



インプレース アップグレード

Red Hat OSP13 から OSP16 へのインプレース アップグレードは、Cisco ACI OpenStack Plug-in 5.2(1) 以降でサポートされています。アップグレードプロセスは、『*FRAMEWORK FOR UPGRADES (13 TO 16.2) Red Hat*』ガイドで説明されている手順に関連しており、主にこの手順に基づいています。

この章で説明する各手順は必須であり、アップグレードに必要です。『*FRAMEWORK FOR UPGRADES (13 TO 16.2)*』ガイドの関連手順が示されています。

- [カスタム ACI リポジトリの削除 \(1 ページ\)](#)
- [ロールのカスタマイズ \(2 ページ\)](#)
- [Open vSwitch の互換性 \(3 ページ\)](#)
- [Cisco コンテナの構築 \(4 ページ\)](#)
- [アップグレードの準備 \(5 ページ\)](#)
- [アップグレードの収束 \(6 ページ\)](#)

カスタム ACI リポジトリの削除

カスタム ACI リポジトリを削除するには、次の手順を実行します。

始める前に

『*FRAMEWORK FOR UPGRADES (13 TO 16.2) Red Hat*』ガイドの「構成可能なサービスとパラメータの更新」セクションまでの手順に従います。

手順

ステップ 1 次の Ansible プレイブックを作成して、オーバークラウドノードの現在の yum リポジトリから ACI リポジトリを削除します。

```
---
- name: Remove ACI Repo
  hosts: overcloud
  become: yes
  tasks:
```

```
- name: remove_acirepo
  ansible.builtin.file:
    path: /etc/yum.repos.d/ciscoaci.repo
    state: absent
```

ステップ2 アンダークラウドからプレイブックを実行します。

```
ansible-playbook -i ~/inventory.yaml <name of playbook file>
```

ロールのカスタマイズ

ロールをカスタマイズするには、次の手順を使用します。

『*FRAMEWORK FOR UPGRADES (13 TO 16.2)*』ガイドの「*Updating composable services in custom routes_data files*」セクションで、カスタム ロール テンプレートを使用してシスコの構成可能サービスを追加します。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。

手順

次のコマンドを使用してプレイブックを実行します。

```
ansible-playbook -i ~/inventory.yaml
/opt/ciscoaci-tripleo-heat-templates/tools/generate_ciscoaci_role_data.yaml
```

これにより、`/home/stack/templates` ディレクトリに新しい `custom_roles_data.yaml` ファイルが作成されます。

カスタム ロールファイルを使用している場合は、上記の手順の代わりに、コントローラおよびコンピューティング ロールにサービスを追加する必要があります。

コントローラ ロールに次のサービスを追加します。

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

コンピューティング ロールに次のサービスを追加します。

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

Open vSwitch の互換性

『*FRAMEWORK FOR UPGRADES (13 TO 16.2) Red Hat*』ガイドの「アップグレード中の *Open vSwitch* 互換性の維持」セクションをスキップしてください。次に、シスコ固有の設定環境 `yml` ファイル (`ciscoaci-config.yml`) の例を示します。

```
# A Heat environment file which can be used to enable a
# a Neutron Cisco Aci backend on the controller, configured via puppet
resource_registry:

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

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

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

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

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

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

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

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

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

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

    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

parameter_defaults:

    EC2MetadataIp: 1.100.1.1
    ControlPlaneDefaultRoute: 1.100.1.1
    OvercloudControllerFlavor: control
    OvercloudComputeFlavor: compute

    DockerInsecureRegistryAddress: ["ostack-pt-1-s1-ucloud-13.ctlplane.localdomain:8787",
    "1.100.1.1:8787"]

    NeutronCorePlugin: 'ml2plus'
```

```

NeutronServicePlugins: 'group_policy,ncp,apic_aim_l3'
NeutronEnableIsolatedMetadata: true
NeutronEnableForceMetadata: true
NeutronPhysicalDevMappings: physnet1:eth1,physnet2:eth2
EnablePackageInstall: true
ACIScopeNames: true
ACIApicHosts: 10.30.120.190
ACIApicUsername: admin
ACIApicPassword: noir0123
ACIApicSystemId: ostack-pt-1-s1
ACIMechanismDrivers: 'apic_aim'
ACIApicEntityProfile: sauto_ostack-pt-1-s1_aep
ACIApicInfraVlan: 3701
ACIApicInfraSubnetGateway: 10.0.0.30
ACIApicInfraAnycastAddr: 10.0.0.32
ACIOpflexUplinkInterface: bond1
ACIYumRepo: http://1.100.1.1:8787/v2/__acirepo

ACIOpflexEncapMode: vxlan
NeutronNetworkVLANRanges: physnet1:1751:1800
ACIOpflexVlanRange: 751:800
HeatEnginePluginDirs:
/usr/lib64/heat,/usr/lib/heat,/usr/local/lib/heat,/usr/local/lib64/heat,/usr/lib/python2.7/site-packages/gbpautomation/heat

ACIVpcPairs: 101:102
NeutronPluginMl2PuppetTags: 'neutron_plugin_ml2,neutron_plugin_cisco_aci'
AciVmmMcastRanges: 225.2.1.1:225.2.255.255
AciVmmMulticastAddress: 225.2.10.3

#Below parameters are only needed when installing OpenShift on Openstack
ACIOpflexInterfaceType: 'ovs'
ACIOpflexInterfaceMTU: 8000

```

Cisco コンテナの構築

Cisco コンテナを構築するには、次の手順を使用します。

『*FRAMEWORK FOR UPGRADES (13 TO 16.2)* Red Hat』ガイドの「標準オーバークラウドのアップグレード」セクションに進む前に、シスコ固有のコンテナを構築する必要があります。

手順

ステップ 1 アンダークラウドから OSP13 Cisco Tripleo パッケージを削除し、新しい OSP16 RPM をインストールします。

例：tripleo-ciscoaci-16.2-1054.noarch.rpm をインストールするには、次のコマンドを使用します。

```

sudo yum remove tripleo-ciscoaci
sudo yum install ./tripleo-ciscoaci-16.2-1054.noarch.rpm

```

ステップ 2 アップストリーム コンテナ レジストリにログインします。たとえば、Red Hat のアップストリーム コンテナ レジストリを使用している場合は、次のコマンドを実行します。

```

sudo podman login registry.redhat.io

```

ステップ3 アップストリーム コンテナ レジストリにログインした後、次を使用して OSP16 の ACI コンテナ ビルド スクリプトを実行します。

```
sudo /opt/ciscoaci-tripleo-heat-templates/tools/build_openstack_aci_containers.py -z
openstack-ciscorpms-repo-16.2-1006.tar.gz
```

このスクリプトは、`/home/stack/templates/ciscoaci_containers.yaml` ファイルを作成します。このファイルは、シスコ固有のアップストリーム サービスまたは変更されたアップストリーム サービスのコンテナ イメージへのマッピングを提供します。

ステップ4 OSP16 のコンテナを構築した後、次のコマンドを使用して移行 (Stein) コンテナを構築します。

```
sudo /opt/ciscoaci-tripleo-heat-templates/tools/build_transitional_aci_containers.py -z
openstack-ciscorpms-repo-15.0-995.tar.gz --force
```

このスクリプトは、OSP15 用の ACI 移行コンテナを構築し、`/home/stack/templates/ciscoaci_containers_stein.yaml` ファイル内のコンテナイメージへのシスコ固有のサービスのマッピングを提供します。

アップグレードの準備

『*FRAMEWORK FOR UPGRADES (13 TO 16.2) Red Hat*』ガイドの「*Running the overcloud upgrade preparation*」セクションに、Cisco ACI 統合に固有のファイルが含まれていることを確認します。その内容は次のとおりです。

- カスタム ロール ファイル
- Cisco ACI OSP16 コンテナ マッピング Heat 環境ファイル
- Cisco ACI OSP15 コンテナ マッピング Heat 環境ファイル
- Cisco ACI 固有の構成環境ファイル

次に、シスコ固有のテンプレートを使用した **upgrade prepare** コマンドの例を示します。

```
source ~/stackrc
openstack overcloud upgrade prepare \
    --templates /home/stack/tripleo-heat-templates \
    -r /home/stack/templates/custom_roles_data.yaml \
    -e /home/stack/tripleo-heat-templates/environments/network-isolation.yaml \
    -e /home/stack/templates/containers-prepare-parameter.yaml \
    -e /home/stack/templates/network-environment.yaml \
    -e /home/stack/templates/ciscoaci_containers.yaml \
    -e /home/stack/templates/ciscoaci_containers_stein.yaml \
    -e /home/stack/templates/ciscoaci-config.yaml \
    -e /home/stack/templates/rhsm.yaml \
    -e /home/stack/templates/upgrades-environment.yaml -y
```

アップグレードの収束

『*FRAMEWORK FOR UPGRADES (13 TO 16.2) Red Hat*』ガイドの「*Synchronizing the overcloud stack*」セクションに、Cisco ACI 統合に固有のファイルが含まれていることを確認します。その内容は次のとおりです。

- カスタム ロール ファイル
- Cisco ACI OSP16 コンテナ マッピング Heat 環境ファイル
- Cisco ACI OSP15 コンテナ マッピング Heat 環境ファイル
- Cisco ACI 固有の構成環境ファイル

次に、Cisco 固有の yaml ファイルを使用した **upgrade converge** コマンドの例を示します。

```
source ~/stackrc
openstack overcloud upgrade converge \
    --templates /home/stack/tripleo-heat-templates \
    -r /home/stack/templates/custom_roles_data.yaml \
    -e /home/stack/tripleo-heat-templates/environments/network-isolation.yaml \
    -e /home/stack/templates/containers-prepare-parameter.yaml \
    -e /home/stack/templates/network-environment.yaml \
    -e /home/stack/templates/ciscoaci_containers.yaml \
    -e /home/stack/templates/ciscoaci_containers_stein.yaml \
    -e /home/stack/templates/ciscoaci-config.yaml \
    -e /home/stack/templates/rhsm.yaml \
    -e /home/stack/templates/upgrades-environment.yaml -y
```

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。