Cisco CloudCenter インテグレーション ガイド、リリース 4.6.x

初版:2016 年 11 月 7 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、米国シスコシステムズ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。

ACI

概要

CloudCenter のユーザは、設定済みアプリケーション プロファイルを使用して、アプリケーションのインフラストラクチャに依存しないモデルを作成できます。モデル化すると、Cisco CloudCenter プラットフォームと Cisco アプリケーション セントリック インフラストラクチャ（ACI）が連携し、自動化されたエンドツーエンドのプロビジョニングをアプリケーションのコンピューティング、ストレージ、およびネットワーク設定と、それらに必要な一連のコンポーネントに提供します。

ACI の基礎

ACI ポリシー モデルの詳細については、『Cisco ACI Fundamentals Guide』を参照してください。

可用性

CloudCenter と ACI の統合は **VMware** クラウド環境で使用できます。

CloudCenter 4.6 は次の APIC リリースをサポートしています。

- Cisco APIC リリース 1.0
- Cisco APIC リリース 1.1
- Cisco APIC リリース 1.2

利点

CloudCenter と ACI の統合により、次の利点がもたらされます。

- ACI のポリシー オブジェクトの完全に自動化された作成機能を使用する。
- プログラミングやアプリケーション コードの変更、またはクラウドに固有のスクリプトの記述、あるいは、ネットワークに関する特別な専門知識を必要とすることなく、ネットワークのマイクロセグメンテーションによるセキュリティと効率性を実現する。
- 完全に統合された Cisco ACI ネットワーク ポリシーと設定により、ユーザがアプリケーションをセルフサービスまたはオンデマンドによって導入し管理する。

統合の要件

CloudCenter プラットフォームは、オーバーレイ インフラストラクチャのエンドツーエンドでのプロビジョニングとアプリケーションの導入を自動化します。ACI では、これに次のリソースのプロビジョニングと管理が含まれています。

> CloudCenter で設定する APIC テナントにこれらのリソースを作成する権限があることを確認します。

- アプリケーション ネットワーク プロファイル（ANP）
- エンドポイント グループ（EPG）
- コントラクト
- サブジェクト/フィルタ

APIC でアプリケーションをプロビジョニングし、設定するための CloudCenter プラットフォームの前提条件として、まず、Cisco ACI の動作環境を設定するための次の要件を満たします。

- リーフ スイッチ プロファイル、スイッチ セレクタ、インターフェイス プロファイル、およびポリシー グループ
- VLAN プール
- VMware の仮想マシン マネージャ(VMM)ドメイン
- 外部インターネット接続用に L3 アウトで設定された新しいテナントとブリッジのドメインへのルーティングが可能な IP サブネット
- ルーティング プロトコル
- VRF

APIC の要件

HTTP と HTTPS の両方で動作する Cisco Application Policy Infrastructure Controller(Cisco APIC)機能。

- HTTPS:デフォルトでは、Cisco APIC は、UI と REST APIS のいずれでも HTTPS のみをリッスンします。APIC は、APIC ホスト名に対応する有効な SSL 証明書により設定されていることを確認します。
- HTTP:APIC の HTTP アクセスを有効にし、ホスト名または IP アドレスのいずれかを使用してアクセス可能であることを確認します。

環境の健全性を確保するには、次の手順を実行します。

1. APIC UI を使用して、EPG が 1 つある新しいアプリケーション ネットワーク プロファイルを手動で追加します。
2. 新しい VMware 仮想分散スイッチ(vDS)ポート グループがプロビジョニングされており、APIC UI に表示されることを確認します。
3. vCenter の UI を使用して、作成済みのポート グループをポイントするネットワークが設定された新しい VM をプロビジョニングするか、または複製します。
4. *ストリクト モード*で操作する場合、VM への SSH/RDP にはアクセスできません。
    a. ポート 22/3389 の*コントラクト*を EPG となるプロバイダーを使用してステップ 1 から作成します。
    b. ステップ 4a で作成した*コントラクト*が使用する新しい L3 アウト設定を作成します。
5. ステップ 3 で起動した VM に SSH/RDP でアクセスし、CloudCenter バンドル リポジトリと AMQP サーバにアクセスできることを確認します。

CloudCenter の要件

ACI の拡張機能で使用される CCO は、対応する APIC エンドポイントにアクセス可能である必要があります。

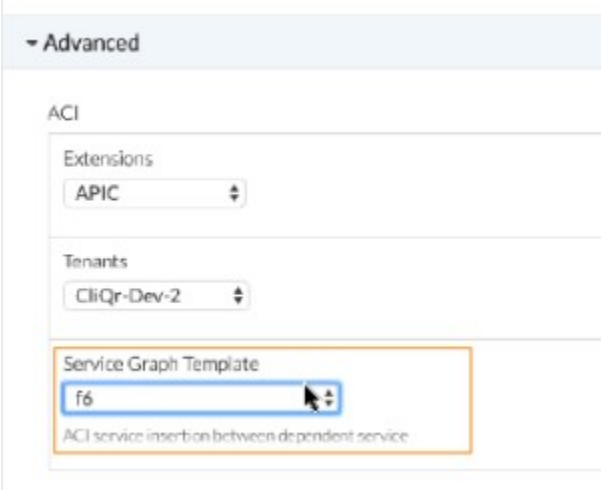
VMware vSphere の要件

要件	詳細
VMware vCenter 5.0/5.5 の動作環境	VMware vSphere の最小バージョンは v5.0 ですが、vSphere v5.5 U2 が最適です。
CloudCenter プラットフォームは、VMware のプライベート データセンターへの仮想マシンのプロビジョニングを自動化します。	CloudCenter プラットフォームで vCenter を設定するには、***アクセス クレデンシャル***が必要です。
すべての ESX ホストは、ACI リーフ スイッチに物理的に接続されている必要があります。	データセンターに必要なインストール要件は次のとおりです。 • 中規模のインスタンスを 10 以上実行できる物理 ESX ホスト • ESX クラスタ(クラスタが 1 つのホストのみで構成されている可能性もあります) • 100 GB 以上の空き容量があるデータストア(または DRS サポート用のデータストア クラスタ)
ESXi ホストが Cisco UCS ベースの場合	• CMM の VLAN を vNIC テンプレートにマッピングする必要があります。 • ファブリックからのアップリンクは、リーフ スイッチへのトランキング VLAN と相互接続する必要があります。

拡張機能の使用

ACI 拡張機能を作成して CCO の機能を拡張し、ACI 環境でネットワークをプロビジョニングすることができます。その後に、拡張機能を使用して次の CloudCenter リソースを設定できます。

- **導入環境フロー**:ACI の拡張機能は、導入環境でも統合されるため、各クラウド アカウントがその拡張機能を使用するかどうかを決定できます。CCM がクラウド プロバイダーに要求を行う必要はありません。詳細については、導入環境のデフォルトを参照してください。
- **アプリケーションの導入レベル**:アプリケーションを導入するときにアプリケーション プロファイルに組み込まれるテナントと VMM のドメインを設定します。[External Routed Network] フィールド(レイヤ 3 アウト)を APIC 設定に合わせて設定すると、そのテナント ネットワークに接続することができます。
- **外部サービスとしての ACI**:外部サービスを含んでいるアプリケーションを導入する場合は、このサービス層の [Advanced] セクションで ACI 拡張機能を設定すると、APIC サービス グラフ テンプレートを使用することができます。



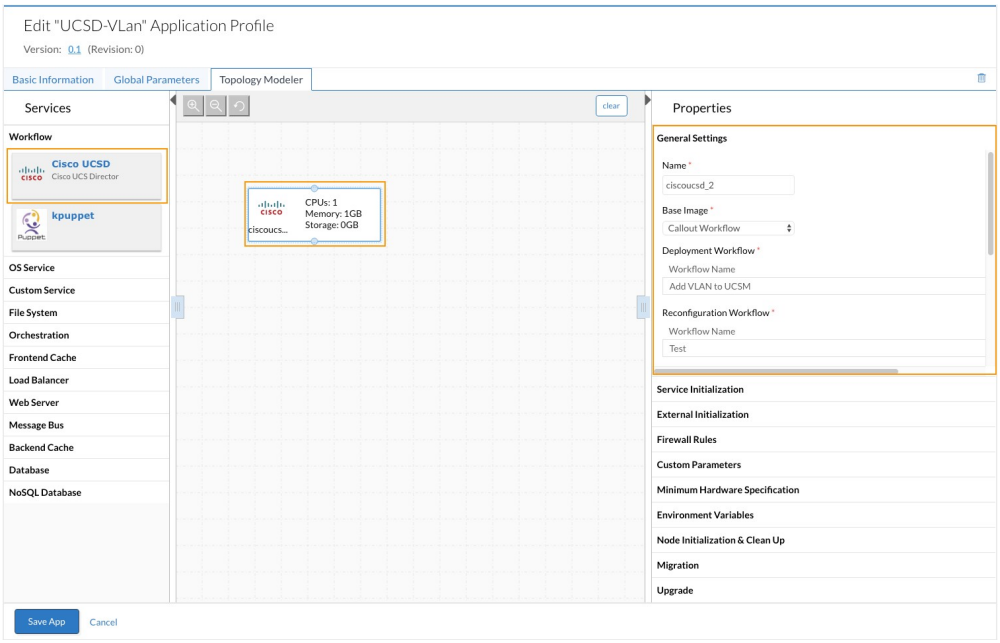
Cisco UCSD

- 概要
- 可用性
- 制限事項
- 統合の要件
- UCSD ワークフローのサポート
- カスタム サービスとしての UCSD の設定

概要

CloudCenter プラットフォームは、ユニファイド コンピューティング システム ディレクタ(UCSD)コールアウトのワークフローを呼び出せるようにする Cisco UCSD の統合を実装します。ユーザは Cisco UCSD サービスを CloundCenter のトポロジ モデラーにドラッグ アンド ドロップし、1 つまたは複数の UCSD コールアウトを使用してトポロジを作成できます。これにより、企業は UCSD コールアウトを使用するアプリケーションの混在トポロジを作成することができ、次の利点を得ることができます。

- 企業はガバナンスとワークフロー管理に CloudCenter を使用できる。
- システム管理者は UCSD を使用して物理ストレージをプロビジョニングできる。



可用性

CloudCenter 4.6 は、次の UCSD リリースをサポートしています。

- Cisco UCS Director リリース 5.1
- Cisco UCS Director リリース 5.2
- Cisco UCS Director リリース 5.3

制限事項

この統合を使用する場合は、次の制限事項に注意してください。

- USCD のこの統合の実装により、ネットワーク アプライアンスのストレージ設定をプロビジョニングできるようになる（CloudCenter プラットフォームについてテストおよび確認済み）。

  現在、UCSD では VM をプロビジョニングできます。

- この機能は、限定されたお客様に対してテストおよび実装されている。
- このバージョンでは、このページで明示的に説明した統合のみがサポートされている。
- この統合には Worker1 のアプライアンスは必要ない。
- この統合では、CCM アプライアンスおよび CCO アプライアンスは使用できない。

統合の要件

Cisco UCSD と統合するには、CloudCenter のシステム管理者は次の条件を順守する必要があります。

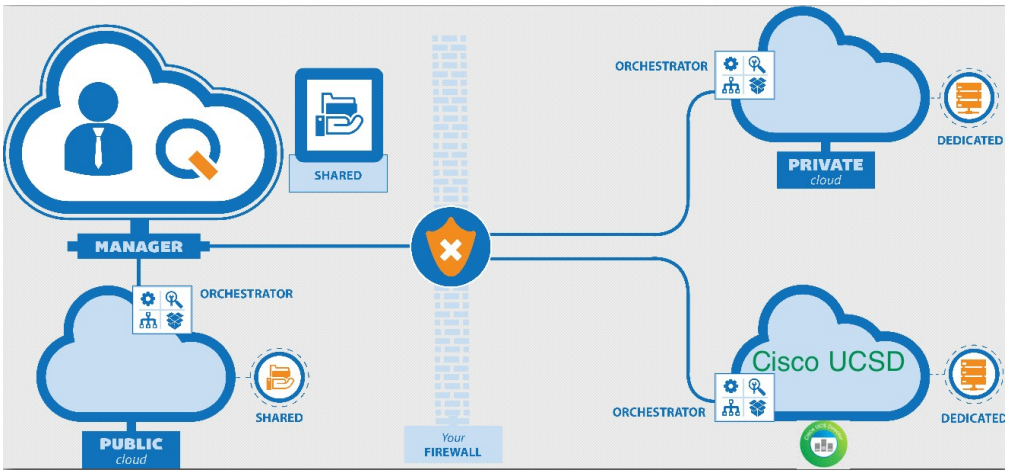
- Cisco UCSD アカウントと環境にアクセスする管理者機能。

  企業内で UCSD を統合する場合は、エンドツーエンドの導入を実行するための UCSD 環境に対するアクセス権が CloudCenter プラットフォームに必要です。

- CloudCenter プラットフォームによって呼び出される UCSD のワークフローのリストに関する知識。

  CloudCenter プラットフォームでは、UCSD が公開するコールアウト フローを使用するため、オーケストレーション プロセスを抽象化します。CloudCenter プラットフォームは、クラウドのガバナンスと管理を目的としてのみ、公開された UCSD パラメータを使用します。

- Cisco UCSD の各インスタンスは、アプリケーションをモデル化する前に、CCO と関連付ける必要がある。



- 現時点では、1 つの CloudCenter で 1 つの UCSD インスタンスをサポートする。
- CloudCenter の UCSD の各実装には、関連付けられた物理イメージのエントリが CloudCenter プラットフォーム内に必要です（このエントリは、論理イメージが使用されない場合でも、ダミーのプレースホルダとなります）。

UCSD ワークフローのサポート

UCSD には、ワークフローの概念があります。これらのワークフローは、企業や導入環境により異なります。一例として、UCSD の統合を使用してストレージをプロビジョニングする場合のワークフローでは、CloudCenter は現在、次のワークフローを呼び出します。

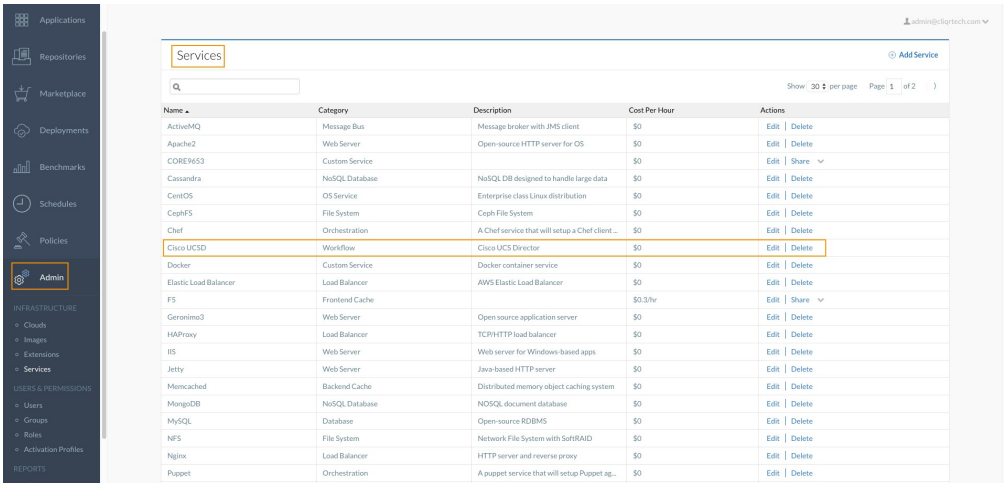
- 新しい記憶域の作成
- 記憶域の存在の検証
- 既存の記憶域の更新
- 記憶域の削除
- 記憶域に関する情報の取得

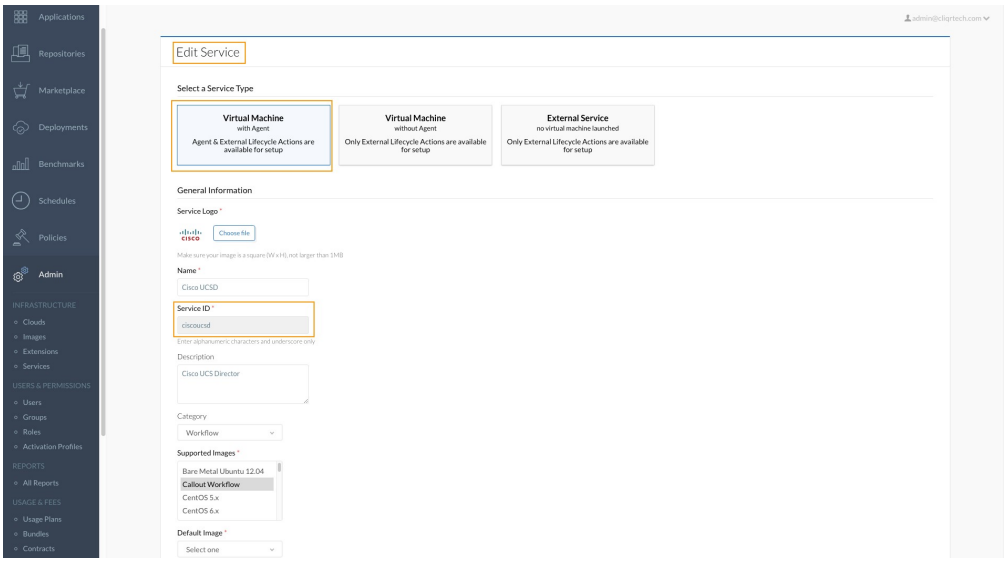
カスタム サービスとしての UCSD の設定

現在の UCSD のワークフローは、企業の追加のストレージ作業域の作成とメンテナンスに固有のものです。UCSD が 1 つの CCO に関連付けられているため、記憶域に関する情報にアクセスするたびに、CCO は許可された UCSD ワークフローを取得します。

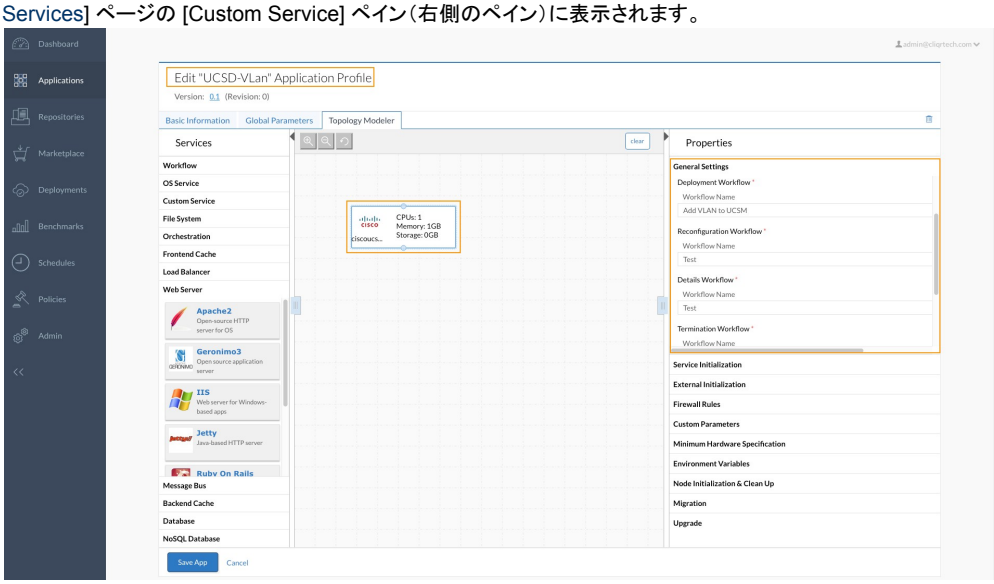
UCSD をカスタム サービスとして設定、定義、使用するには、次のプロセスを実行します。

1. UCSD クラウドを設定します。クラウドの設定を参照してください。
   [Cloud Type]：**UCSD**
   [Cloud Family]：**Cisco UCSD**
   [Region]：ユーザ設定が可能な名前
2. ダミーの UCSD 物理イメージを設定し（「イメージ ID の設定」を参照）、このイメージをマッピングします。
3. **Cisco UCSD** サービスを編集し、許可された UCSD ワークフロー パラメータを定義します。カスタム サービスの定義を参照してください。



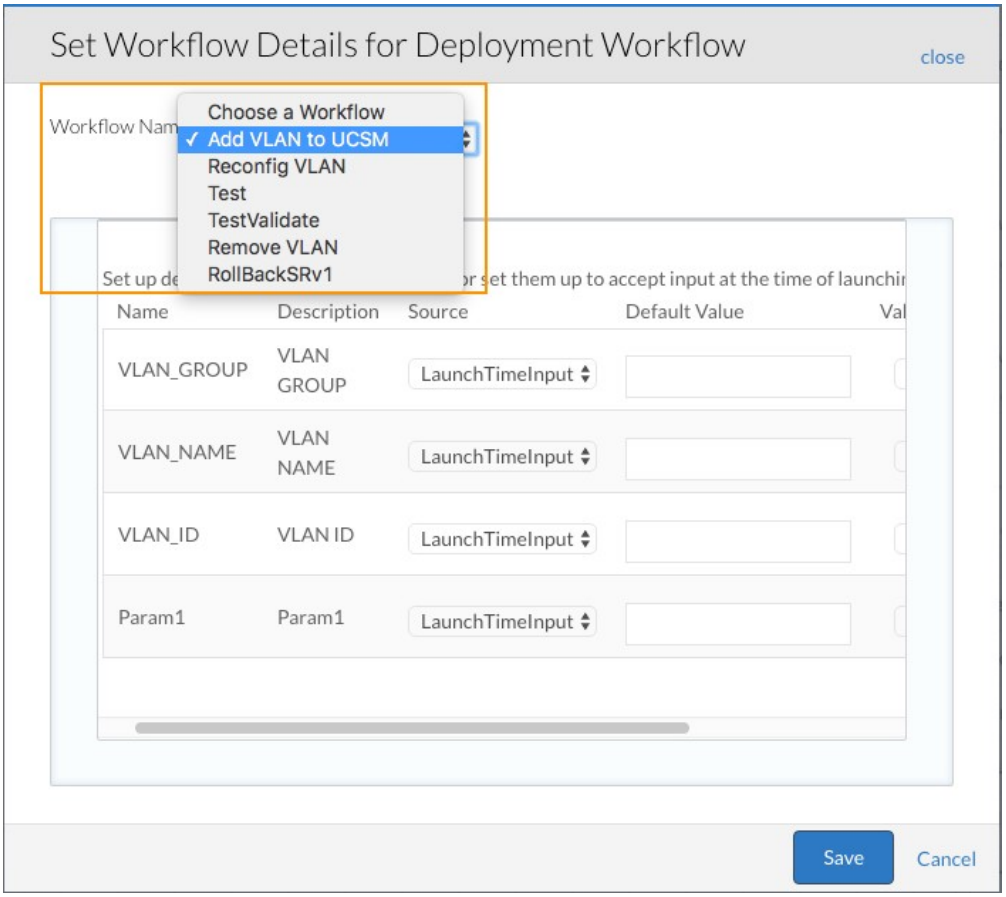


4. アプリケーションをモデル化し、UCSD サービスを使用します。UCSD サービスを選択すると、定義した UCSD サービスのパラメータが [Topology Services] ページの [Custom Service] ペイン（右側のペイン）に表示されます。

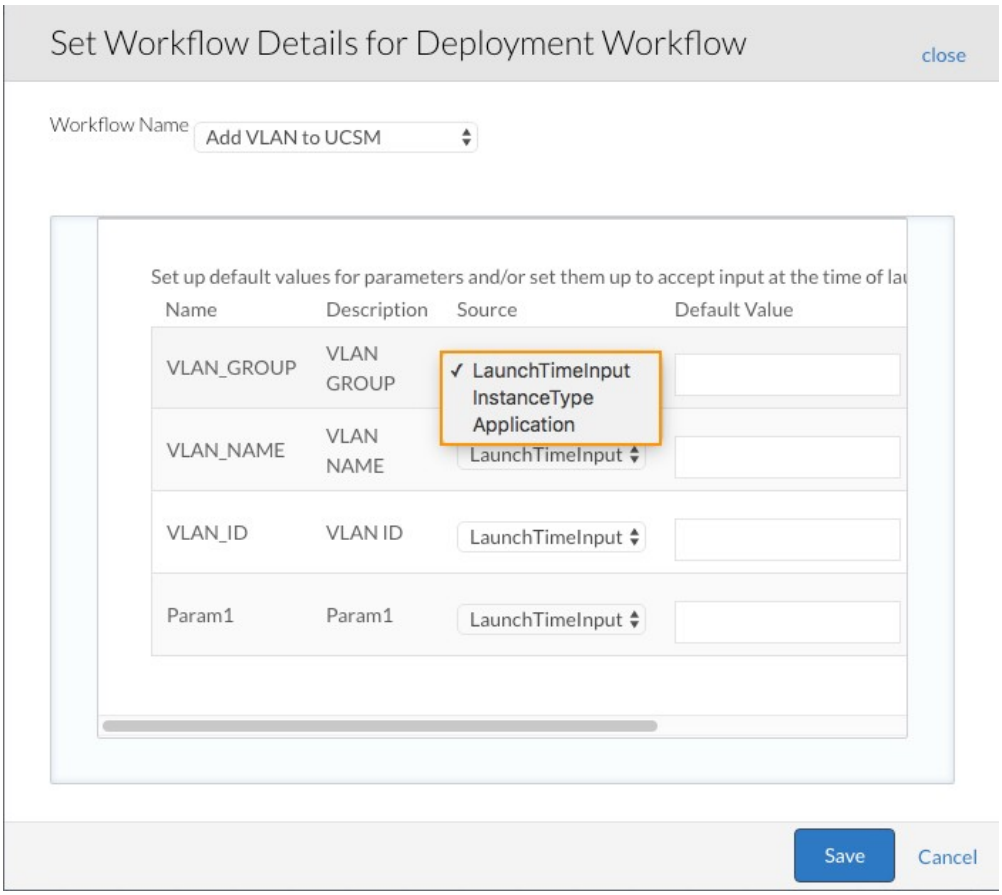


5. 必要に応じて、追加のグローバル パラメータを追加できます。[Topology Modeler] > [Global Parameters] を表示します。

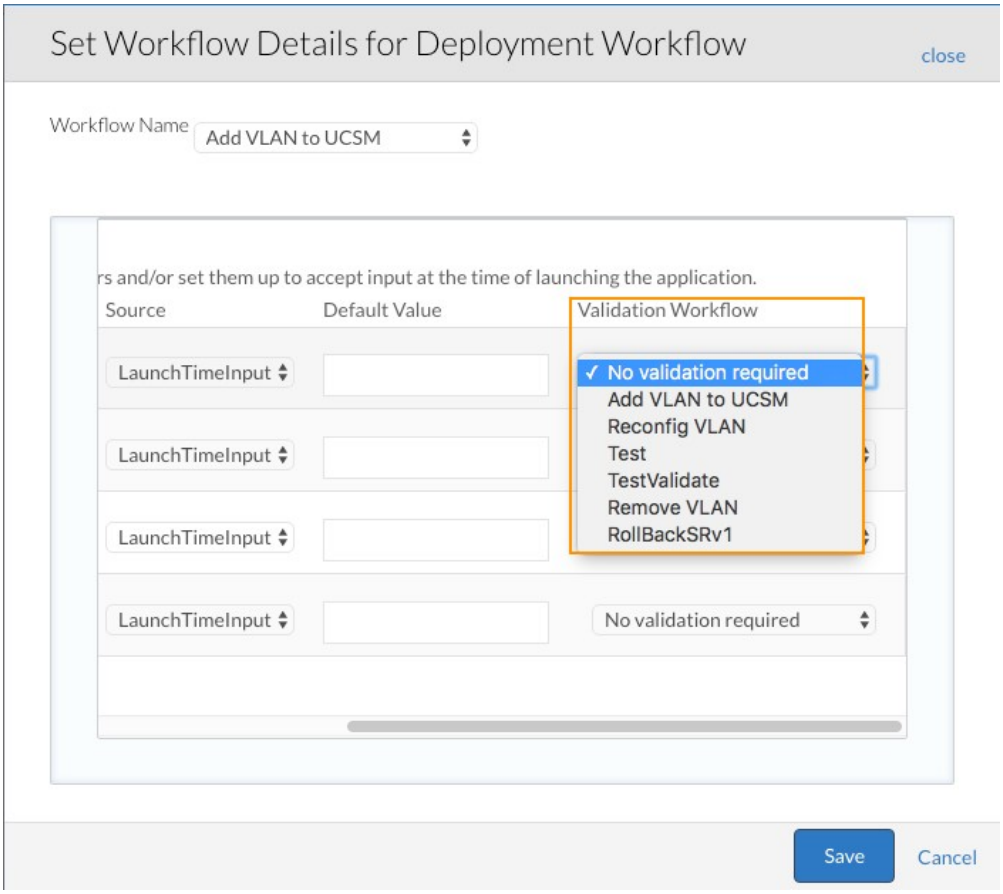
6. 設定済みの UCSD のワークフローのリストが [General Settings] セクションに表示されます。設定済みのワークフローをクリックすると、その特定のワークフローに関連付けられているパラメータが表示されます。



7. ドロップダウン リストからこのワークフローの各パラメータに ソースを選択します。
   - [LaunchTimeinput]：ユーザによるアプリケーション導入時に、このパラメータをオーバーライドできることを示します。
   - [InstanceType]：アプリケーション導入時にインスタンス タイプを選択すると、そのインスタンス タイプに関連付けられているストレージ サイズがこのパラメータの値になります。
   - [Application]：導入時に使用されるデフォルト値を入力します（オーバーライドできません）。



8. 該当する場合は、ドロップダウン リストから必要な [Validation Workflow] をパラメータごとに選択することもできます。このリストは、導入ごとに固有のものであり、該当するワークフローのみを表示するようにフィルタリングすることもできます。



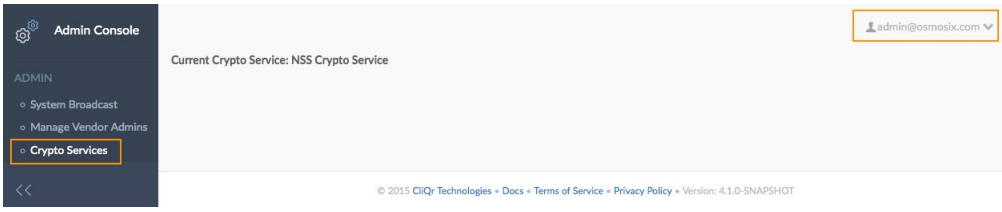
9. UCSD アプリケーションを起動します。

これで、UCSD をカスタム サービスとして設定し、起動しています。

CloudHSM

概要

CloudCenter プラットフォームは AWS クラウド ハードウェア セキュリティ モジュール（CloudHSM）をサポートします。このモジュールは、セキュア キー ストレージを提供し、不正開封防止ハードウェア モジュール内での暗号操作を可能にするハードウェア アプライアンスです。



要件

CloudHSM を使用するには、システム管理者としてログインする際に、次の要件に従う必要があります。

- Luna プロバイダー（lunaProvider.jar ファイル）用に CCM を設定し、このファイルが */usr/local/tom cat/lib* ディレクトリにコピーされており、Tomcat がこのライブラリを使用できるようにします。Tomcat を再起動して、lunaProvider ライブラリが自動的に使用されるようにする必要があります。
- 各テナントは、一意の暗号キーが必要です。
- CloudHSM サーバとやり取りする CCM インスタンスは、CCM と同じ VPC 内に存在している必要があります。
- CloudHSM への接続を再確立する前に、CCM をリブートします。

その他の参考資料

1. Safenet Luna WebHelp
2. AWS CloudHSM
3. AWS CloudHSM 開始ガイド
4. AWS CloudHSM フォーラム
5. Connecting Multiple VPCs with EC2 Instances（SSL）

SMTP メール サーバ

概要

ここでは、SMTP メール サーバ サーバを設定し、CCM を使用して電子メールを送信する方法について説明します。

> ⚠ CloudCenter プラットフォームは、TLS ポートをサポートしていません。CloudCenter プラットフォームは、SSL ポートのみをサポートして SMTP メール サーバを設定します。

> ⓘ パスワード文字列を暗号化形式で追加します。

メール プロパティ ファイル

Mail.properties ファイルは、ローカル Tomcat サーバ（/usr/local/tomcat/webapps/ROOT/WEB-INF/mail.properties）で入手できます。

```
# The hostname or IP address of your SMTP server
# Currently mob-gen.com email domain is hosted by gmail
# Gmail requires smtp over ssl, do not modify these settings
mail.smtp.host=smtp.gmail.com
mail.smtp.auth=true
mail.smtp.port=465
mail.smtp.socketFactory.port=465
mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
mail.smtp.socketFactory.fallback=false

# Email user to authenticate to gmail
mail.user.number=1

mail.user.1=<your_osmosix_email_addr>
mail.password.1=<your_email_password>
from.mail.user.1=<your_cloudcenterrtech_email_addr>
from.mail.username.1=
```

SMTP 設定プロセス

CCM の SMTP メール サーバを設定するには、次のプロセスを実行します。

1. mail.properties ファイルを開きます。
   vi /usr/local/tomcat/webapps/ROOT/WEB-INF/mail.properties
2. このファイル内で必要な設定を変更します。
   たとえば、Gmail を使用する場合は、次の行のみを変更します。

   ```
   mail.user.1=<your_osmosix_email_addr>

   mail.password.1=<your_email_password>

   from.mail.user.1=<your_cloudcentertech_email_addr>

   from.mail.username.1=
   ```

   同様に、mail.properties ファイルで必要な行を変更して、CCM からメールにアクセスできるようにします。
3. Tomcat を再起動します。
   /etc/init.d/tomcat restart

コールアウト スクリプト

概要

VM の導入ライフサイクルのさまざまな段階で、CloudCenter プラットフォームは、プロビジョニング プロセスの動作を制御する機能をサポートします。動作を制御する各種ライフサイクル ポイントを「*トピック*」と呼びます。動作はトピックに割り当てる*コールアウト*経由でスクリプトにより制御します。コールアウトに共通する使用例は、IP アドレスの割り当て（IPAM）トピック時に IP アドレスを取得し、IP シャットダウン トピック（IPAM2）時に IP アドレスの割り当てを解除するための IPAM ツールの照会です。この使用例の実装例については、「infoblox の統合」ページを参照してください。

コールアウトは CCO ごとに設定され、その CCO からプロビジョニングされたすべての VM に適用されることに注意してください。異なる動作が必要な場合は、コールアウト スクリプト内に制御ロジック（if/then/case）を使用できます。

各コールアウトは、コンフィギュレーション ファイル（callout.conf）と実行するスクリプトの 2 つの主要部分から構成されています。これらのファイルは、CCO の /usr/local/osmosix/callout/<name> パスに配置されています。使用するサブフォルダの名前は任意ですが、ベスト プラクティスはコールアウトの対象であるトピックの名前を使用することです。例：/usr/local/osmosix/callout/ipam/<files>

CloudCenter プラットフォームから実行すると、コールアウト スクリプトは、クラウドの種類、導入環境など、使用可能なさまざまな環境変数へアクセスすることができます。

> ⚠ 完全なリストは、/usr/local/osmosix/callout/<name>/logs/<timestamp> のコールアウト スクリプト ログで入手できます。

コールアウト スクリプトは同じパラメータと着信変数を使用します。各スクリプトは異なる変数を公開するため、同時に使用することはできません。必要に応じて任意のスクリプトを実行できます。

サポート対象のコールアウト トピック

これらのスクリプトのそれぞれについて、以下の項で説明します。

> ✓ 各スクリプトの説明に直接リンクしている前出の黙示を使用してください。

スクリプト トピック	説明	サポート対象のクラウド
vmNaming	各ノードを起動する前に呼び出されます。 このスクリプトは、各ジョブに（スクリプトに挿入された）すべての名前変数（アプリケーションの名前、階層の名前、選択したイメージ）を提供します。	OpenStack、VMware、および AWS。 サポート対象のクラウドの場合のサポート対象のコールアウトの詳細については、VM (Node) Name Config を参照してください。
ipam	ネットワークおよび OS に固有の設定	OpenStack、VMware、および AWS。
preDestroy	ノードが破棄される直前に呼び出されます。	OpenStack、VMware、および AWS。
ipamDealloc	ノードが破棄される直前に呼び出されます。	OpenStack、VMware、および AWS。

callout.conf のサポート対象の属性

このコールアウト スクリプトは、各行で <key>=<value> を使用して、標準的な Java プロパティ ファイル形式をサポートします。

コールアウト スクリプトの環境変数

コールアウト スクリプトは入力パラメータの環境変数を受け入れます。変数のリストはノードの種類によって異なります。次の表に例を表示します。

変数	サンプル値または種類
eNV_osName	Linux、Windows
eNV_vmName	vmNaming モジュールによって渡された文字列、または CloudCenter プラットフォームによって自動生成された文字列
eNV_JOB_ID	アプリケーション VM（のみ）を識別する整数
eNV_launchUserId	スクリプト/モジュールを起動する担当者のユーザ ID を識別する整数
eNV_launchUserName	スクリプト/モジュールを起動する担当者のユーザ名を識別する文字列

アプリケーション VM に対するすべてのジョブ アプリケーション設定も、eNV 変数として使用できます。詳細については、CloudCenter の定義済みパラメータを参照してください。

ベスト プラクティス
デバッグ レベルをオンにしてデバッグ ログを確認し（ログファイルの検出を参照）、使用可能なすべての変数のリストを表示します。

各コールアウト スクリプトの設定

Callout.conf ファイルで各スクリプトを個別に設定します。

これらのコールアウト スクリプトそれぞれを、***テナント レベルではなく***、リージョン レベルで CCO ごとに設定できます。次に、vmNaming コールアウト スクリプトを追加する設定手順の例を示します。

クラウド リージョンのコールアウトまたは属性を変更する場合は、CCO を再起動して変更を有効にする必要があります。

コールアウト 追加次のプロセスを実行します。

1. CCO に次のディレクトリを作成します。

   ```
   /usr/local/osmosix/callout/vmname/
   ```

2. このディレクトリに次のファイルを作成します。

   ```
   /usr/local/osmosix/callout/vmname/callout.conf
   ```

3. `Create a file for the script:`

   ```
   /usr/local/osmosix/callout/vmname/<script name>
   ```

4. アクセス許可を必ず実行します。

   ```
   chmod 777 <script>
   ```

5. Callout.conf ファイルでこのファイルを参照します。

vmNaming コールアウト スクリプト

vmNaming スクリプトのサポート対象の環境変数：

変数	サンプル値または種類
eNV_JOB_ID	整数（アプリケーション VM のみ）
eNV_launchUserId	整数
eNV_launchUserName	string

vmNaming スクリプトのサポート対象のキー：

CloudCenter の必須キー	説明
vmName	VM の名前

vmNaming コールアウト スクリプトの出力例：

run.sh

```
#!/bin/bash

echo "vmName=`uuidgen`"
```

このスクリプトで VM の名前を変更することができます。OpenStack、VMware、および AWS クラウドの CCM UI を使用した VM の名前変更については、VM（ノード）名の設定を参照してください。

IPAM コールアウト スクリプト

統合の一環として、IPAM を作成し、CCO の起動時に動的に呼び出されたコールアウト スクリプトを組み込みます。モジュールは動的にロードまたはリロード（自動ロード）したり、CCO 起動時にロードしたりできます。デフォルトでは、自動ロードは無効になっています。

IPAM モジュールのコールアウト スクリプトには、次のパラメータなどが含まれています。

- DNS server list
- DNS サフィックス リスト
- vNIC の数
- vNIC の IP アドレスの数
- vNIC のネットマスクの数
- VM name

スクリプトを実行すると、そのクラウドに対するすべての導入で IPAM モジュールが管理する IP アドレスが検出されます。

> IPAM との統合のためのコールアウト スクリプト オプションは VMware クラウドでのみ使用できます。

コールアウト スクリプトのパスは /usr/local/osmosix/callout で、各モジュールはスクリプト パスの下のサブフォルダです。

例

```
UserClusterName="cluster01"
eNV_Cloud_Setting_UserDataCenterName="dc02"
eNV_NumTasks="1" eNV_UseBatchTaskList="0"
eNV_Cloud_Setting_UserResourcePoolName="resourcepool1"
eNV_Cloud_Setting_UserClusterName="cluster01"
```

サポート対象のプロパティ

各コールアウト スクリプトに対して出力される複数のキーと値のペア。

Key	説明	必須かどうか
osHostname	ホスト名	はい
DnsServerList	DNS サーバ リスト（カンマ区切り）	
DnsSuffixList	DNS サフィックス リスト（カンマ区切り）	
nicCount	vNIC の数	はい
nicIP_n	n の vNIC の IP アドレスの数	はい
nicNetmask_n	n の vNIC のネットマスクの数	はい
nicGateway_n	n の vNIC ゲートウェイ（CCO） IP アドレスの数	
nicDnsServerList_n	n の vNIC の DNS サーバのリスト（カンマ区切り）の数	はい
nicUseDhcp_0	IPAM スクリプトの一部として、nicIP_n と nicDnsServerList_n にダミー値を指定します。ただし、これらの値は DHCP の設定によって上書きされます。	IPAM コールアウトを使用しており、アドレッシングが DHCP を使用するように割り当てられている場合は必須。
ANY	このプロパティは、`reinject` の設定が true の場合にサポートされます。 例：myCustomParam=myValue	

Linux に固有の IPAM のプロパティ

各コールアウト スクリプトに対して出力される複数のキーと値のペア。

Key	説明	必須かどうか
domainName	Linux ドメイン名 domainName を指定しない場合は、CloudCenter プラットフォームでデフォルト値の **mydomain** に設定します。 デフォルト値の **mydomain** を使用しない場合は、この値を必ず指定してください。	はい
hwClockUTC	H/W クロック の UTC	はい
timeZone	タイム ゾーン	はい

Windows に固有の IPAM のプロパティ

Key	説明	必須かどうか
timeZoneId	このタイム ゾーンの Windows インデックス ID。 Windows に固有の Vmware IPAM 設定スクリプトの場合、導入完了後に有効な変更のみが未定の期間にわたって表示される場合があることに注意してください。	はい
fullName	管理者ユーザの名前	はい
組織	組織の名前（文字列）	はい
productKey	Windows のプロダクト キー	はい
setAdminPassword	管理者パスワード	はい
changeSid	Microsoft SID に対する true または false の値 VMware に固有のスクリプトでは、Windows XP 以降の Windows バージョンを使用している場合は、changeSid オプションを必ず **true** に設定してください。	はい
deleteAccounts	true または false の値	はい
dynamicPropertyName	任意のプロパティの予約済みの名前の所有者	はい
dynamicPropertyValue	任意のプロパティの予約済みの値の所有者	はい
custSpec	導入のコールアウト ページで使用する VMware でのゲスト カスタマイズ仕様。	いいえ
domainName	自動的にドメインを結合するために使用	いずれかのドメイン値が欠落している場合は、ワークグループ キーが必要です。 3 つのドメイン値がすべて存在する場合、ワークグループは必要ありません。
domainAdminName	自動的にドメインを結合するために使用	
domainAdminPassword	自動的にドメインを結合するために使用	
workgroup	VM を配置するワークグループ。	

Windows に固有の例

run.sh

```
#!/bin/bash

echo "setAdminPassword=abcd"
echo "timeZoneId=10 *"
echo "fullName=Enterprise ABCD"
echo "organization=ABCD"
echo "productKey=..."
echo "changeSid=true"
```

IPAM コールアウト スクリプトの例

run.sh

```
#!/bin/bash

echo "DnsServerList=8.8.8.8,10.0.0.100"

echo "nicCount=2"
echo "nicIP_0=10.0.0.100"
echo "nicDnsServerList_0=1.2.3.4,5.6.7.8"
echo "nicGateway_0=10.0.0.1"
echo "nicNetmask_0=255.255.255.0"

echo "nicIP_1=192.1.0.100"
echo "nicDnsServerList_1=10.10.10.10"
echo "nicGateway_1=192.1.0.1"
echo "nicNetmask_1=255.255.255.0"

echo "domainName=test.org"
echo "hwClockUTC=true"
echo "timeZone=Canada/Eastern"
echo "osHostname=testhost1"
```

AWS および OpenStack の設定について
AWS と OpenStack のクラウド設定では、次のパラメータはサポートされていません。

- domainName
- hwClockUTC
- timeZone
- osHostname

nicCount パラメータでは、一度に 1 つの NIC の使用のみが許可されます。

VM の設定に複数の NIC が含まれている場合は、NIC ごとに 1 回、コールアウト スクリプトが呼び出されます。

preDestroy コールアウト スクリプト

preDestroy スクリプトでは、ネットワーク設定のプロビジョニング解除の*直前に*システムに通知できるようにします。

このスクリプトからの出力は通知にすぎないため、CloudCenter プラットフォームはこの通知を検索しません。

IPAM2 コールアウト スクリプト

IPAM2 スクリプトでは、環境をクリーンアップできるようにし、reinject 設定によってサポートされているカスタム プロパティのみと連動します。

IPAM2 の例

run.sh

```
#!/bin/bash

./delete_record_by_ip.sh $IP
```

このスクリプトからの出力は通知にすぎないため、CloudCenter プラットフォームはこの通知を検索しません。

コールアウト ワークフローの例

IPAM コールアウトの例

```
#!/usr/bin/env python
import infoblox, sys, requests, os, random
requests.packages.urllib3.disable_warnings()

#Assign command line arguments to named variables
hostname = os.environ['vmName']
domain = "vm.cloudcenter.com"
fqdn = hostname + "."+ domain
network = "10.49.18.0/23" #sys.argv[2]
netmask = "255.255.254.0"
gateway = "10.49.19.254"
dns_server = "10.48.112.33,10.52.112.19"

#Setup connection object for Infoblox
iba_api = infoblox.Infoblox('10.49.9.163', 'admin', 'infoblox', '1.4',
iba_dns_view='VM-view', iba_network_view='default', iba_verify_ssl=False)

try:
 #Create new host record with supplied network and fqdn arguments
 ip = iba_api.create_host_record(network, fqdn)
 print "DnsServerList="+dns_server
 print "nicCount=1"
 print "nicIP_0=" + ip
 print "nicDnsServerList_0="+dns_server
 print "nicGateway_0="+gateway
 print "nicNetmask_0="+netmask
 print "domainName="+domain
 print "HWClockUTC=true"
 print "timeZone=Canada/Eastern"
 print "osHostname="+hostname
 print "infobloxFQDN="+fqdn
except Exception as e:
 print e
```

Infoblox

概要

CloudCenter プラットフォームは、IP アドレス管理（IPAM）の統合をサポートし、導入に向けての IP アドレスの管理を行います。

ここでは、CCO 上での複数のコールアウト スクリプトの実行による、IPAM プロバイダーである Infoblox との統合について説明します。各コールアウト スクリプトの詳細については、「コールアウト スクリプト」を参照してください。

CloudCenter プラットフォームが IP アドレスと DNS 名の割り当てについてどのように Infoblox をサポートするかの簡単なデモンストレーションについては、Infoblox の統合に関するビデオを参照してください。

IPAM 統合の前提条件

クラウド サポートの詳細については、「コールアウト スクリプト」を参照してください。

> ⚠ クラウド リージョンのコールアウトまたは属性を変更する場合は、CCO を再起動して変更を有効にする必要があります。

IPAM モジュールの設定

1. モジュール テンプレートを取得して、**/usr/local/osmosix/callout** に保存するには、CloudCenter サポートにお問い合わせください。
2. テスト環境に応じて、コールアウト スクリプトを変更します。
3. スリープ ジョブをモデル化し、環境変数を追加し、ジョブを実行してコールアウト ログを確認し、変数が正しくエクスポートされることを確認します。
4. 多層 Web アプリケーションをモデル化し、環境変数を追加し、ジョブを実行してコールアウト ログを確認し、変数が正しくエクスポートされることを確認します。

Infoblox API

Infoblox モジュールと統合するには、Infoblox-API-Python モジュールを使用します。

次に、Infoblox-API-Python モジュールを使用するスクリプトの例を示します。このスクリプトには、**python-requests 2.5 が必要**です。また、このスクリプトはコールアウトから直接呼び出すことができます。

createHost.py

```
#!/usr/bin/env python
import infoblox, sys
#Check to see if command line included enough arguments.
if (len(sys.argv) < 3):
 print "Usage: createHost.py <fqdn> <network CIDR>"
 quit()
#Assign command line arguments to named variables
fqdn = sys.argv[1]
network = sys.argv[2]
#Setup connection object for Infoblox
iba_api = infoblox.Infoblox('10.110.1.45', 'admin', 'infoblox', '1.6', 'default',
'default', False)
try:
 #Create new host record with supplied network and fqdn arguments
    ip = iba_api.create_host_record(network, fqdn)
    print "nicCount=1"
    print "nicIP_1=" + ip
except Exception as e:
    print e
```

コールアウト コンフィギュレーション ファイル

次に、Infoblox アプリケーションとの統合コールアウトの例を示します。

callout.conf

```
name=infoblox
type=exec
topic=ipam
debug=true
executable=createHost.py
reinject=true
disabled=false
```

Jenkins

概要

事前にモデル化された Jenkins プロジェクト（Git/SVN からソース コードを取得する Maven など）では、CloudCenter Jenkins プラグインを使用して CloudCenter と統合できます。

このファイルを手動でコピーする必要はありません。Jenkins のユーザがこのプラグインを使用できるように、CloudCenter がダウンロード URL を提供します。ダウンロード場所については、CloudCenter サポートにお問い合わせください。

CloudCenter Jenkins プラグイン

CloudCenter Jenkins プラグインは、サポート対象のクラウド 上での導入をユーザが Jenkins サーバから直接開始できるようにして、Jenkins と CloudCenter 間の完全な統合を実現します。

さらに、ユーザは各階層に対してアップグレード スクリプトを指定することで、既存の導入をアップグレードできます。

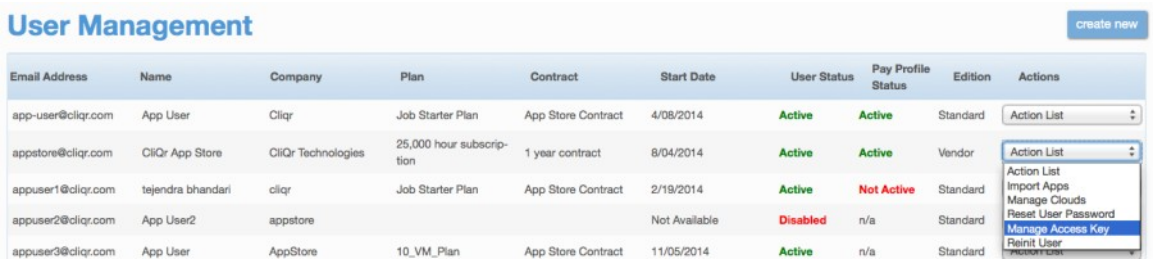
前提条件

- Jenkins に慣れていない場合は、ソース リポジトリとして Github を使用し、Jenkins で Maven プロジェクトを設定します。詳細については、http://www.youtube.com/watch?v=ITQGi5jzjvo を参照してください。
- CloudCenter Jenkins プラグインを使用するには、サポート対象の Jenkins バージョン値が **1.624** 以上である必要があります。
- Jenkins サーバに必要な Java のバージョンは **Java 7** である必要があります。

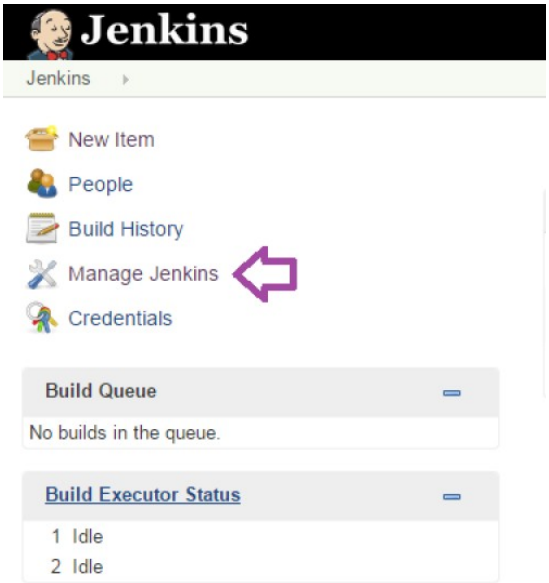
CloudCenter Jenkins プラグインのインストール

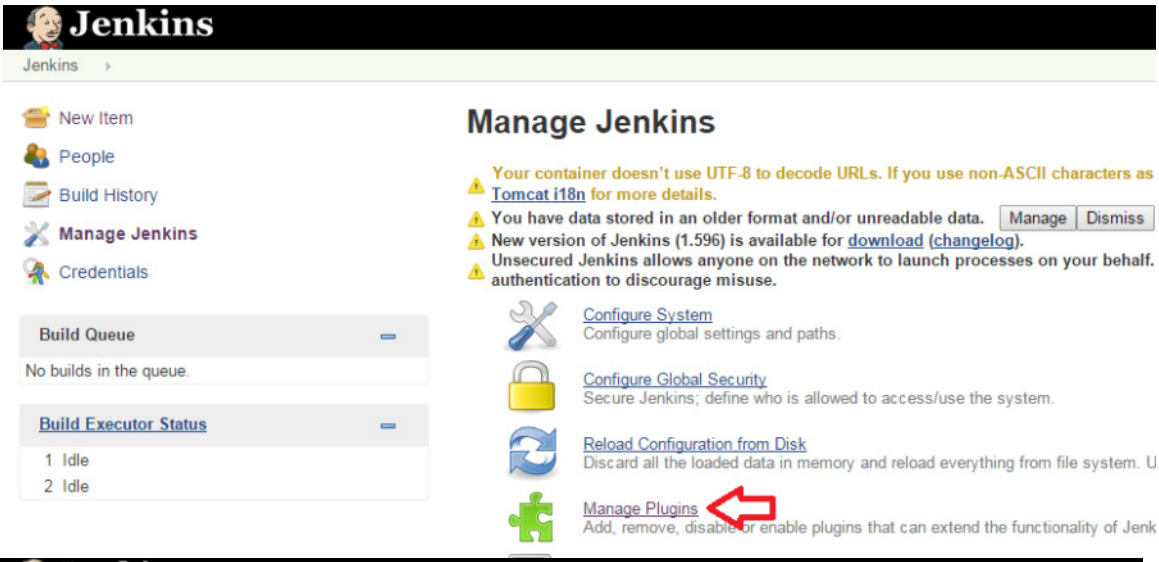
CloudCenter Jenkins プラグインをインストールするには、次の手順を実行します。

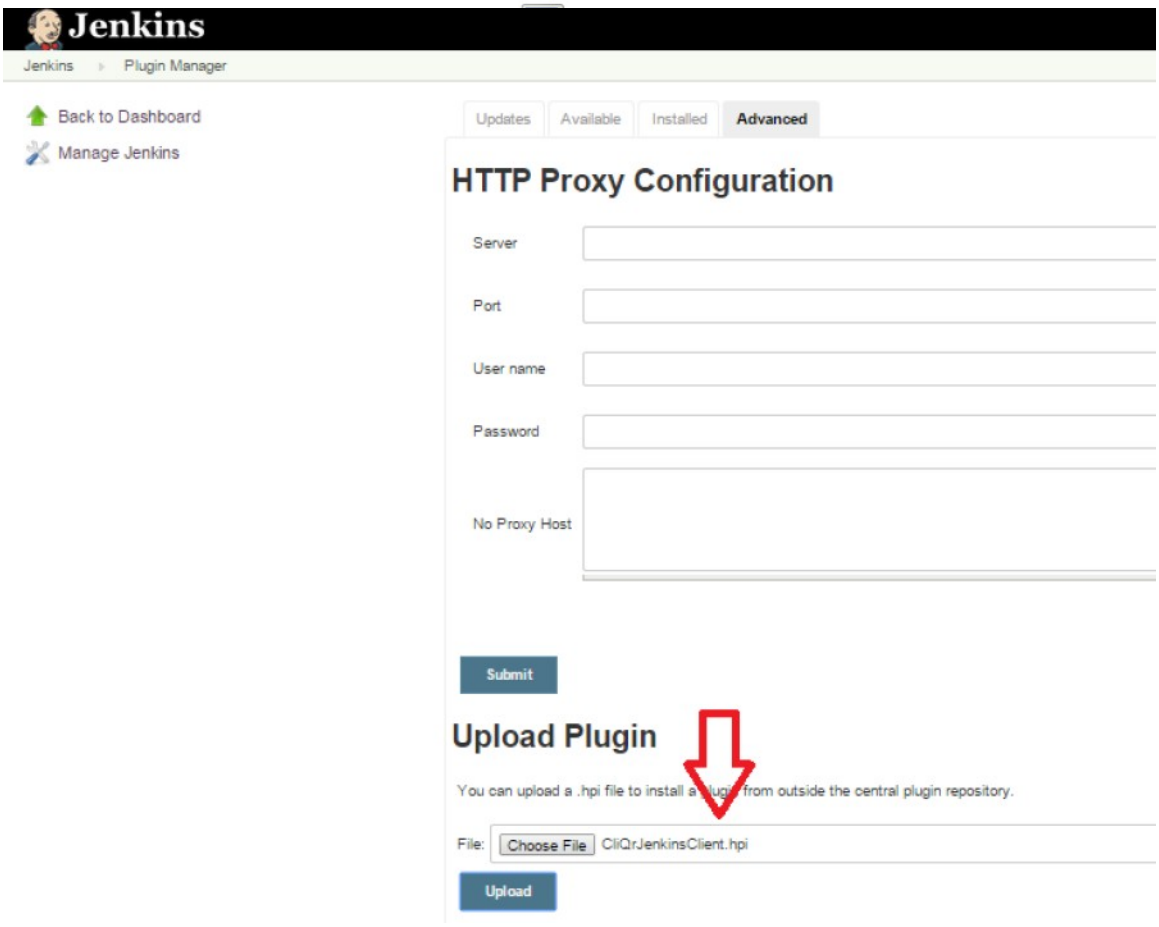
1. CloudCenter サポートに問い合わせて、CloudCenter Jenkins プラグインのダウンロード リンクを取得します。
2. 管理者クレデンシャルを使用して CCM にログインします。
3. Jenkins ユーザ用に API 管理（アクセス）キーを生成します。

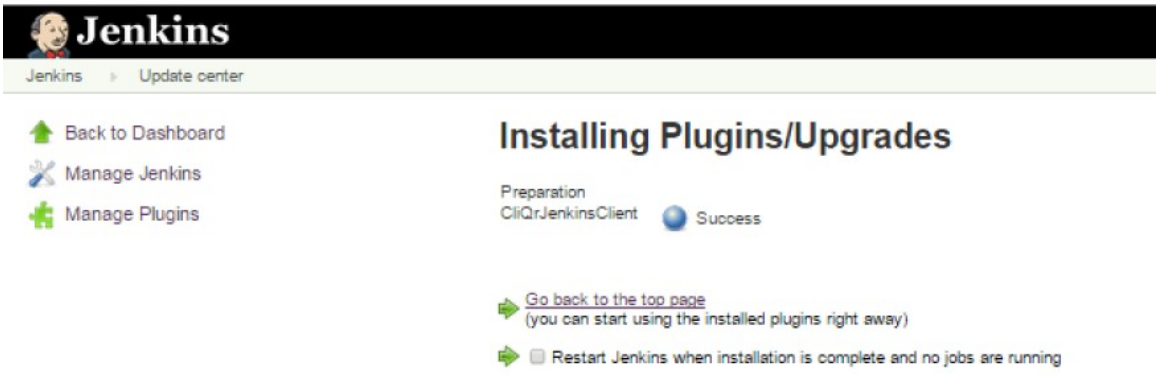


4. アプリケーションをモデル化し、このユーザが Jenkins ビルド サーバからアーティファクトにアクセスできるようにします。
5. Jenkins で、[Manage Jenkins] > [Manage Plugins] > [Advanced] > [Upload Plugin] に移動し、CloudCenter Jenkins プラグインをアップロードして、インストールします。

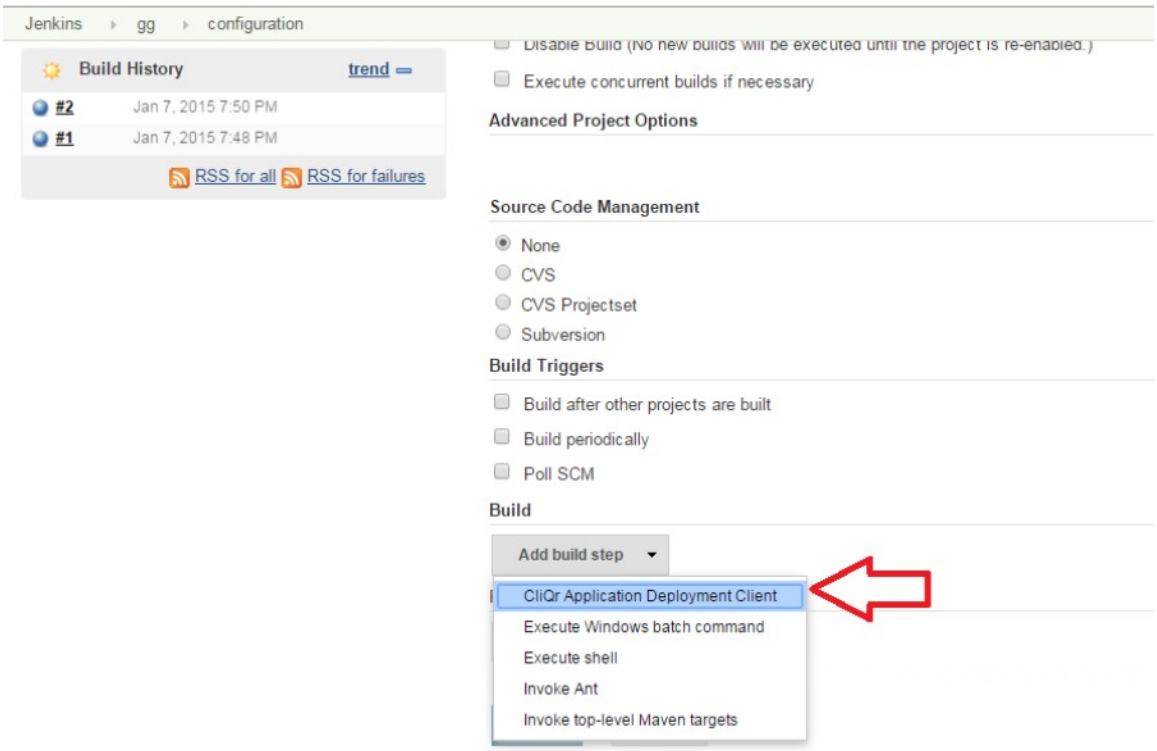



Jenkins
Jenkins
New Item
People
Build History
Manage Jenkins
Credentials
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Manage Jenkins
Your container doesn't use UTF-8 to decode URLs. If you use non-ASCII characters as
Tomcat i18n for more details.
You have data stored in an older format and/or unreadable data.
Manage
Dismiss
New version of Jenkins (1.596) is available for download (changelog).
Unsecured Jenkins allows anyone on the network to launch processes on your behalf.
authentication to discourage misuse.
Configure System
Configure global settings and paths.
Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.
Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system.
Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenk


Jenkins
Jenkins
Plugin Manager
Back to Dashboard
Manage Jenkins
Updates
Available
Installed
Advanced
HTTP Proxy Configuration
Server
Port
User name
Password
No Proxy Host
Submit
Upload Plugin
You can upload a .hpi file to install a plugin from outside the central plugin repository.
File:
Choose File
CliQrJenkinsClient.hpi
Upload


Jenkins
Jenkins
Update center
Back to Dashboard
Manage Jenkins
Manage Plugins
Installing Plugins/Upgrades
Preparation
CliQrJenkinsClient
Success
Go back to the top page
(you can start using the installed plugins right away)
Restart Jenkins when installation is complete and no jobs are running

6. CliQrJenkinsPlugin をインストールした後に、既存または新規のプロジェクトに移動してポストビルド ステップを設定し、Maven プロジェクトを使用して Git/SVN からソースコードを Jenkins ビルドに取得します。



7. CloudCenter アプリケーション導入クライアントを Jenkins に設定して、ビルド システムからの統合と導入（新規または既存ノードでのアップグレード）を続行します。

[CloudCenter Application Deployment Client] ページのパラメータのリストを次の表に示します。

パラメータ	説明
CloudCenter CCM URL	CCM サーバの IP アドレス。信頼できる証明書（詳細については、証明書の認証を参照してください）が CCM サーバに追加されていることを確認します（自己署名証明書 *ではなく*）。
Username	[Manage Access Key] セクションにリストされたユーザ名（詳細については API 認証を参照してください）。
AccessKey	[Manage Access Key] セクションにリストされたユーザの API 管理キー（詳細については、API 管理キーを参照してください）。
Deploy to Project	このフラグを使用して、一般的な導入環境ではなく、プロジェクトに展開します（詳細については、プロジェクトとフェーズを参照してください）。

パラメータ	説明
プロジェクト名	**Deploy to Project** フラグを選択した場合は、このパラメータが表示され、プロジェクトのリストが CCM サーバから取得されます。このパラメータ値に基づいて、このプロジェクトに関連付けられているアプリケーションのみが、[App lication Version] フィールドから除外されます。
Project Phase	プロジェクト名に基づいて、そのプロジェクトのすべてのフェーズのドロップダウン リストが表示されます。[Deployment Environment] フィールドにこの値を使用し、プロジェクトをフィルタリングします。
Application Name	CCM UI にリストされたアプリケーションのドロップダウン リストから、導入に必要なアプリケーションを選択します。 クレデンシャルを入力した後は、アプリケーション管理 API API のロードにしばらく時間がかかる場合があるため、その間は待機します。
Application Version	アプリケーションに基づいて、このドロップダウン リストからアプリケーションのバージョンを選択します。
Deployment Environment	アプリケーションの導入に必要な導入環境を選択します。CloudCenter は各導入のデフォルト設定（デフォルト クラウド、デフォルト インスタンス タイプなど）を使用するため、これらのデフォルト設定すべてを必ず確認してください。
Cloud Type	導入環境に存在するクラウド タイプのドロップダウン リストからクラウド タイプを 1 つ選択します。
AppParameters	グローバル パラメータとして渡すキーと値のペアのカンマ区切りのリスト。次に例を示します。 abcd=wow、cdef=cliqrRocks CloudCenter には、$BUILD_ID、$BUILD_NUMBER、$BUILD_TAG、$JOB_NAME が含まれています。また、使用可能であれば、他のプラグインの $BUILD_TIMESTAMP も含まれています。 このセクションで変数を追加して、他のジョブが共有しているパラメータを取得することができます。例： abc=${BUILD_PARENT_NUMBER} または abc=$BUILD_PARENT_NUMBER また、キーと値のパラメータ ペアが複数行含まれている $WORKSPACE/appParams ファイルからパラメータを取得するオプションもあります。それらのパラメータを使用すると、CCM UI に表示せずにパスワードやその他の機密情報を渡すことができます。
Binaries to be Copied	アーティファクト リポジトリ、外部リポジトリ、または外部ホストからコピーする必要があるファイル パスまたはフォルダ パスのカンマ区切りのリスト。
外部ロケーションへのバイナリのコピー	• External Host：外部 IP アドレスまたはリポジトリの IP またはパブリック DNS • HostUsername：このホストに対する SCP を可能にする SSSH ログイン ユーザ名 • Password：認証が実行される上記ユーザのパスワード。 • Target Folder on Above Host：ファイルまたはフォルダがコピーされる外部ホスト上のターゲット ロケーション。 /tmp/app1 が指定場所である場合、すべてのバイナリが /tmp/app1/latest で使用できるようになります。
Create a new Deployment on every Build	導入に対して一度しか実行できません。 このオプションではすべてのビルドにまったく新しい導入を作成します。
Update an Existing Deployment	導入に対して一度しか実行できません。 • 同じプロジェクトの以前のビルド時に CloudCenter Jenkins プラグインから起動した以前の導入を更新します。 • 以前の導入が進行中で（ジョブ）、実行状態になっていない場合、CloudCenter Jenkins プラグインは導入が実行状態になるまで待機してから更新をトリガーします（詳細については 導入と VM の状態を参照してください）。 • 以前の導入がエラーとなった場合、またはその導入が CCM サーバから停止またはキャンセルされた場合、CloudCenter Jenkins プラグインは更新プロセスの一部として新しい導入を起動します。 • これが最初のビルドである場合は、CloudCenter Jenkins プラグインが新しい導入を作成し、次の成功したビルドから既存の導入を使用します。 • UpdateScripts：ここに記載する順序で実行される tierName:scriptToExecute スクリプトのカンマ区切りのリスト。例：AppCluster:/shared/app/petclinic/update.sh, Database:/shared/app/updatemysql.sh, AppCluster:/shared/app/startServer.sh

jenkinsBuildId マクロ

更新スクリプトでは、更新導入時に*コピーするバイナリ*をポイントする引数として、$BUILD_ID または %jenkinsBuildId% を渡すことができます。

> %jenkinsBuildId% マクロは、アプリケーションに固有のマクロではありません。この CloudCenter で定義されたマクロは、Jenkins プラグインを使用して起動する導入に適用されます。

jenkinsBuildId マクロは、アプリケーションの導入の userenv に Jenkins のビルド ID を渡すために主に使用されます。Jenkins プラグインがトリガーする導入では userenv に自動的に jenkinsBuildId が設定され、リポジトリ/ストレージ内の正しいバイナリをポイントするために使用されます。たとえば、Web サーバに /shared/app/petclinic/latest/ に設定された以前の war ファイル パスがある場合、この war ファイル（petclinic.war）はこのマクロを使用して /shared/app/petclinic/ %jenkinsBuildId%/petclinic.war をポイントします。

更新導入の場合、既存の導入に渡す必要がある値を変更します。これは、userenv に導入時の古い jenkinsBuildId があるためです。

CloudCenter がターゲット ロケーションで作成するフォルダ名は、ランダムなタイムスタンプ値ではなく、jenkinsBuildId 値を使用するようになります。

Create a New Deployment on Every Build

このオプションは、すべてのビルドにまったく新しい導入を作成します。

Update an Existing Deployment

既存の導入を更新した場合は、次のようになります。

- 同じプロジェクトの以前のビルド時に Jenkins プラグインから起動した以前の導入を更新します。
- 以前の導入ジョブがまだ進行中で、実行状態になっていない場合、プラグインは実行状態になるまで待機してから更新をトリガーします。
- 以前の導入がエラー状態で終了するか、またはその導入が CCM から停止またはキャンセルされた場合、プラグインは更新の一部として新たな導入を起動します。
- はじめてのビルドの場合、このプラグインは新規導入を作成し、次に成功したビルドには、この既存の導入を使用します。
- セキュリティ グループ エラーになった場合は、複数のシステムが同じアカウントにアクセスしていることを必ず確認してください。

SSO

SSO AD の統合

概要

一部の企業には独自の Active Directory（AD）や、その他の類似するセットアップがあり、それらのクレデンシャルを使用して外部アプリケーションやプラットフォームにログインする場合があります。CloudCenter は直接的な AD 認証をサポートしていません。その代わりに、サービス プロバイダー（SP）としての CloudCenter と、ADFS などのお客様の ID プロバイダー（IDP）との間でシングル サインオン（SSO）を使用して統合をサポートします。

CloudCenter は各ベンダーがテナントとしてモデル化されているマルチテナント モデルをサポートします。テナントにはルート階層ツリーの構造が 1 つあります。各テナントには、一連の固有のユーザが存在します。それらのユーザの 1 人がテナント管理者（ルート管理者またはプラットフォーム管理者とも呼ぶ）であり、特殊な管理者権限を持っています。

CloudCenter は LDAP や AD に対して直接認証されません。

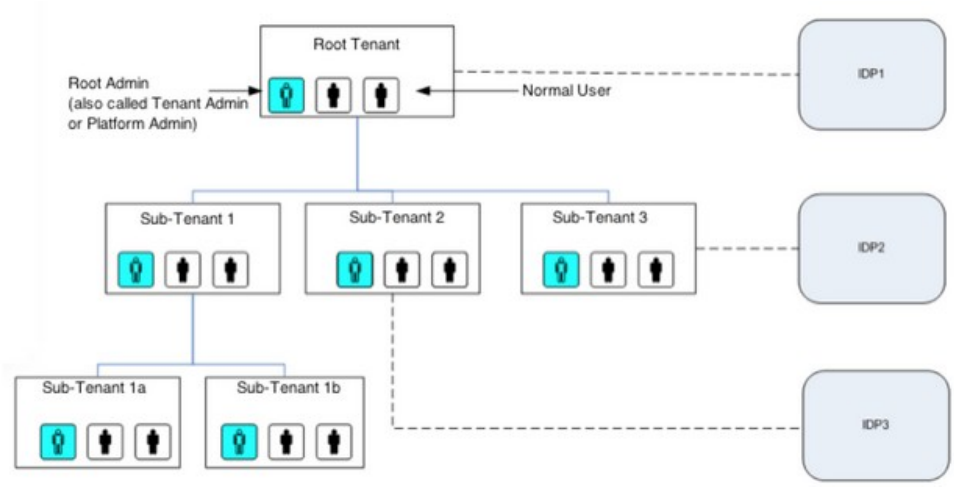
CloudCenter は、SAML 2.0 プロトコル（Ping ID、ADFS、Shibboleth など）をサポートする SSO ID プロバイダー（IDP）を通じて LDAP や AD とのみやり取りします。

CloudCenter を使用して SSO を実装するには、次が必要です。

1. CCM を設定して、認証を SSO IDP にリダイレクトする必要があります。
2. また、ユーザ アクティベーション プロファイルに対して、（SAML IDP によって返された）追加のユーザのカスタム プロパティをマッピングする必要もあります。
3. 次のステップを正常に実行すると、CloudCenter は適切なユーザ グループ メンバーシップと追加のロールおよび権限を自動的に割り当てます。

ユーザ認証

外部 ID プロバイダー（IDP）によって認証されるユーザ（ユーザ X）とは別に、ユーザ X には CCM VM のユーザ データベース内に対応する存在が必要です。SSO 環境では、ユーザ X が IDP によって認証され、最初に CCM VM を使用すると、プラットフォーム管理者がテナントとテナント管理者を作成していた場合においては、ユーザ X の認証が CCM ユーザ データベース内に作成されます。



各テナントは、次のように独自の SSO をポイントできます。

- 専用のエイリアス ホスト名を持ち、外部 IDP を使用してユーザを認証するように、各テナントを設定できる。
- 各テナントとユーザには、外部組織とユーザに関連付けるための *externalId* がある。

削除されたユーザの処理

IDP データベースからユーザを削除すると、削除されたユーザは CloudCenter にはログインできませんが、設定や関連付けられた依存関係は CloudCenter プラットフォームにそのまま維持されます。

SAML SSO の統合

- 概要
- CloudCenter サポート
- SAML 認証設定
- その他の参考資料

概要

CloudCenter は LDAP や AD に対して直接認証されません。

CloudCenter は、SAML 2.0 プロトコル（Ping ID、ADFS、Shibboleth など）をサポートする SSO ID プロバイダー（IDP）を通じて LDAP や AD とのみやり取りします。

CloudCenter を使用して SSO を実装するには、次が必要です。

1. CCM を設定して、認証を SSO IDP にリダイレクトする必要があります。
2. また、ユーザ アクティベーション プロファイルに対して、（SAML IDP によって返された）追加のユーザのカスタム プロパティをマッピングする必要もあります。
3. 次のステップを正常に実行すると、CloudCenter は適切なユーザ グループ メンバーシップと追加のロールおよび権限を自動的に割り当てます。

CCM インスタンスは Security Assertion Markup Language (SAML) 2.0 SSO を Spring Security SAML Extension を通じてサポートします。

システム管理者は、ルート レベルまたはテナント レベルで SAML 統合を設定できます。この統合を正確に設定するには、導入に応じて、ルート テナントまたはサブテナントに関する次の情報が必要です。

- テナント設定情報
- アクティベーション プロファイル情報

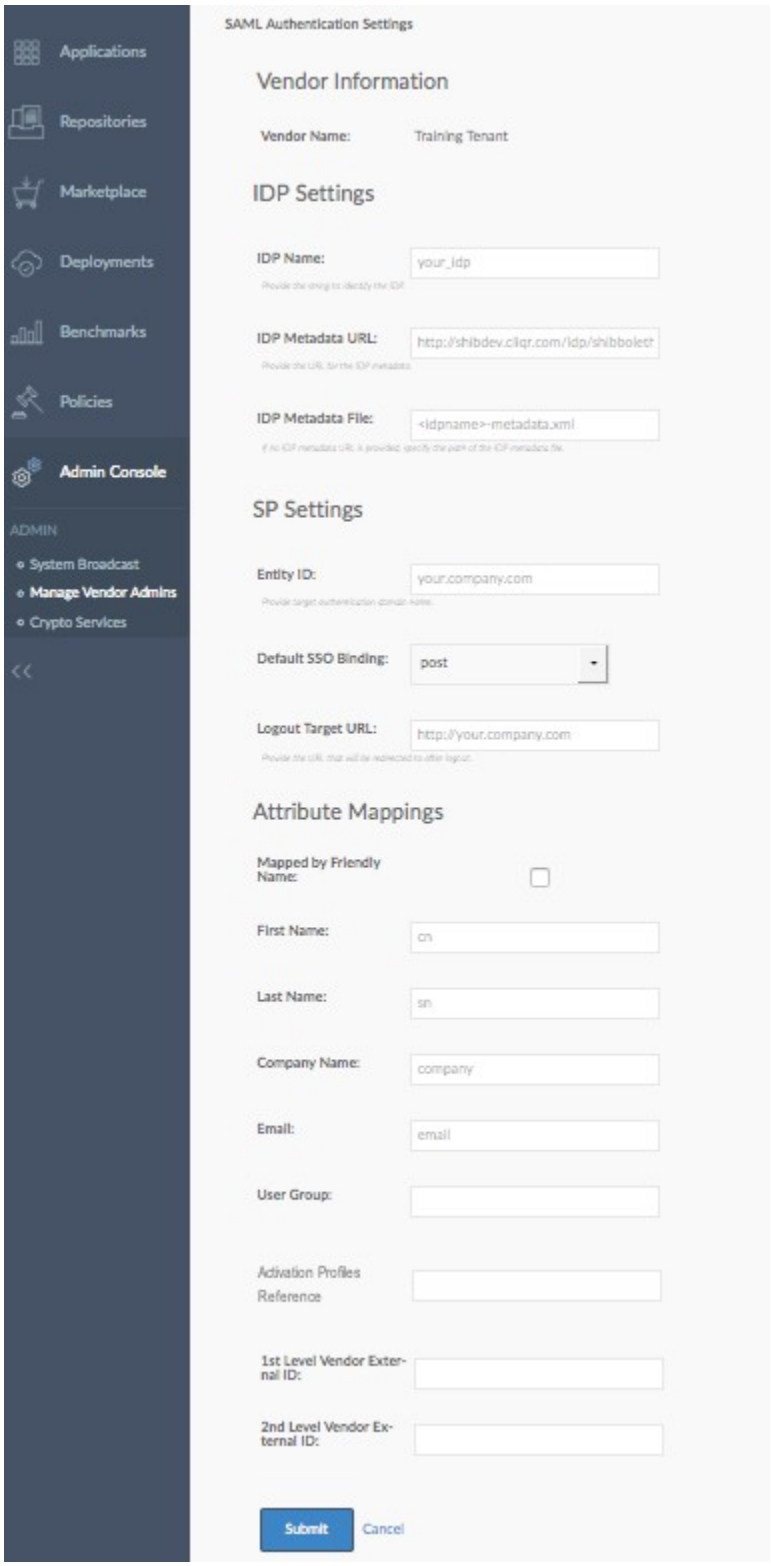
CloudCenter サポート

詳細については、CloudCenter サポートにお問い合わせください。

SAML 認証設定

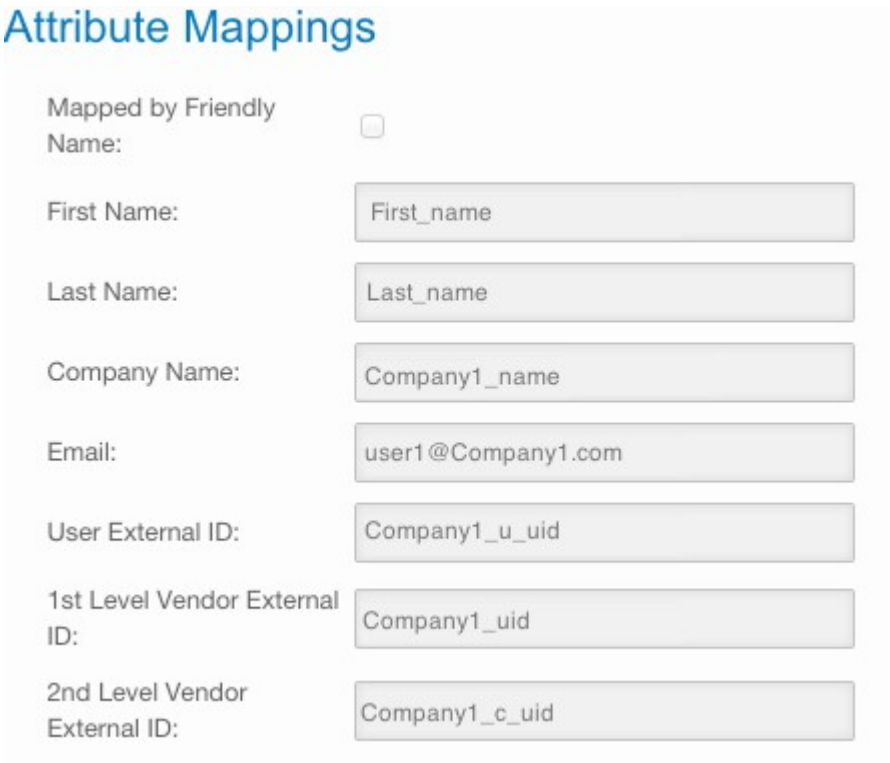
テナントを設定し、SSO を使用するには、次の手順を実行します。

1. テナントを作成します（サブテナントの設定を参照してください）。
   a. [Short Name]：空白文字および特殊文字を使用せずに文字列を入力します。
   b. [External Id]：テナントが関連付けられている外部システムの組織の ID を入力します。
   c. Tenant：そのテナントの CCM サーバのドメイン名。これは、SSO の観点から、サービス プロバイダー（SP）のエンド ポイントして機能します。
2. 新しく作成したテナント管理者としてログインし、アクティベーション プロファイルを作成します。
3. [Tenant Info] タブをクリックし、新しく作成したアクティベーション プロファイルをデフォルトのアクティベーション プロファイルとして選択します。
4. IDP 情報を収集して、IDP が提供するサブジェクト属性を確認し、ユーザ プロパティのマッピング プランに対して IDP サブジェクト属性を作成します。
5. システム管理者としてログインし（[Admin Users] > [Login as a System Admin] を参照）、[Manage Vendor Admins] タブをクリックして [Authentication Settings] アクション項目を選択します。
6. [IDP Settings] セクション、[SP Settings] セクション、および [Attribute Mappings] セクションに情報を入力して [Submit] をクリックします。



- [IDP Metadata]：CloudCenter と IDP 間相互の信頼関係を構築します。
- [Entity ID]：この認証のターゲット ドメイン名
- [Logout URL]：会社の SAML ページにログインする場合は、ログインしたユーザが SAML ページからログアウトする際にそのユーザをリダイレクトするページの URL を指定する必要があります。
- [Attribute Mapping] セクション：CloudCenter 内のユーザ属性にマッピングされる IDP からのフィールドです。これらのフィールドについて不明な点がある場合は、IDP に問い合わせてください。少なくとも、名、姓、電子メール アドレス、外部ユーザ ID は入力する必要があります。
- [User Group]：IDP からフィールド名を指定します。これは、ユーザが属するグループを決定するために使用されます。
- [Activation Profiles Reference]：デフォルト プロファイルの代わりに関連付けられたアクティベーション プロファイルを選択するには、メタデータで属性を指定します。
- [1st/2nd Level Vendor External ID]：SSO 自己署名ユーザの正しいテナントへの配置を自動化するために使用します。

- SSO ユーザの適切なテナントへの配置を自動化するためにサブテナントを作成する場合に、外部 ID の設定とともに使用します。次に例を示します。



  - 3 つのレベルの階層を持つ組織の場合：
    - 最上位組織 = **ルート組織**
      - 組織に関係なく各ユーザには一意のユーザ ID（UID）があります。
    - 第 1 レベルの組織（サブテナント）= **Company1**
      - **Company1** の各ユーザには一意の外部識別子、**Company1_uid** があります。
    - 第 2 レベルの組織（サブテナント）= **Company1_c**（それぞれの会社には複数の部門または顧客が存在する場合と、存在しない場合があります）
      - 各部門または顧客ユーザには一意の外部識別子、**Company1_c_uid** があります。

  - **Company1_uid** と **Company1_c_uid** の組み合わせを使用してユーザの組織を見つけます。
    - **Company1_uid** と **Company1_c_uid** が*ない*ユーザはルート組織に配置されます。
    - **Company1_uid** *のみ*のユーザは次のように配置されます。
      - 第 1 レベルのサブテナント
      - 外部 ID がユーザの **Company1_uid** と一致
    - **Company1_uid** と **Company1_c_uid** の*両方*を持つユーザは次のように配置されます。
      - 第 2 レベルのサブテナント
      - 外部 ID
        - 第 2 レベルのサブテナントがユーザの **Company1_c_uid** と一致
        - 第 1 レベルのサブテナントがユーザの **Company1_uid** と一致

- デフォルトの動作では、ベンダー ID にマッピングされたフィールドの値が空白の場合、ユーザは**ルート** テナントに配置されます。
- デフォルトのマッピングを設定し、[Vendor ID] フィールドに値を指定することによってユーザをブロックすることはできますが、テナントの値に一致しません。

> ⚠ 第 1 レベルと第 2 レベルのベンダーの値と、いずれのテナントの外部 ID にも一致しない外部 ID をマッピングした場合、それらのユーザにはアクセス権は与えられません。

7. SP メタデータをダウンロードし、それを IDP 管理者に送信して SP を登録します。

その他の参考資料

IDP ループの SP グループへのマッピングの例については、Ping ID ナレッジ ベースを参照してください。

Shibboleth SSO 統合の使用例

CloudCenter サポート

詳細については、CloudCenter サポートにお問い合わせください。

CloudCenter は LDAP や AD に対して直接認証されません。

CloudCenter は、SAML 2.0 プロトコル（Ping ID、ADFS、Shibboleth など）をサポートする SSO ID プロバイダー（IDP）を通じて LDAP や AD とのみやり取りします。

CloudCenter を使用して SSO を実装するには、次が必要です。

1. CCM を設定して、認証を SSO IDP にリダイレクトする必要があります。
2. また、ユーザ アクティベーション プロファイルに対して、（SAML IDP によって返された）追加のユーザのカスタム プロパティをマッピングする必要もあります。
3. 次のステップを正常に実行すると、CloudCenter は適切なユーザ グループ メンバーシップと追加のロールおよび権限を自動的に割り当てます。

Shibboleth のインストール

1. Tomcat 6 で Ubuntu のベース イメージを準備します。

   次に、Tomcat6 の手順を示します（Tomcat 7 には制限があるため）。

   Jetty 7 も使用できますが、Jetty を使用する場合は次の手順とは異なります。

2. 最新の Shibboleth IDP ソフトウェアを http://shibboleth.net/downloads/identity-provider/latest/ から /shib-distro/ にダウンロードします。
3. 次のようにアーカイブを解凍します。
   sudo unzip /opt/shib-disto/shibboleth.zip
4. 承認済みのディレクトリから Tomcat にファイルをコピーします。
   sudo cp /shib-disto/shibboleth/endorsed/* /usr/local/tomcat6/endorsed/".
   これを承認済みのディレクトリにする必要がある場合があります。
   sudo mkdir /usr/local/tomcat6/endorsed
5. Tomcat6-dta-ssl-1.0.0.jar をダウンロードし、TOMCAT_HOME/lib にコピーします。
   sudo cp /tmp/tomcat6-dta-ssl-1.0.0.jar /usr/local/tomcat6/lib
6. Shibboleth をインストールします。
   a. **cd /shib-distro/shibboleth/**
   b. **sudo ./install.sh**
   c. **Enter** キーを押してデフォルトのインストール パスである /opt/shibboleth-idp を承認します。
   d. **Enter** キーを押してサーバの完全修飾ドメイン名（FQDN）（idp01.cloudcenter.com など）を承認します。
   e. パスワードを入力して、キーストーンを作成します。
7. 属性要求の [Certificate] をオフにします（詳細については、属性要求に対する Web サーバ CA 検証の無効化を参照してください）。または、SP（サービス プロバイダー）（この使用例では CCM サーバ）からの証明書が Shibboleth で信頼されていることを確認します。
   a. ルートとして、java のセキュリティ プロバイダーである Shibboleth-AnyCert.jar ファイルを JRE lib/ext ディレクトリにコピーします。
   ```
   wget http://www.switch.ch/aai/downloads/Shibboleth-AnyCert.jar
   Ccp Shibboleth-AnyCert.jar ${JAVA_HOME}/jre/lib/ext/Shibboleth-AnyCert.jar
   ```
   b. Security.provider セクションに行を追加して、次のセキュリティ プロバイダーを ${JAVA_HOME}/jre/lib/security/java.security に追加します。
   ```
   security.provider.7=edu.internet2.middleware.shibboleth.quickInstallIdp.
   AnyCertProvider
   ```

   コネクタ全体を設定するときは、ステップ c. は、下記の手順 8 に含まれています。既存のトラストストアがない場合は、最初にトラストストアを作成する必要があります。

c. ${CATALINA_HOME}/conf/server.xml ファイルで、次の例に示すように、AnyCert トラストストア アルゴリズムに使用する、クライアント認証が必要な Attribute Authority Connector を設定します。

```
<Connector
    truststoreFile="/etc/shibboleth/truststore.jks"
    truststorePass="$TRUSTSTORE_PASSWORD$"
    truststoreAlgorithm="AnyCert"
/>
```

Tomcat がこのコネクタにトラストストアを使用することはないはずですが、ダミー証明書であっても、少なくても 1 つの証明書を含むトラストストアを指定することも必要です。

8. Tomcat server.xml にコネクタを追加します。
   a. **sudo vi /usr/local/tomcat6/conf/server.xml**
   b. 次のように、コネクタを追加します。「PASSWORD」は、インストール時に IDP に入力したパスワードに置き換えます。shibboleth を別の場所にインストールした場合は、赤で示したパスを必ず更新します。

```
<Connector port="443"
           protocol="org.apache.coyote.http11.Http11Protocol"
           SSLImplementation="edu.internet2.middleware.security.tomcat6.
             DelegateToApplicationJSSEImplementation"
           scheme="https"
           SSLEnabled="true"
           clientAuth="want"
           keystoreFile="/opt/shibboleth-idp/credentials/idp.jks"
           keystorePass="PASSWORD"
 truststoreFile="/opt/shibboleth-idp/credentials/trustore.jks"
 truststorePass="osmosix"
 truststoreAlgorithm="AnyCert"
/>
```

9. WAR ファイルを展開する ...webapps/ROOT/ の場所に WAR ファイルをコピーする代わりに、Shibboleth ではコンテキスト展開フラグメントの使用を推奨します。
   a. ファイル TOMCAT_HOME/conf/Catalina/localhost/idp.xml "sudo vi /usr/local/tomcat6/conf/Catalina/localhost/idp,xml を作成します。
   b. 次のコンテキスト情報を追加します。別の場所にインストールした場合は、赤で示したパスを更新します。

```
<Context docBase="/opt/shibboleth-idp/war/idp.war"
         privileged="true"
         antiResourceLocking="false"
         antiJARLocking="false"
         unpackWAR="false"
         swallowOutput="true" />
```

Shibboleth の設定

Shibboleth には基本的に 4 つのコンフィギュレーション ファイルがあります。すべてのファイルが /opt/shibboleth-idp/conf/ にあります。詳細については、IdPAuthUserPass を参照してください。

- attribute-resolver.xml では、認証ソース（この例では Active Directory）とユーザにプルする属性を定義します。
- attribute-filter.xml では、SP にリリースするユーザ属性と、リリース先の SP を定義します。
- Handler.xml では、ユーザ認証/セッションの処理方法を定義します。
- Login.config では、ユーザの認証方法を定義します。

1. cd /opt/shibboleth-idp/conf
2. login.config に次の行を追加します。赤で示された項目を更新して Active Directory の設定に合わせます。
   a. ホストは Active Directory のグローバル カタログです。フェールオーバーや冗長性を提供するために複数のサーバをスペースで区切って入力することができます。認証用に複数のサーバを使用する場合は、同じ一連のサーバを使用して属性を解決する必要があります。
   b. ベースは、検索ベースです。ドメイン名が反映されるように更新します。すべてのユーザが AD 内のデフォルトの Users フォルダに存在している場合は、cn=Users,dc=cloudcentertech,dc=local を使用できます。ユーザ アカウントがドメインの異なる領域にある場合は、ルートまたはディレクトリを使用する必要がある場合があります。
   c. このアカウントの username\password については、CloudCenter サポートにご連絡ください。
   d. 複数の LDAP（AD）ディレクトリに対して、または同じディレクトリ内で異なる検索ベースで認証を行う必要がある場合は、login.config を設定します。詳細については、IdPAuthUserPass を参照してください。
   e. ログイン ページの例は、*src/main/webapp/login.jsp* の IdP ディストリビューションにあります。ログイン ページをカスタマイズする方法については、IdPAuthUserPassLoginPage を参照してください。

```
hibUserPassAuth {
edu.vt.middleware.ldap.jaas.LdapLoginModule required
host="ad01.cloudcenter.com"
port="3268"
base="dc=cloudcentertech,dc=local"
ssl="false"
userFilter="sAMAccountName={0}"
serviceUser="saml@cloudcentertech.local"
serviceCredential="CloudCenterWin2day!"
subtreeSearch="true"
referral="follow";
};
```

3. Handler.xml を変更します。
   a. UsernamePassword LoginHandler を見つけ、有効にします。LoginHandler を有効にするセクションの後の <!-- before the section and the --> を削除します。

```
<LoginHandler xsi:type="UsernamePassword"
   jaasConfigurationLocation="file:///opt/shibboleth-idp/conf/login.config">
<AuthenticationMethod>rn:oasis:names:tc:SAML:2.0:ac:classes:
PasswordProtectedTransport</AuthenticationMethod>
</LoginHandler>
```

   b. RemoteUser Login Handler を見つけ、無効にします。コメント アウトするには、次のセクションの末尾に <!-- in front and --> を配置します。

```
<LoginHandler xsi:type="RemoteUser"
</LoginHandler>
```

4. attribute-resolver.xml を編集します。
   a. LDAP データ コネクタを追加し、ユーザおよびそれらの属性の解決に使用する Active Directory をポイントします。以下の情報を次のファイルに入力します。
   <!-- LDAP Connector -->
   一意の DataConnecter ID を選択し、複数のソースに対するユーザ認証を保証して、さらにコネクタを追加できるようにします。また、ユーザを解決するために属性を追加する場合は、ID を参照する必要もあります。

```
<resolver:DataConnector id="cloudcenterLDAP" xsi:type="LDAPDirectory" xmlns="urn:mace:shibboleth:2.0:resolver:dc"
ldapURL="ldap://10.100.1.220:3268" baseDN="dc=cloudcentertech,dc=local"
principal="saml@cloudcentertech.local"
principalCredential="CloudCenterWin2day!">
<FilterTemplate>
<![CDATA[
(sAMAccountName=$requestContext.principalName)
]]>
</FilterTemplate>
<LDAPProperty name="java.naming.referral" value="follow"/>

</resolver:DataConnector>
```

   b. 高度な設定（コネクタ、フィルタリングなどに冗長 LDAP サーバを使用）については、ResolverLDAPDataConnector を参照してください。
   c. 名前識別子の属性を追加します。ファイルには、transientId を使用するように属性がすでに設定されています。永続的 ID（sAMAccountName にマッピング）を設定する必要があります。これは、CloudCenter プラットフォームは外部 ID をこの属性にマップするためです（この ID が一時的なものである場合は問題を引き起こす可能性があります）。
   d. Dependency ref に作成したコネクタを参照していることを確認します。

```
<resolver:AttributeDefinition id="sAMAccountName" xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
        sourceAttributeID="sAMAccountName">

    <resolver:Dependency ref="cloudcenterLDAP" />

    <resolver:AttributeEncoder xsi:type="enc:SAML1StringNameIdentifier" nameFormat="urn:mace:shibboleth:1.0:nameIden
tifier"/>

    <resolver:AttributeEncoder xsi:type="enc:SAML2StringNameID" nameFormat="urn:oasis:names:tc:SAML:2.0:nameid-for
mat:persistent"/>

  </resolver:AttributeDefinition>
```

e. 該当するすべてのユーザ属性に属性定義リゾルバを追加します。
   - CloudCenter プラットフォームには、ユーザの名、姓、電子メール、およびユーザ ID（UID）の 4 つの属性が必要です。
   - ルート テナント レベルで SSO を設定し、第 1 レベルのサブテナントに対して機能させる場合は、ユーザが属しているべきテナントをマークする属性が必要です。
   - SSO を第 2 レベルのサブテナントに対して機能させるには、属性がもう 1 つ必要です。

f. 作成したデータ コネクタのしたにこれを追加して 4 つの必須属性をプルします。

> Dependency ref= の下に作成したコネクタを参照します。複数のコネクタを使用する場合は、属性リゾルバの複数のコピーで Dependency ref を変更します。

```
<resolver:AttributeDefinition id="mail" xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
sourceAttributeID="mail">
<resolver:Dependency ref="cloudcenterLDAP" />
<resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
name="urn:mace:dir:attribute-def:mail" />
<resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
name="urn:oid:0.9.2342.19200300.100.1.3" friendlyName="mail" />
</resolver:AttributeDefinition>

<resolver:AttributeDefinition id="givenName" xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
sourceAttributeID="givenName">
<resolver:Dependency ref="cloudcenterLDAP" />
<resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
name="urn:mace:dir:attribute-def:givenName" />
<resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
name="urn:oid:0.9.2342.19200300.100.1.3" friendlyName="givenName" />
</resolver:AttributeDefinition>

<resolver:AttributeDefinition id="sn" xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
sourceAttributeID="sn">
<resolver:Dependency ref="cloudcenterLDAP" />
<resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
name="urn:mace:dir:attribute-def:sn" />
<resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
name="urn:oid:0.9.2342.19200300.100.1.3" friendlyName="sn" />
</resolver:AttributeDefinition>

<resolver:AttributeDefinition id="uid" xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
sourceAttributeID="uid">
<resolver:Dependency ref="cloudcenterLDAP" />
<resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
name="urn:mace:dir:attribute-def:uid" />
<resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
name="urn:oid:0.9.2342.19200300.100.1.3" friendlyName="uid" />
</resolver:AttributeDefinition>
```

g. 必須のすべての属性（少なくても 4 つの必須 CloudCenter 属性）がグローバル カタログにパブリッシュされます。すべての属性が自動的に公開されるとは限りません。追加の属性を使用した場合は、それらの属性もグローバル カタログにパブリッシュされることを確認します。

h. AD 内のデフォルト属性を参照し、Global Catalog カラムを組み込むと、デフォルトでグローバル カタログに存在するかどうかを確認することができます。

i. 属性の複製を変更するには、グローバル カタログと一部の属性セットを参照してください。

j. 新しいスキーマ クラスまたは属性の定義を追加します。

k. SSO を設定して複数の AD ドメインからユーザを認証する場合は、IdPMultipleLDAP を参照してください。

5. attribute-filter.xml を編集します。
   a. 次の AttributeRules を追加して、SP に解決する属性（この例では CloudCenter）をリリースします。AttributeFilterPolicy の後に追加して一時 ID を任意のユーザにリリースします。
   b. 他の属性をプルした場合は、ルールを追加してそれらもリリースします。

```
<afp:AttributeRule attributeID="sAMAccountName">
<afp:PermitValueRule xsi:type="basic:ANY"/>
</afp:AttributeRule>

<afp:AttributeRule attributeID="mail">
<afp:PermitValueRule xsi:type="basic:ANY"/>
</afp:AttributeRule>

<afp:AttributeRule attributeID="sn">
<afp:PermitValueRule xsi:type="basic:ANY"/>
</afp:AttributeRule>

<afp:AttributeRule attributeID="givenName">
<afp:PermitValueRule xsi:type="basic:ANY"/>
</afp:AttributeRule>

<afp:AttributeRule attributeID="description">
<afp:PermitValueRule xsi:type="basic:ANY"/>
</afp:AttributeRule>
```

6. Tomcat を再起動します。
   cd /usr/local/tomcat6/bin ./shutdown.sh
   ./startup.sh

ADFS SAML SSO の統合

概要

CloudCenter は LDAP や AD に対して直接認証されません。

CloudCenter は、SAML 2.0 プロトコル（Ping ID、ADFS、Shibboleth など）をサポートする SSO ID プロバイダー（IDP）を通じて LDAP や AD とのみやり取りします。

CloudCenter を使用して SSO を実装するには、次が必要です。

1. CCM を設定して、認証を SSO IDP にリダイレクトする必要があります。
2. また、ユーザ アクティベーション プロファイルに対して、（SAML IDP によって返された）追加のユーザのカスタム プロパティをマッピングする必要もあります。
3. 次のステップを正常に実行すると、CloudCenter は適切なユーザ グループ メンバーシップと追加のロールおよび権限を自動的に割り当てます。

CCM インスタンスは Security Assertion Markup Language (SAML) 2.0 SSO を Spring Security SAML Extension を通じてサポートします。

システム管理者は、ルート レベルまたはテナント レベルで SAML の統合を設定できます。この統合を正確に設定するには、導入に応じて、ルート テナントまたはサブテナントに関する次の情報が必要です。

- テナント情報
- アクティベーション プロファイル情報

ドメインおよびポータルの確認

次の情報が正確であることを確認します。

- CCM のタイムゾーンと時刻（および関連付けによるその他すべてのアプライアンス）が AD ドメイン コントローラと一致している。
- FQDN ポータル ページのログイン（https://cloud.core.enterpise.com など）が正確である。

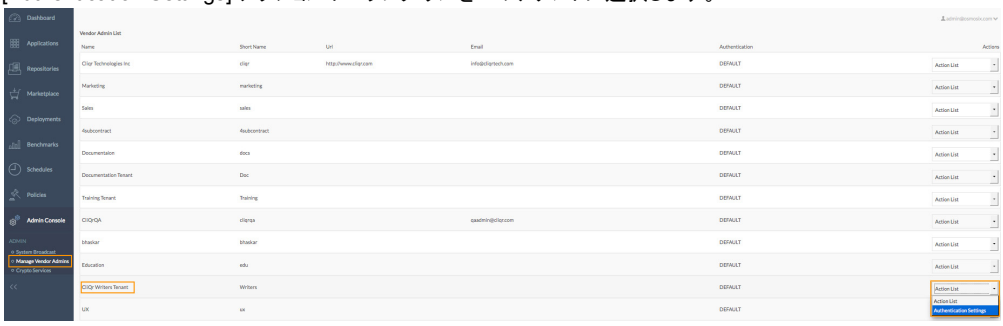
CloudCenter サポート

詳細については、CloudCenter サポートにお問い合わせください。

SAML 認証設定

テナントを設定し、SSO を使用するには、次の手順を実行します。

1. テナントを作成します（サブテナントの設定を参照してください）。
    a. [Short Name]：空白文字および特殊文字を使用せずに文字列を入力します。
    b. [External Id]：テナントが関連付けられている外部システムの組織の ID を入力します。
    c. Tenant：そのテナントの CCM サーバのドメイン名。これは、SSO の観点から、サービス プロバイダー（SP）のエンド ポイントして機能します。
2. 新しく作成したテナント管理者としてログインし、アクティベーション プロファイルを作成します。
3. [Vendor Info] タブをクリックし、新しく作成したアクティベーション プロファイルをデフォルトのアクティベーション プロファイルとして選択します。
4. システム管理者としてログインし（[Admin Users] > [Login as a System Admin] を参照）、[Manage Vendor Admins] タブをクリックして [Authentication Settings] アクション ドロップダウンをこのテナントに選択します。



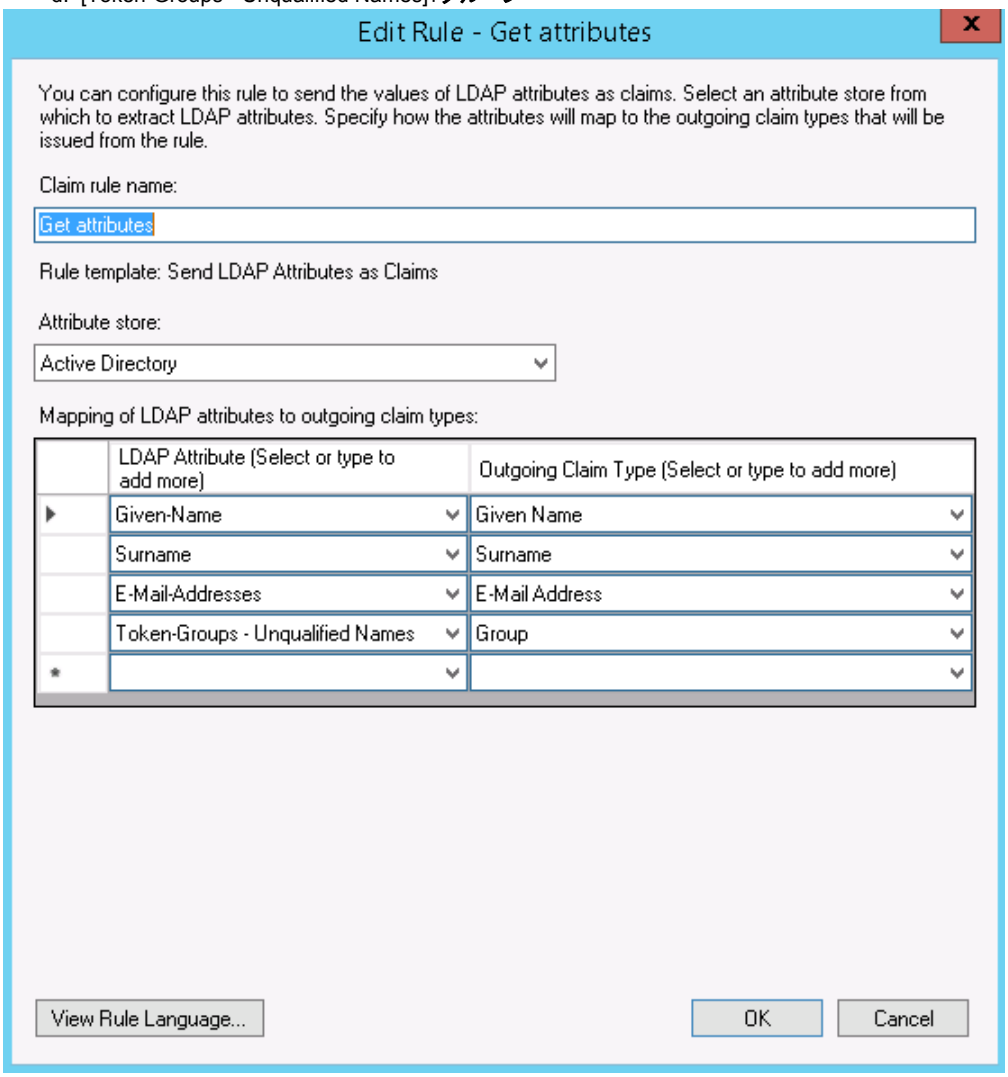
5. [IDP Settings] に情報を入力します。
    a. [IDP Name]（名前の例ではサポート対象の AD ドメインを示しています）
    b. [IDP Metadata URL]：CloudCenter プラットフォームと IDP 間の相互の信頼関係を確立します（現在、HTTPS はサポートされていないため、HTTP を使用します）。
    c. [IDP Metadata File]（該当する場合）。
6. [SP Settings] に情報を入力します。
    a. [Entity Id]：この認証のターゲット ドメイン名（ログイン ページの DNS 名が必要です）
    b. **デフォルトの SSO** バインディングは、ポスト時はそのままにします
    c. [Logout Target URL]：会社の SAML ページにログインする場合は、ログインしたユーザが SAML ページからログアウトする際にそのユーザをリダイレクトするページの URL を指定する必要があります（[Entity ID] と同じである可能性があります）。
7. [Attribute Mappings] セクションに情報を入力します。これらは、CloudCenter プラットフォーム内のユーザ属性にマッピングされる IDP からのフィールドです。これらのフィールドについて不明な点がある場合は、IDP に問い合わせてください。少なくとも、名、姓、電子メール アドレス、外部ユーザ ID は入力する必要があります。
    a. **名**のマッピングを入力します（http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname）。
    b. **姓**のマッピングを入力します（http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname）。
    c. **電子メール**のマッピングを入力します（http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress）。
    d. **ユーザ グループ**のマッピングを入力します（http://schemas.xmlsoap.org/claims/Group）。
    e. メタデータ ファイルをダウンロードします。
8. [Submit] をクリックします。`

ADFS の信頼設定

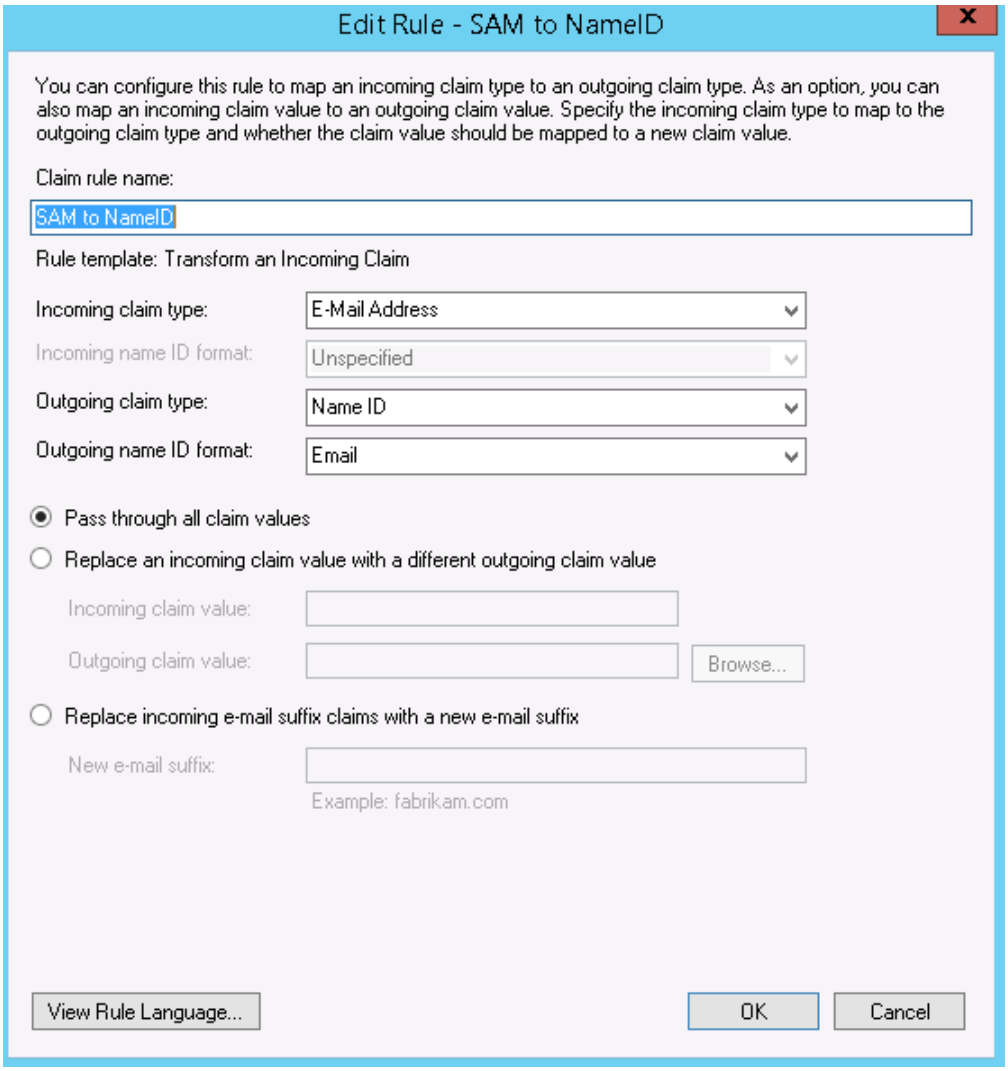
ADFS の信頼設定を行い、対応する要求規則を編集するには、次の手順を実行します。

1. AD FS Manager で、[AD FS] > [Trust Relationship] > [Relying Party Trusts] を選択し、[Add Relying Party Trust] をクリックして Add Relying Party Trust ウィザードを開きます。
2. [Welcome] ページで [Start] をクリックします。
3. [Select Import Data from a file] ページで、sp-xxxxx.xml を参照し、選択します。
4. [Next] をクリックします。
5. [Display name] に表示名を指定します。
6. [Next] をクリックします。
7. [Configure Multi-factor Authentication Now?] ページで [I do not want to configure multi-factor authentication settings for this relying party trust at this time] を選択します。
8. [Next] をクリックします。
9. [Choose Issuance Authorization Rules] ページで、[Permit all users to access this relying party] を選択します。
10. [Next] をクリックします。

11. [Ready to Add Trust] ページで、新しい信頼当事者証明のプロパティを入力し、[Next] をクリックして、信頼当事者証明情報を保存します。
12. [Finish] ページで、[Close] をクリックします。このアクションが [Edit Claim Rules] ボックスに自動的に表示されます。
13. [Properties] をクリックします。
14. [Advanced] タブの [Secure hash algorithm] で [SHA-1] を選択し、[OK] をクリックします。
15. リストで、要求規則を作成する信頼をクリックします。
16. 選択した信頼を右クリックし、[Edit Claim Rules] をクリックします。
17. [Select Rule Template] ページの [Claim rule template] でリストから [Send LDAP Attributes as Claims] をクリックし、[Next] をクリックします。
18. [Configure Rule] ページの [Claim rule name] の表示名フィールドに ***Get Attributes*** と入力します。
19. [Mapping of LDAP attributes to outgoing claim types] で、次の **LDAP 属性**を選択し、対応する**出力方向の要求の種類**タイプをドロップダウン リストから選択します。
    a. [Given-Name]：姓名の**名**
    b. [Surname]：姓名の**姓**
    c. [E-Mail-Addresses]：**電子メール アドレス**
    d. [Token-Groups - Unqualified Names]：**グループ**



20. [OK] をクリックします。
21. 別の規則を入力方向の要求のトランスフォーム テンプレートに追加します。[Select Rule Template] ページの [Claim rule template] で、リストから [Transform an Incoming Claim] を選択し、[Next] をクリックします。
22. NameID に対して規則に SAM という名前を指定して、次の値をマッピングします。
    a. [Incoming claim type]：**電子メール アドレス**
    b. [Outgoing claim type]：**名前 ID**
    c. [Outgoing name ID format]：**Email**



23. [OK] をクリックします。

これで、ADFS SAML SSO の統合が設定されました。

ServiceNow

- アーキテクチャの概要
- 統合の概要
- インストールおよび設定

CloudCenter：ServiceNow の統合のアーキテクチャの概要

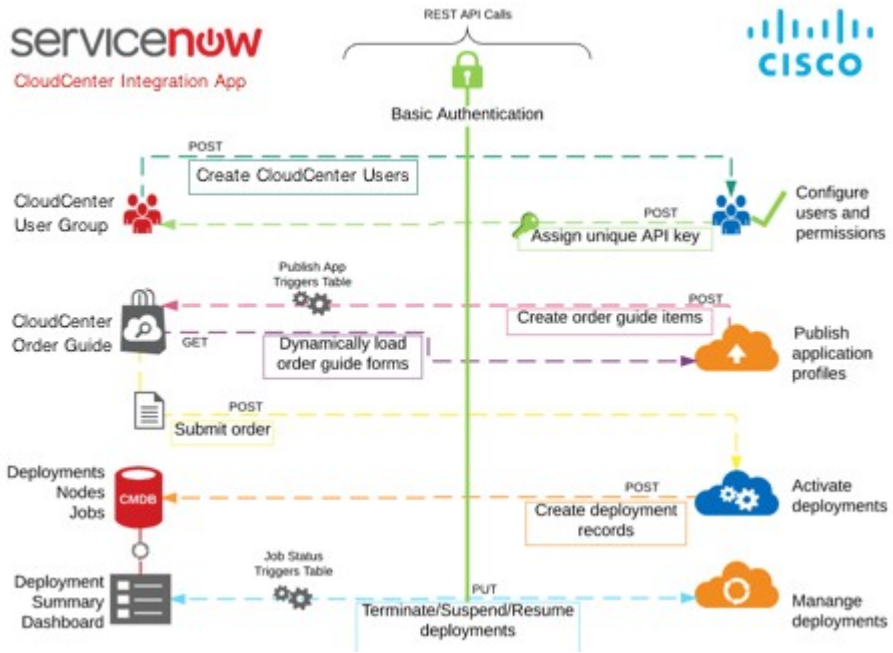
- 概要
- アーキテクチャ
- 統合の要件
- 機能

概要

Cisco Cloud Center には ServiceNow によって認定された統合アプリケーションがあり、ServiceNow の App Store からダウンロードできます。統合アプリは、ServiceNow の CMS 内と、「プライベート アプリケーション スコープ」空間内で開発されます。これによって ServiceNow 環境の他の部分に対するリソースの可用性が制限されますが、統合アプリがグローバル スコープに影響を与えず、顧客の ServiceNow 環境の全体的な整合性を維持するようにします。

アーキテクチャ

次の図に、Cisco CloudCenter と ServiceNow 間の通信を示します。すべての通信が Rest API コールを通じて行われます。



- **ユーザの作成**:ServiceNow のユーザ レコードは、設定プロセス時に設定されたグループに追加されます。ユーザがグループに追加されると、アウトバンド REST が実行され、対応するユーザ レコードを CloudCenter に作成してアクティベートし、CloudCenter API への後続のコールの認証に使用される API キーが ServiceNow ユーザに割り当てられます。
- **パブリッシュ**:アプリケーションが CloudCenter から ServiceNow にパブリッシュされると、ServiceNow に対してインバンド REST コールが実行され、アプリケーション プロファイルが作成されます。ServiceNow への最初のパブリッシュの後、アプリケーションが CloudCenter で更新されている場合は、管理者は ServiceNow にアプリケーションを再パブリッシュする必要があります。
- **注文**:アプリケーションがカタログを通じて要求されると、関連付けられた導入の設定項目(CI)と論理 CI グループが作成されます。その後、アウトバウンド REST コールが実行され、導入 CI が更新され、関連付けられたノードとジョブ CI が作成されます。
- **導入管理**:CloudCenter に導入関連の更新がある場合は、インバンド REST コールが実行され、それらの更新が関連付けられた CCI にプッシュされます。

統合の要件

コンポーネント	要件	詳細
ServiceNow	バージョン サポート	統合アプリケーションでは、ServiceNow がジュネーブまたはヘルシンキに存在している必要があります。ヘルシンキのサポートは統合アプリのバージョン 1.2 から開始されました。
	ポート	CloudCenter Manager は、MID サーバが使用されていない場合は、SSL を介して ServiceNow に直接到達でき、またその逆も可能である必要があります。
	MID サーバ(任意)	MID サーバのサポートは、ServiceNow から CCM へのインバンド トラフィックに使用できます。CCM から ServiceNow へのアウトバンド トラフィックは直接使用できる必要があります。
Cisco CloudCenter	CloudCenter の動作環境	詳細については、「Installation」を参照してください。
	バージョン サポート	統合アプリケーションは、Cisco CloudCenter 4.6.0 以降をサポートしています。
	ポリシー定義	実装ガイドに定義されているとおり、アプリケーションのパブリッシュにはアクション ポリシーを Cisco CloudCenter に作成する必要があります。
	テナント管理者の API キー	ServiceNow から CloudCenter への API 通信の場合。

機能

- **MID Server**:統合は、MID サーバの有無を問わず、ServiceNow 環境をサポートします。
- **アプリケーション セキュリティ**:すべての API キーが*パスワード(2 方向暗号化)*内部タイプを使用して暗号化されます。CloudCenter へのコールには、ユーザに関連付けられた API キーを使用した基本的な認証が必要です。
- **アプリケーション コンポーネント**:統合アプリは、多数のスクリプト インクルード、ビジネス ルール、および一部の ServiceNow テーブルの拡張などのテーブルを作成します。
- **変更されたコンポーネント**:統合は ServiceNow 内の*発注書*コンポーネントを変更します。
- **アプリケーションのカスタマイズ**:統合アプリケーションをカスタマイズできます。顧客が自社のローカルインストールに変更を行った場あい、それらの変更は、今後の製品の更新によってローカルの変更が上書きされた場合を考え、アップデート設定で維持されるようにすることを推奨します。
- **製品(アプリケーション)カタログ**:統合では Angular を使用して開発された新しい製品カタログが追加されます。統合アプリは OOB ServiceNow を使用しません。
- **見積もりの生成**:統合は CloudCenter に保存されたイメージまたはクラウド コストのデータに依存してチェックアウト フォームに見積もりを提供します。見積もりを概算として使用し、開発コストを決定します。
- **導入の運用コスト**:統合は、各開発の運用コストを最初の見積もりプロセス時に取得したコスト データに基づいて維持します。

CloudCenter と ServiceNow の統合の概要

概要

Cisco CloudCenter を使用すると、サービス マネージャとアプリケーション所有者は、アプリケーション プロファイルを任意の場所に導入し、それらを ServiceNow にパブリッシュすると、簡単にモデルを作成できます。

ServiceNow を使用すると、ユーザはサポート対象のデータセンターまたはクラウドへの導入を、各オプションのコストの対照比較とともに要求することができます。

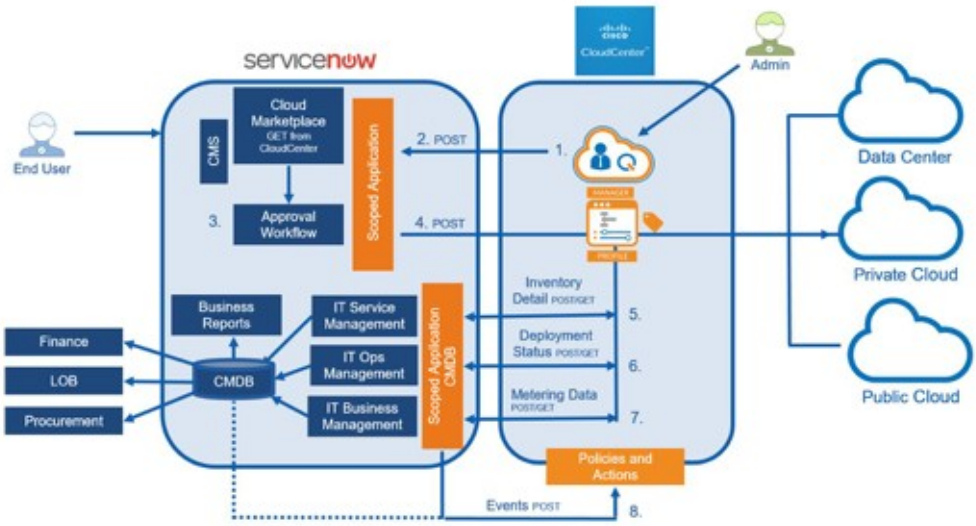
Cisco CloudCenter と ServiceNow の統合は、IT サービス管理(ITSM)、IT 運用管理(ITOM)、および IT ビジネス管理(ITBM)プロセスを拡張してアプリケーションの導入と管理を組み込む一方で、企業全体の管理機能とガバナンス機能を追加します。

統合の詳細

Cisco CloudCenter は ServiceNow によって認定され、ServiceNow の App Store からダウンロードできる統合アプリケーションを提供します。この統合により、CloudCenter で作成されたアプリケーションやサービスが ServiceNow のカスタム カタログにパブリッシュされて、指定された ServiceNow ユーザが使用できるようになります。これらのユーザはカタログを参照し、アプリケーションを設定してカートに追加し、クラウド間で導入コストを比較し、希望するクラウドにアプリケーションを導入することができます。

導入のためにアプリケーションを送信すると、統合アプリ内でそのステータスをトラッキングできます。さらに、*一時停止*、*再開*、または*終了*などのライフサイクル アクションをアクティブな導入上で取得でき、CloudCenter からリアルタイムで更新されます。ユーザは、ServiceNow を使用したまま、導入済みの仮想マシンに SSH または RDP で接続することもできます。

これらの機能はすべて、ServiceNow の CMS 内で完全に機能し、開発された一連のサンプル UI ページを実装することによって可能となります。ユーザ インターフェイスは、分かりやすく、カスタマイズが可能であるとともに、クラウドの導入と管理のほとんどを簡単にできるようにします。統合アプリのセットアップ ウィザードの一部として、各 ServiceNow 環境が CloudCenter 環境と簡単にリンクされ、ServiceNow と Cisco CloudCenter 間のすべての通信が保護され、REST API コールを通じて実行されます。



利点

CloudCenter-ServiceNow 統合アプリケーションには次の利点があります。

- **ServiceNow による認定**：統合アプリケーションは ServiceNow によって認定されており、ServiceNow のお客様は App Store からダウンロードできます。このアプリケーションは、グローバル スコープではなく、ServiceNow アプリケーションのスコープ内で開発されており、既存の ServiceNow の実装への影響が最小限に押さえられています。
- **DevOps を促進**：新しいインフラストラクチャやアプリケーションの導入をさらに効率的に取得できるようにすることで、DevOps を促進します。
- **使いやすさ**：シンプルかつ効果的なユーザ インターフェイスによって、クラウドの導入の複雑さを排除します。ServiceNow のエンド ユーザに単一の場所を提供することで、追加ツールのためのトレーニングの必要性を削減します。
- **コスト削減**：導入や終了をスケジュールすることで、コストを低減し、インフラストラクチャ リソースを最適化します。
- **互換性の向上**：統合アプリケーションは MID サーバを利用することで ServiceNow 環境との互換性が得られます。
- **アプリケーションのパブリッシュ**：サービス マネージャはクラウドに依存しないアプリケーション プロファイルを Cisco CloudCenter モデラ内で設定済みのイメージやインポートしたイメージ、アプリケーション サービス、コンテナを使用して簡単にモデル化し、それを ServiceNow に対してパブリッシュして使用されるようにすることができます。
- **サービス要求**：ServiceNow では、指定されたユーザがアプリケーションおよびサービスを選択し、ショッピング カートに追加し、開始日と終了日を入力して、利用可能なクラウドごとに提示されるコストの対照比較に基づいて導入場所を選択できます。
- **ワークフローの許可**：予想コストとともに要求を承認者にルーティングすることで、その意思決定を促進します。
- **導入の自動化**：ServiceNow は Cisco CloudCenter ソリューションを呼び出し、ターゲットのクラウド API との直接通信を通じてクラウドのインフラストラクチャ リソースをプロビジョニングします。次に、アプリケーション イメージ、サービス、およびデータを導入し、設定します。
- **ステータスの更新**：ServiceNow は Cisco CloudCenter ソリューションからステータスと進行中のインフラストラクチャ パフォーマンスの更新を受け取ります。サードパーティ製アプリケーションのパフォーマンス モニタリングおよびサービス保証ツールからの情報を使用して、ServiceNow はインシデントをログに記録するか、または新しい修復ワークフローを開始することができます。

機能

- Cisco CloudCenter から ServiceNow へのアプリケーションのワンクリック パブリッシュ。
- 単一の仮想マシン、または完全なアプリケーション スタックのプロビジョニングと管理。
- カスタマイズを容易にするために Angular 内で開発され、OOB ServiceNow カタログにはリンクされていない製品（アプリケーション）カタログの提供。
- ショッピング カートのエクスペリエンス。
- 発注時に設定されたクラウドの導入コストの対照比較。
- 後で使用するための見積書の保存、または財務処理のためのダウンロード。
- 承認ワークフローと電子メール通知。
- ガバナンスを目的とした Cisco CloudCenter 内でのタグの使用。
- 展開および終了のスケジューリング。
- アクティブな導入の一時停止、再開、終了などのライフサイクル アクション。
- 導入済みの仮想マシンへのリモート接続。
- ServiceNow と Cisco CloudCenter 間の通信に REST API を使用。

ServiceNow と CloudCenter のインストールと設定

可用性

CloudCenter 4.6 は次の **ServiceNow** のリリースをサポートします。

- ServiceNow、Geneva
- ServiceNow、Helsinki

CloudCenter 4.6 は次の **Integration - CloudCenter** のリリースをサポートします。

- Integration - CloudCenter リリース 1.0
- Integration - CloudCenter リリース 1.1
- Integration - CloudCenter リリース 1.2

Cisco CloudCenter の設定

ServiceNow の統合用に CloudCenter を設定するには、段階的なプロセスに従う必要があります。

1. カスタム アクションの作成

Cisco CloudCenter から ServiceNow にアプリケーション プロファイルをパブリッシュするカスタム アクションを作成する必要があります。
カスタム アクションを作成するには、Cisco CloudCenter で次の手順を実行します。

1. 管理者権限を使用して、Cisco CloudCenter にログインします。
2. 左側のナビゲーションから [Policies] を選択します。
3. [Custom Actions] タブを選択します。
4. [Add Custom Action] をクリックし、各フィールドを次のように定義します。

フィールド	値
名前	ServiceNow へパブリッシュ
Visible to User	イネーブル
オブジェクト	アプリケーション
Action Type	Web サービスを呼び出す
プロトコル	HTTPS
Web Service API	<yourServiceNowInstance.com>/api/now/table/x_cqt_cliqr_publish_app_trigger
Username	Rest.admin（このユーザは後で ServiceNow に作成されます）
Password	Rest.admin ユーザのパスワードを入力

フィールド	値
Http Request Type	POST
コンテンツ タイプ	JSON
本体	`{ "app_id": "%appId%", "app_name": "%appName%", "latest_app_version": "%latestAppVersion%", "owner_id": "%ownerId%", "owner": "%owner%" }`

2. ポリシーの作成

3 つのアクション ポリシーを作成して、アプリケーションの導入のステータスを ServiceNow に伝達する必要があります。各ポリシーについては、次に示す共通設定と個別のポリシー設定を使用します。

新しいそれぞれのポリシーを作成するには、Cisco CloudCenter で次のステップを実行します。

1. 左側のナビゲーションから [Policies] を選択します。
2. [Policies] タブを選択します。
3. [Add Action Policy] をクリックし、各フィールドを次のように定義します。

共通設定

フィールド	値
名前	下記の「*個別のポリシーの設定*」の項を参照してください。
Execute For	アプリケーションの導入
On Event	下記の「*個別のポリシーの設定*」の項を参照してください。
Action Type	Web サービスを呼び出す
プロトコル	HTTPS
Web Service API	<yourServiceNowInstance.com>/api/now/table/x_cqt_cliqr_job_status_trigger
Username	Rest.admin（このユーザは後で ServiceNow に作成されます）
Password	Rest.admin ユーザのパスワードを入力
Http Request Type	POST
コンテンツ タイプ	JSON
本体	下記の「*個別のポリシーの設定*」の項を参照してください。
Auto Enabled for shared users	イネーブル
Restrict users from disabling this Policy	イネーブル

個別のポリシーの設定

ポリシー（Policy）	イベント時	本体
1. SNOW_job_status_changed	ステータス変更	`{ "job_id": "%jobId%", "job_name": "%jobName%", "job_type": "%jobType%", "app_name": "%appName%", "owner": "%owner%", "status": "%status%", "changed_on": "%ChangedOn%", "new_status": "%NewStatus%" }`
2. SNOW_job_deployed	Deployed	`{ "job_id": "%jobId%", "job_name": "%jobName%", "job_type": "%jobType%", "app_name": "%appName%", "owner": "%owner%", "status": "%status%", "deployed_on": "%DeployedOn%" }`
3. SNOW_job_canceled	Cancelled（キャンセル）	`{ "job_id": "%jobId%", "job_name": "%jobName%", "job_type": "%jobType%", "app_name": "%appName%", "owner": "%owner%", "status": "%status%", "cancelled_on": "%CancelledOn%" }`

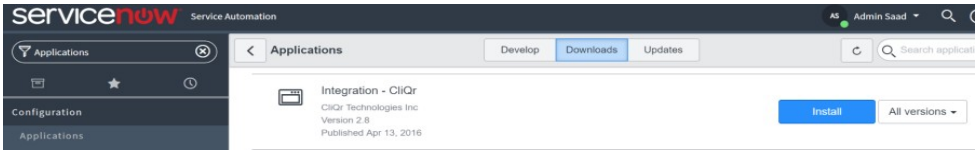
ServiceNow の設定

CloudCenter-ServiceNow 統合アプリケーションを設定するには、段階的なプロセスに従う必要があります。

1. CloudCenter-ServiceNow 統合アプリケーションのダウンロードとインストール

CloudCenter-ServiceNow 統合アプリケーションをダウンロードしてインストールするには、次の手順を実行します。

1. CloudCenter-ServiceNow 統合アプリケーションをダウンロードする要求を送信します。
   a. https://store.servicenow.com Web サイトにアクセスします。
   b. **Integration - CloudCenter** を検索します。
   c. カタログから [Integration - CloudCenter] アプリケーションを選択します。
   d. [Contact Seller] で販売者を選択して、アプリケーションをダウンロードする承認を受けます。
2. ServiceNow に CloudCenter アプリケーションをインストールします。
   a. アプリケーションが承認されたら、左側のフィルタ オプションで [System Applications] > [Applications] > [Downloads] > [Integration - CloudCenter] に移動します。
   b. [All versions] をクリックします。
   c. 適切なバージョンを選択し、インストールします。



2. CloudCenter-ServiceNow 統合アプリケーションの設定

CloudCenter-ServiceNow 統合アプリケーションを設定するには、次の手順を実行します。

1. Procurement プラグインをイネーブルにします。
   a. フィルタを使用し、[Plugins] に移動します。
   b. プラグイン モジュール内で [Procurement] を検索します。
   c. [Procurement - com.snc.procurement] を見つけてアクティベートします。
   d. 完了したら、[Close and Reload Form] をクリックします。

2. 証明書をインストールします。

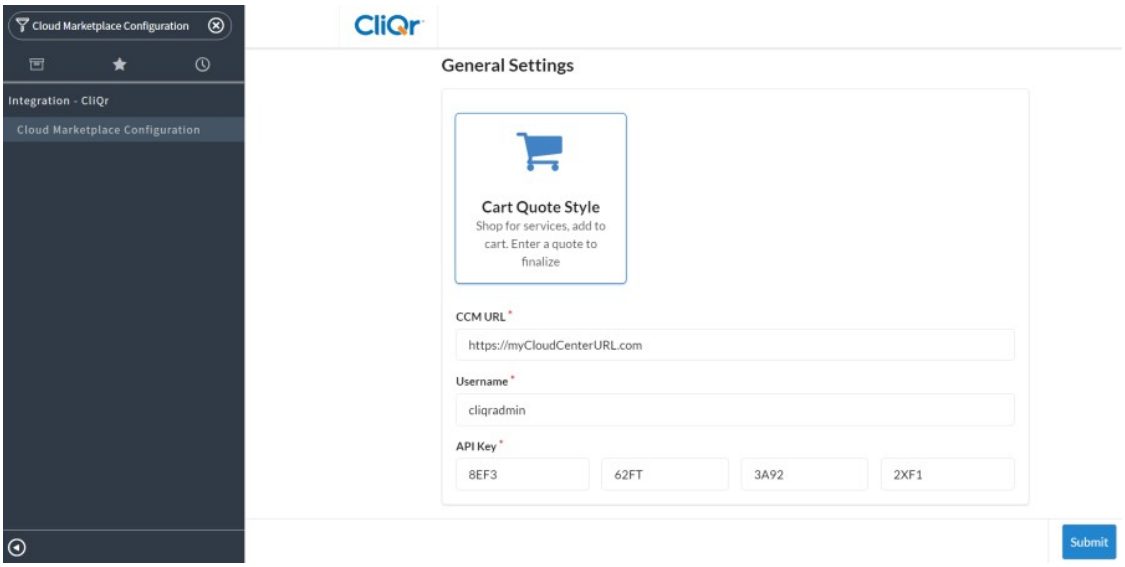
> 自己署名証明書を Cisco CloudCenter で使用する場合は、ServiceNow に証明書を追加して通信を有効にする必要があります。

   a. フィルタを使用して、[System Definition] > [Certificates] に移動します。
   b. 種類に [Trust Store Cert] を使用して、新しい X.509 証明書を追加します。
3. Cisco CloudCenter との接続を設定します。

> このステップでは、ネットワーク上の ServiceNow 環境と通信できる CloudCenter Manager（CCM）があると仮定します。

   a. フィルタを使用し、[Cloud Marketplace Configuration] に移動します。
   b. Cisco CloudCenter との接続を確立するには、[General Settings] セクションの次の必須フィールドに入力します。

General Settings	その他の情報
CCM URL	https://<*ciscoCloudCenterURL*>.com 形式を使用して、Cisco CloudCenter の URL を入力します。
Username	Cisco CloudCenter のデフォルトのユーザ名である **cliqradmin** を入力するか、または別の管理者アカウントを入力します。
API Key	上記で使用した管理者アカウントの API キーを入力します。API キーを取得するには、Cisco CloudCenter にログインします。[Admin] > [Users] に移動し、使用するアカウントの [Manage API Key] をクリックします。

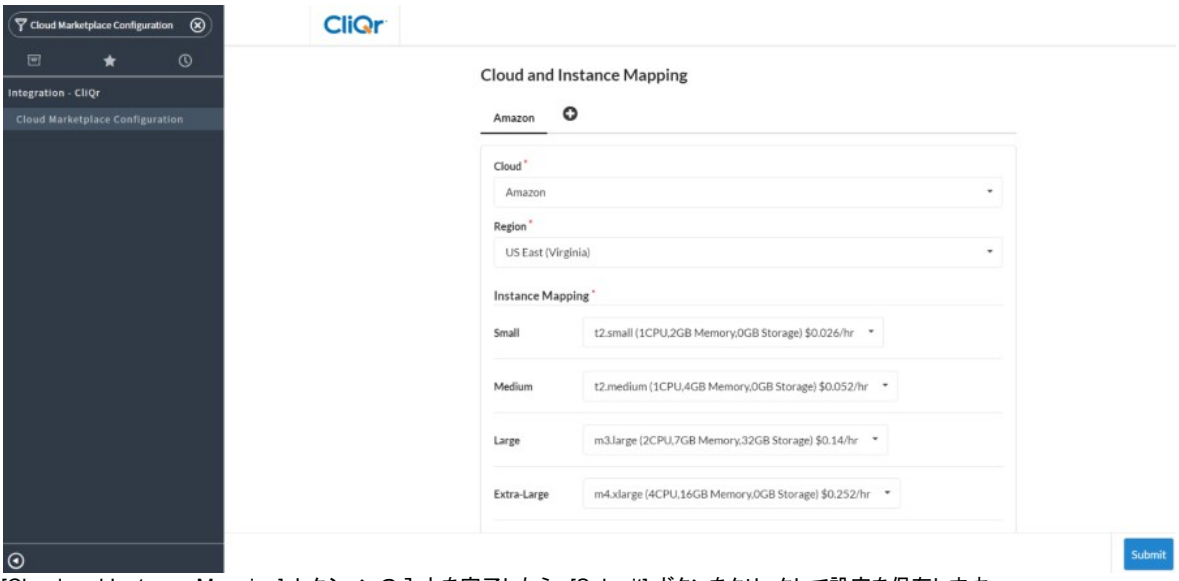


c. [General Settings] セクションに入力したら、[Submit] をクリックします。[Cloud Marketplace Configuration] ページがリロードされ、[Cloud and Instance Mapping] という新しいセクションにアクセスできるようになります。同じページに留まり、次の項のステップを実行します。

> この手順は重要です。[Cloud and Instance Mapping] セクションが表示されない場合は、ServiceNow は Cisco CloudCenter と通信できません。

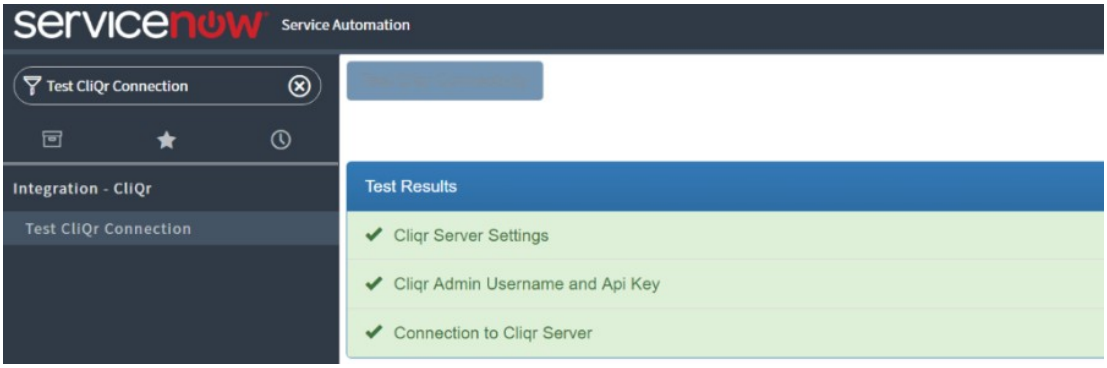
4. クラウドとインスタンスのマッピングを設定します。
   a. [Cloud and Instance Mapping] セクションの次の必須フィールドに入力します。参考までに、下のスクリーンショットを参照してください。

クラウドとインスタンスのマッピング	その他の情報
クラウド	ドロップダウンを使用してクラウドを選択します。Cisco CloudCenter で有効になっているクラウドのみを選択できます。
領域	ドロップダウンを使用して、Cisco CloudCenter で有効になっているリージョンを選択します。
Instance Mapping	統合アプリケーションの注文書には、ユーザがインスタンスのサイズを選択できるようにシンプルなアプローチが利用されています。このドロップダウンを使用して、小規模、中規模、大規模、超大規模に適したインスタンス タイプ値を関連付けます。追加のクラウドを設定するには、プラス アイコンをクリックします。



   b. [Cloud and Instance Mapping] セクションの入力を完了したら、[Submit] ボタンをクリックして設定を保存します。

5. 接続を確認します。
   a. フィルタを使用して、[Integration - CloudCenter] > [Test CliQr Connection] に移動します。
   b. [Test CliQr Connectivity] をクリックします。
   c. 次の図に示すように、3 つのテストがすべて成功する必要があります。



6. 統合ユーザを作成します。
   a. ServiceNow で、新しいユーザを次のように作成します。

フィールド	値
ユーザ ID	rest.admin
名	Rest
姓	Admin
電子メール	不要
Password	<enter a secure password>
Web service access only	イネーブル
Internal integration user	イネーブル
Roles	ロール u_cliqr_admin、procurement_admin、rest_service を追加します。

7. 発注書（PO）を作成します。

> 統合アプリケーション内の注文書に入力するには、有効な PO が必要です。

   a. フィルタを使用して、[Procurement] > [Orders] > [Purchase Orders] にある [Purchase Orders] に移動します。
   b. 新しいポリシー マップを次のように作成します。

フィールド	その他の情報
PO Number	[Number] フィールドから自動生成番号を入力します。
値	発注書の適切な値を入力します。
Assigned To	任意のユーザを選択します。設定を検証するため、後でこのフィールドを更新する予定です。
タイプ	クラウドのマーケットプレイスを選択します。

8. 設定後のチェックリストに記載されている項目に従って、統合を確認します。
   a. [System Security] > [Users and Group] の下にある次のグループとロールを確認します。

Group Name	Roles	その他の情報
Cloud Marketplace Service Instance Owners	x_cqt_cliqr.service_instance_owner	このグループに属するユーザはサービス インスタンスの所有者であり、ダッシュボードの [Approvals] タブも確認します。少なくとも 1 人のユーザがこのグループに属している必要があります。
Cloud Marketplace Consumers	u_cliqr_requester	カタログを参照し、承認要求を許可/拒否するには、ユーザがこのグループに含まれている必要があります。このグループにユーザを（CliQr グループとともに）追加すると、Cisco CloudCenter にユーザ アカウントが作成され、このユーザに API キーが生成されます。[Integration - CloudCenter] > [User Properties and Integration - CloudCenter] > [User API Keys] に移動してこの要件を確認します。
CliQr	rest_service procurement_user	この統合に使用した Cisco CloudCenter のアクティベーション プロファイルごとに個別のグループを作成します。最初のグループは **CliQr** という名前にする必要があります。 追加のグループ（必要な場合）は、**CliQr *<activation profile tag>*** という名前にする必要があります。

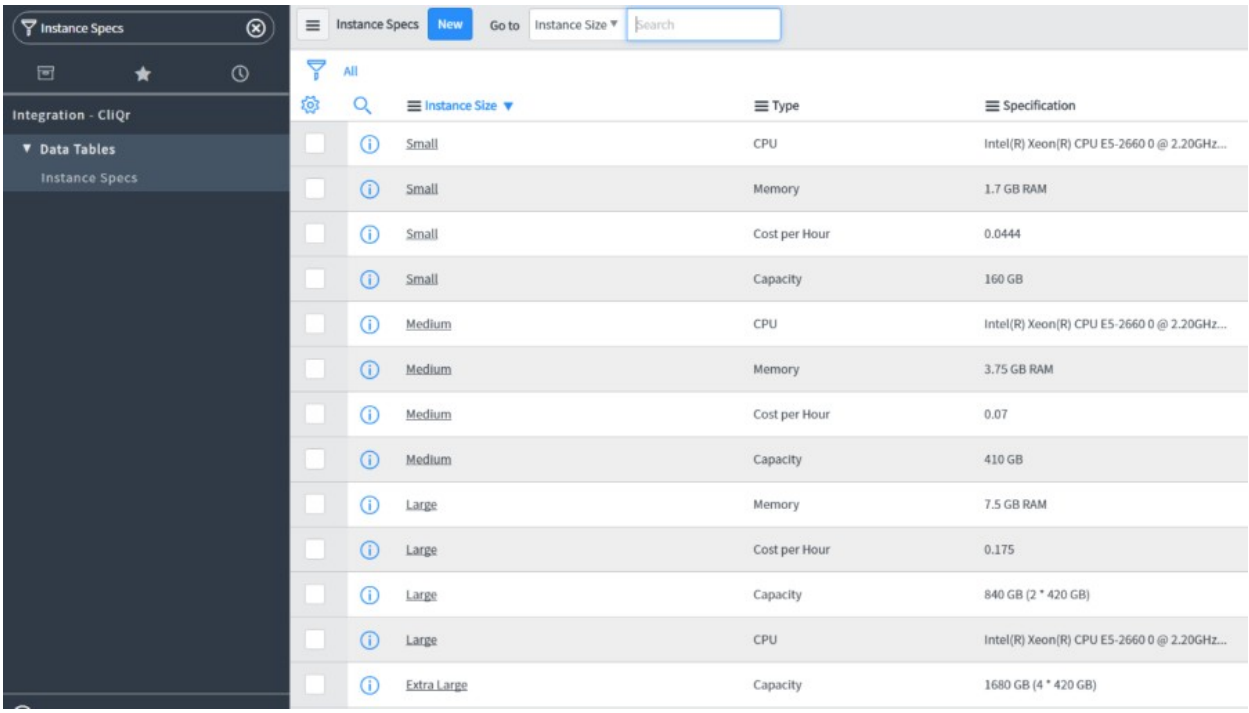
   b. [Integration - CloudCenter] > [Group Properties] の下にある次のグループ プロパティを確認します。

Group	名前	値	その他の情報
CliQr	activiation_profile	1	必要に応じて、追加のグループのプロパティを作成するには、次の形式を使用します。 **グループ**：CliQr ***<activation profile tag>*** **名前**：activation_profile **値**：***<cliqr_profile_id}***> cliqr_profile_id は Cisco CloudCenter のアクティベーション プロファイルの ID です。これは、1 や 2 のような数値です。グループ関連アクティベーション プロファイルは、注文ガイドの項目についてグループ メンバーが参照できるオプションを定義します。API の [View Activation Profiles] を使用して、ID を取得します。Cisco CloudCenter にアクティベーション プロファイルがない場合は、上に示したリンク先にある手順に従って、アクティベーション プロファイルを 1 つ作成する必要があります。

9. [Integration - CloudCenter] > [Settings] > [Properties] の下にある次のプロパティを確認します。

プロパティ名	説明
default_published_app_group	このプロパティのデフォルト値は Cloud Marketplace Consumers です。これは、Cisco CloudCenter からパブリッシュされたアプリケーションが関連付けられるデフォルトのグループです。ユーザ グループに基づいてアプリケーションをフィルタリングするための要件がない場合、このグループには常にデフォルト値を設定する必要があります。
max_order_quantity	統合アプリケーションの注文書では、ユーザがドロップダウン リストから各アプリケーションの数量を選択できます。このカンマ区切りのリストを使用して、数量を選択するためのドロップダウン値を設定します。デフォルト値は 1、2、3、4、5 です。

10. [Integration - CloudCenter] > [Data Tables] > [Instance Specs] の下にある [Instance Specs] の表にサンプル データが含まれていることを確認します。このデータは、インスタンス サイズ（小規模、中規模、大規模、または超大規模）を選択した時点でユーザに対して注文書上に提示されます。



11. 電子メールを有効にし、MID サーバをサポートするオプションの設定手順は次のとおりです。
    a. 承認ワークフローの電子メール通知に対して電子メールを有効にする必要があります。既存の ServiceNow のインストールにはすでにこれが設定されている場合があります。
        i. フィルタを使用して、[System Properties] > [Email Properties]（ServiceNow のバージョンによっては [Email]）に移動します。
        ii. アウトバウンドとインバンドの電子メール送信が有効になっていることを確認します。
    b. MID サーバに対するサポートは次のとおりです。
        i. [MID Server] > [Capabilities] に移動し、REST、SSH、および SOAP の機能が追加されていることを確認します。
        ii. [Integration - CloudCenter] > [Settings] > [Properties] の下に次のプロパティを作成します。

プロパティ名	値
Mid_server_name	<configured_rest_server_name>

CloudCenter と ServiceNow との統合の検証

CloudCenter-ServiceNow 統合アプリケーションを検証するには、次の手順を実行します。

1. テスト ユーザを作成します。
    a. インストールを確認するには、ServiceNow に次のテスト ユーザ アカウントを作成します。ユーザ アカウントを作成するには、電子メール アドレスが必要です。

ユーザ名	グループ メンバーシップ	説明
consumer.user	i. Cloud Marketplace Consumers ii. CliQr	このテスト ユーザを使用して注文を送信します。
serviceOwner.user	i. Cloud Marketplace Consumers ii. Cloud Marketplace Service Instance Owners	このテスト ユーザを使用して、所有者の観点から注文要求を承認または拒否します。
financeOwner.user	i. Cloud Marketplace Consumers ii. Cloud Marketplace Service Instance Owners	このユーザを使用して、財務の観点から注文要求を承認または拒否します。

    b. 新しい発注書が上記の [Create a Purchase Order] セクションに作成されました。この発注書の [Assigned To] フィールドをユーザ financeOwner.user に更新します。
2. アプリケーション プロファイルを ServiceNow にパブリッシュします。
    a. Cisco CloudCenter の [Applications] タブでアプリケーション プロファイルを見つけます。
    b. アプリケーションのドロップダウンをクリックし、テナント内のすべてのユーザと**共有**します。
    c. アプリケーションのドロップダウンをクリックし、[Publish to ServiceNow] を選択します。
    d. ServiceNow で、[Integration - CloudCenter] > [Data Tables] > [Application Profiles] に移動します。パブリッシュ済みのアプリケーションを含むレコードがあるはずです。

3. 発注するには、次の手順を実行します。
    a. https://<*yourServiceNowInstance*>.com/cloud-marketplace に移動します。
    b. consumer.user ユーザとしてログインします。
    c. [Store] をクリックします。
    d. カタログからアプリケーションを選択し、注文を設定して送信します。必ず、サービス インスタンスの所有者として **serviceOwner.user** を選択し、financeOwner.user に割り当てられている PO を選択してください。
    e. serviceOwner.user としてログインし、次に financeOwner.user としてログインして要求を承認します。
    f. 注文が承認されると、指定された導入開始日時に基づいてその注文が導入されます。
    g. [Service Instances] タブを参照し、注文のステータスが導入済み（Deployed）であることを確認します。

追加の注意事項

統合アプリのインストールによって、タスク テーブルにフィールドが追加されます。これは線形動作であるため、環境内の既存のデータに応じて、ロードにしばらく時間がかかる場合があります。