco_lldp.yaml

OS::TripleO::Services::OVNDBs: OS::Heat::None
OS::TripleO::Services::OVNController: OS::Heat::None
OS::TripleO::Services::OVNMetadataAgent: OS::Heat::None
OS::TripleO::Services::ComputeNeutronL3Agent: OS::Heat::None
OS::TripleO::Services::NeutronL3Agent: OS::Heat::None
```

ステップ 4 Cisco ACI 証明書ベースの認証を使用するには、X.509 証明書を持つローカル ユーザーを作成し、パラメータ `ACIApicPrivateKey` および `ACIApicCertName` を使用して、Cisco ACI リソース ファイルで証明書とキーを指定します。

『[Cisco APIC Security Configuration Guide, Release 5.1\(x\)](#)』の「Creating a Local User and Adding a User Certificate」の項を参照してください。

(注) 証明書ベースの認証を使用する場合は、`ACIApicPassword` パラメータを指定しないでください。

ステップ 5 オーバークラウドを展開します。

オーバークラウドを展開する場合は、`-r` オプションを使用して作成したカスタムロールのデータ ファイルを含めます。サイト固有の環境ファイルに加えて、Cisco ACI 環境ファイルと Cisco ACI コンテナ YAML ファイルも環境リストに含めます。

例：

```
openstack overcloud deploy --templates /home/stack/tripleo-heat-templates -r
/home/stack/templates/aci_roles_data.yaml -e
/home/stack/tripleo-heat-templates/environments/network-isolation.yaml -e
```

```
/home/stack/templates/overcloud_images.yaml -e  
/home/stack/templates/network-environment.yaml -e  
/home/stack/templates/ciscoaci_containers.yaml -e /home/stack/templates/ciscoaci-env.yaml  
-e /home/stack/templates/rhel-registration-resource-registry.yaml -e  
/home/stack/templates/environment-rhel-registration
```

上記の例は、Cisco ACI テンプレートとロールの使用方法を示しています。他のテンプレートは、インストール構成によって異なる場合があります。カスタムテンプレートの作成とネットワーク環境テンプレートの自動生成については、Red Hat のガイドラインに従ってください。

参照

- Red Hat Web サイトでの Red Hat OpenStack プラットフォームの Director のインストールと使用方法。
- Cisco.com の特定のポートに [EPG](#) を展開するためのドメイン、接続エンティティプロファイル、および [VLAN](#) の作成のナレッジベースの記事。
- 「[Cisco ACI ファブリックの初期化の例](#)」も参照してください。

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。