



Cisco ACI 向け Red Hat OpenStack プラットフォーム インストールガイド

初版：2015年11月09日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】 シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワーク トポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2015 Cisco Systems, Inc. All rights reserved.



目次

はじめに v

対象読者 v

表記法 v

関連資料 vii

マニュアルに関するフィードバック vii

OpenStack グループベース ポリシー用 Cisco ACI プラグインのインストール 1

前提条件 1

グループベース ポリシー RPM のダウンロード 2

YUM リポジトリのセットアップ 2

サーバの準備 2

グループベース ポリシーを使用する OpenStack インストールの変更 4

OpenStack コントローラのインストールおよび設定 4

ネットワーク コントローラのインストールおよび設定 7

OpenStack Neutron ML2 用 Cisco ACI プラグインのインストール 11

前提条件 11

OpenStack Neutron Modular Layer2 用 Cisco ACI プラグインをインストールするためのワークフロー 12

設置場所の準備 13

サーバの準備 13

OpenStack 修正のインストール 15

OpFlex エージェントおよび APIC OVS エージェントのインストール 19

OpFlex の設定 19

ホスト エージェントおよびサービス エージェントの起動 21

外部接続の設定 22

その他の参考資料 23

packstack コマンドを使用した GBP での OpenStack のインストール 23



はじめに

この前書きは、次の項で構成されています。

- [対象読者](#), [v ページ](#)
- [表記法](#), [v ページ](#)
- [関連資料](#), [vii ページ](#)
- [マニュアルに関するフィードバック](#), [vii ページ](#)

対象読者

このガイドは、アプリケーションセントリック インフラストラクチャ ファブリック の設定および維持に携わるネットワーク管理者およびシステム管理者を対象としています。

表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
<i>italic</i>	イタリック体の文字は、ユーザが値を入力する引数です。
[x]	省略可能な要素（キーワードまたは引数）は、角カッコで囲んで示しています。
[x y]	いずれか1つを選択できる省略可能なキーワードや引数は、角カッコで囲み、縦棒で区切って示しています。

表記法	説明
{x y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波カッコで囲み、縦棒で区切って示しています。
[x {y z}]	角カッコまたは波カッコが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角カッコ内の波カッコと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
variable	ユーザが値を入力する変数であることを表します。イタリック体を使用できない場合に使用されます。
string	引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。

例では、次の表記法を使用しています。

表記法	説明
screen フォント	スイッチが表示する端末セッションおよび情報は、screen フォントで示しています。
太字の screen フォント	ユーザが入力しなければならない情報は、太字の screen フォントで示しています。
イタリック体の screen フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
<>	パスワードのように出力されない文字は、山カッコ (<>) で囲んで示しています。
[]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符 (!) またはポンド記号 (#) がある場合には、コメント行であることを示します。

このマニュアルでは、次の表記法を使用しています。



(注) 「注釈」です。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。

**注意**

「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

**警告****安全上の重要事項**

「危険」の意味です。人身事故を予防するための注意事項が記述されています。装置の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止策に留意してください。各警告の最後に記載されているステートメント番号を基に、装置に付属の安全についての警告を参照してください。

これらの注意事項を保管しておいてください。

関連資料

APIC の機能と運用については、『*Cisco Application Centric Infrastructure Fundamentals*』を参照してください。

マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、apic-docfeedback@cisco.com までご連絡ください。ご協力をよろしく願います。



第 1 章

OpenStack グループベース ポリシー用 Cisco ACI プラグインのインストール

この章の内容は、次のとおりです。

- [前提条件, 1 ページ](#)
- [グループベース ポリシー RPM のダウンロード, 2 ページ](#)
- [YUM リポジトリのセットアップ, 2 ページ](#)
- [サーバーの準備, 2 ページ](#)
- [グループベース ポリシーを使用する OpenStack インストールの変更, 4 ページ](#)

前提条件

開始するには、お使いのコンピューティング環境が次の前提条件を満たしている必要があります。

- Cisco Application Centric Infrastructure (ACI) ファブリックが機能している必要があります。
『*Cisco APIC Getting Started Guide*』を参照してください。
- OpenStack コントローラ、ネットワーク ノード、およびコンピューティング ノードの記録を作成します。
- OpenStack コントローラはリーフに接続できません。
- ネットワーク ノードおよびコンピューティング ノードはリーフに接続する必要があります。
- Red Hat Enterprise Linux (RHEL) 7 または RHEL 7.1 がサーバーにインストールされている必要があります。

Red Hat Enterprise Linux のマニュアルを参照してください。

- OpenStack (OSP) 6 リポジトリへのアクセス権が必要です。

グループベース ポリシー RPM のダウンロード

ここでは、シスコのグループベース ポリシー（GBP）RPM をダウンロードする方法について説明します。

手順

-
- ステップ 1 シスコの Web サイトにアクセスします。
 - ステップ 2 [Download Software] を選択します。
 - ステップ 3 リリース バージョンと `filename_release.tar` ファイルを選択します。
 - ステップ 4 [Download] をクリックします。
 - ステップ 5 ターミナル ウィンドウから、tar ファイルをダウンロードしたディレクトリに移動します。
 - ステップ 6 `filename_release.tar` ファイルを解凍します。

```
$ tar -xvf filename_release.tar
```
-

YUM リポジトリのセットアップ

この項では、RPM を使用して YUM リポジトリをセットアップする方法について説明します。

手順

RPM を使用して YUM リポジトリをセットアップします。

サーバの準備

手順

-
- ステップ 1 グループベース ポリシー（GBP）RPM に対して YUM リポジトリを有効にします。
 - ステップ 2 OpenStack 設定のネットワーク ノードおよびコンピューティング ノードが、ダイレクトモードまたは VPC/ボンドモードでリーフに接続されていることを確認します。
 - ステップ 3 APIC のインフラ VLAN タグ付きサブインターフェイスがサーバに定義されている必要があります。これはリーフに接続されているインターフェイスのサブインターフェイスです。次に例を示します。
 - eth1 はリーフに接続されます。

- APIC はデフォルトの VLAN 4093 で設定されます。
- システムには、サブインターフェイス eth1.4093 が必要です。
/etc/sysconfig/network-scripts/ifcfg-eth1.4093

例：

```
PERSISTENT_DHCLIENT=1
DHCPRELEASE=1
DEVICE=eth1.4093
ONBOOT=yes
PEERDNS=yes
NM_CONTROLLED=no
HWADDR=<mac-address-of-eth1>
TYPE=Ethernet
BOOTPROTO=dhcp
VLAN=yes
ONPARENT=yes
MTU=1600
```

このインターフェイスでは、APIC インフラストラクチャ ネットワークからの DHCP アドレスを要求します。サーバにはこのインターフェイスの `dhclient` 設定が必要です。上記の例では、`/etc/dhcp/dhclient-eth1.4093.conf` ファイルに次の内容が含まれている必要があります。

```
end dhcp-client-identifier 01:<mac-address>;
request subnet-mask, domain-name, domain-name-servers, host-name;
send host-name <server-host-name>;

option rfc3442-classless-static-routes code 121 = array of unsigned integer 8;
option ms-classless-static-routes code 249 = array of unsigned integer 8;
option wpad code 252 = string;

also request rfc3442-classless-static-routes;
also request ms-classless-static-routes;
also request static-routes;
also request wpad;
also request ntp-servers;
```

マルチキャスト ルートの設定には、このサブインターフェイスを使用してください。/etc/sysconfig/network-scripts/route-eth1.4093 ファイルに次の内容が含まれている必要があります。

```
ADDRESS0=224.0.0.0
NETMASK0=240.0.0.0
GATEWAY0=0.0.0.0
METRIC0=1000
```

グループベース ポリシーを使用する OpenStack インストールの変更

ここでは、グループベース ポリシー（GBP）を使用するための OpenStack インストールの変換で必要な変更について説明します。

OpenStack がインストール済みで、Neutron が設定可能である必要があります。

OpenStack コントローラのインストールおよび設定

ここでは、グループベース ポリシー（GBP）および Application Policy Infrastructure Controller（APIC）を使用するために、既存の OpenStack 設定を変換する方法について説明します。

手順

ステップ 1 次のパッケージをインストールします。

- python-pip
- python-pbr
- neutron-opflex-agent
- apicapi
- openstack-neutron-gbp
- python-gbpclient
- openstack-dashboard-gbp
- openstack-heat-gbp

ステップ 2 `/etc/neutron/neutron.conf` ファイルのパラメータ値を更新します。

パラメータ	値	注
「DEFAULT」セクション		
apic_system_id	<string>	
service_plugins	'group_policy,servicechain,apic_gbp_l3,lbaas'	
「opflex」セクション		
networks	'*'	
「ml2_cisco_apic」セクション		

パラメータ	値	注
apic_hosts	APIC コントローラの IP アドレス 例 : 1.1.2.1、1.1.2.1,1.1.2.2、または 1.1.2.3	カンマで区切られた APIC コントローラの IP アドレスです。
apic_username	APIC コントローラのユーザ名	
apic_password	APIC コントローラのパスワード	
apic_domain_name	<string>	
use_vmm	True	
apic_use_ssl	True	
apic_clear_node_profiles	True	
enable_aci_routing	True	
enable_arp_flooding	True	
apic_name_mapping	<user_name user_uuid>	1 つのファブリックに複数の OpenStack インスタンスがある場合は <i>user_uuid</i> を使用します。
apic_entity_profile	<string>	
apic_vmm_domain	<string>	
apic_app_profile_name	<string>	
apic_provision_infra	True	
apic_provision_hostlinks	True	APIC にインターフェイスプロファイル、スイッチプロファイル、ポリシーグループがすでに存在する場合は、 <i>False</i> に設定します。
apic_vpc_pairs	VPC ペアがない場合は空白のままにします。 VPC ペアがある場合は、スイッチペアをリストします。 例 : 101:102,103:104	101、102、103、および 104 は、APIC に登録されたスイッチ ID です。 スイッチ ID 101 と 102、103 と 104 が VPC ペアです。

パラメータ	値	注
「group_policy」セクション		
policy_drivers	'implicit_policy,apic'	
「group_policy_implicit_policy」セクション		
default_ip_pool	<network>/<mask bits> 例 : 192.168.0.0/16	

- a) コンピューティング ノードが接続されたスイッチごとに、`apic_switch:<switch_id>` という新しいセクションを作成します。このセクションに、コンピューティング ノードと対応するスイッチ インターフェイスを `<compute_fqdn>=<switch_port>` 形式でリストします（非 VPC の場合）。

例 :

```
[apic_switch:201]
fab102-compute-1.noironetworks.com=1/33
fab102-compute-2.noironetworks.com=1/34
```

- b) コンピューティング ノードが接続されたスイッチごとに、`apic_switch:<switch_id>` という新しいセクションを作成します。このセクションに、コンピューティング ノードと対応するスイッチ インターフェイスを `<compute_fqdn>=vpc-<mod>-<port>/<vpc_policy_group_name>` 形式でリストします（VPC の場合）。

例 :

```
[apic_switch:101]
compute-1.noiro.lab=vpc-1-17/bundle-101-1-17-and-102-1-17
[apic_switch:102]
compute-1.noiro.lab=vpc-1-17/bundle-101-1-17-and-102-1-17
[apic_switch:103]
compute-2.noiro.lab=vpc-1-18/bundle-101-1-18-and-102-1-18
[apic_switch:104]
compute-2.noiro.lab=vpc-1-18/bundle-101-1-18-and-102-1-18
```

compute-1 と compute-2 は、vPC を使用してファブリックに接続されています。

- compute-1 には、スイッチ 101 のポート 1/17 とスイッチ 102 のポート 1/17 にメンバーが接続されている bond0 があります。VPC のファブリック上のポリシー グループ名は bundle-101-1-17-and-102-1-17 です。
- compute-2 には、スイッチ 103 のポート 1/18 とスイッチ 104 のポート 1/18 にメンバーが接続されている bond0 があります。VPC のファブリック上のポリシー グループ名は bundle-102-1-18-and-102-1-18 です。

ステップ 3 /etc/neutron/plugins/ml2/ml2_conf.ini ファイルのパラメータ値を更新します。

パラメータ	値
「ml2」セクション	
type_drivers	"opflex,local,flat,vlan,gre,vxlan"
tenant_network_types	"opflex"
mechanism_drivers	"apic_gbp"
「ovs」セクション	
local_ip	<ip_address_of_openstack_controller>

ステップ 4 /etc/heat/heat.conf ファイルのパラメータ値を更新します。

パラメータ	値
「DEFAULT」セクション	
plugin_dirs	/usr/lib/python2.7/site-packages/gbpautomation/heat

ネットワークコントローラのインストールおよび設定

ここでは、グループベースポリシー（GBP）および Application Policy Infrastructure Controller（APIC）を使用するために、既存の OpenStack 設定を変換する方法について説明します。

手順

ステップ 1 次のパッケージをインストールします。

- python-pip
- python-pbr
- openstack-neutron-gbp
- openvswitch-gbp
- neutron-opflex-agent
- agent-ovs

ステップ 2 /etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini ファイルのパラメータ値を更新します。

パラメータ	値
「ovs」セクション	
enable_tunneling	False
integration_bridge	<bridge-name> 例 : br-int

- a) ovs_neutron_plugin.ini ファイルから「tunnel_bridge」、「local_ip」、「vxlan_udp_port」および「tunnel_types」パラメータを削除します。

ステップ 3 /etc/neutron/dhcp_agent.ini ファイルのパラメータ値を更新します。

パラメータ	値
「DEFAULT」セクション	
ovs_integration_bridge	ovs_neutron_plugin.ini ファイルで指定されている統合ブリッジの名前。 例 : br-int
enable_isolated_metadata	True

ステップ 4 「openvswitch」カーネルモジュールをアンロードし、「openvswitch-gbp」パッケージの一部としてインストールされた新しいモジュールに置き換えて、OpFlex を設定します。

```
/usr/sbin/rmmod openvswitch
/usr/sbin/modprobe openvswitch
```

- a) 「openvswitch」プロセスを再起動します。
`service openvswitch restart`
- b) OpFlex エージェントのコンフィギュレーションファイルを設定します。OpFlex エージェントは、/etc/opflex-agent-ovs/opflex-agent-ovs.conf ファイルから設定を読み取ります。設定は JSON 形式です。
サンプルファイル内で強調されている値はサイト固有の値に変更してください。

```
{
  "log": {
    "level": "debug2"
  },

  "opflex": {
    "domain": "comp/prov-OpenStack/ctrlr-[ostack]-ostack/sw-InsiemeLSoid",
    "name": "sg1-compute-1",
    "peers": [
```

```

        {"hostname": "10.0.0.30", "port": "8009"}
    ],
    "ssl": {
        "mode": "enabled",
        "ca-store": "/etc/ssl/certs/"
    }
},

"endpoint-sources": {
    "filesystem": ["/var/lib/opflex-agent-ovs/endpoints"]
},

"renderers": {
    "stitched-mode": {
        "ovs-bridge-name": "br-int",
        "encap": {
            "vxlan" : {
                "encap-iface": "br-int_vxlan0",
                "uplink-iface": "bond0.4093",
                "uplink-vlan": 4093,
                "remote-ip": "10.0.0.32",
                "remote-port": 8472
            }
        },
        "forwarding": {
            "virtual-router": {
                "enabled": true,
                "mac": "00:22:bd:f8:19:ff",
                "ipv6": {
                    "router-advertisement": "false"
                }
            },
            "virtual-dhcp": {
                "enabled": "true",
                "mac": "00:22:bd:f8:19:ff"
            }
        },
        "flowid-cache-dir": "/var/lib/opflex-agent-ovs/ids"
    }
}
}
}

```

強調されている値	説明
ostack	両方の値をOpenStack コントローラのインストールおよび設定、(4 ページ) のステップ 2 で定義した apic_system_id で置き換えます。
sg1-compute-1	ホスト名で置き換えます。
br-int	この手順のステップ 2 で定義した OVS の統合ブリッジ名で置き換えます。

強調されている値	説明
bond0.4093	bond0 をリーフに接続されたインターフェイス名で置き換え、4093 を APIC のインフラ VLAN で置き換えます。
4093	APIC のインフラ VLAN で置き換えます。

ステップ 5 agent-ovs サービスを再起動します。

```
service agent-ovs restart
```

ステップ 6 すべてのコンピューティング ノードで、ステップ1、2、および 4b を繰り返します。



第 2 章

OpenStack Neutron ML2 用 Cisco ACI プラグインのインストール

この章の内容は、次のとおりです。

- [前提条件, 11 ページ](#)
- [OpenStack Neutron Modular Layer 2 用 Cisco ACI プラグインをインストールするためのワークフロー, 12 ページ](#)
- [設置場所の準備, 13 ページ](#)
- [サーバの準備, 13 ページ](#)
- [OpenStack 修正のインストール, 15 ページ](#)
- [ホストエージェントおよびサービスエージェントの起動, 21 ページ](#)
- [外部接続の設定, 22 ページ](#)

前提条件

開始するには、次の前提条件を満たしている必要があります。

- Red Hat Linux、OpenStack、および Application Policy Infrastructure Controller (APIC) の十分な知識がある。
- Cisco Application Centric Infrastructure (ACI) ファブリックが機能している。
- サーバまたは仮想マシンで Red Hat または CentOS バージョンを実行している (リポジトリ「サーバ」)。
- コントローラまたはネットワーク ノード、およびコンピューティング ノードが ACI リーフスイッチに接続されている (ネットワーク ノードとコントローラが同じノード上にある場合、本マニュアルで使われる「ネットワーク ノード」はコントローラを指します)。

- OpenStack の RPM ディストリビューション (RDO) または Red Hat 7 以降の Red Hat OpenStack (RHOS) 6 (Juno) で EPEL がイネーブルになっており、可能であればデモ ネットワークがプロビジョニングされていない (EPEL をイネーブルにしてデモ ネットワークをディセーブルにするには、応答ファイルに次の行を追加します)。

```
CONFIG_USE_EPEL=y  
CONFIG_PROVISION_DEMO=n
```

- OpenStack に ネットワーク/ルータ および 仮想マシン が作成されていない (OpenStack にある既存の ネットワーク/ルータ および 仮想マシン は削除してください)。

OpenStack Neutron Modular Layer 2 用 Cisco ACI プラグインをインストールするためのワークフロー

次のワークフローを使用して、OpenStack Neutron Modular Layer 2 (ML2) 用 Cisco ACI プラグインをインストールします。

- 1 設置場所を準備します。
[設置場所の準備, \(13 ページ\)](#) を参照してください。
- 2 サーバを準備します。
[サーバの準備, \(13 ページ\)](#) を参照してください。
- 3 OpenStack 修正をインストールします。
[OpenStack 修正のインストール, \(15 ページ\)](#) を参照してください。
- 4 OpFlex エージェントと Application Policy Infrastructure Controller (APIC) OVS エージェントをインストールします。
[OpFlex エージェントおよび APIC OVS エージェントのインストール, \(19 ページ\)](#) を参照してください。
- 5 OpFlex を設定します。
[OpFlex の設定, \(19 ページ\)](#) を参照してください。
- 6 ホスト エージェントと サービス エージェント を起動します。
[ホスト エージェント および サービス エージェント の起動, \(21 ページ\)](#) を参照してください。
- 7 次の手順を実行して、コンピューティング ノード をすべて設定します。
 - a [サーバの準備, \(13 ページ\)](#)
 - b [OpFlex エージェント および APIC OVS エージェント のインストール, \(19 ページ\)](#)
 - c [OpFlex の設定, \(19 ページ\)](#)
- 8 外部接続を設定します。
[外部接続の設定, \(22 ページ\)](#) を参照してください。

設置場所の準備

YUM リポジトリを作成して、Red Hat OpenStack プラットフォームをインストールする場所を準備する必要があります。

手順

-
- ステップ 1** cisco.com から apic-m12 および opflex RPM をダウンロードします。
- ステップ 2** httpd サービスをインストールして起動します。
- ```
yum install httpd
service httpd start
```
- ステップ 3** RPM を /var/www/html ディレクトリに移動します。
- ```
# mv apic_m12 opflex /var/www/html
```
- ステップ 4** リポジトリを作成します。たとえば、apic_m12 パッケージが apic_m12 ディレクトリにあり、opflex パッケージが opflex ディレクトリにある場合は、次のコマンドを使用します。
- ```
createrepo /var/www/html/opflex
createrepo /var/www/html/apic_m12
```
- 

## サーバの準備

ネットワーク ノードとしても機能するコントローラを含む、すべてのサーバについて次の手順を実行します。この手順の設定例では、次の内容を前提とします。

- eth1 はリーフに接続されます。
- Application Policy Infrastructure Controller (APIC) はデフォルトの VLAN 4093 (インフラ VLAN) で設定されます。この番号は、Application Policy Infrastructure Controller に設定されている VLAN の番号で置き換えてください。

### 手順

- 
- ステップ 1** net-tools をインストールし、ifconfig および route コマンドを有効にします。
- ```
# yum install net-tools
```
- ステップ 2** wget をインストールします。
- ```
yum install wget
```
- ステップ 3** /etc/yum.repos.d/opflex.repo コンフィギュレーションファイルを編集して次の行を追加することで、パッケージの YUM リポジトリを有効にします。
- ```
[opflex]
name=opflex repo
```

```
baseurl=http://172.29.172.13/opflex
failovermethod=priority
enabled=1
gpgcheck=0
[ml2-apic]
name=opflex repo
baseurl=http://172.29.172.13/ml2-apic
failovermethod=priority
enabled=1
gpgcheck=0
```

この変更によって opflex および apic_ml2 リポジトリを追加することができます。

ステップ 4 リポジトリが正常に機能することを確認します。

```
# yum makecache
```

エラーがなければ、リポジトリは正常に機能しています。エラーが発生した場合は、先に進む前に問題をトラブルシューティングしてください。

ステップ 5 OpenStack 設定のネットワーク ノードおよびコンピューティング ノードを、ダイレクト モードまたは VPC/ボンド モードでリーフ スイッチに接続します。

ステップ 6 APIC のインフラ VLAN タグ付きサブインターフェイスをサーバで定義します。これはリーフに接続されているインターフェイスのサブインターフェイスです。

ステップ 7 インターフェイス コンフィギュレーション ファイル `/etc/sysconfig/network-scripts/ifcfg-eth1` に次の行を追加することで、メイン インターフェイス (eth1) の MTU を 1600 に設定します。

```
MTU=1600
```

ステップ 8 eth1 インターフェイスにバウンスします。

```
# ifdown eth1
# ifup eth1
```

ステップ 9 次の内容を含む `/etc/sysconfig/network-scripts/ifcfg-eth1.4093` ファイルを作成して、VLAN 4093 の VLAN サブインターフェイスを設定します。

```
PERSISTENT_DHCLIENT=1
DHCPRELEASE=1
DEVICE=eth1.4093
ONBOOT=yes
PEERDNS=yes
NM_CONTROLLED=no
HWADDR=<mac address of eth1, or self-derived; see note below>
TYPE=Ethernet
BOOTPROTO=dhcp
VLAN=yes
ONPARENT=yes
MTU=1600
```

(注) LLDP フレームのスイッチへの送信に LLDP 設定を使用するには、親インターフェイスと同じ MAC アドレスは使用できません。LLDP 転送を有効にした場合に未定義の動作を引き起こす原因となるため、親インターフェイスの MAC アドレスは使用しないでください。

eth1 インターフェイスでは、APIC インフラストラクチャ ネットワークの DHCP アドレスを要求します。サーバが正しい DHCP オプションを提供するためには、このインターフェイスの dhclient 設定が必要です。

ステップ 10 次の内容を含む /etc/dhcp/dhclient-eth1.4093.conf ファイルを作成します。

```
send dhcp-client-identifier 01:<mac-address of .4093 interface>;
request subnet-mask, domain-name, domain-name-servers, host-name;
send host-name <server-host-name>;

option rfc3442-classless-static-routes code 121 = array of unsigned integer 8;
option ms-classless-static-routes code 249 = array of unsigned integer 8;
option wpad code 252 = string;

also request rfc3442-classless-static-routes;
also request ms-classless-static-routes;
also request static-routes;
also request wpad;
also request ntp-servers;
```

ステップ 11 次の内容を含む /etc/sysconfig/network-scripts/route-eth1.4093 ファイルを作成して、マルチキャストルートのサブインターフェイスを設定します。

```
ADDRESS0=224.0.0.0
NETMASK0=240.0.0.0
GATEWAY0=0.0.0.0
METRIC0=1000
```

ステップ 12 サーバで LLDPD を有効にします。LLDPD パッケージをインストールします。

```
# cd /etc/yum.repos.d/
# wget http://download.opensuse.org/repositories/home:vbernat/RHEL_7/home:vbernat.repo
# yum install lldpd
```

ステップ 13 スイッチへの LLDP フレームの送信に問題があるため、/etc/sysconfig/lldpd ファイルの LLDP_OPTIONS パラメータに "-r" を付加してフレームをブロックします。

```
LLDPD_OPTIONS="-r"
```

何らかの理由で LLDP 転送が必要な場合は、APIC のインフラ VLAN サブインターフェイスが親インターフェイスの MAC アドレスを継承しないことを確認し、この手順は省略してください。

ステップ 14 LLDPD を有効にして起動します。

```
# service lldpd start
# systemctl enable lldpd
```

OpenStack 修正のインストール

OpenStack コントローラまたはネットワーク ノードに、追加の OpenStack 修正パッケージをインストールする必要があります。パッケージには EPEL リポジトリが必要です。EPEL が有効な状態で OpenStack をインストールしなかった場合は、EPEL を手動で有効にする必要があります。

- eth1 はリーフに接続されます。

- Application Policy Infrastructure Controller (APIC) はデフォルトの VLAN 4093 (インフラ VLAN) で設定されます。この番号は、Application Policy Infrastructure Controller に設定されている VLAN の番号で置き換えてください。

手順

- ステップ 1** 次のパッケージをインストールします。
- ```
yum install python-pip
yum install python-pbr
yum install neutron-opflex-agent apicapi neutron-ml2-driver-apic
```
- ステップ 2** oslo-concurrency との依存関係の欠落に関するエラーが表示された場合は、/etc/yum.repos.d/epel-testing.repo ファイルを修正して [epel-testing] セクションに「enabled=1」を設定することで、epel-testing リポジトリを有効にしてください。
- ```
[epel-testing]
name=Extra Packages for Enterprise Linux 7 - Testing - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/testing/7/$basearch
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=testing-epel7&arch=$basearch
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
```
- (注) epel-testing に独自のリポジトリ ファイルが存在することがあります。この場合、epel-testing を有効にするには、/etc/yum.repos.d/epel-testing.repo ファイルの編集が必要な可能性があります。
- ステップ 3** /etc/neutron/neutron.conf ファイルを編集して service_plugins パラメータを更新します。
- ```
service_plugins = cisco_apic_l3,lbaas,metering
```
- ステップ 4** /etc/neutron/plugins/ml2/ml2\_conf.ini ファイルを編集して、mechanism\_drivers、type\_drivers、および tenant\_network\_type パラメータを更新します。
- ```
mechanism_drivers = cisco_apic_ml2
type_drivers = opflex,local,flat,vlan,gre,vxlan
tenant_network_types = opflex
```
- a) 「network_vlan_ranges」を含むすべての行をコメントアウトします。
- ステップ 5** /etc/neutron/dhcp_agent.ini ファイルを編集して dhcp_driver パラメータを更新します。
- ```
dhcp_driver = apic_ml2.neutron.agent.linux.apic_dhcp.ApicDnsmasq
```
- a) DEFAULT セクションに次のオプションが存在することを確認してください。
- ```
[DEFAULT]
ovs_integration_bridge = br-int
enable_isolated_metadata = True
```
- ステップ 6** DHCP エージェントを再起動します。
- ```
systemctl restart neutron-dhcp-agent.service
```

**ステップ 7** /etc/neutron/plugins/ml2/ml2\_conf\_cisco.ini ファイルを編集して次のパラメータを設定します。

```
[DEFAULT]
apic_system_id = <any string>

[opflex]
networks = '*'

[ml2_cisco_apic]
apic_hosts = comma-separated IP address list of APIC controllers
 10.1.2.1 or 10.1.2.1,10.1.2.2,10.1.2.3
apic_username = Username for APIC controller
apic_password = Password for 'apic_username'
apic_domain_name = <string>
use_vmm = True
apic_use_ssl = True
apic_clear_node_profiles = True
enable_aci_routing = True
enable_arp_flooding = True
apic_name_mapping = use_name (If there are multiple OpenStack instances per
 fabric, use use_uuid)
apic_entity_profile = <any string>
apic_vmm_domain = <any string>
apic_app_profile_name = <any string>
apic_function_profile = <any string>

apic_sync_interval = 0
apic_agent_report_interval = 30
apic_agent_poll_interval = 2
apic_provision_infra = True
apic_provision_hostlinks = True (Set to False if the interface profiles, switch
 profiles, policy groups are pre-existing on the APIC)

apic_vpc_pairs = leave blank if not VPC, otherwise
 101:102,103:104 (101,102,103 and 104 are switch ids as
 registered in APIC and pc pair between switch ids 101 and
 102, 103 and 104)

[group_policy]
policy_drivers = implicit_policy,apic

[group_policy_implicit_policy]
default_ip_pool = 192.168.0.0/16
```

**ステップ 8** コンピューティングノードが接続されたスイッチごとに、[apic\_switch:<switch id>] という名前のセクションを作成し、このセクションにコンピューティングノードおよび対応するスイッチインターフェイスをリストします。

VPC を使用していない場合は、次の形式を使用してください。

```
<compute_fqdn>=<switch port>
```

例：

```
[apic_switch:201]
fab102-compute-1.noironetworks.com=1/5
fab102-compute-2.noironetworks.com=1/6
```

VPC を使用している場合は、次の形式を使用してください。

```
<compute_fqdn>=vpc-<mod>-<port>/<vpc_policy_group_name>
```

次の例は、以下のように設定されます。

- compute-1 と compute-2 は、VPC を使用してファブリックに接続されます。
- compute-1 には、スイッチ 101 のポート 1/17 とスイッチ 102 のポート 1/17 にメンバーが接続されている bond0 があります。VPC のファブリック上のポリシー グループ名は「bundle-101-1-17-and-102-1-17」です。
- compute-2 には、スイッチ 103 のポート 1/18 とスイッチ 104 のポート 1/18 にメンバーが接続されている bond0 があります。VPC のファブリック上のポリシー グループ名は「bundle-102-1-18-and-102-1-18」です。

/etc/neutron/plugins/ml2/ml2\_conf\_cisco.ini ファイルのエントリは次のとおりです。

```
[apic_switch:101]
compute-1.noiro.lab=vpc-1-17/bundle-101-1-17-and-102-1-17
[apic_switch:102]
compute-1.noiro.lab=vpc-1-17/bundle-101-1-17-and-102-1-17
[apic_switch:103]
compute-2.noiro.lab=vpc-1-18/bundle-101-1-18-and-102-1-18
[apic_switch:104]
compute-2.noiro.lab=vpc-1-18/bundle-101-1-18-and-102-1-18
```

**ステップ 9** /usr/lib/systemd/system/neutron-server.service ファイルを編集して ExecStart 行に「--config-file /etc/neutron/plugins/ml2/ml2\_conf\_cisco\_apic.ini」を追加することで、/etc/neutron/plugins/ml2/ml2\_conf\_cisco.ini ファイルを読み込む neutron-server プロセスを設定します。追加された行は次のようになります

```
ExecStart=/usr/bin/neutron-server
--config-file /usr/share/neutron/neutron-dist.conf
--config-file /etc/neutron/neutron.conf
--config-file /etc/neutron/plugin.ini
--config-file /etc/neutron/plugins/ml2/ml2_conf_cisco.ini
--log-file /var/log/neutron/server.log
```

**ステップ 10** サービス コンフィギュレーションをリロードします。

```
systemctl daemon-reload
```

**ステップ 11** ネットワーク ノード（ネットワーク コントローラがない場合はコントローラ ノード）で neutron-server サービスを再起動します。

```
service neutron-server restart
```

## OpFlex エージェントおよび APIC OVS エージェントのインストール

ネットワークノード、またはコントローラノード（ネットワークコントローラがない場合）で、OpFlex エージェントと Application Policy Infrastructure Controller (APIC) OVS エージェントをインストールします。

### 手順

**ステップ 1** OpFlex エージェントと APIC OVS エージェントをインストールします。

```
yum install neutron-opflex-agent
yum install agent-ovs
```

**ステップ 2** /etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini ファイルを編集して次の行を追加します。

```
[ovs]
enable_tunneling = False
integration_bridge = br-int
local_ip = <ip address of openstack controller>
```

a) 次のオプションをコメントアウトまたは削除します。

- tunnel\_bridge
- local\_ip
- vxlan\_udp\_port
- tunnel\_types

## OpFlex の設定

ネットワークノード、またはコントローラノード（ネットワークコントローラがない場合）で、OpFlex エージェントをインストールした後に設定を行う必要があります。

### 手順

**ステップ 1** デフォルトの Open vSwitch エージェントをディセーブルにします。

```
systemctl disable neutron-openvswitch-agent
service neutron-openvswitch-agent stop
```

**ステップ 2** シスコの修正 Open vSwitch パッケージをインストールします。

```
yum install openvswitch-gbp
```

**ステップ 3** インストールした Open vSwitch パッケージが、前に存在したカーネルバージョンと異なる場合は、サーバをリブートします。

**ステップ 4** Open vSwitch モジュールをリロードします。

```
rmmmod openvswitch
modprobe openvswitch
```

**ステップ 5** OpFlex エージェントのコンフィギュレーションファイルを設定します。OpFlex エージェントは、`/etc/opflex-agent-ovs/opflex-agent-ovs.conf` ファイルから設定を読み取ります。設定は JSON 形式です。次のサンプルファイルで、`<apic_system_id>` を以前の ini ファイルの設定値に変更し、`<hostname of the system>` を適切なホスト名に変更します。インフラ VLAN と VLAN インターフェイスおよびブリッジの名前が正しいことを確認します。

```
{
 "log": {
 "level": "debug2"
 },

 "opflex": {
 "domain": "comp/prov-OpenStack/ctrlr-<apic_system_id>-<apic_system_id>/sw-InsiemeLSoid",

 "name": "<hostname of this system>",
 "peers": [
 {"hostname": "10.0.0.30", "port": "8009"}
],
 "ssl": {
 "mode": "enabled",
 "ca-store": "/etc/ssl/certs/"
 }
 },

 "endpoint-sources": {
 "filesystem": ["/var/lib/opflex-agent-ovs/endpoints"]
 },

 "renderers": {
 "stitched-mode": {
 "ovs-bridge-name": "br-int",

 "encap": {
 "vxlan": {
 "encap-iface": "br-int_vxlan0",
 "uplink-iface": "eth1.4093",
 "uplink-vlan": 4093,
 "remote-ip": "10.0.0.32",
 "remote-port": 8472
 }
 }
 },
 "forwarding": {
 "virtual-router": {
 "enabled": true,
 "mac": "00:22:bd:f8:19:ff",
 "ipv6": {
 "router-advertisement": "false"
 }
 }
 }
 }
}
```



## 外部接続の設定

コンピューティング ノードの設定後に、外部接続を設定する必要があります。

### 手順

- ステップ 1** ネクストホップ IP アドレスが 1.101.2.1 の場合は、Cisco Application Centric Infrastructure (ACI) L3Out ルータの IP アドレスを 1.101.2.2/30、ホストの SNAT プールを 2.101.2.1/24 に設定して、コントローラの設定にこれらの IP アドレスが含まれていることを確認します。

```
[apic_external_network:Datacenter-Out]
router_id=1.0.0.2
switch=101
port=1/2
```

```
encap=vlan-1011
gateway_ip=1.101.2.1
cidr_exposed=1.101.2.2/30
host_pool_cidr = 2.101.2.1/24
```

- a) 既存のレイヤ3アウトと既存の外部エンドポイントグループを代わりに使用する場合は、コントローラの設定に既存の情報が含まれていることを確認します。

```
[apic_external_network:Datacenter-Out]
preexisting=True
external_epg=Datacenter-Out-Epg
host_pool_cidr = 2.101.2.1/24
```

この例では、「Datacenter-Out-Epg」が既存の外部エンドポイントグループです。

- ステップ 2** すべてのコンピューティング ノードおよびネットワーク ノードで、SNAT を必要とする外部セグメントに「opflex\_external\_segment」という名前の新しいセクションを作成して次の行を追加し、SNAT アドレス割り当て/ゲートウェイが設定に含まれていることを確認します。

```
ip_gateway = <same as host_pool_cidr on the controller>
ip_address_range <start,end range from host_pool_cidr on the controller for the IP addresses
to be used for SNAT from this host>
```

例：

```
[opflex_external_segment:Datacenter-Out]
ip_gateway = 2.101.2.1/24
ip_address_range = 2.101.2.2
```



付録

# A

## その他の参考資料

---

この章の内容は、次のとおりです。

- [packstack コマンドを使用した GBP での OpenStack のインストール](#), 23 ページ

## packstack コマンドを使用した GBP での OpenStack のインストール

この項では、packstack コマンドを使用して GBP により OpenStack 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** Juno リリースから **packstack** をインストールします。  
Red Hat のマニュアルを参照してください。
- ステップ 2** 以下からリポジトリの複製を作ります。  
<https://github.com/noironetworks/gbp-packstack.git>
- ステップ 3** ファイルを  
/usr/lib/python2.7/site-packages/packstack/puppet/templates/puppet\_templates  
ディレクトリにコピーします。
- ステップ 4** ファイルを /usr/lib/python2.7/site-packages/packstack/plugins/plugins ディレクトリにコピーします。
- ステップ 5** gbp ディレクトリを /usr/share/openstack-puppet/modules ディレクトリにコピーします。  
これで GBP 関連のパラメータが packstack でサポートされるようになりました。応答ファイルを生成し、サイトの要件に合わせて応答ファイルを変更できます。packstack のマニュアルは、次の URL で確認できます。  
<https://wiki.openstack.org/wiki/Packstack>

