Cisco CloudCenter インストレーション 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。

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. シスコの商標の一覧は、http://www.cisco.com/go/trademarks でご確認いただけます。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)

© 2016 Cisco Systems, Inc. All rights reserved.

CloudCenter のインストール

- プラットフォーム アーキテクチャ
- CloudCenter のコンポーネント
- 設置の概要
- インストール方法
- コンポーネント モードとロール
- フェーズ 1:インフラストラクチャの準備
- フェーズ 2:ネットワーク ルールの設定
    - ネットワーク ルールの概要
    - AMQP ネットワーク ルール
    - バンドル ストアのネットワーク ルール
    - CCM ネットワーク ルール
    - CCO ネットワーク ルール
    - DOCKER_REGISTRY ネットワーク ルール
    - Ext_Script ネットワーク ルール
    - Guacamole ネットワーク ルール
    - モニタ ネットワーク ルール
    - パッケージ ストアのネットワーク ルール
    - プロキシの設定
- フェーズ 3:ネットワーク コンプライアンス チェックの実行
- フェーズ 4:コンポーネントのインストール
    - インストーラの概要
    - パッケージ ストアのインストール(条件付き)
    - バンドル ストアのインストール(オプション)
    - CCM のインストール(必須)
    - CloudCenter リージョンごとのインストール(必須)
    - ヘルス モニタのインストール(オプション)
    - 専用 Docker レジストリのインストール(オプション)
    - カスタム イメージのインストール
- メモ
- アプライアンスの設定オプション

CloudCenter プラットフォーム アーキテクチャ

Cisco CloudCenter™ は、データセンター、プライベート クラウド、パブリック クラウド環境へのアプリケーション スタックの展開およびその間での管理の完全なライフサイクル アプローチを安全に提供する、ハイブリッド クラウド管理プラットフォームです。

CloudCenter 独自のアプローチによって、企業はクラウドおよびユーザの統一された管理とガバナンスを使用して、クラウド環境でアプリケーションをモデル化、導入および管理することができます。

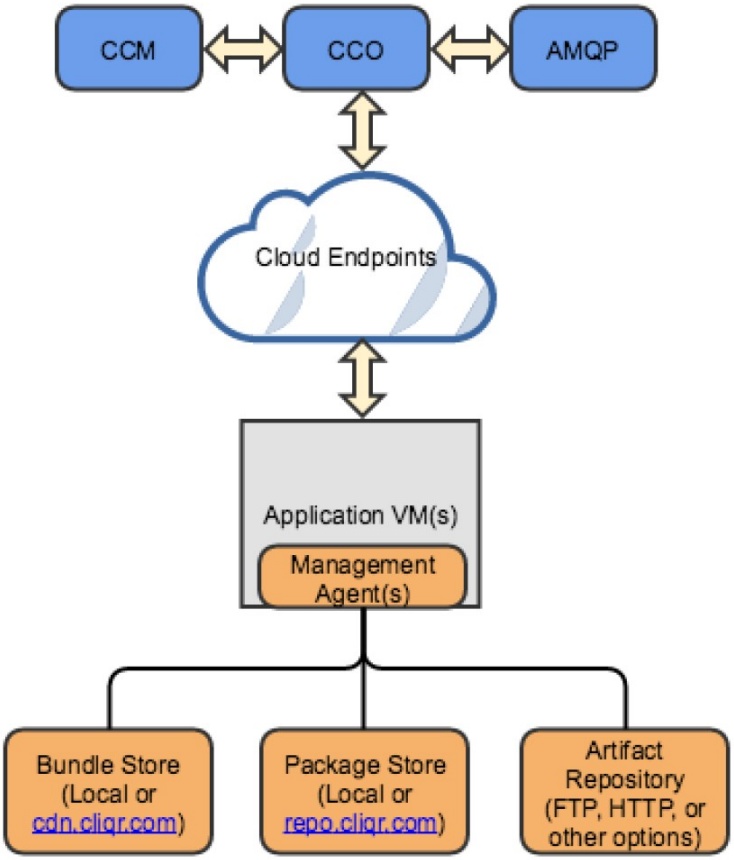
CloudCenter は基盤となるクラウド環境からアプリケーションを抽象化するアプリケーション定義された独自のテクノロジーを提供します。この抽象化により、各アプリケーションの導入および管理に関する要件にインフラストラクチャを適合させることができます。

CloudCenter ソリューションの機能は、いくつかの主要な CloudCenter のコンポーネントを含む、独自の特許テクノロジーによってもたらされます。

コンポーネントの概要

- 概要
- 必要な CloudCenter のコンポーネント
  - CCM
  - クラウド リージョン別
    - CCO
    - AMQP
    - 専用外部スクリプト Executor（条件付き）
- オプションの CloudCenter のコンポーネント
  - 専用 Guacamole サーバ（クラウド リージョンごと）
  - モニタ
  - バンドル ストア
  - パッケージ ストア
  - サービス用の専用 Docker レジストリ
  - ベース OS イメージの準備
- ネットワーク アーキテクチャ

概要



必要な CloudCenter のコンポーネント

CCM

マネージャ、または管理サーバと呼ばれることもある CloudCenter Manager（CCM）は、ユーザが導入をモデル化、移行および管理するためのダッシュボードとして機能し、いずれかの導入モデルを使用しているクラウドとユーザの統一された管理およびガバナンスのための、集中管理層です。

詳細については、「CCM」を参照してください。

クラウド リージョン別

*クラウド リージョン*とは、単一のパブリック クラウド、データセンター、または CloudCenter でサポートされるプライベート クラウドのことです。各クラウド リージョンはクラウドを設定するときに CCM UI で識別されます。

CloudCenter でサポートされるクラウド

CloudCenter でサポートされるクラウドのリストについて：

- 「Public Clouds」を参照してください
- 「Datacenters and Private Clouds」を参照してください

CloudCenter プラットフォームにより管理する必要がある各 *クラウド リージョン*に対し、次の CloudCenter のコンポーネントをインストール、セットアップおよび設定する必要があります。

<table>
<tr><th>クラウド リージョンのコンポーネント</th><th>説明</th></tr>
<tr><td>CCO</td><td>CloudCenter Orchestrator（CCO）（オーケストレータとも呼ばれる）は、ユーザに対して透過的であり、データセンター、プライベート クラウド、またはパブリック クラウド環境に配置されます。これにより、基礎となるクラウド インフラストラクチャからアプリケーションを切り離し、クラウド インフラストラクチャの複雑さを切り離します。

アプリケーション ネットワークが完全に分離され、出力接続がない場合を除き、CloudCenter にはクラウド リージョンまたはデータセンターあたり 1 CCO が必要です。CloudCenter は独立したネットワークをプライベート クラウドとして処理し、別の CCO が必要です。

各 CCO は CCM に登録する必要があります。CCO は CCO が導入されるクラウドに関係なく、CCM と直接通信します。

詳細については、「CCO」を参照してください。</td></tr>
<tr><td>AMQP</td><td>CloudCenter プラットフォームには CCO とエージェント間の Advanced Message Queuing Protocol（AMQP）ベースの通信も含まれます。CloudCenter プラットフォームは、AMQP を実装するためのオープン ソース メッセージ ブローカとして RabbitMQ を使用します。

アプリケーション VM が分離されたネットワーク（Amazon の VPC など）で実行されている場合、アプリケーション VM が RabbitMQ に接続できるように、適切な NAT ルール（出力のみが必要）を設定してください。</td></tr>
<tr><td>専用外部スクリプト Executor（条件付き）</td><td>CloudCenter プラットフォームは、アプリケーションのライフサイクルの各段階において実行されるカスタム スクリプトを使用する機能を提供します。これらのカスタム スクリプトは Docker ベース スクリプト Executor 内で実行し、各 CCO で使用されている OS によって異なります。

<table><tr><th>CCO サーバの ベース OS イメージ</th><th>スクリプト Executor のセットアップ</th></tr><tr><td>RHEL 7.x および CentOS 7.x</td><td>CloudCenter に使用される OS で CCO にデフォルトで組み込まれています。</td></tr><tr><td>RHEL 6.x または CentOS 6.x</td><td>専用のスクリプト Executor を設定する必要があります。これはこれらのバージョンを使用した OS 向けのスタンドアロン Docker コンテナです。</td></tr></table>
拡張性に関する考慮事項
また、増大する拡張性に関する考慮事項に対処するために、CCO サーバに RHEL 7x または CentOS 7x を使用する、リージョン用の専用のスクリプト Executor を設定することもできます。</td></tr>
</table>

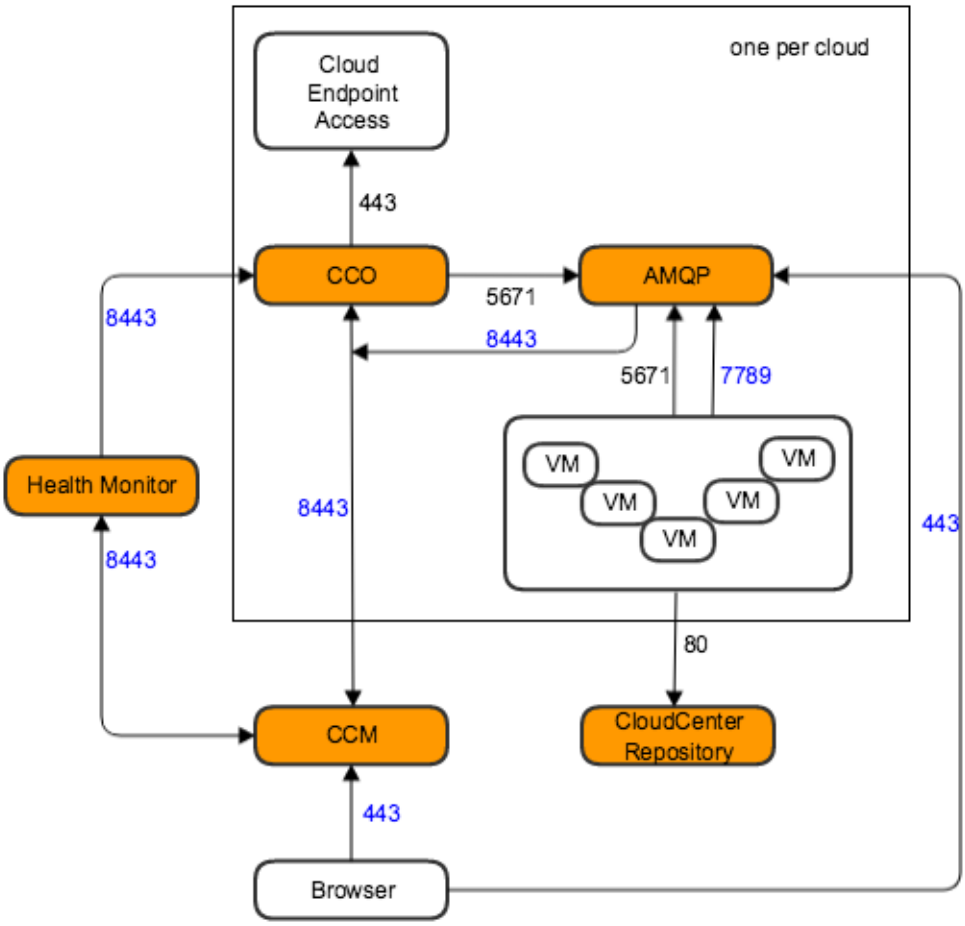
オプションの CloudCenter のコンポーネント

CloudCenter プラットフォームにより管理される必要があるすべての *クラウド リージョン* では、以下の CloudCenter コンポーネントをインストール、セットアップおよび設定することもできます。

オプションの CloudCenter のコンポーネント	説明
専用 Guacamole サーバ（クラウド リージョンごと）	CloudCenter プラットフォームは、Guacamole サーバを使用して、アプリケーション ライフサイクル プロセス中に開始された仮想マシン（VM）への Web ベース SSH/VNC/RDP を可能にします。Guacamole コンポーネントは、AMQP サーバにデフォルトで組み込まれています。 AMQP の IP アドレスにエンドユーザによりルーティング可能にさせないように、クラウド リージョンの専用（スタンドアロン）Guacamole サーバをセットアップすることを選択できます。
モニタ	クラウド ヘルス モニタ（モニタ）はオプションの独立型の CloudCenter のコンポーネントで、クラウドと CCO のヘルス ステータスを監視し、2 つのサブコンポーネントで構成されます。 • **ヘルス チェック マネージャ**：監視対象クラウドにポーリングし、ヘルス ステータス情報を収集します。 • **ヘルス ステータス クエリー サービス**：クラウドのヘルス ステータス データを報告する一連の Web サービス。 追加情報については、「Cloud Health Monitor」を参照してください。

オプションの CloudCenter のコンポーネント	説明
バンドル ストア	バンドル ストアは、エージェントやサービス バンドルを含むリポジトリです。アプリケーション オーケストレーション プロセスの一部として CloudCenter で開始する VM（ワーカーまたはアプリケーション VM）でこれらのバンドルをダウンロードしてインストールする必要があります。 • **インターネット接続使用**：デフォルトのバンドル ストアは（cdn.cliqr.com）にホストされ、アプリケーション VM がインターネットにアクセスできる CloudCenter の導入では、デフォルトのバンドル ストアを使用できます。 • **分離環境**：インターネットへの接続が制限された環境では、ローカル バンドル ストアを作成し、CCO に登録します。 追加情報については、「Bundle Store」を参照してください。
パッケージ ストア	パッケージ ストアは、すべてのアプリケーション サービス（設定済みサービス）のバイナリと、CloudCenter 自体のインストールに必要な複数のコンポーネントのバイナリを含むリポジトリです。 • **インターネット接続使用**：デフォルトのパッケージ ストアは repo.cliqrtech.com にホストされ、アプリケーション VM がインターネットにアクセスできる CloudCenter の導入で、デフォルトのパッケージ ストアを使用できます。 • **分離環境**：インターネットへの接続が制限された環境では、ローカル パッケージ ストアを作成し、CCO に登録します。 追加情報については、「Package Store」を参照してください。
サービス用の専用 Docker レジストリ	コンテナ レベルで VM ベースのサービスまたは外部サービスを使用したカスタム サービスを定義すると、隔離 Docker コンテナを起動することもできます。 • **インターネット接続使用**：サービス イメージのダウンロードには Docker ハブを使用します • **分離環境**：インターネットへの接続が制限された環境では、専用 Docker コンテナを設定するか、またはパッケージ ストア オプションを使用します。 追加情報については、「Dedicated Docker Registry Installation (Optional)」および「External Service」を参照してください。
ベース OS イメージの準備	また、さまざまなクラウドとリージョンの CloudCenter 対応イメージを作成するために CloudCenter インストーラのバイナリを使用することもできます。 • **Linux**：追加情報については、「On a Linux Image」を参照してください。 • **Windows**：追加情報については、「On a Windows Image」を参照してください。 追加情報については、「Base OS Images」を参照してください。

ネットワーク アーキテクチャ



設置の概要

- インストール コンポーネント
- ブラウザでのツリー ペインの表示と展開
- インストール ワークフロー

インストール コンポーネント

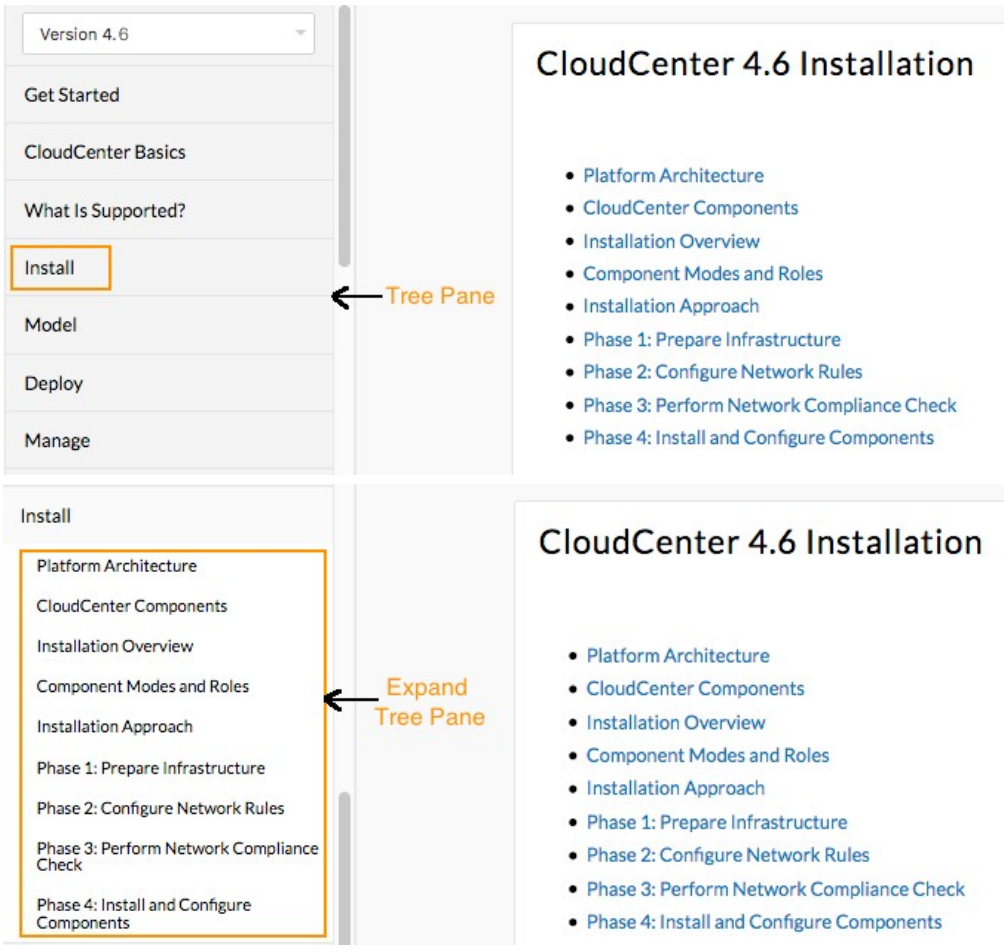
CloudCenter プラットフォームをインストールするには、次のコンポーネントをインストールして設定する必要があります。

No.	コンポーネント(「CloudCenter Components」を参照)	必須かどうか
1	CloudCenter Manager(CCM)	必須
2	CloudCenter リージョン それぞれのパブリック クラウド リージョン、プライベート クラウド、または CloudCenter 導入でサポートする必要があるデータセンターに対し、次のコンポーネントをインストールする必要があります。	必須
2a	CloudCenter Orchestrator(CCO)	必須
2b	メッセージ ブローカ(AMQP)	必須

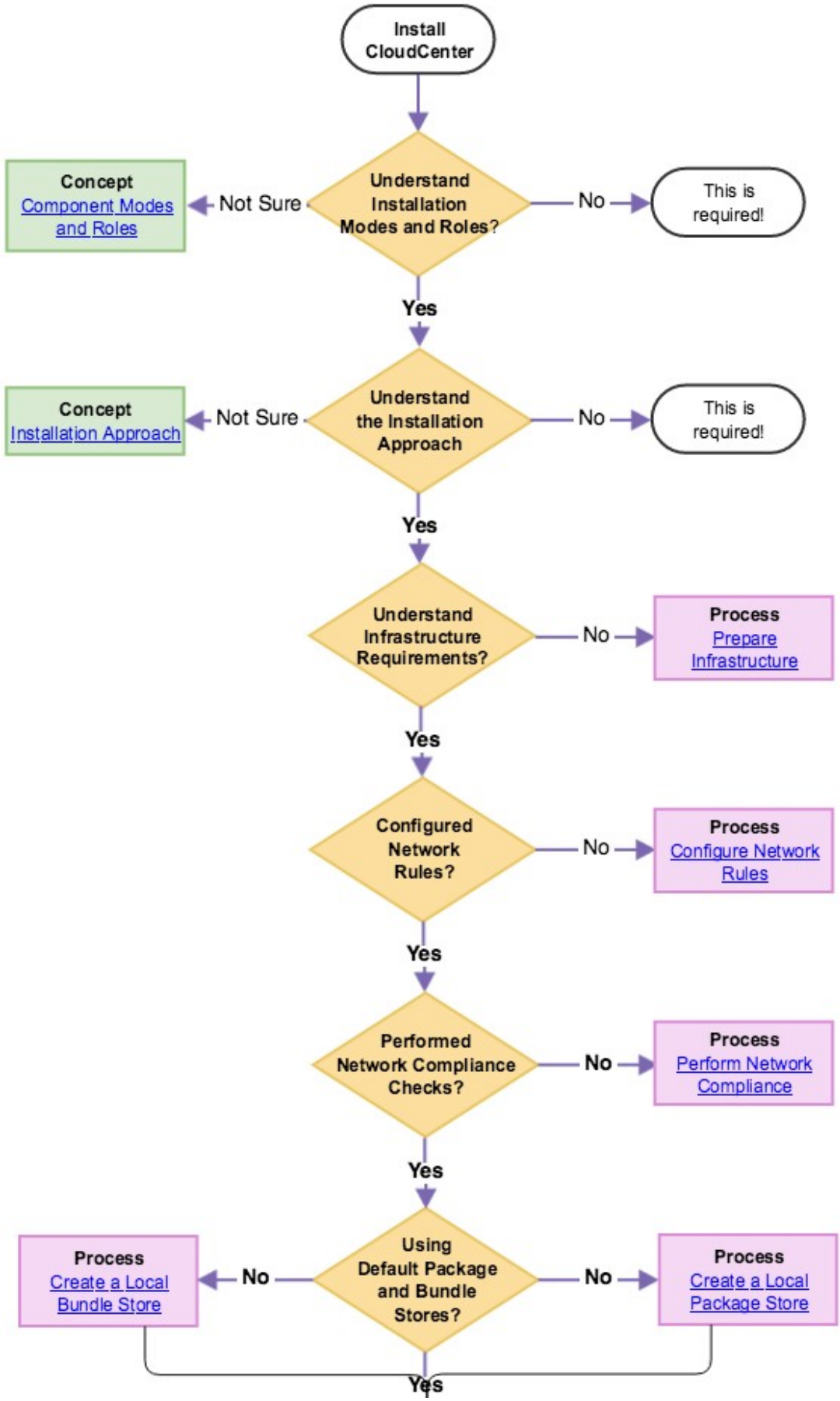
No.	コンポーネント(「CloudCenter Components」を参照)	必須かどうか
2c	専用外部スクリプト Executor(OS バージョンに基づく)	条件
2d	専用 Guacamole サーバ	任意
3	ヘルス モニタ	任意
4	バンドル ストア	任意
5	パッケージ ストア	条件
6	サービス用の専用 Docker レジストリ	任意
7	Linux ベースの OS イメージ	任意
8	Windows ベースの OS イメージ	任意

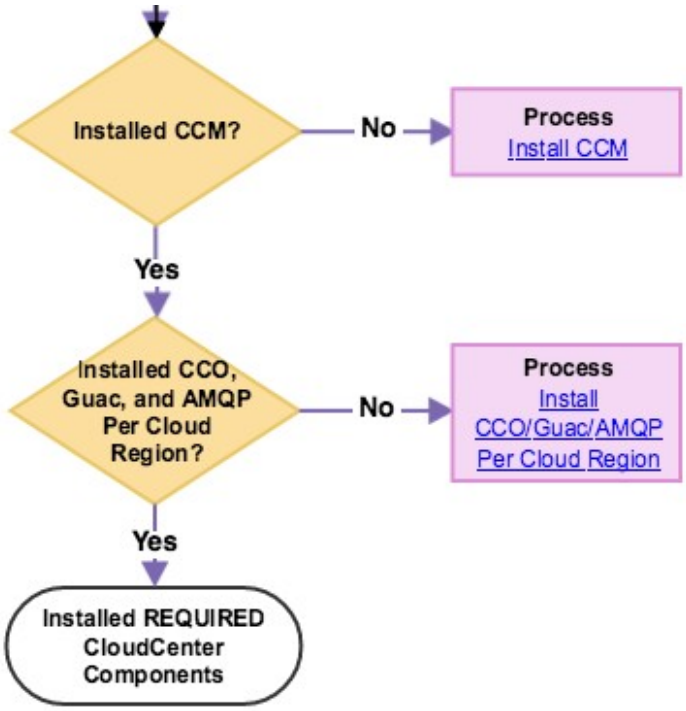
ブラウザでのツリー ペインの表示と展開

インストール セクションの各ページ間を移動できる柔軟性が重要です。同時に、CloudCenter を正しい順序でインストールし、プロセスに従う必要があります。CloudCenter のインストール プロセスを円滑に進めるために、次の図に示すように、ブラウザでツリー ペインが表示され、展開されていることを確認します。



インストール ワークフロー





インストール方法

CloudCenter プラットフォームをインストールするには、次のフェーズを完了する必要があります。

フェーズ	説明
1	**インフラストラクチャの準備** VM、ロード バランサ、仮想 IP（VIP）または Elastic IP（EIP）を起動および識別し、CloudCenter のコンポーネントをインストールして設定します。
2	**ネットワーク ルールの設定** さまざまなコンポーネント間の通信を可能にするネットワーク ルールを設定します。
3	**ネットワーク コンプライアンス チェックの実行** ネットワーク ルールが適切に設定されていることを確認するには、ネットワーク コンプライアンス チェックを実行します。

フェーズ	説明
4	**コンポーネントのインストールおよび設定** 仮想マシンでインストーラを実行して、CloudCenter コンポーネントをインストールし、コンフィギュレーション ウィザードを起動して、コンポーネントのプロパティを設定します。 1. パッケージ ストアのインストール（条件付き） 2. バンドル ストアのインストール（オプション） 3. CCM のインストール（必須） a. CCM 非 HA b. CCM 非 HA スタンドアロン c. CCM HA 4. CloudCenter リージョンごとのインストール（必須） a. 専用外部スクリプト Executor（条件付き） b. 専用 Guacamole サーバ（オプション） c. AMQP（必須） i. AMQP 非 HA ii. AMQP HA d. CCO（必須） i. CCO 非 HA ii. CCO HA 5. ヘルス モニタのインストール（オプション） a. 非 HA のモニタ b. HA のモニタ 6. 専用 Docker レジストリのインストール（オプション） 7. カスタム イメージのインストール（「Management Agent Java Compatibility」を参照） a. Linux イメージの場合 b. Windows イメージの場合

コンポーネント モードとロール

- 概要
- CCM
- クラウド リージョン
  - CCO
  - AMQP
  - 専用外部スクリプト Executor
  - 専用 Guacamole
- モニタ
- バンドル ストア
- パッケージ ストア
- サービス用の専用 Docker レジストリ
- Linux ワーカ イメージ
- Windows ワーカ イメージ

概要

このセクションでは、独自のインフラストラクチャ要件とともに CloudCenter の各コンポーネントに使用できる各種のモードについて説明します。

- **ロール**：各インフラストラクチャ要素を特定します。これらのロールは、それぞれのインストールおよび設定手順で広く使用されます。

  > 特定のロールに関連する IP アドレスを特定するのに、ドキュメントでは ***ROLE*_IP** 形式の規則に従っています。たとえば、ロール **CCM** に関連付けられた仮想マシン（VM）の IP アドレスは ***CCM_IP*** として参照されます

- **モード：**拡張性と高可用性の要件に基づいて、さまざまなモードの CloudCenter のコンポーネントを設定できます。モードごとに独自のインフラストラクチャ要件（VM、ロード バランサ、仮想/Elastic IP）があります。
  - 非 HA スタンドアロン：Web サーバとデータベースが 2 つの個別の VM にインストールされます。
  - **非 HA**：1 つの VM が Web サーバとデータベースで設定されています。
  - HA：2 つの個別のインスタンスにそれぞれ Web サーバとデータベースが含まれ、高可用性のために設定されています
- **タイプ**：各モードとロールに必要なインフラストラクチャのタイプを特定します。タイプには、VM、ロード バランサ、Web サーバ、VM イメージなどがあります。
- **OS オプション**：このモードで使用可能な設定済みのベース OS イメージ。

CCM

モード	ロール	タイプ	OS オプション	目的
非 HA	CCM	VM	• CentOS7 • CentOS6 • RHEL7 • RHEL6	CCM Web サーバとデータベース
非 HA スタンドアロン	CCM_SA	VM		スタンドアロン CCM – Web サーバのみ
非 HA スタンドアロン	MGMTPOSTGRES	VM		スタンドアロン PostgreSQL データベース サーバのみ
HA	CCM_SA_PRIMARY	VM		スタンドアロン プライマリ CCM – Web サーバのみ
HA	CCM_SA_SECONDARY	VM		スタンドアロン セカンダリ CCM – Web サーバのみ
HA	MGMTPOSTGRES_MASTER	VM		PostgreSQL のマスター データベース サーバ
HA	MGMTPOSTGRES_SLAVE	VM		PostgreSQL スレーブ データベース サーバ
HA	CCM_LB	ロード バランサ	-	両方の CCM VM 間での着信要求のバランス
HA	MGMTPOSTGRES_VIP	仮想 IP（VIP）または Easltic IP（EIP）	-	いずれかのサーバがダウンすると、アクティブ サーバへの動的なスイッチングを有効にするように、PostgreSQL サーバの 1 つに接続します

クラウド リージョン

使用するモードによって、次のように異なります。

- 非 HA：1 つの VM
- HA：**既存のロード バランサがあると仮定して、**2 つの VM をセットアップして、高可用性用に設定する必要があります。

> 既存のロード バランサ VM がなければ、この目的のために、高可用性を必要とする各コンポーネントに対し 3 つ目の VM が必要です。

CCO

モード	ロール	タイプ	OS オプション	目的
非 HA	CCO	VM	• CentOS7 • CentOS6 • RHEL7 • RHEL6	オーケストレータ
HA	CCO_PRIMARY	VM		プライマリ オーケストレータ
HA	CCO_SECONDARY	VM		セカンダリ オーケストレータ
HA	CCO_LB	ロード バランサ	-	両方の CCO VM 間での着信要求のバランス

AMQP

モード	ロール	タイプ	OS オプション	目的
非 HA	AMQP	VM	• CentOS7 • CentOS6 • RHEL7 • RHEL6	RabbitMQ ベースのメッセージ キュー
HA	AMQP_PRIMARY	VM		プライマリ RabbitMQ ベースのメッセージ キュー
HA	AMQP_SECONDARY	VM		セカンダリ RabbitMQ ベースのメッセージ キュー
HA	AMQP_LB	ロード バランサ	-	両方の AMQP VM 間での着信メッセージのバランス

専用外部スクリプト Executor

モード	ロール	タイプ	OS オプション	目的
非 HA	EXT_SCRIPT_EXECUTOR	VM	• CentOS7 • RHEL7	Docker ベースの隔離スクリプト実行環境

専用 Guacamole

モード	ロール	タイプ	OS オプション	目的
非 HA	GUAC	VM	• CentOS7 • CentOS6 • RHEL7 • RHEL6	Web ベース SSH/VNC/RDP を可能にする Guacamole ベースのサーバ

モニタ

モード	ロール	タイプ	OS オプション	目的
非 HA	MON	VM	• CentOS7 • CentOS6 • RHEL7 • RHEL6	クラウド リージョンのヘルスをモニタし、CCM に報告します
HA	MON_PRIMARY	VM		(プライマリ)クラウド リージョンのヘルスをモニタし、CCM に報告します
HA	MON_SECONDARY	VM		(セカンダリ)クラウド リージョンのヘルスをモニタし、CCM に報告します
HA	MON_LB	ロード バランサ	-	両方のモニタ VM 間での着信要求のバランス

バンドル ストア

タイプ	ロール	OS オプション	目的
HTTP Web サーバ	BUNDLE_STORE	-	ホストの CloudCenter エージェント バイナリおよび設定済みサービス スクリプト

パッケージ ストア

タイプ	ロール	OS オプション	目的
VM	PACKAGE_STORE	• CentOS7 • CentOS6 • RHEL7 • RHEL6	設定済みサービス スクリプトのバイナリおよび CloudCenter のその他のコンポーネント(JDK、PostgreSQL など)の設定に必要なソフトウェアのホスト

サービス用の専用 Docker レジストリ

タイプ	ロール	OS オプション	目的
VM	DOCKER_REGISTRY	• CentOS7 • CentOS6 • RHEL7 • RHEL6	設定済みサービスに対する Docker ベースのホスト コンテナ イメージ

Linux ワーカ イメージ

タイプ	ロール	OS オプション	目的
VM イメージ	LINUX_WORKER_*OS_VERSION*	• CentOS7 • CentOS6 • RHEL7 • RHEL6 • Ubuntu 12.04 • Ubuntu14.04	アプリケーションのオーケストレーション中に VM を起動するときに使用するマシン イメージ

Windows ワーカ イメージ

タイプ	ロール	OS オプション	目的
VM イメージ	WINDOWS_WORKER_*OS_VERSION*	• Win2k8、 • Win2k12	アプリケーションのオーケストレーション中に VM を起動するために使用するマシン イメージ

フェーズ 1:インフラストラクチャの準備

- インフラストラクチャの起動
- Amazon の特徴
- OpenStack の特徴
- CloudCenter 記述子 JSON ファイル
- ハードウェア要件

インフラストラクチャの起動

目的のコンポーネント モードに基づいて、さまざまなコンポーネントに使用されるインフラストラクチャ要素を起動し、確認します。インフラストラクチャを設定するときは、次のようなクラウドごとの特徴を考慮します。

AWS または OpenStack クラウド環境では、次の段階(「フェーズ 2:ネットワーク ルールの設定」)で通常このタスクを実行する際にセキュリティ グループの設定をスキップできます。

CloudCenter コンポーネントの VM を起動するために使用されるインスタンスは、管理される VM の数に応じて、最小で **2 CPU、4GB RAM、10GB ストレージ**が必要です。詳細については、「*ハードウェア要件*」のセクション(後述)を参照してください。

CloudCenter をインストールするときに、サーバ名または IP アドレスを必要とするさまざまな設定項目があります。**IP アドレスをより適切に管理できるように、DNS 名を使用してサーバを特定します。**

Amazon の特徴

仮想 IP(VIP)またはマスター/スレーブ セットアップ用の Elastic IP(EIP)を有効にするために特定の IAM ロールを使用して、**MGMTPOSTGRES_MASTER** と **MGMTPOSTGRES_SLAVE** の VM を起動します(「コンポーネントのモードとロール」を参照)。

- IAM ロールを使用する理由

  アクセス キーや秘密キーなどを指定する代わりに、IAM ロールの使用によってインスタンス タイプすべてを管理できます。デフォルトではこの機能は無効なので、このボタンを [ON] に変更することで、明示的に有効にする必要があります。追加情報については、「AWS Configurations」>「*IAM and STS*」を参照してください。

- 仮想 IP または Elastic IP の選択

  VIP または EIP の使用に関するガイドライン:

  - **VIP を使用:**
    - 同じ VPC から MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE の両方を起動する場合。
    - VIP を使用している場合、VPC 内でのサブネットの 1 つから未使用の IP を選択します
  - **EIP を使用:**
    - 別の VPC から MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE の両方を起動する場合。
    - EIP を使用している場合、VPC としてスコープを使用して AWS クラウド コンソールから新しい IP を割り当てます。
    - 追加情報については、「VPC peering」を参照してください。

- VIP または EIP に特有な IAM ロールの作成

ポリシー	IAM ロール

VIP	<pre>{
	"Version": "2012-10-17",
	"Statement": [{
	"Action": [
	"ec2:AssignPrivateIpAddresses",
	"ec2:UnassignPrivateIpAddresses",
	"ec2:DescribeInstances",
	"ec2:DescribeNetworkInterfaces"
	],
	"Effect": "Allow",
	"Resource": "*"
	}]
	}</pre>

ポリシー	IAM ロール

EIP	<pre>{
	"Version": "2012-10-17",
	"Statement": [{
	"Action": [
	"ec2:DescribeInstances",
	"ec2:DescribeNetworkInterfaces",
	"ec2:AssociateAddress",
	"ec2:DisassociateAddress",
	"ec2:DescribeAddresses"
	],
	"Effect": "Allow",
	"Resource": "*"
	}]
	}</pre>

OpenStack の特徴

マスター/スレーブ セットアップ用の仮想 IP（VIP）を使用して、**MGMTPOSTGRES_MASTER** と **MGMTPOSTGRES_SLAVE** の VM を起動します（「コンポーネントのモードとロール」を参照）。

- OpenStack サーバにアクセスできる VM にインストールされた OpenStack Neutron クライアントへのアクセス
  1. OpenStack のコマンドライン クライアントをインストールします。
  2. **MGMTPOSTGRES_MASTER** と **MGMTPOSTGRES_SLAVE VM** を起動します
- ここをクリックして展開します
  1. VIP マスター/スレーブの設定用に未使用の IP を確認します。
  2. 次のコマンドを使用して、この IP で新しいポートを作成します。

     ```
     `neutron port-create --fixed-ip ip_address=<VIP> <network-name>`
     ```

  3. MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE VM のポートの ID を取得します。

     ```
     `neutron port-list`
     ```

  4. **fixed_ips** の MGMTPOSTGRES_MASTER および MGMTPOSTGRES_SLAVE IP に対応するリストから **id** をメモします。
  5. 前のステップでメモした各ポート ID に対して次のコマンドを実行し、VIP を使用して MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE に関連付けられているポートを更新します。

     ```
     `neutron port-update <port-id> --allowed-address-pairs type=dict
     list=true ip_address=<VIP>`
     ```

- ホスト名の設定

  起動されたすべての VM で、ホスト名を目的のホスト名に更新します。ロールに一致するホスト名を選択します。たとえば、*CCM_SA_PRIMARY.mydomain.com* です。

  PCS 認証が失敗する原因になるので、データベース サーバのホスト名にアンダースコアを含めることはできません。

  VM のインストールおよび設定後にホスト名を変更すると、未知の問題を引き起こす可能性があります。

- ホスト名の解決の設定

  ホスト名を更新したら、次のコマンドを実行して、VM ホスト名が解決可能であることを確認してください。

  ```
  hostname -i
  ```

  VM の名前が解決可能であれば、**/etc/hosts** ファイルを編集し、VM のホスト名を追加します。例を示します。*<hostname> 127.0.0.1*

- ネットワーク ルーティングのループバック

  ネットワーク ルーティングのループバックは CCM を参照します。これは指定した導入の NAT の背後にあり、内部ネットワークのサーバは外部 IP を使用して CCM にアクセスできないという制限があります。

  このシナリオでは、登録した CCO と AMQP サーバの /etc/hosts ファイルに行を追加して、CCM の内部プライベート IP を提供する必要があります。

  次に例を示します。

  CCM DNS 名が *ccm.example.com* で、NAT の背後にあり、内部プライベート IP アドレスが 12.16.0.5 で、外部パブリック IP アドレスが 54.1.2.5 である場合、ローカルの /etc/hosts ファイルに次の行を追加します

  12.16.0.5 ccm.example.com

  上記の例で使用したホスト名（ccm.example.com）はフェーズ 4 で CCM を設定する際に、***Public DNS*** として設定されているものと一致させる必要があります。

CloudCenter 記述子 JSON ファイル

CloudCenter のコンポーネントすべてについてインフラストラクチャを設定した後は、CloudCenter のすべてのコンポーネントを、各モードとロールのインフラストラクチャ要素に対応するコンポーネントのモードおよび IP アドレスとともにリストする CloudCenter 記述子 JSON ファイルを作成します。この記述子ファイルはネットワーク準拠チェックのためにフェーズ 3 で使用されます。このセクションでは、コンポーネント モードの共通の組み合わせに基づき、サンプルの記述子ファイルを提供します。

全体のファイル構造は次の要因によって異なります。

- **各コンポーネントのモード**
- **クラウド リージョンの数**
- **条件/オプション コンポーネント、リポジトリなどの使用。**
- **このファイルで使用される固有のリージョン名。これらの名前は、すべてのクラウドやデータセンター名と一致する必要がないことに注意してください。これらの文字列は、ネットワーク コンプライアンス チェックの実行および結果のレポートにのみ使用されます**

ローカルのバンドル ストア、パッケージ ストア、または専用 Docker レジストリを設定している場合は、リポジトリ値をサーバ値と交換します（特定の設定に対するそのほかの変更も同様）。

DNS 名を設定したら、IP アドレスではなく、DNS 名を使用します。

- 記述子ファイル：例 1

```
{
    "CloudCenterComponents": {
        "CCM": {
            "CCM_IP": "CCM.Company1.com",
            "mode": "NON-HA "
        },
        "CloudRegions": [{
            "components": {
                "AMQP": {
                    "AMQP_IP": "AMQP.Company1.com",
                    "mode": "NON-HA"
                },
                "CCO": {
                    "CCO_IP": "CCO.Company1.com",
```

```
                    "mode": "NON-HA"
                }
            },
            "name": "AWSEast"
        }],
        "REPOS": {
            "BUNDLE_STORE": "http://cdn.cliqr.com",
            "DOCKER_REGISTRY": "http://repo.cliqrtech.com:5000",
            "PACKAGE_STORE": "http://repo.cliqrtech.com"
        }
    }
}
```

- 記述子ファイル：例 2

```
{
    "CloudCenterComponents": {
        "CCM": {
            "CCM_SA_IP": "CCM.Company2.com",
            "MGMTPOSTGRES_IP": "MGMTPOSTGRES_SA.Company2.com",
            "mode": "STANDALONE"
        },
        "CloudRegions": [{
            "components": {
                "AMQP": {
                    "AMQP_IP": "AMQP.Company2.com",
                    "mode": "NON-HA"
                },
                "CCO": {
                    "CCO_IP": "CCO.Company2.com",
                    "mode": "NON-HA"
                }
            },
            "name": "AWSEast"
        }],
        "REPOS": {
            "BUNDLE_STORE": "http://cdn.cliqr.com",
            "DOCKER_REGISTRY": "http://repo.cliqrtech.com:5000",
            "PACKAGE_STORE": "http://repo.cliqrtech.com"
        }
    }
}
```

- 記述子ファイル：例 3

```
{
    "CloudCenterComponents": {
        "CCM": {
            "CCM_LB_IP": "169.0.1.1",
            "CCM_SA_PRIMARY_IP": "169.0.1.2",
            "CCM_SA_SECONDARY_IP": "169.0.1.3",
            "MGMTPOSTGRES_MASTER_IP": "169.0.1.4",
            "MGMTPOSTGRES_SLAVE_IP": "169.0.1.5",
            "MGMTPOSTGRES_VIP": "169.0.1.6",
            "mode": "HA"
        },
        "CloudRegions": [{
            "components": {
                "AMQP": {
                    "AMQP_LB_IP": "169.0.1.7",
                    "AMQP_PRIMARY_IP": "169.0.1.8",
                    "AMQP_SECONDARY_IP": "169.0.1.9",
                    "mode": "HA"
                },
                "CCO": {
                    "CCO_LB_IP": "169.0.1.10",
                    "CCO_PRIMARY_IP": "169.0.1.11",
                    "CCO_SECONDARY_IP": "169.0.1.12",
                    "mode": "HA"
                },
                "Dedicated_External_Script_Executor": {
                    "EXT_SCRIPT_EXECUTOR_IP": "169.0.1.13",
                    "mode": "NON-HA"
                },
                "Dedicated_Guacamole_Server": {
                    "GUAC_IP": "169.0.1.14",
                    "mode": "NON-HA"
                }
            },
            "name": "AWSEast"
        }, {
            "components": {
                "AMQP": {
                    "AMQP_LB_IP": "169.0.1.15",
                    "AMQP_PRIMARY_IP": "169.0.1.16",
                    "AMQP_SECONDARY_IP": "169.0.1.17",
                    "mode": "HA"
                },
                "CCO": {
                    "CCO_LB_IP": "169.0.1.18",
                    "CCO_PRIMARY_IP": "169.0.1.19",
                    "CCO_SECONDARY_IP": "2.2.2.2",
                     "mode": "HA"
                },
                "Dedicated_External_Script_Executor": {
                     "EXT_SCRIPT_EXECUTOR_IP": "169.0.1.20",
                     "mode": "NON-HA"
                },
                "Dedicated_Guacamole_Server": {
                     "GUAC_IP": "169.0.1.21",
                     "mode": "NON-HA"
                }
            },
            "name": "MyOpenStack"
        }],
```

```
        "MON": {
            "MON_LB": "169.0.1.22",
            "MON_PRIMARY": "169.0.1.23",
            "MON_SECONDARY": "169.0.1.24",
            "mode": "HA"
        },
        "REPOS": {
            "BUNDLE_STORE": "http://cdn.cliqr.com",
            "DOCKER_REGISTRY": "http://repo.cliqrtech.com:5000",
            "PACKAGE_STORE": "http://repo.cliqrtech.com"
        }
    }
}
```

ハードウェア要件

CloudCenter のコンポーネントには、企業向けに管理する VM インスタンスの数に基づく特定のハードウェア要件があります。

- CCM

  ### CCM ハードウェア要件

VM の合計数	単一 CCM ハードウェア要件
< 500	2 CPU、4GB メモリ、50GB ストレージ
501 ～ 2000	2 CPU、8GB メモリ、100GB ストレージ
2001 ～ 10000	4 CPU、16GB メモリ、200GB ストレージ
> 10000	4 CPU、32GB メモリ、500GB ストレージ

- CCO

  ### CCO ハードウェア要件

  ハードウェア要件は、クラウド間の差異により、各クラウドで異なることがあります。

VM の合計数	単一 CCO のハードウェア要件（各クラウドで異なる可能性あり）
< 500	2 CPU、4GB メモリ、50GB ストレージ
501 ～ 2000	2 CPU、8GB メモリ、100GB ストレージ
2001 ～ 10000	4 CPU、16GB メモリ、200GB ストレージ
> 10000	4 CPU、32GB メモリ、500GB ストレージ

- AMQP

  ### AMQP ハードウェア要件

  AMQP のコンポーネントのハードウェア仕様。

VM の合計数	単一 AMQP のハードウェア要件（各クラウドで異なる可能性あり）
< 500	2 CPU、4GB メモリ、50GB ストレージ
501 ～ 2000	2 CPU、8GB メモリ、100GB ストレージ
2001 ～ 10000	4 CPU、16GB メモリ、200GB ストレージ
> 10000	4 CPU、32GB メモリ、500GB ストレージ

- その他のイメージ

その他のイメージのハードウェア要件

追加のコンポーネントのハードウェア仕様（必要な場合）

追加コンポーネント	ハードウェア要件（各クラウドで異なる可能性あり）
クラウド ヘルス モニタ（オプション）	2 CPU、4GB メモリ、50GB ストレージ
バンドル ストア（オプション）	2 CPU、4GB メモリ、50GB ストレージ
パッケージ ストア（オプション）	2 CPU、4GB メモリ、100GB ストレージ

ディスク サイズを増やす

VM のパーティションのサイズを増やすには、該当するクラウドのマニュアルを参照してください。

通常、ルート ディスクはクラウド イメージによって制御され、CloudCenter からはサイズ変更されません。ほとんどのクラウドには自動化されたサイズ変更オプションはありません。アプリケーション VM のルート ディスクは導入に使用されるイメージと必ず同じサイズです。追加情報については、「Manage Instance Types」を参照してください。

フェーズ 2：ネットワーク ルールの設定

ネットワーク ルールの概要

このフェーズでは、さまざまなコンポーネント間の通信を可能にするネットワーク ルールを設定する必要があります。

このページのネットワーク設定は、コンポーネントの相互通信のためのポートの最小要件について示しています。すべてのコンポーネントがすべてのポートを介して相互に通信できる環境（通常は POC 環境またはプライベート データセンター）ではこのフェーズを省略できます。

実稼働環境は、このセクションで指定されたポート経由でのみ通信できるようにすることで、セキュリティで保護されます。

セキュリティ グループ

各 VM のネットワーク ルールを設定するには、CloudCenter のコンポーネント間の適切な相互接続を設定する必要があります。

AWS または OpenStack 上で開始される VM に対し、セキュリティ グループを設定し、VM に関連付けることによって、このタスクを達成しました。

- すべてのポートの要件は TCP プロトコルを使用します。
- コンポーネントと HTTPS アクセスの間のすべての通信では、SSL プロトコルとして TLS を使用します。

このセクションの表は、各コンポーネントのロールのネットワーク要件を示します。

CCM ネットワーク ルール

CCM ネットワーク ルール

CCM

CCM ポート

ポート	方向	リモート送信元	注
80	入力（オプション）	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	HTTP から HTTPS へのリダイレクション用。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	UI/API アクセス用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	CCO_IP または CCO_PRIMARY_IP および CCO_SECONDARY_IP	CCO と CCM VM 間の双方向通信用。
	出力	CCO_IP または CCO_LB_IP	⚠ CloudCenter の導入でサポートされるすべてのクラウド リージョンに必要です。
	入力（オプション）	MON_IP または MON_PRIMARY_IP および MON_SECONDARY_IP	モニタ CM および CCM VM 間の双方向通信用。
	出力（オプション）	MON_IP または MON_LB_IP	
	入力	AMQP_IP または AMQP_PRIMARY_IP および AMQP_SECONDARY_IP	Guacamole による Web SSH/VNC 用。

CCM_SA

CCM_SA ポート

ポート	方向	リモート送信元	注
80	入力（オプション）	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	HTTP から HTTPS へのリダイレクション用。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	UI/API アクセス用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	CCO または CCO_PRIMARY、CCO_SECONDARY	CCO と CCM VM 間の双方向通信用。
	出力	CCO または CCO_LB	
	入力（オプション）	MON または MON_PRIMARY および MON_SECONDARY	モニタ VM と CCM VM 間の双方向通信用。
	出力（オプション）	MON、MON_LB	
	入力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY	Guacamole による Web SSH/VNC 用。
5432	出力	MGMTPOSTGRES	データベースへの通信用。

MGMTPOSTGRES

MGMTPOSTGRES ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
5432	入力	CCM_SA	CCM スタンドアロン VM からの着信接続用。

CCM_SA_PRIMARY および CCM_SA_SECONDARY

CCM_SA_PRIMARY および CCM_SA_SECONDARY ポート

ポート	方向	リモート送信元	注
443	入力	CCM_LB	CCM ロード バランサ VM からの着信接続用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
	入力/出力	CCM_SA_PRIMARY または CCM_SA_SECONDARY	CCM プライマリおよびセカンダリ VM 間の静的ファイル同期用。
8443	入力	CCM_LB	CCM ロード バランサ VM からの着信接続用。
	出力	CCO または CCO_LB	CCO VM への通信用。
	出力（オプション）	MON または MON_LB	モニタ VM への通信用。
5703	入力/出力	CCM_SA_PRIMARY または CCM_SA_SECONDARY	HA のデータの処理の内部実装用。
5432	出力	MGMTPOSTGRES または MGMTPOSTGRES_VIP	データベースへの通信用。

MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE

MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
5432	入力	CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM スタンドアロン VM からの着信接続用。
	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	マスターとスレーブのデータベース VM 間の通信用。
5405（UDP）	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	高可用性を実現するための Corosync Cluster Engine グループの通信システム用。
2224	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	高可用性を実現するための両方のデータベース VM 間のペースメーカー クラスタリング用。
3121	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	
21064	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	

CCM_LB

CCM_LB ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
80	入力（オプション）	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	HTTP から HTTPS へのリダイレクション用。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	UI/API アクセス用。
	出力	CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM プライマリおよびセカンダリ VM 間の通信用。
8443	出力	CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM プライマリおよびセカンダリ VM 間の通信用。
	入力	CCO または CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCO VM からの通信用。
	入力（オプション）	MON または MON_PRIMARY、MON_SECONDARY	モニタ VM からの通信用。
	入力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY	Guacamole による Web SSH/VNC 用。

CCO ネットワーク ルール

CCO ネットワーク ルール

CCO ポート

CCO ポート

ポート	方向	リモート送信元	注
8443	入力/出力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCO と CCM VM 間の双方向通信用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
すべて	出力（条件付き）	クラウド リージョンのエンドポイント、スクリプト ソース	クラウド リージョンのエンドポイント アクセスと外部サービスで定義されているスクリプト/パッケージのダウンロード用。
2375	出力（条件付き）	EXT_SCRIPT_EXECUTOR	外部スクリプトを実行するための Docker コンテナ エンジン アクセス用。

CCO_PRIMARY と CCO_SECONDARY

CCO_PRIMARY と CCO_SECONDARY ポート

ポート	方向	リモート送信元	注
8443	入力/出力	CCO_LB	CCO と CCM VM 間の双方向通信用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
すべて	出力（条件付き）	クラウド リージョンのエンドポイント、スクリプト ソース	クラウド リージョンのエンドポイント アクセスと外部サービスで定義されているスクリプト/パッケージのダウンロード用。
2375	出力（条件付き）	EXT_SCRIPT_EXECUTOR	外部スクリプトを実行するための Docker コンテナ エンジン アクセス用。
5701	入力/出力	CCO_PRIMARY、CCO_SECONDARY	HA のデータの処理の内部実装用。

CCO_LB

CCO_LB ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM VM から CCO への通信用。
	出力	CCO または CCO_PRIMARY、CCO_SECONDARY	CCO のロード バランサから CCO VM への通信用。

AMQP ネットワーク ルール

AMQP ネットワーク ルール

AMQP

AMQP ポート

ポート	方向	リモート送信元	注
22	入力(オプション)	許可された SSH の送信元 IP	トラブルシューティングの目的。
5671	入力	• CCO または CCO_PRIMARY、CCO_SECONDARY • ワーカー VM IP の範囲	CCO VM からの通信および起動された VM からの通信用。
7789	入力	ワーカー VM IP の範囲	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
7788	入力/出力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY、AMQP_LB	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
443	入力	0.0.0.0/0(またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲)	起動された VM の SSH/VNC および RDP アクセス用。
8443	出力	• CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONADRY • CCO または CCO_PRIMARY、CCO_SECONDARY	起動された VM の SSH/VNC アクセス用。AMQP VM の Guacamole サーバはこのポートによって CCM および CCO VM と通信します。

AMQP_PRIMARY と AMQP_SECONDARY

AMQP_PRIMARY と AMQP_SECONDARY ポート

ポート	方向	リモート送信元	注
22	入力(オプション)	許可された SSH の送信元 IP	トラブルシューティングの目的。
5671	入力	• CCO または CCO_PRIMARY、CCO_SECONDARY • ワーカー VM IP の範囲	CCO VM からの通信および起動された VM からの通信用
7789	入力	ワーカー VM IP の範囲	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行
7788	入力/出力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
443	入力	0.0.0.0/0(またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲)	起動された VM の SSH/VNC および RDP アクセス用。
8443	出力	• CCM または CCM_SA または CCM_LB • CCO または CCO_LB	起動された VM の SSH/VNC アクセス用。AMQP の Guacamole サーバはこのポートで CCM および CCO と通信します。
4369	入力/出力	AMQP_PRIMARY、AMQP_SECONDARY	AMQP プライマリおよびセカンダリ VM 間の通信用。
25672	入力/出力	AMQP_PRIMARY、AMQP_SECONDARY	AMQP プライマリおよびセカンダリ VM 間の通信用。

AMQP_LB

AMQP_LB ポート

ポート	方向	リモート送信元	注
22	入力(オプション)	• 許可された SSH の送信元 IP	トラブルシューティングの目的。
5671	入力	• CCO または CCO_PRIMARY、CCO_SECONDARY • ワーカー VM IP の範囲	CCO VM からの通信および起動された VM からの通信用。
7789	入力	ワーカー VM IP の範囲	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。

ポート	方向	リモート送信元	注
7788	入力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY、AMQP_LB	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。

外部スクリプト ネットワーク ルール

EXT_SCRIPT_EXECUTOR ネットワーク ルール

EXT_SCRIPT_EXECUTOR ポート

EXT_SCRIPT_EXECUTOR ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
2375	入力	CCO または CCO_PRIMARY、CCO_SECONDARY	CCO VM の Docker エンジン エンドポイントへのアクセス用

Guacamole ネットワーク ルール

GUAC ネットワーク ルール

GUAC ポート

GUAC ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
7789	入力	ワーカー VM IP の範囲	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
7788	入力/出力	GUAC	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	起動された VM の SSH/VNC および RDP アクセス用。
8443	出力	• CCM または CCM_SA または CCM_LB • CCO または CCO_LB	起動された VM の SSH/VNC アクセス用。AMQP VM の Guacamole サーバはこのポートで CCM および CCO VM と通信します。

モニタ ネットワーク ルール

モニタ ネットワーク ルール

MON ポート

MON ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力/出力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY、CCM_LB	CCM およびモニタ VM 間の双方向通信用です。
	出力	CCO または CCO_LB	CCO VM へのアクセス用。

MON_PRIMARY と MON_SECONDARY ポート

MON_PRIMARY と MON_SECONDARY ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	MON_LB	モニタ ロード バランサからの通信用。
	出力	CCO または CCO_LB	CCO VM へのアクセス用。

MON_LB ポート

MON_LB ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM VM からの通信用。
	出力	MON_PRIMARY、MON_SECONDARY	モニタ VM へのアクセス用。

バンドル ストアのネットワーク ルール

バンドル ストアのネットワーク ルール

BUNDLE_STORE ポート

BUNDLE_STORE ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	SSH アクセス用。
80	入力	ワーカー VM IP の範囲	アプリケーション VM（ワーカー）のブートストラップ スクリプトとエージェント パッケージのダウンロード用

パッケージ ストアのネットワーク ルール

PACKAGE_STORE ネットワーク ルール

PACKAGE_STORE ポート

PACKAGE_STORE ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	SSH アクセス用。
	出力	パブリック リポジトリ サーバ：repo.cliqrtech.com	すべての OS、コンポーネントとサービス パッケージの同期用
80	入力	すべてのコンポーネントとワーカー VM IP の範囲	アプリケーション VM（ワーカー）のブートストラップ スクリプトとエージェント パッケージのダウンロード用
5000	入力（条件付き）	ワーカー VM IP の範囲	アプリケーション VM（ワーカー）のコンテナ サービスのイメージのダウンロード用

Docker レジストリ ネットワーク ルール

DOCKER_REGISTRY ネットワーク ルール

DOCKER_REGISTRY ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	SSH アクセス用。
5000	入力	ワーカー VM IP の範囲	ワーカー VM のコンテナ サービス イメージのダウンロード用。

プロキシの設定

プロキシ サーバがインターネットに接続する必要がある場合は、このプロセスに従ってください。

1. プロキシ設定を反映するようにプロキシ環境を変更します。
2. CloudCenter プラットフォームでは次の行が CCM および CCO の JAVA_OPTS に追加されることが必要です。
   -Dhttp.proxyHost=<proxy_hostname> -Dhttp.proxyPort=<port_number>
3. アプリケーション VM は、追加の設定が必要になることがあります。詳細については、CloudCenter のサポート チームにお問い合わせください。

ネットワーク ルールの概要

ネットワーク ルールの概要

このフェーズでは、さまざまなコンポーネント間の通信を可能にするネットワーク ルールを設定する必要があります。

このページのネットワーク設定は、コンポーネントの相互通信のためのポートの最小要件について示しています。すべてのコンポーネントがすべてのポートを介して相互に通信できる環境（通常は POC 環境またはプライベート データセンター）ではこのフェーズを省略できます。

実稼働環境は、このセクションで指定されたポート経由でのみ通信できるようにすることで、セキュリティで保護されます。

セキュリティ グループ

各 VM のネットワーク ルールを設定するには、CloudCenter のコンポーネント間の適切な相互接続を設定する必要があります。

AWS または OpenStack 上で開始される VM に対し、セキュリティ グループを設定し、VM に関連付けることによって、このタスクを達成しました。

- すべてのポートの要件は TCP プロトコルを使用します。
- コンポーネントと HTTPS アクセスの間のすべての通信では、SSL プロトコルとして TLS を使用します。

AMQP ネットワーク ルール

AMQP ネットワーク ルール

AMQP

AMQP ポート

ポート	*方向*	*リモート送信元*	*注*
22	*入力（オプション）*	*許可された SSH の送信元 IP*	*トラブルシューティングの目的。*
5671	*入力*	• *CCO または CCO_PRIMARY、CCO_SECONDARY* • *ワーカー VM IP の範囲*	*CCO VM からの通信および起動された VM からの通信用。*

ポート	*方向*	*リモート送信元*	*注*
7789	*入力*	*ワーカー VM IP の範囲*	*起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。*
7788	*入力/出力*	*AMQP または AMQP_PRIMARY、AMQP_SECONDARY、AMQP_LB*	*起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。*
443	*入力*	*0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）*	*起動された VM の SSH/VNC および RDP アクセス用。*
8443	*出力*	• *CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONADRY* • *CCO または CCO_PRIMARY、CCO_SECONDARY*	*起動された VM の SSH/VNC アクセス用。AMQP VM の Guacamole サーバはこのポートによって CCM および CCO VM と通信します。*

AMQP_PRIMARY と AMQP_SECONDARY

AMQP_PRIMARY と AMQP_SECONDARY ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
5671	入力	• CCO または CCO_PRIMARY、CCO_SECONDARY • ワーカー VM IP の範囲	CCO VM からの通信および起動された VM からの通信用
7789	入力	ワーカー VM IP の範囲	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行
7788	入力/出力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	起動された VM の SSH/VNC および RDP アクセス用。
8443	出力	• CCM または CCM_SA または CCM_LB • CCO または CCO_LB	起動された VM の SSH/VNC アクセス用。AMQP の Guacamole サーバはこのポートで CCM および CCO と通信します。
4369	入力/出力	AMQP_PRIMARY、AMQP_SECONDARY	AMQP プライマリおよびセカンダリ VM 間の通信用。
25672	入力/出力	AMQP_PRIMARY、AMQP_SECONDARY	AMQP プライマリおよびセカンダリ VM 間の通信用。

AMQP_LB

AMQP_LB ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	• 許可された SSH の送信元 IP	トラブルシューティングの目的。
5671	入力	• CCO または CCO_PRIMARY、CCO_SECONDARY • ワーカー VM IP の範囲	CCO VM からの通信および起動された VM からの通信用。
7789	入力	ワーカー VM IP の範囲	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
7788	入力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY、AMQP_LB	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。

バンドル ストアのネットワーク ルール

バンドル ストアのネットワーク ルール

BUNDLE_STORE ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	SSH アクセス用。
80	入力	ワーカー VM IP の範囲	アプリケーション VM（ワーカー）のブートストラップ スクリプトとエージェント パッケージのダウンロード用

CCM ネットワーク ルール

CCM ネットワーク ルール

CCM ポート

ポート	方向	リモート送信元	注
80	入力（オプション）	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	HTTP から HTTPS へのリダイレクション用。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	UI/API アクセス用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	CCO_IP または CCO_PRIMARY_IP および CCO_SECONDARY_IP	CCO と CCM VM 間の双方向通信用。
	出力	CCO_IP または CCO_LB_IP	⚠ CloudCenter の導入でサポートされるすべてのクラウド リージョンに必要です。
	入力（オプション）	MON_IP または MON_PRIMARY_IP および MON_SECONDARY_IP	モニタ CM および CCM VM 間の双方向通信用。
	出力（オプション）	MON_IP または MON_LB_IP	
	入力	AMQP_IP または AMQP_PRIMARY_IP および AMQP_SECONDARY_IP	Guacamole による Web SSH/VNC 用。

CCM_SA

CCM_SA ポート

ポート	方向	リモート送信元	注
80	入力（オプション）	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	HTTP から HTTPS へのリダイレクション用。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	UI/API アクセス用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	CCO または CCO_PRIMARY、CCO_SECONDARY	CCO と CCM VM 間の双方向通信用。
	出力	CCO または CCO_LB	
	入力（オプション）	MON または MON_PRIMARY および MON_SECONDARY	モニタ VM と CCM VM 間の双方向通信用。
	出力（オプション）	MON、MON_LB	
	入力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY	Guacamole による Web SSH/VNC 用。
5432	出力	MGMTPOSTGRES	データベースへの通信用。

MGMTPOSTGRES

MGMTPOSTGRES ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
5432	入力	CCM_SA	CCM スタンドアロン VM からの着信接続用。

CCM_SA_PRIMARY と CCM_SA_SECONDARY

CCM_SA_PRIMARY および CCM_SA_SECONDARY ポート

ポート	方向	リモート送信元	注
443	入力	CCM_LB	CCM ロード バランサ VM からの着信接続用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
	入力/出力	CCM_SA_PRIMARY または CCM_SA_SECONDARY	CCM プライマリおよびセカンダリ VM 間の静的ファイル同期用。
8443	入力	CCM_LB	CCM ロード バランサ VM からの着信接続用。
	出力	CCO または CCO_LB	CCO VM への通信用。
	出力（オプション）	MON または MON_LB	モニタ VM への通信用。
5703	入力/出力	CCM_SA_PRIMARY または CCM_SA_SECONDARY	HA のデータの処理の内部実装用。
5432	出力	MGMTPOSTGRES または MGMTPOSTGRES_VIP	データベースへの通信用。

MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE

MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
5432	入力	CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM スタンドアロン VM からの着信接続用。
	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	マスターとスレーブのデータベース VM 間の通信用。
5405（UDP）	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	高可用性を実現するための Corosync Cluster Engine グループの通信システム用。
2224	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	高可用性を実現するための両方のデータベース VM 間のペースメーカー クラスタリング用。
3121	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	
21064	入力/出力	MGMTPOSTGRES_MASTER、MGMTPOSTGRES_SLAVE	

CCM_LB

CCM_LB ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
80	入力（オプション）	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	HTTP から HTTPS へのリダイレクション用。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	UI/API アクセス用。
	出力	CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM プライマリおよびセカンダリ VM 間の通信用。

ポート	方向	リモート送信元	注
8443	出力	CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM プライマリおよびセカンダリ VM 間の通信用。
	入力	CCO または CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCO VM からの通信用。
	入力（オプション）	MON または MON_PRIMARY、MON_SECONDARY	モニタ VM からの通信用。
	入力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY	Guacamole による Web SSH/VNC 用。

CCO ネットワーク ルール

CCO ネットワーク ルール

CCO ポート

ポート	方向	リモート送信元	注
8443	入力/出力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCO と CCM VM 間の双方向通信用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
すべて	出力（条件付き）	クラウド リージョンのエンドポイント、スクリプト ソース	クラウド リージョンのエンドポイント アクセスと外部サービスで定義されているスクリプト/パッケージのダウンロード用。
2375	出力（条件付き）	EXT_SCRIPT_EXECUTOR	外部スクリプトを実行するための Docker コンテナ エンジン アクセス用。

CCO_PRIMARY と CCO_SECONDARY

CCO_PRIMARY と CCO_SECONDARY ポート

ポート	*方向*	*リモート送信元*	*注*
8443	*入力/出力*	*CCO_LB*	*CCO と CCM VM 間の双方向通信用。*
22	*入力（オプション）*	*許可された SSH の送信元 IP*	*トラブルシューティングの目的。*
すべて	*出力（条件付き）*	*クラウド リージョンのエンドポイント、スクリプト ソース*	*クラウド リージョンのエンドポイント アクセスと外部サービスで定義されているスクリプト/パッケージのダウンロード用。*
2375	*出力（条件付き）*	*EXT_SCRIPT_EXECUTOR*	*外部スクリプトを実行するための Docker コンテナ エンジン アクセス用。*
5701	*入力/出力*	*CCO_PRIMARY、CCO_SECONDARY*	*HA のデータの処理の内部実装用。*

CCO_LB

CCO_LB ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM VM から CCO への通信用。
	出力	CCO または CCO_PRIMARY、CCO_SECONDARY	CCO のロード バランサから CCO VM への通信用。

DOCKER_REGISTRY ネットワーク ルール

DOCKER_REGISTRY ネットワーク ルール

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	SSH アクセス用。
5000	入力	ワーカー VM IP の範囲	ワーカー VM のコンテナ サービス イメージのダウンロード用。

Ext_Script ネットワーク ルール

EXT_SCRIPT_EXECUTOR ネットワーク ルール

EXT_SCRIPT_EXECUTOR ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
2375	入力	CCO または CCO_PRIMARY、CCO_SECONDARY	CCO VM の Docker エンジン エンドポイントへのアクセス用

Guacamole ネットワーク ルール

GUAC ネットワーク ルール

GUAC ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
7789	入力	ワーカー VM IP の範囲	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
7788	入力/出力	GUAC	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	起動された VM の SSH/VNC および RDP アクセス用。
8443	出力	• CCM または CCM_SA または CCM_LB • CCO または CCO_LB	起動された VM の SSH/VNC アクセス用。AMQP VM の Guacamole サーバはこのポートで CCM および CCO VM と通信します。

モニタ ネットワーク ルール

モニタ ネットワーク ルール

MON ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力/出力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY、CCM_LB	CCM およびモニタ VM 間の双方向通信用です。
	出力	CCO または CCO_LB	CCO VM へのアクセス用。

MON_PRIMARY と MON_SECONDARY ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	MON_LB	モニタ ロード バランサからの通信用。
	出力	CCO または CCO_LB	CCO VM へのアクセス用。

MON_LB ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCM VM からの通信用。
	出力	MON_PRIMARY、MON_SECONDARY	モニタ VM へのアクセス用。

パッケージ ストアのネットワーク ルール

PACKAGE_STORE ネットワーク ルール

PACKAGE_STORE ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	SSH アクセス用。
	出力	パブリック リポジトリ サーバ: repo.cliqrtech.com	すべての OS、コンポーネントとサービス パッケージの同期用
80	入力	すべてのコンポーネントとワーカー VM IP の範囲	アプリケーション VM（ワーカー）のブートストラップ スクリプトとエージェント パッケージのダウンロード用
5000	入力（条件付き）	ワーカー VM IP の範囲	アプリケーション VM（ワーカー）のコンテナ サービスのイメージのダウンロード用

プロキシの設定

プロキシの設定

プロキシ サーバがインターネットに接続する必要がある場合は、このプロセスに従ってください。

1. プロキシ設定を反映するようにプロキシ環境を変更します。
2. CloudCenter プラットフォームでは次の行が CCM および CCO の JAVA_OPTS に追加されることが必要です。
   -Dhttp.proxyHost=<proxy_hostname> -Dhttp.proxyPort=<port_number>
3. アプリケーション VM は、追加の設定が必要になることがあります。詳細については、CloudCenter のサポート チームにお問い合わせください。

フェーズ 3：ネットワーク コンプライアンス チェックの実行

- ネットワーク通信
- CloudCenter のコンポーネント VM
- ワーカー VM
- クライアント テスト

ネットワーク通信

すべてのネットワーク ルールが、さまざまな CloudCenter コンポーネント、ワーカー、VM およびエンドユーザ アクセス全体で相互通信用に正しく設定されていることを確認するために、ネットワーク コンプライアンス チェックを実行する必要があります。

CloudCenter のコンポーネント VM

CloudCenter のすべてのコンポーネントについて、ネットワーク コンプライアンス チェックを行うには、「**フェーズ 1：インフラストラクチャの準備**」**の一部として起動されるすべての仮想マシンで**次の手順を実行します。

1. 「フェーズ 1：インフラストラクチャの準備」で作成したデプロイメント記述子 JSON ファイルを /tmp の場所にアップロードします。
2. **network_validator.py**（CloudCenter インストーラ ファイルで利用可能）ファイルと呼ばれるバリデータ スクリプトを /tmp の場所にアップロードします。
3. 次のコマンドを /tmp の場所から実行します。

```
python
network_validator.py --role=<role> --region=<unique region identifier>
--deploymentDetails=<envdeployment descriptor JSON file path>
```

次にこのコマンドの実行例を示します。

```
python
network_validator.py --role=CCM --deploymentDetails=deployment.json
```

```
python
network_validator.py --role=CCO_PRIMARY --region=AmazonEast1
--deploymentDetails=deployment.json
```

```
python
network_validator.py --role=AMQP_LB --region=AmazonEast1
--deploymentDetails=deployment.json
```

4. 各 CCM、CCO、AMQP コンポーネントの network_validator.py スクリプトを実行した後、cloudcenter_validator.json という名前のファイルが、CCM、CCO、AMQP サーバの /tmp の場所に作成されます。次の例は、CCM 用に作成された cloudcenter_validator.json ファイルです。

```
[{
    "TO_ROLE": "CCO",
    "region": "AWSEast",
    "TO_IP": "12.1.4.5",
    "PORT": 8443,
    "RESULT": "Success"
}, {
    "TO_ROLE": "PACKAGE_STORE",
    "TO_IP": "repo.cliqrtech.com",
```

```
    "PORT": 80,
    "RESULT": "Success"
}]
```

5. 次の点を確認するには、このコマンドの結果を確認してください。
   - 障害は発生しませんでした。障害が発生した場合、ネットワーク設定を適切に更新して、テストを繰り返します。
   - すべての必須ネットワーク ルールが正しく設定されました。

ワーカー VM

アプリケーション オーケストレーション プロセスの一部として起動されるワーカー VM が関連する CloudCenter のコンポーネントとリポジトリと通信できるようにするために、ネットワーク コンプライアンス チェックを実行する必要もあります。

すべてのワーカー VM にネットワーク コンプライアンス チェックを実行するには、次の手順を実行します。

1. CloudCenter プラットフォームにより管理されることを意図された任意のクラウド環境のテスト仮想マシンを起動します
2. 「フェーズ 1：インフラストラクチャの準備」で作成したデプロイメント記述子 JSON ファイルを /tmp の場所にアップロードします。
3. /tmp の場所に network_validator.py ファイルというバリデータ スクリプトをアップロードします。
4. 次のコマンドを /tmp の場所から実行します。

```
python network_validator.py --role=AGENT --region=<unique region identifier>
--deploymentDetails=<deployment descriptor JSON file path>
```

次にこのコマンドの実行例を示します。

```
python network_validator.py
--role=AGENT --region=AmazonEast1 --deploymentDetails=deployment.json
```

5. 次の点を確認するには、このコマンドの結果を確認してください。
   - 障害は発生しませんでした。障害が発生した場合、ネットワーク設定を適切に更新して、テストを繰り返します。
   - すべての必須ネットワーク ルールが正しく設定されました。
6. すべてのコンプライアンス チェックが成功したことを確認した後で、すべてのテスト VM を終了します。

クライアント テスト

一部の CloudCenter のコンポーネント（たとえば、CCM、AMQP、GUAC）は CloudCenter UI（*CCM UI* または *UI* ともいいます）にアクセスするエンドユーザがアクセス可能であるか、CloudCenter REST API を呼び出す必要があります。このようなコンポーネントへのクライアント アクセスを確認するには、次の手順を実行します。

1. エンドユーザが CloudCenter プラットフォームを管理する元のネットワークで実行している Linux、Windows（Python 対応）または MacOSX システムを特定します。
2. このクライアント マシンに導入の記述子 JSON ファイルをアップロードします。
3. このクライアント マシンに network_validator.py ファイルをアップロードします。
4. Python スクリプトの場所から次のコマンドを実行します。

```
python network_validator.py --role=CLIENT --deploymentDetails=deployment.json
```

5. 次の点を確認するには、このコマンドの結果を確認してください。
   - 障害は発生しませんでした。障害が発生した場合、ネットワーク設定を適切に更新して、テストを繰り返します。
   - すべての必須ネットワーク ルールが正しく設定されました。

重要:
「*フェーズ 4:CloudCenter のコンポーネントのインストールと設定*」に進む前に、ネットワーク コンプライアンス チェックがすべて正常であることを確認します。

フェーズ 4:コンポーネントのインストールと設定

概要

仮想マシンでインストーラを実行して、CloudCenter コンポーネントをインストールし、コンフィギュレーション ウィザードを起動して、コンポーネントのプロパティを設定します。

パッケージ ストアのインストール(条件付き)

概要

パッケージ ストアは、すべてのアプリケーション サービス(設定済みサービス)のバイナリと、CloudCenter 自体のインストールに必要な複数のコンポーネントのバイナリを含むリポジトリです。

- **インターネット接続使用**:デフォルトのパッケージ ストアは repo.cliqrtech.com にホストされ、アプリケーション VM がインターネットにアクセスできる CloudCenter の導入では、デフォルトのパッケージ ストアを使用できます。
- **分離環境**:インターネットへの接続が制限された環境では、ローカル パッケージ ストアを作成し、CCO に登録します。

追加情報については、「Package Store」を参照してください。

デフォルト パッケージ ストア
インターネットにアクセスできる CloudCenter 環境ではローカルのパッケージ ストアを設定する必要がなく、repo.cliqrtech.com にホストされるデフォルトのパッケージ ストアを使用します。

ローカルのパッケージ ストアの設定

ローカルのパッケージ ストアを設定するには、次の手順に従います。

インターネット アクセス
パッケージ ストアでは、インターネット アクセスが定期的にデフォルトのパッケージ ストア(repo.cliqrtech.com)から OS パッケージを同期できる必要があります。

1. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
2. ログインして ROOT ユーザになります。

```
sudo -i
```

3. インストーラをダウンロードする場所に移動します。
4. この VM に次のファイルをダウンロードするか、またはコピーします。

```
repo_installer.bin
```

5. システム管理者がプリインストールされたパッケージ ストア リポジトリをイメージに提供した場合、マスター リポジトリの場所としてこのリポジトリを設定するために使用する変数をエクスポートします。デフォルトでは、CloudCenter プラットフォームは repo.cliqrtech.com を使用します。

```
export MASTER_REPO=<package_store_location>
```

6. 次のコマンドを使用してリポジトリ（repo）インストーラを実行します。

```
chmod 755 repo_installer.bin
./repo_installer.bin <ostype> <cloudtype> repo
```

次に例を示します。

```
./repo_installer.bin centos7 amazon repo
```

- <ostype>= centos6、centos7、rhel6、rhel7、ubuntu1404
- <cloudtype>= amazon、azure、azurerm、azurepack、bracket、cloudn、cloudstack、google、opsource、openstack、rackspace、softlayer、vmware、vcd

7. 前の手順からの出力の SSH キーをコピーし、CloudCenter サポートへ送信して、その SSH キーをマスター リポジトリへ追加できるようにします。
8. リポジトリ設定ウィザードを起動し、基本プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを太字でハイライトしています。ウィザードを開始し、次の表 Repo に特定した値を入力します。

設定ウィザードのパス

```
/usr/bin/repo_config_wizard.sh
```


今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

ローカルのパッケージ ストアのプロパティ	フィールド	説明
Change Master	リポジトリの URL	同期するこのリポジトリのマスター リポジトリを設定します。デフォルトのマスターは repo.cliqrtech.com です。
Set_Master		マスターのリポジトリをローカルのリポジトリに変更します。ローカルのリポジトリは他のマスター リポジトリと同期されなくなります。この場合、リポジトリをメンテナンスする必要があります（パッケージの更新、イメージ XML の更新など）。
Repo_Syncup		マスター リポジトリと同期し、マスター リポジトリからのすべての変更（ある場合）を更新します。
Add_ssh_key		このリポジトリをマスター リポジトリとして使用する場合は、このオプションを使用して、このリポジトリを指すローカルリポジトリに SSH キーを追加します。

9. リポジトリ ウィザードを終了します。

パッケージ ストアをローカルに設定できました。

バンドル ストアのインストール（オプション）

- 概要
- バンドル ストアの設定

概要

バンドル ストアは、エージェントやサービス バンドルを含むリポジトリです。アプリケーション オーケストレーション プロセスの一部として CloudCenter で開始する VM（ワーカーまたはアプリケーション VM）でこれらのバンドルをダウンロードしてインストールする必要があります。

- **インターネット接続使用**：デフォルトのバンドル ストアは（cdn.cliqr.com）にホストされ、アプリケーション VM がインターネットにアクセスできる CloudCenter の導入では、デフォルトのバンドル ストアを使用できます。
- **分離環境**：インターネットへの接続が制限された環境では、ローカル バンドル ストアを作成し、CCO に登録します。

追加情報については、「Bundle Store」を参照してください。

バンドル ストアの設定

バンドル ストアを設定するには、次の手順に従います。

> **インターネット アクセス**
> インストールにインターネット接続がない場合、すべてのワーカー VM がアクセス可能な HTTP サーバを設定または特定する必要があります。

1. HTTP サーバをセットアップします。

   > この設定は CentOS サーバの Apache2 を前提としています。別の OS/HTTP サーバを使用している場合は、それに応じて次のコマンドを調整します。

2. HTTP サーバのドキュメントのルートを特定します
   a. ディレクトリを ***/etc/httpd/conf*** に変更します
   b. httpd.conf で site-available/default ファイルを確認します。
   c. これらの設定ファイルのいずれかで ***DocumentRoot*** を検索します。通常、これは /var/www または /var/www/html です。
3. ***DocumentRoot*** ディレクトリに移動します。

   ```
   cd <DocumentRoot>
   ```

4. インストールしている CloudCenter リリース（4.6.0 など）を反映するようにディレクトリを作成し、リリース フォルダ レベルの下にバンドル ディレクトリを作成します。

   ```
   mkdir release-4.5.3-20160623
   cd release-4.5.3-20160623
   mkdir bundle
   ```

5. バンドル ディレクトリに移動します。

   ```
   cd bundle
   ```

6. bundle_artifacts.zip をコピーまたはダウンロードします
7. bundle_artifacts.zip file ファイルを解凍します。

   ```
   unzip bundle_artifacts.zip
   ```

8. リポジトリの場所を表すように設定ファイルを更新します。

```
chmod 755 ./set_bundle_location.sh
./set_bundle_location.sh http://your_bundle_directory_URL/bundle
```

次に例を示します。

```
./set_bundle_location.sh http://example.com/bundle/
```

⚠ コマンドの末尾に「/」を含めないと、プロセスのどこかの時点でエラーが発生します。

CCM のインストール（必須）

CCM インストーラの概要

マネージャ、または管理サーバと呼ばれることもある CloudCenter Manager（CCM）は、ユーザが導入をモデル化、移行および管理するためのダッシュボードとして機能し、いずれかの導入モデルを使用しているクラウドとユーザの統一された管理およびガバナンスのための、集中管理層です。

詳細については、「CCM」を参照してください。

CCM 非 HA

1. CCM：コアおよび CCM インストーラの実行

CCM：コアおよび CCM インストーラの実行

コアおよび CCM インストーラを実行するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに CCM インストーラ ファイルをダウンロードします
   - core_installer.bin
   - ccm-installer.jar
   - ccm-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup
export CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> ccm
```

次に例を示します。

```
./core_installer.bin centos7 amazon ccm
```

構文：

<ostype>= centos6、centos7、rhel6、rhel7

<cloudtype>= amazon、openstack、vmware

d. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

e. /tmp ディレクトリに移動します。

```
cd /tmp
```

f. アプライアンスのインストーラを実行して CCM を設定します。

```
java -jar ccm-installer.jar ccm-response.xml
```

g. CCM VM をリブートします。

2. CCM：CCM プロパティを設定します

CCM ウィザードのプロパティ

CCM ウィザードのプロパティを設定するには、次の手順に従います。

a. CCM ウィザードを呼び出し、基本プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. ウィザードを開始し、次の表に特定した値を入力します。

CCM ウィザードのパス

```
/usr/local/cliqr/bin/ccm_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

CCM プロパティ	フィールド	説明
Mail	• **SMTP Host** • **SMTP Port** • **SMTP Auth**	メール通知を送信する SMTP サーバの詳細。
Mail_User	• **メール ユーザ** • **[Password]** • **ユーザから** • **表示名**	メール通知を送信するメール認証と設定の詳細

CCM プロパティ	フィールド	説明
Server_info	• **パブリック DNS**	• CCM の DNS または IP。 • CCM VM との通信に CCO VM で使用されます。
	• モニタ URL	• モニタ VM の完全な URL。たとえば、https://<MON または MON_LB IP アドレス>:8443 です。 • HTTPS プロトコルである必要があります。 • モニタ VM からのヘルス ステータスの取得に CCM VM で使用されます。
	• Hazelcast IP	• CCM VM のプライベート IP アドレス。 • CloudCenter プラットフォームによって内部で使用されます
	• **外部 URL**	• 非 HA CCM シナリオのオプション
Config_App_Logo	N/A	アプリケーション プロファイル テンプレートで使用されます。
ESB_Info	N/A	Enterprise Service Bus（ESB）、CloudCenter アプライアンスにデフォルトではインストールされないオプション コンポーネントをインストールした場合にのみ必要。
Network	• Hostname • インターフェイス	これらの設定を変更しなかった場合はデフォルトを使用します。

c. CCM 設定ウィザードを終了します。

CCM 非 HA スタンドアロン

CCM 非 HA スタンドアロン

1. MGMTPOSTGRES：コア インストーラの実行

MGMTPOSTGRES コア インストーラを実行するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに CCM インストーラ ファイルをダウンロードします
   - core_installer.bin
   - ccm-installer.jar
   - ccm-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup
export CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> mgmtpostgres
```

次に例を示します。

```
./core_installer.bin centos7 amazon mgmtpostgres
```

構文：

<ostype>= centos6、centos7、rhel6、rhel7

<cloudtype>= amazon、openstack、vmware

2. MGMTPOSTGRES：CCM_SA からの DB アクセスを設定します

MGMTPOSTGRES に対する CCM_SA からのデータベース アクセスを設定するには、次の手順に従います。

a. CCM ウィザードを呼び出し、CCM_SA へのデータベース アクセスを許可する CCM IP を設定します。

CCM ウィザードのパス

```
/usr/local/cliqr/bin/ccm_config_wizard.sh
```

b. PostgreSQL データベース アクセスを設定するために、CCM IP フィールドに *CCM_SA_IP* を入力します。
c. 変更を確認し、CCM 設定ウィザードを終了します。

3. CCM_SA：コアおよびアプライアンス インストーラの実行

CCM_SA のコアおよびアプライアンス インストーラを実行するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに CCM インストーラ ファイルをダウンロードします
   - core_installer.bin
   - ccm-installer.jar
   - ccm-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup
export CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> ccm_sa
```

次に例を示します。

```
./core_installer.bin centos7 amazon ccm_sa
```

構文：

<ostype> = centos6、centos7、rhel6、rhel7

<cloudtype> = amazon、openstack、vmware

d. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

e. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

f. 次の手順で、/tmp ディレクトリに移動し、ccm-response.xml ファイルを変更します。

応答ファイルのエントリ	値	注
<entry key="db_host" value="localhost"/>	localhost を PostgreSQL データベースの IP または DNS に置き換えます。(***MGMTPOSTGRES_IP***)	リモート データベースに接続するために、スタンドアロン CCM(CCM_SA)に必要

g. アプライアンスのインストーラを実行して CCM を設定します。

```
java -jar ccm-installer.jar ccm-response.xml
```

h. CCM VM をリブートします。

4. CCM_SA:CCM プロパティを設定します

CCM ウィザードのプロパティ

CCM ウィザードのプロパティを設定するには、次の手順に従います。

a. CCM ウィザードを呼び出し、基本プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. ウィザードを開始し、次の表に特定した値を入力します。

CCM ウィザードのパス

```
/usr/local/cliqr/bin/ccm_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

CCM プロパティ	フィールド	説明
Mail	• **SMTP Host** • **SMTP Port** • **SMTP Auth**	メール通知を送信する SMTP サーバの詳細。
Mail_User	• **メール ユーザ** • **[Password]** • **ユーザから** • **表示名**	メール通知を送信するメール認証と設定の詳細
Server_info	• **パブリック DNS**	• CCM の DNS または IP。 • CCM VM との通信に CCO VM で使用されます。
	• モニタ URL	• モニタ VM の完全な URL。たとえば、https://<MON または MON_LB IP アドレス>:8443 です。 • HTTPS プロトコルである必要があります。 • モニタ VM からのヘルス ステータスの取得に CCM VM で使用されます。
	• Hazelcast IP	• CCM VM のプライベート IP アドレス。 • CloudCenter プラットフォームによって内部で使用されます
	• **外部 URL**	• 非 HA CCM シナリオのオプション
Config_App_Logo	N/A	アプリケーション プロファイル テンプレートで使用されます。
ESB_Info	N/A	Enterprise Service Bus(ESB)、CloudCenter アプライアンスにデフォルトではインストールされないオプション コンポーネントをインストールした場合にのみ必要。
Network	• Hostname • インターフェイス	これらの設定を変更しなかった場合はデフォルトを使用します。

c. CCM 設定ウィザードを終了します。

CCM HA

CCM HA

1. MGMTPOSTGRES_MASTER と SLAVE：コア インストーラの実行と、SSH 通信の設定

MGMTPOSTGRES_MASTER と SLAVE：コア インストーラの実行と、SSH 通信の設定

コア インストーラを実行し、MGMTPOSTGRES_MASTER と SLAVE の SSH 通信を設定するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに CCM インストーラ ファイルをダウンロードします
   - core_installer.bin
   - ccm-installer.jar
   - ccm-response.xml

c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup
export CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> mgmtpostgres
```

次に例を示します。

```
./core_installer.bin centos7 amazon mgmtpostgres
```

構文：

<ostype> = centos6、centos7、rhel6、rhel7

<cloudtype> = amazon、openstack、vmware

d. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

e. MGMTPOSTGRES_SLAVE でステップ 1 ～ 4 を実行します。
f. MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE 間で SSH キーを交換します

i. MGMTPOSTGRES_MASTER で、新しい SSH キーを生成するには、次のコマンドを実行します。

CloudCenter リージョンごとのインストール（必須）

各リージョン向け

各クラウド リージョン（パブリック クラウド、プライベート クラウド、データセンター）に対し CloudCenter 導入により管理される次のコンポーネントを必ず設定します。

専用外部スクリプト Executor（条件付き）:

- CloudCenter プラットフォームは、アプリケーションのライフサイクルの各段階において実行されるカスタム スクリプトを使用する機能を提供します。これらのカスタム スクリプトは Docker ベース スクリプト Executor 内で実行し、各 CCO で使用されている OS によって異なります。
- RHEL 7.x および CentOS 7.x：CloudCenter で使用される OS のデフォルトで CCO に組み込まれています。
- RHEL 6.x または CentOS 6.x：専用 Executor スクリプトを設定する必要があります。これはこれらのバージョンを使用した OS 向けのスタンドアロン Docker コンテナです。

これは、スタンドアロンの Docker ベースの外部スクリプト Executor をインストールする条件付きのステップです。

この手順は、CCO VM に CentOS7 または RHEL7 イメージを使用している場合は*オプション*です。

この手順は、CCO VM に CentOS6 か RHEL6 イメージを使用している場合は**必須**です。

専用外部スクリプト Executor のインストール

EXT_SCRIPT_EXECUTOR：コア インストーラの実行

1. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
2. /tmp フォルダに Docker インストーラ ファイル core_installer.bin をダウンロードします。
3. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup export
CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> docker
```

次に例を示します。

```
./core_installer.bin centos7 amazon docker
```

構文：

- *<ostype>= centos7、rhel7*
- *<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、vcd*

4. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

5. Docker VM をリブートします。

専用 Guacamole サーバ（オプション）

これは、エンドユーザに AMQP サーバを提示したくない場合に、導入を選択できるオプション コンポーネントです。デフォルトでは、Guacamole サーバは AMQP と同じ VM にインストールされます。

- CloudCenter プラットフォームは、Guacamole サーバを使用して、アプリケーション ライフサイクル プロセス中に開始された仮想マシン（VM）への Web ベース SSH/VNC/RDP を可能にします。Guacamole コンポーネントは、AMQP サーバにデフォルトで組み込まれています。
- AMQP の IP アドレスにエンドユーザによりルーティング可能にさせないように、クラウド リージョンの専用（スタンドアロン）Guacamole サーバをセットアップすることを選択できます。

1. GUAC：コア インストーラの実行

GUAC：コア インストーラの実行

a. *VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。*
b. */tmp フォルダに Docker インストーラ ファイルをダウンロードします。*
   - *core_installer.bin*
   - *cco-installer.jar*
   - *conn_broker-response.xml*
c. *次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。*

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup export
CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> rabbit
```

次に例を示します。

```
./core_installer.bin centos7 amazon rabbit
```

構文：
   - *<ostype>= centos7、rhel7*
   - *<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、vcd*

d. *core_installer.bin ファイルを削除します。*

```
rm core_installer.bin
```

e. *ログオフし、ルート ユーザとして再度ログインして、JAVA home を設定します。*

```
exit
sudo -i
```

f. */tmp ディレクトリに移動します。*

```
cd /tmp
```

g. *アプライアンスのインストーラを実行して GUAC を設定します。*

```
java -jar cco-installer.jar conn_broker-response.xml
```

h. *GUAC VM をリブートします。*

2. GUAC：Guacamole の CCM/CCO プロパティの設定

GUAC：Guacamole の CCM/CCO プロパティの設定

a. *GUA ウィザードを起動して、CCO と CCM プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。*
b. *GUA ウィザードを起動します。*

GUA ウィザードのパス
/usr/local/cliqr/bin/gua_config_wizard.sh

> ⚠ ***今後の参照のためにこれを書き留めてください。***
> *後で使用できるように、「**メモ**」セクションを印刷したものにフィールドの詳細を書き留めてください。*

c. *CCM および CCO VM のプロパティの設定*

グループ	*ホスト*	*使用可能な IP アドレス*
CCM_Info	*CCM ホスト*	***CCM_IP** または **CCM_SA_IP** または **CCM_LB_IP***
CCO_Info	*CCO ホスト*	***CCO_IP** または **CCO_LB_IP***

d. *変更を確認し、GUA 設定ウィザードを終了します。*

AMQP（必須）

AMQP 非 HA

1. AMQP：コアおよびアプライアンス インストーラの実行

AMQP：コアおよびアプライアンス インストーラの実行

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに AMQP インストーラ ファイルをダウンロードします
   - core_installer.bin
   - cco-installer.jar
   - conn_broker-response.xml

c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup export
CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> rabbit
```

次に例を示します。

```
./core_installer.bin centos7 amazon rabbit
```

構文:

<ostype>= centos6、*centos7*、*rhel6*、*rhel7*

<cloudtype>= amazon、*azure*、*azurerm*、*azurepack*、*google*、*opsource*、*openstack*、*softlayer*、*vmware*、*vcd*

d. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

e. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

f. /tmp ディレクトリに移動します。

```
cd /tmp
```

g. アプライアンスのインストーラを実行して AMQP を設定します。

```
java -jar cco-installer.jar conn_broker-response.xml
```

h. AMQP VM をリブートします。

2. AMQP：Guacamole サーバの CCM/CCO プロパティの設定

AMQP：Guacamole サーバの CCM/CCO プロパティの設定

専用 GUAC 設定の場合
この GUA の設定ウィザードの指示は、専用 Guacamole サーバを設定している場合は必要ではありません。

a. GUA ウィザードを起動して、CCO と CCM プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。

GUA ウィザードを起動します。

GUA ウィザードのパス

```
/usr/local/cliqr/bin/gua_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

b. CCM および CCO VM のプロパティの設定

グループ	ホスト	使用可能な IP アドレス
CCM_Info	CCM ホスト	***CCM_IP*** または ***CCM_SA_IP*** または ***CCM_LB_IP***
CCO_Info	CCO ホスト	***CCO_IP*** または ***CCO_LB_IP***

c. 変更を確認し、GUA 設定ウィザードを終了します。

AMQP HA

AMQP HA

1. AMQP_PRIMARY：コアおよびアプライアンス インストーラの実行

AMQP_PRIMARY：コアおよびアプライアンス インストーラの実行

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに AMQP インストーラ ファイルをダウンロードします
   - core_installer.bin
   - cco-installer.jar
   - conn_broker-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup export
CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> rabbit
```

次に例を示します。

```
./core_installer.bin centos7 amazon rabbit
```

構文：

<ostype>= centos6、centos7、rhel6、rhel7

<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、および vcd

d. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

e. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

f. /tmp ディレクトリに移動します。

```
cd /tmp
```

g. アプライアンスのインストーラを実行して AMQP を設定します。

```
java -jar cco-installer.jar conn_broker-response.xml
```

h. AMQP VM をリブートします。

2. AMQP_PRIMARY：Guacamole サーバの CCM/CCO プロパティの設定

AMQP_PRIMARY：Guacamole サーバの CCM/CCO プロパティの設定

専用 GUAC 設定の場合
この GUA の設定ウィザードの指示は、専用 Guacamole サーバを設定している場合は必要ではありません。

a. GUA ウィザードを起動して、CCO と CCM プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. GUA ウィザードを起動します。

GUA ウィザードのパス
/usr/local/cliqr/bin/gua_config_wizard.sh

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

c. CCM および CCO VM のプロパティの設定

グループ	ホスト	使用可能な IP アドレス
CCM_Info	CCM ホスト	***CCM_IP*** または ***CCM_SA_IP*** または ***CCM_LB_IP***
CCO_Info	CCO ホスト	***CCO_IP*** または ***CCO_LB_IP***

d. 変更を確認し、GUA 設定ウィザードを終了します。

3. AMQP_SECONDARY：コアおよびアプライアンス インストーラの実行

AMQP_SECONDARY：コアおよびアプライアンス インストーラの実行

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに AMQP インストーラ ファイルをダウンロードします
   - core_installer.bin
   - cco-installer.jar
   - conn_broker-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup export
CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> rabbit
```

次に例を示します。

```
./core_installer.bin centos7 amazon rabbit
```

構文：

<ostype>= centos6、centos7、rhel6、rhel7

<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、および vcd

d. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

e. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

f. /tmp ディレクトリに移動します。

```
cd /tmp
```

g. アプライアンスのインストーラを実行して AMQP を設定します。

```
java -jar cco-installer.jar conn_broker-response.xml
```

h. AMQP VM をリブートします。

4. AMQP_SECONDARY：Guacamole サーバの CCM/CCO プロパティの設定

AMQP_SECONDARY：Guacamole サーバの CCM/CCO プロパティの設定

専用 GUAC 設定の場合
この GUA の設定ウィザードの指示は、専用 Guacamole サーバを設定している場合は必要ではありません。

a. GUA ウィザードを起動して、CCO と CCM プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。

GUA ウィザードを起動します。

GUA ウィザードのパス

```
/usr/local/cliqr/bin/gua_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

b. CCM および CCO VM のプロパティの設定

グループ	ホスト	使用可能な IP アドレス
CCM_Info	CCM ホスト	***CCM_IP*** または ***CCM_SA_IP*** または ***CCM_LB_IP***
CCO_Info	CCO ホスト	***CCO_IP*** または ***CCO_LB_IP***

c. 変更を確認し、GUA 設定ウィザードを終了します。
d. 設定ウィザードを終了します。

5. AMQP_PRIMARY と AMQP_SECONDARY：クラスタ構成

AMQP_PRIMARY と AMQP_SECONDARY：クラスタ構成

a. 各 VM を起動するために使用するキー ペアを使用して両方の VM インスタンス（AMQP_PRIMARY および AMQP_SECONDARY）に SSH 接続します。
b. .erlang.cookie（/var/lib/rabbitmq にある隠しファイル）は、両方の AMQP サーバで同じであることを確認します。

i. 両者が同じであれば、.erlang.cookie を AMQP のプライマリ サーバから AMQP のセカンダリ サーバの /var/lib/rabbitmq の場所にコピーします（またはその逆）。
ii. プライマリからセカンダリ AMQP サーバへ .erlang.cookie ファイルをコピーした（またはその逆）後で、その .erlang.ccokie ファイルの所有者と権限を変更します。コピーされたサーバ上のファイルの所有者と権限を変更するには、次のコマンドを実行します。

```
sudo -i
cd /var/lib/rabbitmq
chown rabbitmq:rabbitmq .erlang.cookie
chmod 400 .erlang.cookie
```

iii. .erlang cookie ファイルをコピーした後、AMQP プライマリ および AMQP セカンダリ サーバ両方で次のコマンドを実行します。

```
sudo service rabbitmq-server stop
systemctl enable rabbitmq-server.service
sudo service rabbitmq-server start
```

c. 次のコマンドを使用して AMQP サーバをデタッチします。

```
rabbitmq-server -detached
```

d. AMQP_SECONDARY サーバで次のコマンドを実行します。

```
rabbitmqctl stop_app
rabbitmqctl join_cluster rabbit@<AMQP_PRIMARY_HOSTNAME>
rabbitmqctl start_app
```

e. 両方の AMQP サーバの AMQP クラスタの状態を確認します。

```
rabbitmqctl cluster_status
```

f. AMQP サーバのいずれかで次のコマンドを発行して、ミラーリング ポリシーを設定します。

```
rabbitmqctl set_policy ha-all "^cliqr" '{"ha-mode":"all"}' -p /cliqr
```

g. 次のコマンドを実行して、ポリシーの変更を確認します。

```
rabbitmqctl list_policies -p /cliqr
```

h. rabbit アレイに行を追加することで、両方の AMQP サーバの rabbitmq 設定ファイル（/etc/rabbitmq にある）を変更します（autoheal 用）。

```
{
    cluster_partition_handling,
    autoheal
}
#Example:
    [{
    rabbit,
    [{
        cluster_partition_handling,
        autoheal
    }, {
        ssl_listeners,
        [5671]
    }, {
        handshake_timeout,
        300000
    }, {
        ssl_handshake_timeout,
        300000
    }, {
        ssl_options,
        [{
            cacertfile,
            "/etc/rabbitmq/certs/cacert.pem"
        }, {
            certfile,
            "/etc/rabbitmq/certs/cert.pem"
        }, {
            keyfile,
            "/etc/rabbitmq/certs/key.pem"
        }, {
            verify,
            verify_peer
        }, {
            fail_if_no_peer_cert,
            false
        }]
    }]
}]
```

i. 両方の AMQP サーバで rabbitmq を再起動します。

```
/etc/init.d/rabbitmq-server restart
```

6. AMQP_LB

AMQP_LB

AMQP のロード バランシングは、HAProxy、NGiNX、Apache2、または AWS Elastic ロード バランサ（ELB）などのサービスをネイティブで使用できるクラウドで実行できます。ロード バランサ サービスを設定し、AMQP ロード バランシングを確認するには、ポート 5671 でリスニングし、AMQP_PRIMARY および AMQP_SECONDARY サーバの両方の 443 で要求を調整してください。

次のロード バランシング設定が AMQP VM の HAProxy と CentOS7.x VM で行われました。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。

b. ルート ユーザとして HAProxy をインストールします。

```
yum -y install haproxy
```

c. 次のように、HAProxy の設定ファイルを変更します

```
vi /etc/haproxy/haproxy.cfg

# configuration to listen on 5671 and loadbalance

frontend amqps-in
 mode tcp
 log global
 bind *:5671
 default_backend amqps

backend amqps
    mode tcp
    balance roundrobin
    option ssl-hello-chk
    server amqp1 <AMQP_PRIMARY>:5671 check
    server amqp2 <AMQP_SECONDARY>:5671 check
```

d. 5671 ポートにバインドするには、SELinux を無効にする必要があります（SELinux を無効にするには次のコマンドを実行します）。

```
setenforce 0
```

e. HAProxy サービスを開始し、そのステータスを確認します。これはアクティブである必要があります

```
systemctl start haproxy
systemctl status haproxy
sudo -i
ssh-keygen -t rsa
cd ~/.ssh
cat id_rsa.pub >> authorized_keys
```

ii. MGMTPOSTGRES_MASTER から MGMTPOSTGRES_SLAVE の同じ場所に id_rsa ファイル（~/.ssh/id_rsa と ~/.ssh/id_rsa.pub）をコピーします。MGMTPOSTGRES_SLAVE に .ssh ディレクトリがない場合は、ファイルをコピーする前に、次のコマンドを使用して作成します。

```
sudo -i
mkdir -p ~/.ssh
chmod 700 ~/.ssh
```

iii. MGMTPOSTGRES_SLAVE で、次のコマンドを実行し、新しい SSH キーを生成します。

```
sudo -i
chmod 400 ~/.ssh/id_rsa*
cat id_rsa.pub >> authorized_keys
```

iv. 各 VM 上で次のコマンドを実行して、MGMTPOSTGRES_MASTER と MGMTPOSTGRES_SLAVE 間の相互 SSH アクセスを確認します。

```
sudo -i
ssh root@<MGMTPOSTGRES_MASTER/MGMTPOSTGRES_SLAVE>
```

2. MGMTPOSTGRES_MASTER：高可用性の設定

MGMTPOSTGRES_MASTER：高可用性の設定

MGMTPOSTGRES_MASTER の高可用性を設定するには、次の手順に従います。

a. DB 設定ウィザードを呼び出し、データベースの HA を設定します。

CCM ウィザードのパス

```
/usr/local/cliqr/bin/gua_config_wizard.sh
```

b. PostgreSQL データベースの HA を確保するように Postgres HA を設定し、次のように各フィールドに情報を入力します。

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

データベースのプロパティ	フィールド	説明
Configure_Postgres_HA	• マスター ホスト名	マスター データベース VM のホスト名
	• マスター プライベート IP	マスター データベース VM のプライベート IP アドレス
	• スレーブ ホスト名	スレーブ データベース VM のホスト名
	• スレーブ プライベート IP	スレーブ データベース VM のプライベート IP アドレス
	• VIP **または** • EIP	データベースの VIP/EIP IP このオプションを選択するには、マウスを使用します。

詳細を入力すると、データベース サーバは、HA 設定に続いてデータベース サーバ間のレプリケーション設定を開始し、最終的に次のステータス メッセージが表示されます。

- Configuring database for HA ...
- Configuring database for replication

c. CCM 設定ウィザードを終了します。

3. CCM_SA_PRIMARY と CCM_SA_SECONDARY：コア インストーラの実行と SSH 通信の設定

CCM_SA_PRIMARY と SECONDARY：コア インストーラの実行と SSH 通信の設定

コア インストーラを実行し、CCM_SA_PRIMARY と CCM_SA_SECONDARY の SSH 通信を設定するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに CCM インストーラ ファイルをダウンロードします
   - core_installer.bin
   - ccm-installer.jar
   - ccm-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup
export CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> ccm_sa
```

次に例を示します。

```
./core_installer.bin centos7 amazon ccm_sa
```

構文：

<ostype> = centos6、centos7、rhel6、rhel7

<cloudtype> = amazon、openstack、vmware

d. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

e. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

f. CCM_SA_PRIMARY と CCM_SA_SECONDARY 間で SSH キーを交換します
   i. CCM_SA_PRIMARY で、新しい SSH キーを生成するには、次のコマンドを実行します。

```
su - cliqruser
ssh-keygen -t rsa
cd ~/.ssh
cat id_rsa.pub >> authorized_keys
```

   ii. CCM_SA_PRIMARY から CCM_SA_SECONDARY の同じ場所に id_rsa ファイル（~/.ssh/id_rsa と ~/.ssh/id_rsa.pub）をコピーします。CCM_SA_SECONDARY に .ssh ディレクトリがない場合は、ファイルをコピーする前に、次のコマンドを使用して作成します。

```
su - cliqruser
mkdir -p ~/.ssh
chmod 700 ~/.ssh
```

iii. CCM_SA_SECONDARY で、次のコマンドを実行し、新しい SSH キーを生成します。

```
su - cliqruser
chmod 400 ~/.ssh/id_rsa*
cat id_rsa.pub >> authorized_keys
chmod 600 authorized_keys
```

iv. 各 VM 上で次のコマンドを実行して、CCM_SA_PRIMARY と CCM_SA_SECONDARY 間の相互 SSH アクセスを確認します。

```
su - cliqruser
ssh cliqruser@<CCM_SA_PRIMARY/CCM_SA_SECONDARY>
```

4. CCM_SA_PRIMARY：アプライアンス インストールの実行

CCM_SA_PRIMARY：アプライアンス インストールの実行

CCM_SA_PRIMARY に対しアプライアンス インストールを実行するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. 次の手順で、/tmp ディレクトリに移動し、ccm-response.xml ファイルを変更します。

応答ファイルのエントリ	値	注
<entry key="db_host" value="default"/>	db_host の IP アドレスは、CCM のデータベースの設定により異なります。 • スタンドアロン DB：インスタンスの IP アドレスを使用します • HA DB：VIP アドレスを使用します	必須

c. アプライアンスのインストーラを実行して CCM を設定します。

```
java -jar ccm-installer.jar ccm-response.xml
```

d. CCM VM をリブートします。

5. CCM_SA_SECONDARY：アプライアンス インストールの実行

CCM_SA_SECONDARY：アプライアンス インストールの実行

CCM_SA_SECONDARY に対しアプライアンス インストールを実行するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. 次の手順で、/tmp ディレクトリに移動し、ccm-response.xml ファイルを変更します。

応答ファイルのエントリ	値	注
<entry key="db_enable" value="true "/>	値を [false] に変更します。	HA セットアップでは、セカンダリ インスタンスのアプライアンス インストーラはデータベースを更新する必要がありません。
<entry key="db_host" value="default"/>	db_host の IP アドレスは、CCM のデータベースの設定により異なります。 i. スタンドアロン DB：インスタンスの IP アドレスを使用します。 ii. HA DB：VIP アドレスを使用します	必須です。

c. アプライアンスのインストーラを実行して CCM を設定します。

```
java -jar ccm-installer.jar ccm-response.xml
```

d. CCM VM をリブートします。

6. CCM_SA_PRIMARY：高可用性の設定

CCM_SA_PRIMARY：高可用性の設定

CCM_SA_PRIMARY の高可用性を設定するには、次の手順に従います。

a. CCM ウィザードを root ユーザとして起動し、基本的なプロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. ウィザードを開始し、次の表に特定した値を入力します。

CCM ウィザードのパス

```
/usr/local/cliqr/bin/ccm_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

CCM プロパティ	フィールド	説明
Configure_HA	• Primary Node Private IP	プライマリ CCM VM の IP アドレス
	• Secondary Node Private IP	セカンダリ CCM VM の IP アドレス
	• パブリック DNS	• CCM_LB の DNS または IP を使用します • CCM VM との通信に CCO VM で使用されます。

	• Hazelcast IP	• Hazelcast IP はカンマで区切られた、プライマリおよびセカンダリ CCM VM のパブリックまたはプライベート IP アドレスである必要があります。
		• 2 つの CCM VM が：
		  • 同じネットワーク上に設定されている場合、プライベート IP を使用します。
		  • 異なるネットワーク上に設定されている場合、パブリック IP を使用します。

	• 外部 URL	• 非 HA CCM のシナリオのオプション • CCM_LB_IP を使用します

c. 詳細を入力すると、データベース サーバは、HA 設定に続いてデータベース サーバ間のレプリケーション設定を開始し、最終的に次のステータス メッセージが表示されます。
- Configuring CCM HA ...
- Restart server（経過表示バー）
- Configured CCM HA successfully

> ⚠ CCM プライマリ サーバに HA を設定した後、次のメッセージが表示されます。
> *Restarting secondary ccm tomcat failed due to cliqruser may not have sudo permission.*
>
> **この時点で、セカンダリ CCM サーバの tomcat を手動で再起動する必要があります。**

d. CCM セカンダリ サーバで次のコマンドを実行して、Tomcat サーバを再起動します。

```
sudo -i
/etc/init.d/tomcat restart
```

e. CCM 設定ウィザードを終了します。

7. CCM_SA_PRIMARY と CCM_SA_SECONDARY：基本プロパティの設定

CCM ウィザードのプロパティ

CCM ウィザードのプロパティを設定するには、次の手順に従います。

a. CCM ウィザードを呼び出し、基本プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. ウィザードを開始し、次の表に特定した値を入力します。

CCM ウィザードのパス

```
/usr/local/cliqr/bin/ccm_config_wizard.sh
```

> ⚠ **今後の参照のためにこれを書き留めてください。**
> 後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

CCM プロパティ	フィールド	説明
Mail	• **SMTP Host** • **SMTP Port** • **SMTP Auth**	メール通知を送信する SMTP サーバの詳細。
Mail_User	• **メール ユーザ** • **[Password]** • **ユーザから** • **表示名**	メール通知を送信するメール認証と設定の詳細
Server_info	• **パブリック DNS**	• CCM の DNS または IP。 • CCM VM との通信に CCO VM で使用されます。
	• モニタ URL	• モニタ VM の完全な URL。たとえば、https://<MON または MON_LB IP アドレス>:8443 です。 • HTTPS プロトコルである必要があります。 • モニタ VM からのヘルス ステータスの取得に CCM VM で使用されます。
	• Hazelcast IP	• CCM VM のプライベート IP アドレス。 • CloudCenter プラットフォームによって内部で使用されます
	• **外部 URL**	• 非 HA CCM シナリオのオプション
Config_App_Logo	N/A	アプリケーション プロファイル テンプレートで使用されます。
ESB_Info	N/A	Enterprise Service Bus（ESB）、CloudCenter アプライアンスにデフォルトではインストールされないオプション コンポーネントをインストールした場合にのみ必要。
Network	• Hostname • インターフェイス	これらの設定を変更しなかった場合はデフォルトを使用します。

c. CCM 設定ウィザードを終了します。

8. CCM_LB

CCM_LB

CCM の HAProxy で CentOS7.x VM をロード バランシングする設定例です。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. ルート ユーザとして HAProxy をインストールします。

```
yum -y install haproxy
```

c. CCM プライマリ サーバで CCM_LB の haproxy 設定の .pem ファイルを作成します。
   i. 次のコマンドを実行します。

```
sudo -i
cd /usr/local/tomcat/conf/ssl
cat mgmtserver.crt mgmtserver.key >> mgmtserver.pem
cat ca.crt mgmtserver.key >> ca.pem
```

> ⚠ 環境に応じて mgmtserver および ca pem に名前を付けることができますが、.pem 拡張子を必ず追加してください。

   ii. 以前に CCM_LB サーバに作成した mgmtserver.pem および ca.pem ファイルを /etc/haproxy の場所に置きます。

d. 以下に示すとおり、HAProxy の設定ファイルを変更します。

```
vi /etc/haproxy/haproxy.cfg

# configuration to listen on 443 with SSL certs and loadbalance
frontend https-in
    mode http
    log global
    bind *:443 ssl crt /etc/haproxy/mgmtserver.pem ca-file
/etc/haproxy/ca.pem
    default_backend ccms

# configuration to listen on 8443 with SSL certs and loadbalance
frontend httpsalt-in
    mode tcp
    bind *:8443
    default_backend nodes

backend ccms
    balance roundrobin
    mode    http
    log global
    option httplog
    cookie SVR insert preserve nocache
    server  ccm1 < CCM_SA_PRIMARY_IP >:443 check cookie ccm1 ssl
verify none
    server  ccm2 < CCM_SA_SECONDARY_IP >:443 check cookie ccm2 ssl
verify none

backend nodes
    mode tcp
```

```
    balance roundrobin
    option ssl-hello-chk
    server  ccm1 < CCM_SA_PRIMARY_IP >:8443 check
    server  ccm2 < CCM_SA_SECONDARY_IP >:8443 check
```

e. HAProxy サービスを開始し、ステータス応答がアクティブであることを確認します。

```
systemctl start haproxy
systemctl status haproxy
```

CCO（必須）

CCO 非 HA

1. CCO：コアおよびアプライアンス インストーラの実行

CCO：コアおよびアプライアンス インストーラの実行

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに CCO インストーラ ファイルをダウンロードします
   - core_installer.bin
   - cco-installer.jar
   - cco-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup export
  CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> cco
```

次に例を示します。

```
./core_installer.bin centos7 amazon cco
```

構文：

<ostype>= centos6、centos7、rhel6、rhel7

<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、または vcd

d. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

e. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

f. /tmp ディレクトリに移動します。

```
cd /tmp
```

g. アプライアンスのインストーラを実行して CCO を設定します。

```
java -jar cco-installer.jar cco-response.xml
```

h. CCO VM をリブートします。

2. CO:CCO プロパティの設定

CCO:CCO プロパティの設定

a. CCO ウィザードを呼び出し、エージェント バンドル、AMQP サーバ、Guacamole サーバおよび Docker サーバのプロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. CCO ウィザードを起動します。

GUA ウィザードのパス

```
/usr/local/cliqr/bin/cco_config_wizard.sh
```

> **今後の参照のためにこれを書き留めてください。**
> 後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

c. エージェント バンドル、AMQP サーバ、Guacamole サーバおよび Docker VM のプロパティを設定します。

グループ	プロパティ	注
AgentBundle		デフォルトを使用します。 • カスタム バンドルを使用する場合、cdn.cliqr.com をカスタム バンドル ストアの IP または DNS に置き換えます • パッケージ ストアを使用する場合、repo.cliqrtech.com をカスタム パッケージ ストアの IP または DNS に置き換えます
AMQP_Server	• AMQP Server IP • AMQP Port	• ***AMQP_IP*** か ***AMQP_LB_IP*** • 5671
Network	• Hostname	CCO 環境のネットワークの詳細を設定します。これは、VM のプライベート IP を設定するオプションの手順です。VM の IP またはホスト名が事前に設定されていない場合、または既存の IP またはホスト名を上書きする必要がある場合は、一般に次の情報を設定できます。

グループ	プロパティ	注
Guacamole	• Connection Broker Host • Connection Broker Port1 • Connection Broker Port2	• ***AMQP_IP*** か ***AMQP_LB_IP*** • 7788 • 7789
Docker	• Docker Registry URL • Docker CACert URL	• カスタム Docker レジストリが使用されたときだけ設定します • Docker レジストリでカスタム CA 証明書を使用して SSL を使用しているときだけ設定します

d. 変更を確認し、CCO 設定ウィザードを終了します。

CCO HA

CCO HA

1. CCO_PRIMARY：コアおよびアプライアンス インストーラの実行

CCO_PRIMARY：コアおよびアプライアンス インストーラの実行

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに CCO インストーラ ファイルをダウンロードします
   - core_installer.bin
   - cco-installer.jar
   - cco-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup export
CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> cco
```

次に例を示します。

```
./core_installer.bin centos7 amazon cco
```

構文：

<ostype>= centos6、centos7、rhel6、rhel7

<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、または vcd

d. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

e. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

f. /tmp ディレクトリに移動します。

```
cd /tmp
```

g. アプライアンスのインストーラを実行して CCO を設定します。

```
java -jar cco-installer.jar cco-response.xml
```

h. CCO VM をリブートします。

2. CCO_PRIMARY：CCO のプロパティの設定

CCO_PRIMARY：CCO のプロパティの設定

a. CCO ウィザードを呼び出し、エージェント バンドル、AMQP サーバ、Guacamole サーバおよび Docker サーバのプロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。

b. CCO ウィザードを起動します。

CCO ウィザードのパス

```
/usr/local/cliqr/bin/cco_config_wizard.sh
```

⚠ **今後の参照のためにこれを書き留めてください。**
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

c. エージェント バンドル、AMQP サーバ、Guacamole サーバおよび Docker VM のプロパティと HA を設定します。

グループ	プロパティ	注
AgentBundle	–	デフォルトを使用します。 • カスタム バンドルを使用する場合、cdn.cliqr.com をカスタム バンドル ストアの IP または DNS に置き換えます • パッケージ ストアを使用する場合、repo.cliqrtech.com をカスタム パッケージ ストアの IP または DNS に置き換えます
AMQP_Server	• AMQP Server IP • AMQP Port	• ***AMQP_IP*** か ***AMQP_LB_IP*** • 5671
Network	• Hostname	CCO 環境のネットワークの詳細を設定します。これは、VM のプライベート IP を設定するオプションの手順です。VM の IP またはホスト名が事前に設定されていない場合、または既存の IP またはホスト名を上書きする必要がある場合は、一般に次の情報を設定できます。

グループ	プロパティ	注
Guacamole	• Connection Broker Host • Connection Broker Port1 • Connection Broker Port2	• ***AMQP_IP*** か ***AMQP_LB_IP*** • 7788 • 7789
Docker	• Docker Registry URL • Docker CACert URL	• カスタム Docker レジストリが使用されたときだけ設定します • Docker レジストリでカスタム CA 証明書を使用して SSL を使用しているときだけ設定します
Configure_HA	Hazelcast IP List	***CCO_PRIMARY_IP*** と ***CCO_SECONDARY_IP*** のカンマ区切りのリスト

d. 変更を確認し、CCO 設定ウィザードを終了します。

3. CCO_SECONDARY：コアおよびアプライアンス インストーラの実行

CCO_SECONDARY：コアおよびアプライアンスインストーラの実行

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダに CCO インストーラ ファイルをダウンロードします
   - core_installer.bin
   - cco-installer.jar
   - cco-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup export
CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> cco
```

次に例を示します。

```
./core_installer.bin centos7 amazon cco
```

構文：

<ostype>= centos6、centos7、rhel6、rhel7

<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、または vcd

d. core_installer.bin ファイルを削除します。

```
rm core_installer.bin
```

e. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

f. /tmp ディレクトリに移動します。

```
cd /tmp
```

g. アプライアンスのインストーラを実行して CCO を設定します。

```
java -jar cco-installer.jar cco-response.xml
```

h. CCO VM をリブートします。

4. CCO_SECONDARY：CCO のプロパティの設定

CCO_SECONDARY：CCO のプロパティの設定

a. CCO ウィザードを呼び出し、エージェント バンドル、AMQP サーバ、Guacamole サーバおよび Docker サーバのプロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. CCO ウィザードを起動します

CCO ウィザードのパス

```
/usr/local/cliqr/bin/cco_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

c. エージェント バンドル、AMQP サーバ、Guacamole サーバおよび Docker VM のプロパティと HA を設定します。

グループ	プロパティ	注
AgentBundle	–	デフォルトを使用します。 • カスタム バンドルを使用する場合、cdn.cliqr.com をカスタム バンドル ストアの IP または DNS に置き換えます • パッケージ ストアを使用する場合、repo.cliqrtech.com をカスタム パッケージ ストアの IP または DNS に置き換えます
AMQP_Server	• AMQP Server IP • AMQP Port	• ***AMQP_IP*** か ***AMQP_LB_IP*** • 5671
Network	• Hostname	CCO 環境のネットワークの詳細を設定します。これは、VM のプライベート IP を設定するオプションの手順です。VM の IP またはホスト名が事前に設定されていない場合、または既存の IP またはホスト名を上書きする必要がある場合は、一般に次の情報を設定できます。
Guacamole	• Connection Broker Host • Connection Broker Port1 • Connection Broker Port2	• ***AMQP_IP*** か ***AMQP_LB_IP*** • 7788 • 7789
Docker	• Docker Registry URL • Docker CACert URL	• カスタム Docker レジストリが使用されたときだけ設定します • Docker レジストリでカスタム CA 証明書を使用して SSL を使用しているときだけ設定します
Configure_HA	Hazelcast IP List	***CCO_PRIMARY_IP*** と ***CCO_SECONDARY_IP*** のカンマ区切りのリスト

d. 変更を確認し、CCO 設定ウィザードを終了します。

5. CCO_LB

CCO_LB

ロード バランシングは、HAProxy、NGiNX、Apache2、または AWS Elastic ロード バランサ（ELB）などのサービスをネイティブで使用できるクラウドで実行できます。ロード バランサ サービスを設定し、CCO ロード バランシングを確認するには、ポート 443 でリスニングし、CCO_PRIMARY および CCO_SECONDARY サーバの両方のポート 443 で要求を調整してください。

次のロード バランシング設定が CCO VM の HAProxy と CentOS7.x VM で行われました。

a. CCO VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. ルート ユーザとして HAProxy をインストールします。

```
yum -y install haproxy
```

c. 次のように、HAProxy の設定ファイルを変更します

```
vi /etc/haproxy/haproxy.cfg


# configuration to listen on 8443 with SSL certs and loadbalance
frontend httpsalt-in
    mode tcp
    log global
    bind *:8443
    default_backend ccos

backend ccos
    mode tcp
    balance roundrobin
    option ssl-hello-chk
    server  cco1 <CCO_PRIMARY_IP>:8443
    server  cco2 <CCO_SECONDARY_IP>:8443
```

d. HAProxy サービスを開始し、そのステータスを確認します。これはアクティブである必要があります

```
systemctl start haproxy
systemctl status haproxy
```

ヘルス モニタのインストール（オプション）

非 HA のモニタ

1. モニタ:コアおよび CCM インストーラの実行

モニタ:コアおよび CCM インストーラの実行

バンドル モニタを設定するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダにモニタのインストーラ ファイルをダウンロードします。
   - core_installer.bin
   - monitor-installer.jar
   - monitor-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup
export CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> monitor
```

次に例を示します。

```
./core_installer.bin centos7 amazon monitor
```

構文:

<ostype>= centos6、centos7、rhel6、rhel7

*<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、*または *vcd*

d. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

e. /tmp ディレクトリに移動します。

```
cd /tmp
```

f. アプライアンスのインストーラを実行してモニタを設定します。

```
java -jar monitor-installer.jar monitor-response.xml
```

g. モニタ VM をリブートします。

2. モニタ：CCM プロパティの設定

モニタ：CCM プロパティの設定

a. CCM ウィザードを開始し、各モニタ サーバの基本的なプロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. ウィザードを開始し、次の表に特定した値を入力します。

モニタ ウィザード パス

```
/usr/local/cliqr/bin/monitor_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

グループ	プロパティ	注
CCM_Info	• Monitor ID • CCM Hostname/URL • Monitor User	• クラウド プロバイダーで作成したインスタンスとボリュームのヘルスチェックに使用する（英数字の）固有識別子。 • **CCM_IP** または **CCM_SA_IP** または **CCM_LB_IP** • そのユーザにとって有効なクラウド リージョンのヘルス チェックが実行される CCM サーバ上のユーザ ID。すべてのアクティブ化されたクラウド リージョンでヘルス チェックを実行する場合は、これを 2（CloudCenter のルート管理者のユーザ ID）に設定します。特定のクラウド リージョンでヘルスチェックを行う場合は、それらのリージョンを使用して新しいユーザを作成してアクティブ化し、そのユーザのユーザ ID をこのプロパティの値として使用します。ユーザ ID を取得するために、v1 ユーザ API を使用できます。

c. モニタ ウィザードを終了します。

HA のモニタ

HA のモニタ

1. MON_PRIMARY：コア インストーラの実行と SSH 通信の設定

MON_PRIMARY：コア インストーラの実行と SSH 通信の設定

プライマリ ヘルス モニタを設定するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダにモニタのインストーラ ファイルをダウンロードします。
   - core_installer.bin
   - monitor-installer.jar
   - monitor-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup
export CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> monitor
```

次に例を示します。

```
./core_installer.bin centos7 amazon monitor
```

構文：

<ostype>= centos6、centos7、rhel6、rhel7

*<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、*または *vcd*

d. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

e. /tmp ディレクトリに移動します。

```
cd /tmp
```

f. アプライアンスのインストーラを実行してモニタを設定します。

```
java -jar monitor-installer.jar monitor-response.xml
```

g. プライマリ モニタ VM をリブートします。

2. MON_SECONDARY：コア インストーラの実行と SSH 通信の設定

MON_SECONDARY：コア インストーラの実行と SSH 通信の設定

プライマリ ヘルス モニタを設定するには、次の手順に従います。

a. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. /tmp フォルダにモニタのインストーラ ファイルをダウンロードします。
   - core_installer.bin
   - monitor-installer.jar
   - monitor-response.xml
c. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 core_installer.bin

#Set the following only if a local package store is setup
export CUSTOM_REPO=<http://local_package_store ip>

./core_installer.bin <ostype> <cloudtype> monitor
```

次に例を示します。

```
./core_installer.bin centos7 amazon monitor
```

構文：

<ostype>= centos6、centos7、rhel6、rhel7

*<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、*または *vcd*

d. ログオフし、ルート ユーザとして再度ログインして、JAVA Home が設定されたことを確認します。

```
exit
sudo -i
```

e. /tmp ディレクトリに移動します。

```
cd /tmp
```

f. アプライアンスのインストーラを実行してモニタを設定します。

```
java -jar monitor-installer.jar monitor-response.xml
```

g. セカンダリ モニタ VM をリブートします。

3. MON_PRIMARY および MON_SECONDARY：基本プロパティの設定

モニタ：CCM プロパティの設定

a. CCM ウィザードを開始し、各モニタ サーバの基本的なプロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。

b. ウィザードを開始し、次の表に特定した値を入力します。

モニタ ウィザード パス

```
/usr/local/cliqr/bin/monitor_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

グループ	プロパティ	注
CCM_Info	• Monitor ID • CCM Hostname/URL • Monitor User	• クラウド プロバイダーで作成したインスタンスとボリュームのヘルスチェックに使用する（英数字の）固有識別子。 • **CCM_IP** または **CCM_SA_IP** または **CCM_LB_IP** • そのユーザにとって有効なクラウド リージョンのヘルス チェックが実行される CCM サーバ上のユーザ ID。すべてのアクティブ化されたクラウド リージョンでヘルス チェックを実行する場合は、これを 2（CloudCenter のルート管理者のユーザ ID）に設定します。特定のクラウド リージョンでヘルスチェックを行う場合は、それらのリージョンを使用して新しいユーザを作成してアクティブ化し、そのユーザのユーザ ID をこのプロパティの値として使用します。ユーザ ID を取得するために、v1 ユーザ API を使用できます。

c. モニタ ウィザードを終了します。

4. MON_LB

MON_LB

ロード バランシングは、HAProxy、NGiNX、Apache2、または AWS Elastic ロード バランサ（ELB）などのサービスをネイティブで使用できるクラウドで実行できます。ロード バランサ サービスを設定し、ロード バランシングをモニタするには、ポート 443 でのリスニングおよび MON_PRIMARY と MON_SECONDARY の両サーバのポート 443 での要求のバランシングを必ず実行します。

次のロード バランシングの設定は、モニタ VM 用の HAProxy を伴う CentOS7.x VM で行われました。

a. MON VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
b. ルート ユーザとして HAProxy をインストールします。

```
yum -y install haproxy
```

c. 次のように、HAProxy の設定ファイルを変更します

```
vi /etc/haproxy/haproxy.cfg

# configuration to listen on 8443 and loadbalance

frontend httpsalt-in
    mode tcp
    log global
    bind*:8443
    default_backend mons


backend mons
    mode tcp
    balance roundrobin
    option ssl-hello-chk
    server mon1 <MON_PRIMARY_IP>:8443 check
    server mon2 <MON_SECONDARY_IP>:8443 check
```

d. HAProxy サービスを開始し、そのステータスを確認します。これはアクティブである必要があります

```
systemctl start haproxy
systemctl status haproxy
```

専用 Docker レジストリのインストール（オプション）

概要

（VM ベースのサービスを使用して）カスタム サービスを、または（コンテナ レベルで）外部サービスを定義した後は、隔離された Docker コンテナを開始することもできます。

コア インストーラおよびアプライアンス インストーラの実行

Docker レジストリを設定する手順は次のとおりです。

1. VM を起動するために使用するキー ペアを使用して VM インスタンスに SSH 接続します。
2. インストーラ ファイルを /tmp folder: repo_installer.bin にダウンロードします。
3. 次のコマンドを使用して、コア システム コンポーネントを設定するためのコア インストーラを実行します。

```
sudo -i
cd /tmp
chmod 755 repo_installer.bin

#Set the following only if a local package store is setup
export CUSTOM_REPO=<http://local_package_store ip>

./repo_installer.bin <ostype> <cloudtype> monitor
```

次に例を示します。

```
./repo_installer.bin centos7 amazon monitor
```

構文:

<ostype>= centos7、rhel7

<cloudtype>= amazon、azure、azurerm、azurepack、google、opsource、openstack、softlayer、vmware、または vcd

4. core_installer.bin ファイルを削除します。

```
rm repo_installer.bin
```

5. Docker VM をリブートします。

カスタム イメージのインストール

管理エージェントの Java バージョン互換性

- 概要
- Java のバージョン

概要

CloudCenter により管理される各 VM では、そこで動作する管理エージェント バンドル(エージェント)コンポーネントが必要です。CloudCenter プラットフォーム インストーラ ファイル内の各 VM イメージには管理エージェントが含まれます。CloudCenter プラットフォームは、エージェントがインストールされていないことを検出し、導入プロセスの一環として、各 VM にエージェントを自動的にプッシュします。

CCO は、管理エージェントからモニタ情報を受信すると、エージェントに次のタスクを指示します。

- アプリケーション導入タスクを完了する
- プロビジョニング タスク(設定スクリプトの実行など)を実行する
- カスタム クリーンアップ スクリプト(アプリケーションのプロビジョニング解除やシャットダウンなど)を実行する
- ポリシー(自動スケーリング時のミドル ウェア サービスの再設定など)を適用する
- ポリシー要件に基づいてシステム メトリックを収集する
- データをモニタし、ステータスを更新して、システム ハートビートをキープアライブする

CloudCenter プラットフォームは管理エージェントで Java を使用します。次に、対応する CloudCenter リリースの JDK バージョンの詳細を示します。

Java のバージョン

CloudCenter のバージョン	Java のバージョン
CloudCenter 4.0.0	Java 1.7 アップデート 75
CloudCenter 4.5.3	Java 1.7 アップデート 80
CloudCenter 4.6.0	Java 1.8 アップデート 102

カスタム Linux イメージへの CloudCenter ツールのインストール

- 概要
- インストール プロセス
- クラウドごとの特徴
- OS タイプの識別
- インストールの正常終了

概要

さまざまなクラウドおよびリージョンに CloudCenter 対応イメージを作成する場合は、CloudCenter インストーラ バイナリ（LINUX_WORKER_OS_VERSION）を使用します。

⚠ 事前作成されたイメージを変更する（たとえば、OS アップデートの実行時に、新しいツールをインストールして、管理エージェントをアップグレードするなどの）場合、そのイメージには、CloudCenter ツールがインストール済みになっているので注意してください。

CloudCenter ツールがインストール済みのイメージをアップデートする際は、最後のスナップショットを作成する前に、このファイル（/usr/local/osmosix/etc/hostid）を削除する必要があります。

✓ CentOS 7 および RHEL-7 に CloudCenter ツールをインストールする場合にもこの手順を行ってください。

ⓘ Linux マシンに管理エージェントをインストールする場合は、使用する OS およびプラットフォームに関する情報を提供する必要があります。

VMware VM

VMware VM を使用するには、VMware ツールが Linux マシンにすでにインストールされている必要があります。

インストール プロセス

Linux イメージに CloudCenter ツール（「Key Components」を参照）をインストールするには、次の手順に従います。

1. インストーラ バイナリおよびパスを取得するには、CloudCenter サポートにお問い合わせください。CCM VM の /tmp フォルダにインストーラ パッケージ *(worker_installer.bin)* をダウンロードします。
2. VM を起動するのに使用したキー ペアを使用して、アプリケーション VM インスタンスに SSH で接続します。
3. 次のコマンドを実行して CloudCenter ツールをインストールします。

```
sudo -i
cd /tmp
```

4. アプリケーション VM の /tmp フォルダにインストーラ パッケージをダウンロードします。
5. 権限を変更し、アプリケーション VM（ワーカー）インストーラ ファイルを使用して CloudCenter ツールをインストールします。

```
chmod 755 worker_installer.bin
./worker_installer.bin <ostype> <cloudtype> worker1
-- See Syntax
```

構文

ostype = rhel6、rhel7、oel6、centos6、centos7、ubuntu10、ubuntu1204、ubuntu1404、および amazon（使用している OS タイプを識別するには、次の項を参照してください）

cloudtype = amazon、azure、azurepack、azurerm、google、opsource、openstack、softlayer、vmware、および vcd

6. VM インスタンスをクリーンアップし、終了します。

```
rm worker_installer.bin
rm ~/.ssh/authorized_keys
exit
```

クラウドごとの特徴

Azure

Azure インストールの特徴

1. CloudCenter ツールをアプリケーショ ン VM にインストールします。
2. インストールが終了したら、次のコマンドを実行します。
   `waagent -deprovision+user`
3. VM を停止します。
4. イメージをキャプチャします。

OS タイプの識別

OS タイプを確認/識別するには、次の手順に従います。

1. 端末で、（Linux の実装に基づいて）次のいずれかのコマンドを発行します。
   a. **$ cat /etc/issue**
      Ubuntu 14.04.2 LTS \n \l

      または
   b. **$ cat /etc/*-release**
      CentOS release 6.3 (Final)
2. 次のいずれかのコマンドへの応答に基づいて OS タイプを識別できます。

インストールの正常終了

インストーラ コマンドを実行すると、インストール結果が画面に次のように表示されます。

- 成功シナリオ：正常にインストールされたコンポーネントのリストは緑色の文字で示されます。
- 障害シナリオ：各障害の詳細を示すログ ファイルへのパスが表示されます。

Windows イメージへの CloudCenter ツールのインストール

- インストール プロセス
- インストール後の検証

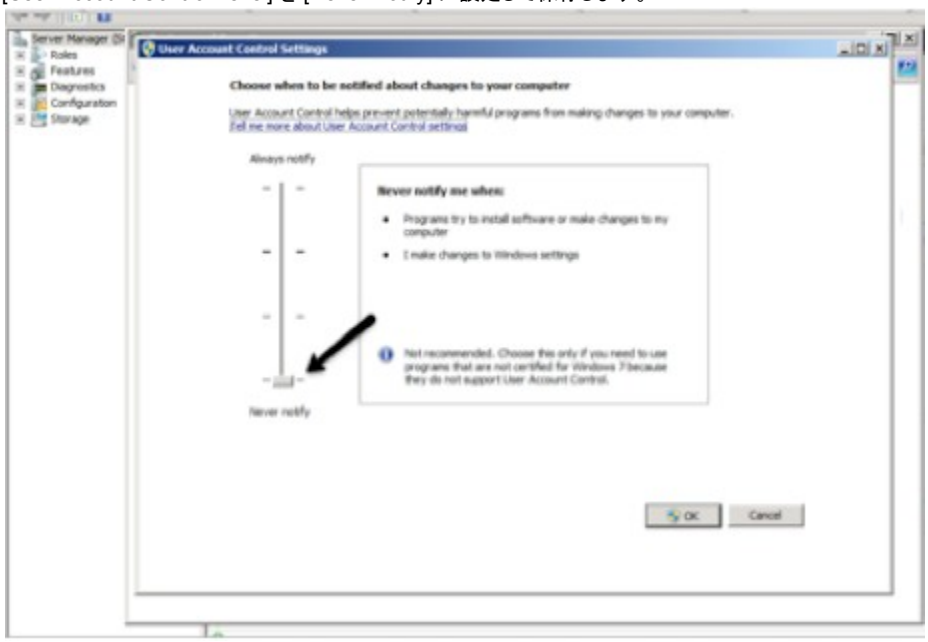
この手順は、クラウドおよび Windows のバージョンによって異なる場合があります。この手順により、WINDOWS_WORKER_*OS_VERSION* イメージをインストールするための参照点が得られます。

インストール プロセス

Windows イメージに CloudCenter ツールをインストールするには、次の手順に従います。

1. CloudCenter サポートに連絡して、インストーラ パッケージ *(cliqr_installer.exe)*を手に入れます。
2. クラウド プロバイダー コンソールからベースの Windows イメージを起動します。次の図に、OpenStack での例を示します。

3. [User Account Control Level] を [Never Notify] に設定して保存します。



4. CloudCenter には Powershell 4.0 が必要です。このバージョンの Powershell を実行していることを確認します。

```
PS C:\> $PSVersionTable.PSVersion

Major Minor Build Revision
----- ----- ----- --------
4 0 -1 -1
```

5. Powershell のコマンドライン ウィンドウで、Powershell ExecutionPolicy をバイパスするコマンドを発行します。
Set-ExecutionPolicy -ExecutionPolicy Bypass

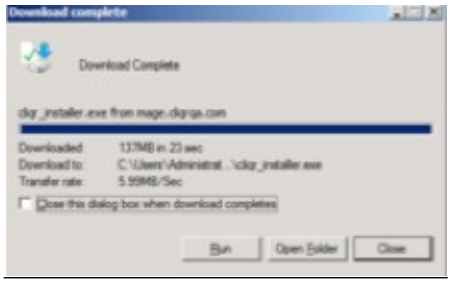
6. `C:\PROGRA~1` パスが `C:\Program Files` に対して解決可能であることを確認します。

```
dir "C:\PROGRA~1"
```

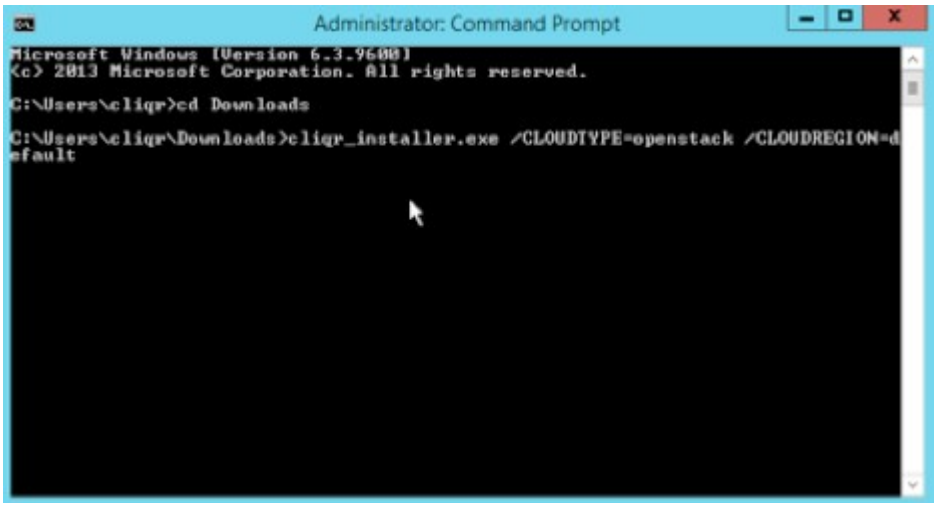
解決可能でない場合、対応するリンクを次のように作成する必要があります。

```
mklink /J "C:\PROGRA~1" "C:\Program Files"
```

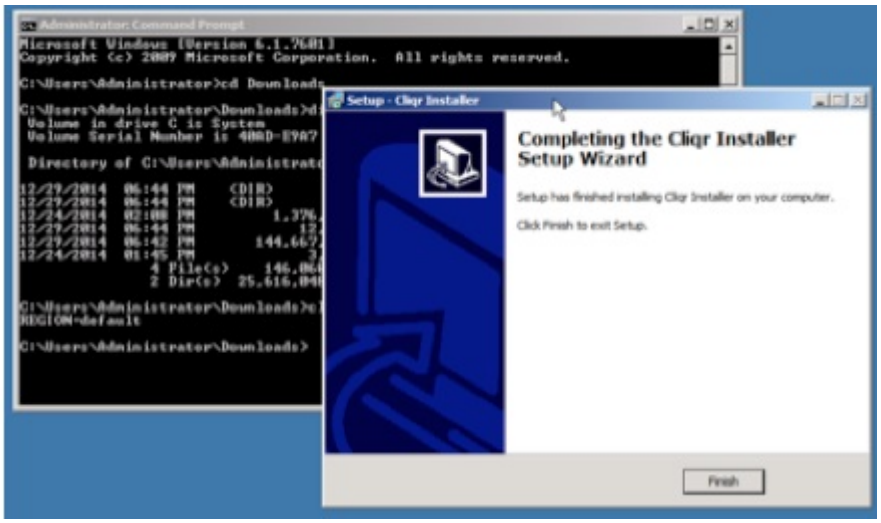
7. CloudCenter サポートの指示に従って、CloudCenter ツールをダウンロードします。



8. コマンド プロンプト ウィンドウを開き、ダウンロードしたファイルを実行します。
C:\Downloads cliqr_installer.exe /CLOUDTYPE= openstack（または他のクラウド）/CLOUDREGION=default
（他のクラウドのオプションとしては、amazon、azure、azurepack、google、openstack、opsource、softlayer、vcd、vmware、cisco があります）



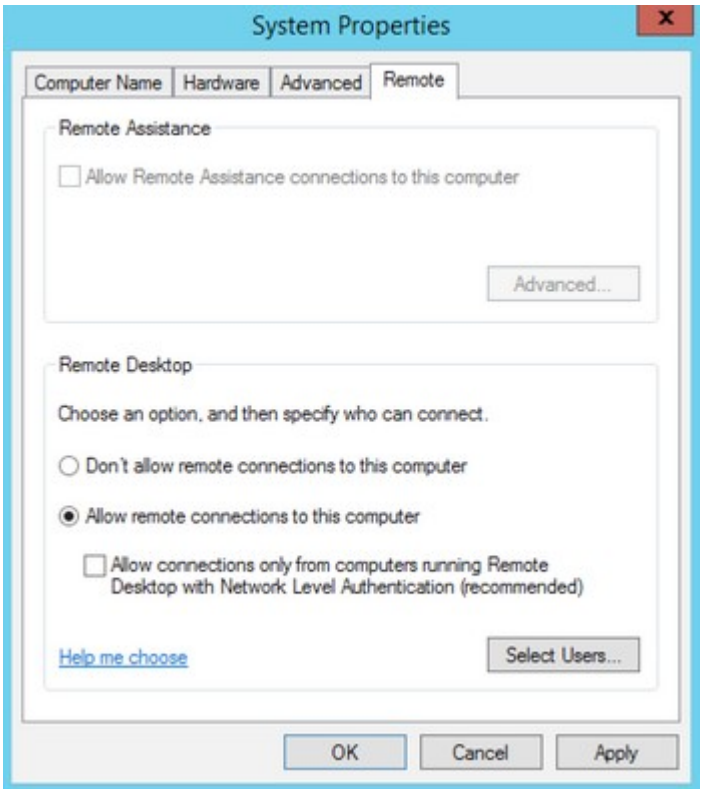
9. インストーラを実行します。導入によっては、IIS のインストールを追加します。



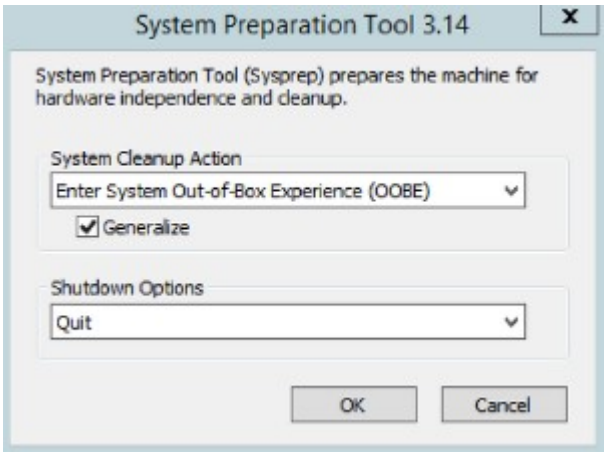
Windows Azure イメージを使用する場合の追加の手順

- Azure 内 Windows 2012 R2 アプリケーション VM イメージを設定する*場合*、次に示す追加の手順が必要です。
  - Azure 内 Windows 2012 R2 ワーカー イメージ
    a. サーバ マネージャを開き、[Add Roles and Features] ウィザードを起動するかまたは開きます。
    b. [Before you begin] ペインで [Next] をクリックします。
    c. [Installation type] ペインで [Role-based or feature-based installation] を選択し、[Next] をクリックします。
    d. [Server Selection] ペインで [Select a server from the server pool] オプションを選択します。
    e. [Server Role] ペインで [Web Server (IIS)] を選択し、[Application Development] の下のすべてのロールを追加します。
    f. [Next] をクリックして [Features] ペインに進みます。
    g. [Internet Information Services] ペインで [Web Management Tools] を展開し、[IIS Management Console] を選択します。
    h. [Add Features] を選択し、以下を選択します。
        i. .NET Framework 3.5 Features の場合、.NET Framework 3.5
        ii. .NET Framework 4.5 Features の場合、ASP.NET 4.5

- installer.exe ファイルを削除します。
- [Control Panel] > [System and Security] > [System] > [Allow Remote Access] に移動します。チェックボックスをオフにし、**ネットワーク レベル認証によってリモート デスクトップを実行しているコンピュータからの接続だけを許可するようにします**。この操作により、CloudCenter UI から VM への RDP が可能になります。

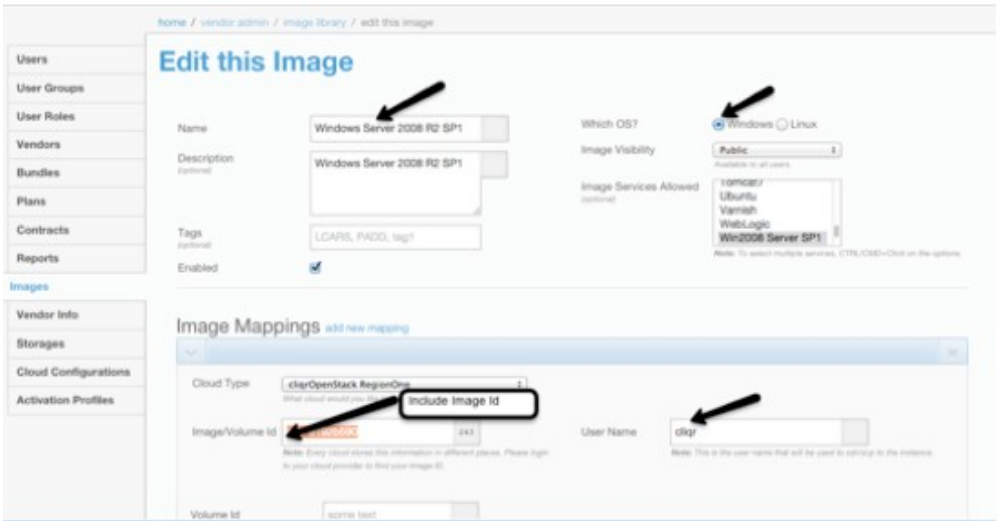


- ファイル エクスプローラで、C:\Windows\System32\Sysprep に移動します。**sysprep** をダブルクリックします。下の図に表示されているオプションを選択していることを確認します。sysprep が実行されると RDP セッションが終了します。



- [Azure Management UI] に移動し、アプリケーション VM イメージをシャットダウンします

10. クラウド コンソールからイメージのスナップショットを作成します。
11. スナップショットを作成したら、CloudCenter の論理イメージにイメージ ID を含めます。

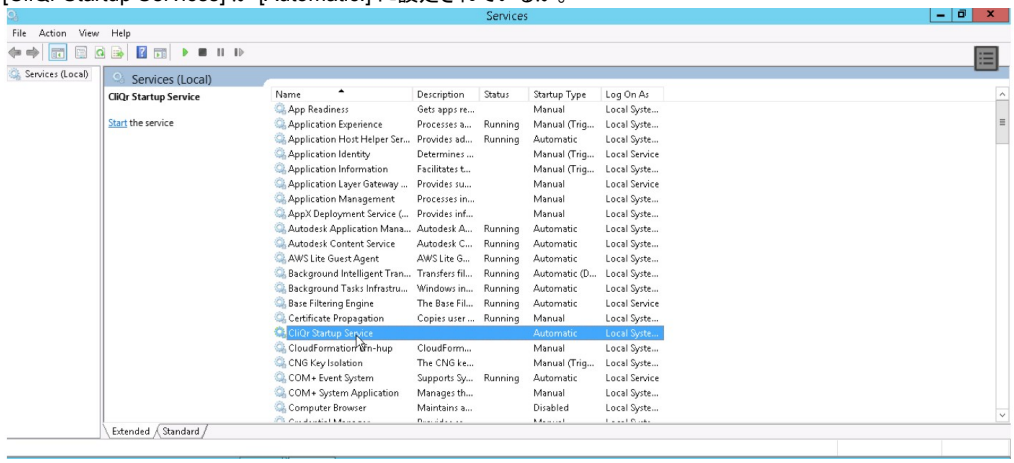


これで、Windows イメージがクラウドにマッピングされました。

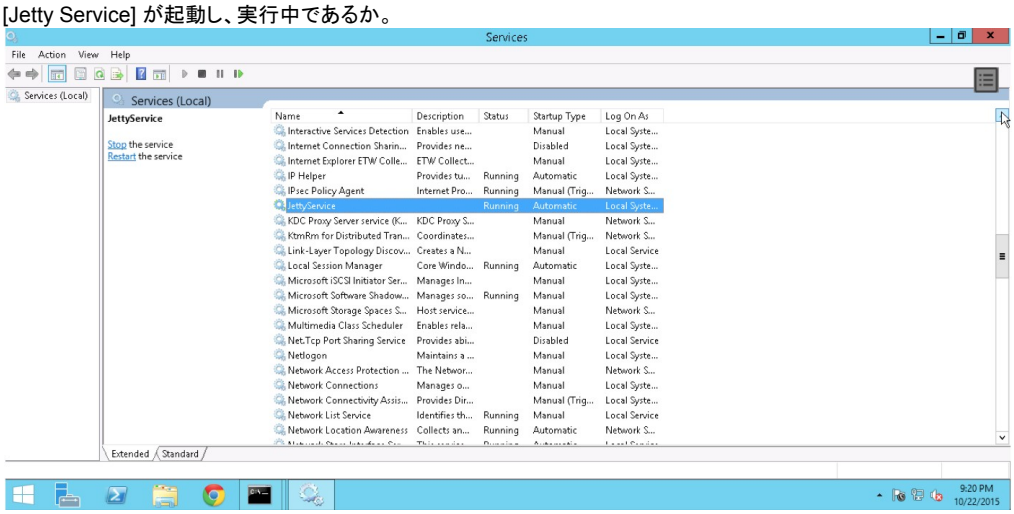
インストール後の検証

エージェントをインストールしたら、次の情報を確認します。

- [CliQr Startup Services] が [Automatic:] に設定されているか。



- [Jetty Service] が起動し、実行中であるか。



この情報が検証されない場合、エージェントが正しくインストールされない可能性があるので注意してください。

メモ

CloudCenter プラットフォームの導入時には、IP アドレスや DNS 名、その他のエントリを、後の使用のためにメモするよう何度が指示されます。この作業はインストール プロセスの一部です。相互参照やフォローアップのためにメモや参照を追跡するのに役立つよう、シスコでは次の表を用意しています。このページを印刷し、*メモ*列に必要な詳細情報を書き留めてください。

追跡リスト	リンク	メモ
☐ 1	パッケージ ストアのインストール（条件付き）	リポジトリの IP アドレス/FQDN
☐ 2	バンドル ストアのインストール（オプション）	インターネットへの接続が制限されている環境の場合は、ローカル バンドル ストアを作成し、それを CCO によって登録します。

☐ 3	CCM のインストール（必須）	非 HA
		• Mail
		  • SMTP Host
		  • SMTP Port
		  • SMTP Auth
		Mail_User
		  • メール ユーザ
		  • [Password]
		  • ユーザから
		  • 表示名
		Server_Info
		  • パブリック DNS
		  • モニタ URL
		  • Hazelcast IP
		  • 外部 URL
		Network
		  • Hostname
		  • インターフェイス

		非 HA スタンドアロン
		• CCM IP
		• Mail
		  • SMTP Host
		  • SMTP Port
		  • SMTP Auth
		• Mail_User
		  • メール ユーザ
		  • [Password]
		  • ユーザから
		  • 表示名
		• Server_Info
		  • パブリック DNS
		  • モニタ URL
		  • Hazelcast IP
		  • 外部 URL
		• Network
		  • Hostname
		  • インターフェイス

		HA
		• マスター データベースのホスト名
		• マスター データベースのプライベート IP
		• スレーブ データベースのホスト名
		• スレーブ データベースのプライベート IP
		• VIP
		• プライマリ CCM IP
		• セカンダリ CCM ノード IP
		• Mail
		  • SMTP Host
		  • SMTP Port
		  • SMTP Auth
		• Mail_User
		  • メール ユーザ
		  • [Password]
		  • ユーザから

追跡リスト	リンク	メモ

		• 表示名
		• Server_Info
		  • パブリック DNS
		  • モニタ URL
		  • Hazelcast IP
		  • 外部 URL
		• Network
		  • Hostname
		  • インターフェイス

☐ 4	CloudCenter リージョンごとのインストール（必須）	各クラウド リージョンはクラウドを設定するときに CCM UI で識別されます。CloudCenter プラットフォームで管理する必要がある *クラウド リージョン*ごとに、GUAC、AMQP、および CCO コンポーネントをインストール、セットアップおよび設定する必要があります。
☐ 4.1	**GUAC**	• CCM のホスト名 • CCO のホスト名

☐ 4.2	**AMQP**	• 非 HA
		  • CCM のホスト名
		  • CCO のホスト名
		• HA
		  • ***プライマリ AMQP***
		  • CCM のホスト名
		  • CCO のホスト名
		  • ***セカンダリ AMQP***
		  • CCM のホスト名
		  • CCO のホスト名

☐ 4.3	**CCO**	• 非 HA
		  • AgentBundle
		  • カスタム バンドルを使用する場合、cdn.cliqr.com をカスタム バンドル ストアの IP または DNS に置き換えます
		  • パッケージ ストアを使用する場合、repo.cliqrtech.com をカスタム パッケージ ストアの IP または DNS に置き換えます
		  • AMQP_Server
		  • AMQP Server IP
		  • AMQP Port
		  • Network
		  • Hostname
		  • Guacamole
		  • Connection Broker Host
		  • Connection Broker Port1
		  • Connection Broker Port2
		  • Docker
		  • Docker Registry URL
		  • Docker CACert URL
		• HA
		  • プライマリ CCO
		  • AgentBundle
		  • カスタム バンドルを使用する場合、cdn.cliqr.com をカスタム バンドル ストアの IP または DNS に置き換えます
		  • パッケージ ストアを使用する場合、repo.cliqrtech.com をカスタム パッケージ ストアの IP または DNS に置き換えます
		  • AMQP_Server
		  • AMQP Server IP
		  • AMQP ポート
		  • ネットワークのホスト名
		  • Guacamole
		  • Connection Broker Host
		  • Connection Broker Port1
		  • Connection Broker Port2
		  • Docker
		  • Docker Registry URL
		  • Docker CACert URL
		  • Hazelcast IP のカンマ区切のリスト
		  • プライマリ CCM IP
		  • セカンダリ CCM ノード IP
		  • セカンダリ CCO

追跡リスト	リンク	メモ

		• AgentBundle
		  • カスタム バンドルを使用する場合、cdn.cliqr.com をカスタム バンドル ストアの IP または DNS に置き換えます
		  • パッケージ ストアを使用する場合、repo.cliqrtech.com をカスタム パッケージ ストアの IP または DNS に置き換えます
		• AMQP_Server
		  • AMQP Server IP
		  • AMQP ポート
		• ネットワークのホスト名
		
		• Guacamole
		  • Connection Broker Host
		  • Connection Broker Port1
		  • Connection Broker Port2
		• Docker
		  • Docker Registry URL
		  • Docker CACert URL
		• Hazelcast IP のカンマ区切のリスト
		  • プライマリ CCM IP
		  • セカンダリ CCM ノード IP

☐ 5	ヘルス モニタのインストール（オプション）	• 非 HA
		  • Monitor ID
		  • CCM Hostname/URL
		  • Monitor User
		• HA
		  • MON_PRIMARY
		  • MON_SECONDARY
		  • CCM Hostname/URL
		  • Monitor User
		  • Monitor ID

☐ 6	専用 Docker レジストリのインストール（オプション）	• ローカル パッケージ ストアの IP

アプライアンスの設定オプション

概要

CloudCenter は、次の 2 つのオプションのいずれかを使用してセットアップできます。

- Installers：インストーラ オプションを使用すると、Cisco CloudCenter でサポートされる任意のモードおよびロールで、さまざまなコンポーネントを柔軟に決定したり使用したりできます。
- Appliances：シスコでは、コンポーネント用 CloudCenter アプライアンスを提供しています。これには、特定のクラウド プロバイダー向けに修正された***特定***のモードおよびロールが含まれています。

提供状況

CloudCenter のメジャー リリースには、次のコンポーネントおよびクラウド プロバイダー用のアプライアンスが含まれています。これらのアプライアンスは、シスコにより、CentOS 7.x ベース イメージ上に構築されています。

クラウド	CCM	CCO	AMQP	Monitor	Worker
Amazon	AMI	AMI	AMI	AMI	AMI
Openstack	QCOW2	QCOW2	QCOW2	QCOW2	QCOW2
VMware	OVA	OVA	OVA	OVA	OVA

アプライアンス アプローチ

アプライアンスを使用して CloudCenter をセットアップするには、次の手順に従います。

1. インフラストラクチャの準備

インフラストラクチャの準備

アプライアンス アプローチのためにインフラストラクチャを準備するには、次の手順に従います。

a. イメージをインポートします。

クラウド	詳細
Amazon	cloudcenter-ami-request メイラー リストに電子メールを送信して、AWS アカウントの起動権限を取得します。
Openstack	各コンポーネント用の QCOW2 イメージ ファイルをインポートします。
VMware	*CliqrTemplates* という名前のディレクトリを作成し、各コンポーネント用の OVA ファイルをインポートします。

b. インスタンスを起動します。
   i. インポート済み/共有アプライアンス イメージを使用して、各コンポーネントのインスタンスを作成します。
   ii. 要件:
      1. CloudCenter セットアップごとに 1 つの CCM とモニタが必要です。

         ワーカー イメージは CCM で定義されるため、このコンポーネントのインスタンスを起動する必要はありません。

      2. クラウド リージョンごとに 1 つの AMQP と CCO が必要です。

c. ホスト名の設定:起動したすべての仮想マシンで、ホスト名を更新します。ロールに一致するホスト名を選択します。次に例を示します。

例

```
CCM.mydomain.com
```

コンポーネントをインストールして設定した*後に*ホスト名を変更すると、不明な問題が発生する場合があります。

d. ホスト名の解決の設定:ホスト名を更新したら、次のコマンドを実行して、VM ホスト名が解決可能であることを確認します。
   i. **hostname -i**
   ii. VM の名前が解決可能であれば、**/etc/hosts** ファイルを編集し、VM のホスト名を追加します。
   次に例を示します。

例

```
<hostname> 127.0.0.1
```

e. ネットワーク ルーティング ループバック
   - ネットワーク アドレス変換(NAT)の背後で実行中の導入済みの CCM を参照します。
   - この設定では、内部ネットワークからのマシンが CCM へのアクセスに外部 IP を使用しないように制限されます。
   - この制限に対処するため、CCO および AMQP サーバの **/etc/hosts** ファイルに行を追加し、CCM の内部プライベート IP を含める必要があります。例:CCM の DNS 名が ccm.example.com で、NAT の背後にあり、内部プライベート IP アドレスが 192.168.20.5 で、その外部パブリック IP アドレスが 54.16.20.5 の場合、ローカルの /etc/hosts ファイルに次の行を入力します。

例

```
192.168.20.5 ccm.example.com
```

> ⚠ CCM を設定する場合、上記で使用されるホスト名（ccm.example.com）は、CCM の設定時に [Public DNS] に設定したものと一致する必要があります。

f. CloudCenter 記述子 JSON ファイルを次のように作成します。
すべての CloudCenter コンポーネントのインフラストラクチャを設定したら、CloudCenter 記述子 JSON ファイルを作成し、そこには、すべての CloudCenter コンポーネントを、各モードおよびロールのインフラストラクチャ要素に対応したモードおよび IP アドレスとともに表示します。この記述子ファイルは、ネットワー コンプライアンス チェック（ステップ 3）で使用されます。次に、いくつかの一般的なコンポーネント モードの組み合わせに基づくサンプル ファイル記述子を示します。

> ⚠ **全体的なファイル構造は、さまざまなコンポーネントのモード、クラウド リージョンの数、条件付き/オプションのコンポーネントの使用、およびリポジトリなどの要因によって異なります。また、ファイルで使用されているリージョン名は一意にする必要がありますが、クラウドやデータセンターの名前と一致する必要はありません。これらの文字列は、次に示すように、ネットワーク コンプライアンス チェックの実行、およびレポート結果にのみ使用されます。**

例

```
CCM- NON - HA
Regions
AWEEast
CCO- NON - HA
AMQP- NON - HA
Repos
Bundle Store - cdn.cliqr.com
Package Store - repo.cliqrtech.com
Docker - repo.cliqrtech.com

{
    "CloudCenterComponents": {
        "CCM": {
            "CCM_IP": "1.1.1.1",
            "mode": "NON-HA-STANDALONE"
        },
        "CloudRegions": [{
            "components": {
                "AMQP": {
                    "AMQP_ IP": "1.1.1.1",
                    "mode": "NON-HA"
                },
                "CCO": {
                    "CCO_IP": "1.1.1.1",
                    "mode": "NON-HA"
                }
            },
            "name": "AWSEast"
        }],
        "REPOS": {
            "BUNDLE_STORE": "http://cdn.cliqr.com",
            "DOCKER_REGISTRY": "http://repo.cliqrtech.com:5000",
            "PACKAGE_STORE": "http://repo.cliqrtech.com"
        }
    }
}
```

2. ネットワーク ルーツの設定

ネットワーク ルールの設定

CCM ポート

CCM ポート

ポート	方向	リモート送信元	注
80	入力（オプション）	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	HTTP から HTTPS へのリダイレクション用。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	UI/API アクセス用。
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力	CCO_IP または CCO_PRIMARY_IP および CCO_SECONDARY_IP	CCO と CCM VM 間の双方向通信用。
	出力	CCO_IP または CCO_LB_IP	⚠ CloudCenter の導入でサポートされるすべてのクラウド リージョンに必要です。
	入力（オプション）	MON_IP または MON_PRIMARY_IP および MON_SECONDARY_IP	モニタ CM および CCM VM 間の双方向通信用。
	出力（オプション）	MON_IP または MON_LB_IP	
	入力	AMQP_IP または AMQP_PRIMARY_IP および AMQP_SECONDARY_IP	Guacamole による Web SSH/VNC 用。

AMQP ポート

AMQP ポート

ポート	方向	リモート送信元	注
22	入力（オプション）	許可された SSH の送信元 IP	トラブルシューティングの目的。
5671	入力	• CCO または CCO_PRIMARY、CCO_SECONDARY • ワーカー VM IP の範囲	CCO VM からの通信および起動された VM からの通信用。
7789	入力	ワーカー VM IP の範囲	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
7788	入力/出力	AMQP または AMQP_PRIMARY、AMQP_SECONDARY、AMQP_LB	起動された VM の SSH/VNC アクセス用。逆プロキシによる実行。
443	入力	0.0.0.0/0（またはアクセス可能なユーザ ブラウザでの適切な IP アドレス範囲）	起動された VM の SSH/VNC および RDP アクセス用。
8443	出力	• CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONADRY • CCO または CCO_PRIMARY、CCO_SECONDARY	起動された VM の SSH/VNC アクセス用。AMQP VM の Guacamole サーバはこのポートによって CCM および CCO VM と通信します。

CCO ポート

CCO ポート

ポート	方向	リモート送信元	注
8443	入力/出力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY	CCO と CCM VM 間の双方向通信用。
22	入力(オプション)	許可された SSH の送信元 IP	トラブルシューティングの目的。
すべて	出力(条件付き)	クラウド リージョンのエンドポイント、スクリプト ソース	クラウド リージョンのエンドポイント アクセスと外部サービスで定義されているスクリプト/パッケージのダウンロード用。
2375	出力(条件付き)	EXT_SCRIPT_EXECUTOR	外部スクリプトを実行するための Docker コンテナ エンジン アクセス用。

モニタリング ポート

MON ポート

ポート	方向	リモート送信元	注
22	入力(オプション)	許可された SSH の送信元 IP	トラブルシューティングの目的。
8443	入力/出力	CCM または CCM_SA または CCM_SA_PRIMARY、CCM_SA_SECONDARY、CCM_LB	CCM およびモニタ VM 間の双方向通信用です。
	出力	CCO または CCO_LB	CCO VM へのアクセス用。

3. ネットワーク コンプライアンス チェックの実行

ネットワーク コンプライアンス チェックの実行

CloudCenter コンポーネント VM、アプリケーション VM、およびエンド ユーザ アクセス間のコンポーネント間通信で使用されるすべてのネットワーク ルールが正確に設定されていることを確認するには、ネットワーク コンプライアンス チェックを実行する必要があります。

CloudCenter のコンポーネント VM

CloudCenter コンポーネントのネットワーク コンプライアンス チェックを実行する場合は、「*1. インフラストラクチャの準備*」**のプロセスの一環として起動した 各 VM** でこの手順を実行します。

a. 「*1. インフラストラクチャの準備*」で作成したデプロイメント記述子 JSON ファイルを /tmp ロケーションにアップロードします。
b. validator.py という名前の検証スクリプト(CloudCenter アプライアンス ファイルで使用可能)を /tmp ロケーションにアップロードします。
c. /tmp ロケーションから次のコマンドを実行します。

コマンド

```
python network_validator.py --role=<role> --region=<unique
region identifier> --deploymentDetails=<envdeployment descriptor
JSON file path>
```

例:

- python network_validator.py —-role=CCM --deploymentDetails=deployment.json
- python network_validator.py —-role=CCO --region=AmazonEast1 --deploymentDetails=deployment.json
- python network_validator.py —-role=AMQP --region=AmazonEast1 --deploymentDetails=deployment.json

d. 結果を検証し、障害がないこと、および、すべての必要なネットワーク ルールが正確に設定されていることを確認します。障害が発生した場合は、ネットワーク設定を適切に更新してテストを繰り返します。

アプリケーション VM

（アプリケーション オーケストレーションの一部として起動した）アプリケーション VM が関連の CloudCenter コンポーネントおよびリポジトリと通信できることを確認するには、次の手順を実行します。

a. CloudCenter で管理する予定のすべてのクラウド環境で、テスト VM を起動します。
b. デプロイメント記述子 JSON ファイルを /tmp ロケーションにアップロードします。
c. validator.py という名前の検証スクリプトを /tmp ロケーションにアップロードします。
d. /tmp ロケーションから次のコマンドを実行します。

コマンド

```
python network_validator.py --role=AGENT --region=<unique region
identifier> --deploymentDetails=<deployment descriptor JSON file
path>
```

例：
python network_validator.py –-role=AGENT --region=AmazonEast1 --deploymentDetails=deployment.json

e. 結果を検証し、障害がないこと、および、すべての必要なネットワーク ルールが正確に設定されていることを確認します。障害が発生した場合は、ネットワーク設定を適切に更新してテストを繰り返します。
f. ネットワーク コンプライアンス チェック テストが成功したら、テスト VM を終了します。

クライアント テスト

UI や REST API アクセスを可能にするため、ユーザが特定の CloudCenter コンポーネント（CCM AMQP や GUAC）にアクセスできる必要があります。これらのコンポーネントへのクライアント アクセスを可能にするには、次の手順を実行します。

a. エンドユーザがアクセスするネットワーク上で、Linux、Windows（Python が有効）、または MacOSX システムが実行していることを確認します。
b. このクライアント システムにデプロイメント記述子ファイルをアップロードします。
c. クライアント システムに validator.py をアップロードします。
d. python スクリプト ロケーションから次のコマンドを実行します。

コマンド

```
python validator.py --role=CLIENT
```

e. 結果を検証し、障害がないこと、および、すべての必要なネットワーク ルールが正確に設定されていることを確認します。障害が発生した場合は、ネットワーク設定を適切に更新してテストを繰り返します。

次のセクションに進む前に、ネットワーク コンプライアンス チェックが成功することを確認します。

4. コンポーネントの設定

コンポーネントの設定

CCM ウィザードのプロパティ

CCM ウィザードのプロパティ

CCM ウィザードのプロパティを設定するには、次の手順に従います。

a. CCM ウィザードを呼び出し、基本プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. ウィザードを開始し、次の表に特定した値を入力します。

CCM ウィザードのパス

```
/usr/local/cliqr/bin/ccm_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

CCM プロパティ	フィールド	説明
Mail	• **SMTP Host** • **SMTP Port** • **SMTP Auth**	メール通知を送信する SMTP サーバの詳細。
Mail_User	• **メール ユーザ** • **[Password]** • **ユーザから** • **表示名**	メール通知を送信するメール認証と設定の詳細
Server_info	• **パブリック DNS**	• CCM の DNS または IP。 • CCM VM との通信に CCO VM で使用されます。
	• モニタ URL	• モニタ VM の完全な URL。たとえば、https://<MON または MON_LB IP アドレス>:8443 です。 • HTTPS プロトコルである必要があります。 • モニタ VM からのヘルス ステータスの取得に CCM VM で使用されます。
	• Hazelcast IP	• CCM VM のプライベート IP アドレス。 • CloudCenter プラットフォームによって内部で使用されます
	• **外部 URL**	非 HA CCM シナリオのオプション
Config_App_Logo	N/A	アプリケーション プロファイル テンプレートで使用されます。
ESB_Info	N/A	Enterprise Service Bus（ESB）、CloudCenter アプライアンスにデフォルトではインストールされないオプション コンポーネントをインストールした場合にのみ必要。
Network	• Hostname • インターフェイス	これらの設定を変更しなかった場合はデフォルトを使用します。

c. CCM 設定ウィザードを終了します。

AMQP - CCM/CCO ウィザードのプロパティ

AMQP：Guacamole サーバの CCM/CCO プロパティの設定

専用 GUAC 設定の場合
この GUA の設定ウィザードの指示は、専用 Guacamole サーバを設定している場合は必要ではありません。

a. GUA ウィザードを起動して、CCO と CCM プロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニューグループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。

GUA ウィザードを起動します。

GUA ウィザードのパス

```
/usr/local/cliqr/bin/gua_config_wizard.sh
```

> **今後の参照のためにこれを書き留めてください。**
> 後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

b. CCM および CCO VM のプロパティの設定

グループ	ホスト	使用可能な IP アドレス
CCM_Info	CCM ホスト	***CCM_IP*** または ***CCM_SA_IP*** または ***CCM_LB_IP***
CCO_Info	CCO ホスト	***CCO_IP*** または ***CCO_LB_IP***

c. 変更を確認し、GUA 設定ウィザードを終了します。

CCO ウィザードのプロパティ

CCO：CCO プロパティの設定

a. CCO ウィザードを呼び出し、エージェント バンドル、AMQP サーバ、Guacamole サーバおよび Docker サーバのプロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. CCO ウィザードを起動します。

GUA ウィザードのパス

```
/usr/local/cliqr/bin/cco_config_wizard.sh
```

> **今後の参照のためにこれを書き留めてください。**
> 後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

c. エージェント バンドル、AMQP サーバ、Guacamole サーバおよび Docker VM のプロパティを設定します。

グループ	プロパティ	注
AgentBundle		デフォルトを使用します。 • カスタム バンドルを使用する場合、cdn.cliqr.com をカスタム バンドル ストアの IP または DNS に置き換えます • パッケージ ストアを使用する場合、repo.cliqrtech.com をカスタム パッケージ ストアの IP または DNS に置き換えます
AMQP_Server	• AMQP Server IP • AMQP Port	• ***AMQP_IP*** か ***AMQP_LB_IP*** • 5671
Network	• Hostname	CCO 環境のネットワークの詳細を設定します。これは、VM のプライベート IP を設定するオプションの手順です。VM の IP またはホスト名が事前に設定されていない場合、または既存の IP またはホスト名を上書きする必要がある場合は、一般に次の情報を設定できます。
Guacamole	• Connection Broker Host • Connection Broker Port1 • Connection Broker Port2	• ***AMQP_IP*** か ***AMQP_LB_IP*** • 7788 • 7789
Docker	• Docker Registry URL • Docker CACert URL	• カスタム Docker レジストリが使用されたときだけ設定します • Docker レジストリでカスタム CA 証明書を使用して SSL を使用しているときだけ設定します

d. 変更を確認し、CCO 設定ウィザードを終了します。

Monitor - CCM ウィザードのプロパティ

モニタ:CCM プロパティの設定

a. CCM ウィザードを開始し、各モニタ サーバの基本的なプロパティを設定します。ウィザードには異なるプロパティを持つ複数のメニュー グループが含まれています。次の表では、各プロパティをリストし、共通のプロパティを**太字**でハイライトしています。
b. ウィザードを開始し、次の表に特定した値を入力します。

モニタ ウィザード パス

```
/usr/local/cliqr/bin/monitor_config_wizard.sh
```

今後の参照のためにこれを書き留めてください。
後で使用できるように、「メモ」セクションを印刷したものにフィールドの詳細を書き留めてください。

グループ	プロパティ	注
CCM_Info	• Monitor ID • CCM Hostname/URL • Monitor User	• クラウド プロバイダーで作成したインスタンスとボリュームのヘルスチェックに使用する(英数字の)固有識別子。 • **CCM_IP** または **CCM_SA_IP** または **CCM_LB_IP** • そのユーザにとって有効なクラウド リージョンのヘルス チェックが実行される CCM サーバ上のユーザ ID。すべてのアクティブ化されたクラウド リージョンでヘルス チェックを実行する場合は、これを 2(CloudCenter のルート管理者のユーザ ID)に設定します。特定のクラウド リージョンでヘルスチェックを行う場合は、それらのリージョンを使用して新しいユーザを作成してアクティブ化し、そのユーザのユーザ ID をこのプロパティの値として使用します。ユーザ ID を取得するために、v1 ユーザ API を使用できます。

c. モニタ ウィザードを終了します。

証明書認証

- 概要
- クライアント証明書
  - 独自の証明書の使用
- コンポーネント証明書
  - 要件
  - CA ファイル セットアップ プロセス

概要

CloudCenter では、次の 2 種類の証明書を使用します。

- **クライアント証明書**:ブラウザ経由のクライアント通信、または CCM サーバによる REST 通信のための CCM UI への要求を認証します。
- **コンポーネント証明書**:コンポーネント導入のための、CloudCenter コンポーネント間(CCO から CCM、および GUA から CCM)の通信を認証します。

いずれかの証明書の更新が必要な場合があります。

クライアント証明書

クライアント証明書は、**example.com.crt** または **cliqrtech.com.crt** を参照します。

CloudCenter のすべてのインストーラおよびアプライアンス パッケージには、そのまますぐに使用できるよう構築されたデフォルトの自己署名証明書が含まれています。ご利用のブラウザが提供するプロセスに従い、CCM を信頼できるアプリケーションとして追加します。

信頼できる認証により、CCM アプリケーションと Web サーバ間の信頼関係が設定されたことが示されます。CCM アプリケーションは、信頼できる Web サーバから要求を受信すると、その Web サーバが要求された認証を取り扱っていると想定します。

独自の証明書の使用

⚠ コンポーネント証明書では、CloudCenter サポートが提供するファイルだけを使用してください。この目的では、独自のファイルは利用できません。

独自のクライアント証明書を使用する場合は、次の手順に従います。

1. *certificate*.crt、*certificate*.key、および *cacertificate*.crt ファイルを用意します。
   たとえば、example.com.crt、example.com.key、および gd_bundle.crt ファイルとします。
2. 証明書ファイルを **/usr/local/tomcat/conf/ssl** フォルダに保存します。
3. **/usr/local/tomcat/conf/server.xml** ファイルを新しい証明書を指すように更新します。
4. CCM サーバを再起動します。

コンポーネント証明書

コンポーネント証明書は、/usr/local/tomcat/conf/ssl フォルダ（このフォルダには、/usr/local/osmosix/ssl/*component* フォルダへのリンクが含まれる）内の、**mgmtserver.crt**（CCM）、**gateway.crt**（CCO）、**monitor.crt**（Monitor）、**gua.crt**（Guacamole）、または **esb.crt** ファイルを参照します。

CloudCenter プラットフォームは、各企業が CloudCenter 証明書サーバを使用して導入ごとに新しい証明書が生成することを要件としています。

⚠ 導入に固有な証明書セットを生成、取得して、CloudCenter コンポーネント間の相互通信を可能にするには、CloudCenter サポートに連絡してください。これらの証明書を各コンポーネントの /usr/local/osmosix/ssl フォルダで更新する必要があります。

独自の証明書を含めるようにクライアント証明書を更新することもできます。その場合は、次の項「*独自の証明書の使用*」を参照してください。

要件

- 新しい証明書形式では、証明書が導入 ID を保持するため、単一の CloudCenter 導入におけるコンポーネント間でセキュアな通信が実現します。
- 新規導入では、CloudCenter 証明書サーバを使用して、CCM、CCO、モニタ、Guacamole（GUA）、およびエンタープライズ サービス バス（ESB）用の新しい証明書を生成する必要があります。
- CloudCenter の導入ごとに、次の要件を満たす証明書を使用する必要があります。
  - 一意の導入 ID である CloudCenter ID（CCID）が CliQr によって自動生成される。
  - 証明書が相互に認証される。
  - 各 CCID が独自の証明書セットを含む。

企業が導入のための証明書を必要とする場合は、導入ごとに一意の CA も必要です。

⚠ 次のいずれかを要求する場合は、CloudCenter サポートに連絡してください。

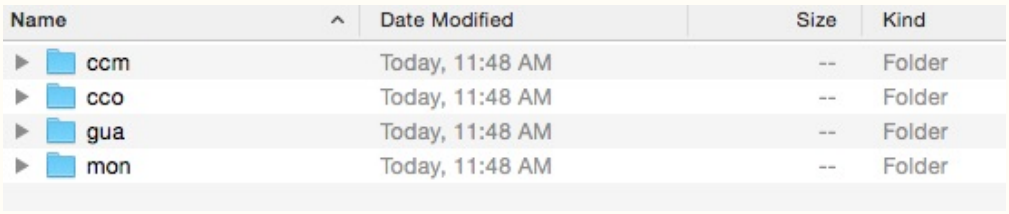
- 各導入のための証明書
  または
- 独自の CA を使用するための証明書署名要求（CSR）

要求があった場合、CloudCenter サポートは、生成された CSR を企業に転送し、企業は独自の CA を使用して証明書に署名できます。

✓ 開発環境やテスト環境では、すぐに作業できるよう構築されているデフォルトの自己署名証明書（すべての CloudCenter インストーラおよびアプライアンス パッケージで使用可能）を使用します。

CA ファイル セットアップ プロセス

これらの証明書は、CloudCenter の標準インストールの一部としては生成されません。導入に固有の（下図のような）証明書セットを生成および取得するには、CloudCenter サポートにお問い合わせください。



これらのコンポーネントのそれぞれ（CCM、CCO、GUA、MON、および ESB）で、次のコマンドを実行します。

1. 生成された証明書パッケージ（uniqueName_certs.zip file）を /tmp フォルダにコピーし、各コンポーネントの証明書をそれぞれ抽出します。

```
cd /usr/local/osmosix/ssl
jar xf /tmp/certs.zip [ccm | cco | gua | mon | esb ]
```

2. エンタープライズ サービス バス（ESB）サーバでは、さらに次のコマンドを実行する必要があります。

```
cd /etc/rabbitmq/certs
jar xf <cert-zip-filename> esb_rabbit
mv ./esb_rabbit/* .
rm -rf esb_rabbit


chown -R rabbitmq:rabbitmq /etc/rabbitmq/certs
chmod 700 /etc/rabbitmq/certs
chmod 600 /etc/rabbitmq/certs/*
```

3. 証明書パッケージをコピーした後、すべてのコンポーネントをリブートします。

CloudCenter の設定とセットアップ

- 管理者アカウントの設定
- クラウドの設定
- インスタンス タイプの管理
- イメージ ID の設定
- イメージのマッピング
- CCM による CCO の登録

管理者アカウントの設定

- 概要
- CloudCenter の管理者アカウントの設定
- 管理者用使用計画の作成
- 管理者用契約の作成
- 管理者への計画の割り当て
- 管理者プロファイルの変更

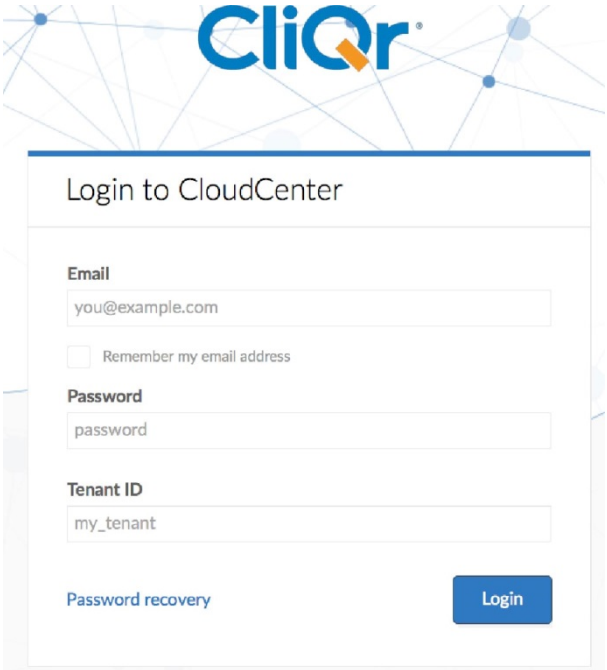
概要

CCM VM を設定したら、システム管理者（「People」参照）は、CloudCenter プラットフォーム管理者（ルート）のアカウントを設定する必要があります。CCM UI を起動した後は、まず使用計画を作成し、CloudCenter の契約で指定した VM の数を指定し、計画を CloudCenter プラットフォーム管理者に割り当てる必要があります。

CloudCenter の管理者アカウントの設定

CloudCenter プラットフォーム管理者を設定するには、次のプロセスに従います。

1. ブラウザを開き、CCM VM の IP アドレス（たとえば、https://<CCM_IP>）を使用して CCM VM に接続します。



2. 有効なクレデンシャルを使用して CloudCenter 管理者としてログインします。初期のクレデンシャルは次のとおりです。
   ユーザ名：admin@cliqrtech.com
   パスワード：cliqr

> **（注）**
> [Account Information] ページに、（まだ使用計画が設定されていない）[Trial Subscription] として現在の計画が表示されます。
>
> [Account Information] ページを表示するには、CCM UI の右上隅にある [user account] リンクをクリックし、ドロップダウンから [Account] を選択します。

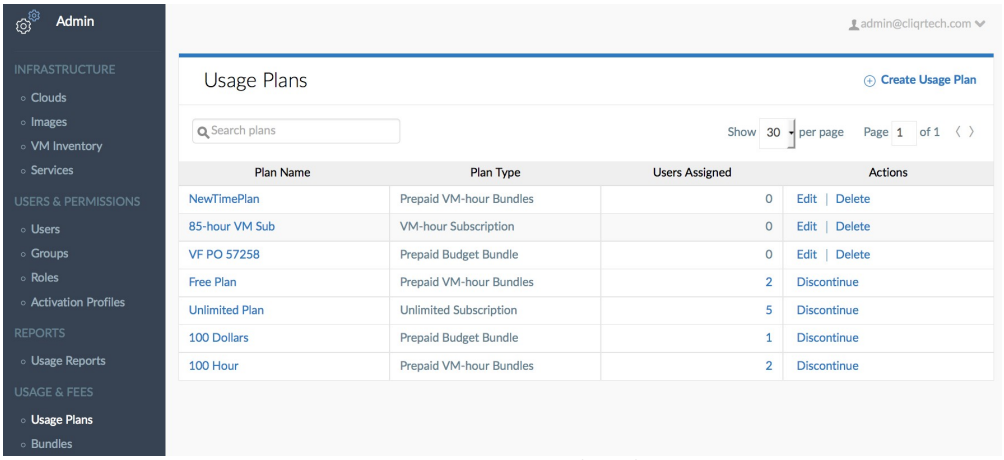
3. デフォルトのクレデンシャルを変更します。

> ログイン後は、すぐにこのデフォルトのクレデンシャルを変更してください。

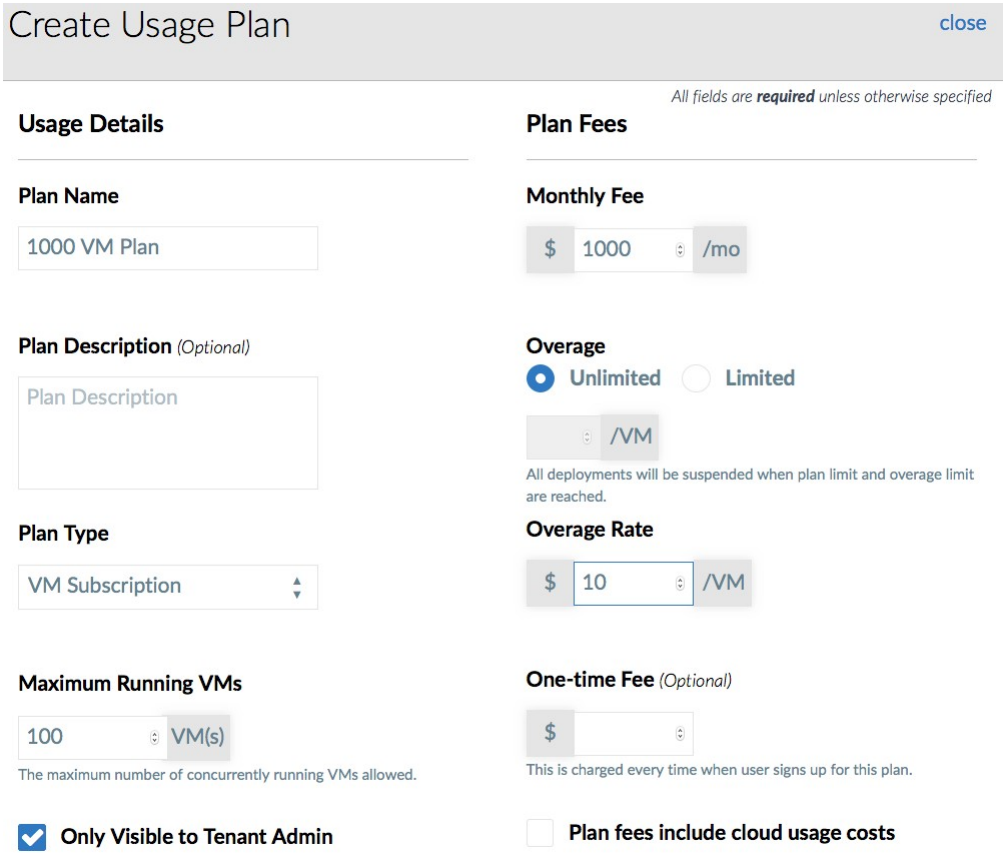
管理者用使用計画の作成

追加情報については、「Usage Plans and Fees」を参照してください。
プラットフォーム管理者用の使用計画を設定するには、次のプロセスに従います。

1. 左ペインで、[Admin] リンクをクリックします。
2. 左ペインの [Usage Plans] をクリックして [Usage Plans] ページにアクセスします。



3. [Create Usage Plan] をクリックして、[Create Usage Plan] ポップアップを開きます。



4. 次のフィールドに、CloudCenter の契約で指定したとおりに入力します。
   a. [Usage Details] セクションに、以下の情報を入力します。
      i. [Plan Name] と [Description] を入力します。
      ii. [Plan Type] フィールドで [VM Subscription] を選択します。
      iii. [Monthly VMs] フィールドで、1 ヵ月に同時に実行させる VM の数を設定します。
   b. [Only Visible to Tenant Admin] チェックボックスをオンに(この設定をルート プラットフォーム管理者だけに適用するように)して計画をテナント管理者だけに表示するようにします。これにより、この管理者は、追加のユーザに対して他の計画をセットアップできるようになります。[Plan Costs] セクションに、以下の情報を入力します。
      i. 計画の [Base Price] を設定します。
      ii. 他の VM の [Overage Rate] を設定します。
      iii. プラットフォーム管理者に [Unlimited] タイプを割り当て、プラットフォーム管理者が制限なしに専用のインストールを管理できるようにします。[Overage Limit] で [Limited] を選択する場合は、計画を制限する VM の数を指定します。
      iv. 必要に応じて、[One-time Fee] および [Storage Rate] を追加します。
      v. [Payment profile is required] のチェックを外します。
   c. [Save] をクリックします。[Usage Plans] ページに、この新しいプランが正常に追加されたことが表示されます。

管理者用契約の作成

管理者を含みすべてのユーザは、CCM UI を使用して続行するには、契約が必要です。各ユーザは、CCM UI を使用し続けるには、契約に同意する必要があります。

管理者用の契約を作成するには、次の手順を実行します。

1. 左ペインで、[Admin] リンクをクリックします。
2. 左ペインの [Contracts] をクリックして [Contracts] ページにアクセスします。
3. [Create Contract] をクリックして、[Create New Contract] ポップアップを開きます。
4. 次のフィールドに、CloudCenter の契約で指定したとおりに入力します。

> ⚠ 現在の CloudCenter ライセンス契約書は、信頼に基づいたものです。

5. 契約書に必要な情報を入力してください。

> ✓ [Only Visible to Admin] チェックボックスをオン（設定をルート プラットフォーム管理者に対してのみ適用）にして計画を管理者だけに表示するようにします。

6. [Submit] をクリックします。

管理者への計画の割り当て

計画を管理者またはユーザに割り当てるには、次の手順に従います。

1. この計画を CloudCenter のプラットフォーム管理者に割り当てるには、左ペインの [Users] タブをクリックし、[User Management] ページにアクセスします。

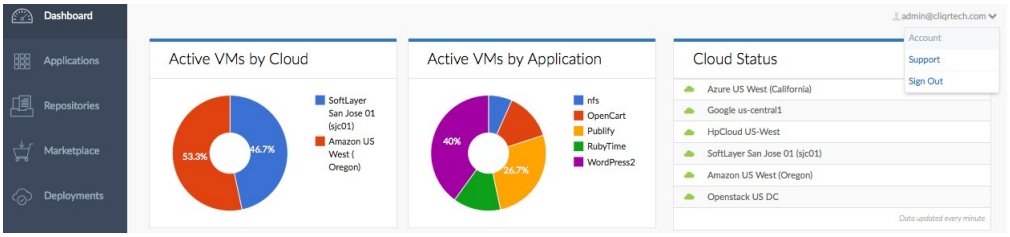
2. CloudCenter Admin（admin@cliqrtech.com）を検索し、[Actions] ドロップダウン リストから [Manage Plans] を選択します。
3. 作成した計画を検索し、このプラットフォーム管理者に計画を割り当てます。
4. root とも呼ばれるプラットフォーム管理者（「People」を参照）として CCM UI に再ログインします。ライセンス契約書の条項に同意するよう求められます。

> ⓘ CloudCenter のプラットフォームを効果的に使用し続けるために、必ず、この時点で契約条項に同意してください。

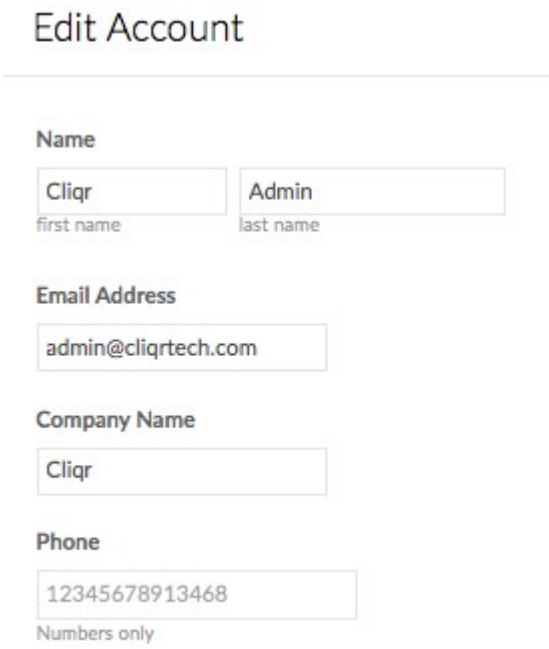
管理者プロファイルの変更

管理者プロファイル（少なくとも、デフォルトのユーザ名とパスワード）を変更するには、次の手順に従います。

1. CCM UI の右上隅に記された管理者の電子メール アドレスをクリックし、[Account] を選択して、このアカウントの [Account Information] ページにアクセスします。

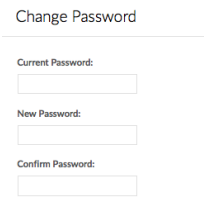


2. [Edit Account] をクリックしてアカウント名、電子メール アドレス、会社名、電話番号を編集します。



3. [Change Password]（[Quick Links] の上）をクリックし、パスワードを変更します。

> **注意**
> CCM UI のアカウント設定のいずれかを変更した場合は、CCM UI からログアウトして、新しいクレデンシャルを使用してログインする必要があります。



これで、CCM および CCO がインストールされ、プラットフォーム管理者（ルート管理者）が追加されました。導入を完了するには、次のセットアップ作業を必ず実行してください。

- クラウドの設定
- イメージ ID の設定
- イメージのマッピング
- CCM による CCO の登録

クラウドの設定

概要

アプリケーションのアプリケーション VM イメージを作成し、イメージ ID を使用してこれらのイメージを参照する必要があります。イメージ ID がない場合、選択したクラウドでイメージ マッピングを設定する際に、クラウドでイメージ ID を参照することができません。

サードパーティ製のイメージに接続するためには、各クラウドの設定に論理マッピングが必要です。詳細は、クラウドによって、また各 CCO を配置する場所によって異なります。各クラウドを論理的にマッピングする必要があります（「Manage Images」を参照）。

注意
プライベート クラウドおよびパブリック クラウドに CloudCenter をインストールする場合は、それぞれのクラウドが互いに双方向接続されていることを確認してください（「Phase 2: Configure Network Rules」を参照）。

クラウドおよびクラウド アカウントの設定

この項では、3 つの異なるクラウドの、クラウド設定の概要とウォークスルー の例を説明します。これらの手順を、CloudCenter 導入時のクラウド設定のガイドラインとして使用してください。これらのクラウド設定の詳細が得られた後は、CCM UI で適用可能なクラウド設定にマッピングします。

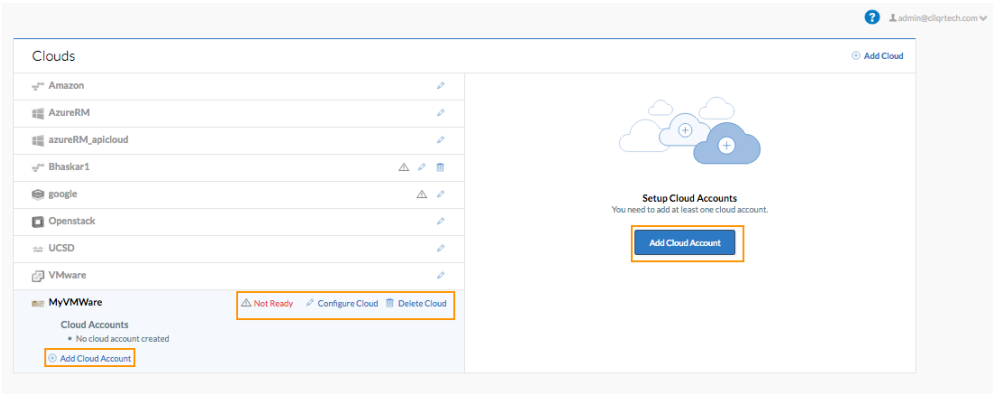
クラウド内の主なコンポーネント間の通信を可能にするには、CCM UI の [Cloud] ページで、クラウド設定に論理マッピングの詳細を適用する必要があります。

- VMware クラウドでの論理マッピング
  - VMware クラウドの設定

    VMware クラウドで論理マッピングを設定するには、次のプロセスに従います。

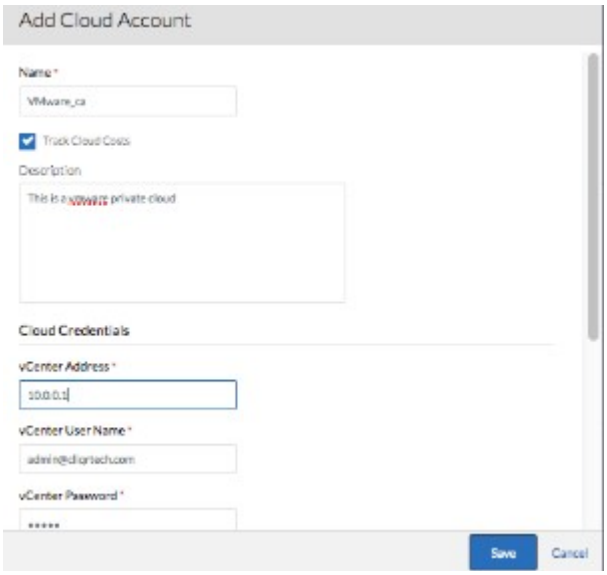
    1. 新しいクレデンシャルを使用して、CloudCenter UI にログインします。
    2. 左ペインに表示されている [Admin] リンクをクリックします。
    3. サイド パネルで [Clouds] をクリックし、[Clouds] ページを表示します。
       a. クラウドを追加するには、[Add Cloud] をクリックし、**VMware** クラウド ディレクタまたは VMware プライベート クラウド ファミリを選択してから、このクラウドの名前と説明を入力し、[Save] をクリックします。

       b. 新しいクラウドを追加すると、**VMware** クラウド ファミリを含む行に鉛筆アイコン、または [Add Cloud Account] ボタンが自動的に表示され、これを使用してこのクラウド ファミリにアカウントを追加できます。



       c. [!] アイコンは、クラウド アカウントが設定されていないことを示します。

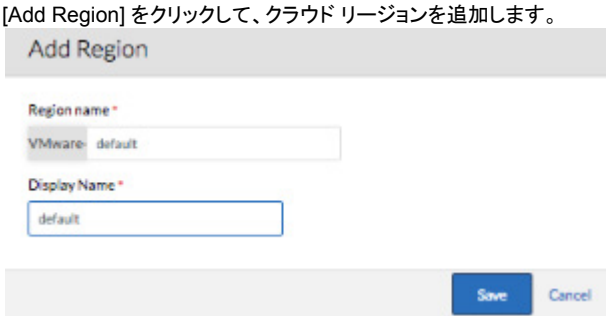
4. [Add Cloud Account] をクリックして、このクラウド ファミリにアカウントを追加します。[Add Cloud Account] ポップアップが表示されます。



> ⚠ vCenter ユーザに完全な管理者権限がない場合は、(CloudCenter を使用する)各ユーザに少なくともアクセス権限があることを確認してください。

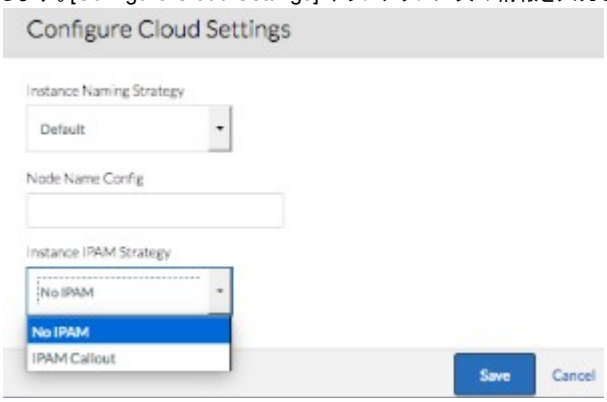
クラウド アカウントの詳細	説明
vCenter Address	このアカウントに設定する IP アドレス。
vCenter User Name/Password	このクラウド アカウントにアクセスするためのクレデンシャル。
VMware NSX	VMware NSX のセキュリティ プロファイルを設定するための、ネットワーク プロトコル プロファイル設定([ON] またはデフォルトの [OFF])。[ON] にした場合は、次のフィールドを設定する必要があります。
Manager Address	NSX マネージャの IP アドレス。
NSX User Name/Password	NSX Manager にアクセスするためのクレデンシャル。

   a. 新しいクラウドに**名前**を割り当てます。
   b. [Track Cloud Costs] チェックボックスをオンにして、クラウド コストを追跡します。
   c. vCenter ユーザの詳細を入力します。
   d. [Save] をクリックし、新しく追加したクラウドが [Cloud Configurations] ページに表示されていることを確認します。
5. クラウド アカウントが正常に追加されたら、[Regions] タブをクリックして、クラウド リージョンを追加します。
6. [Add Region] をクリックして、クラウド リージョンを追加します。



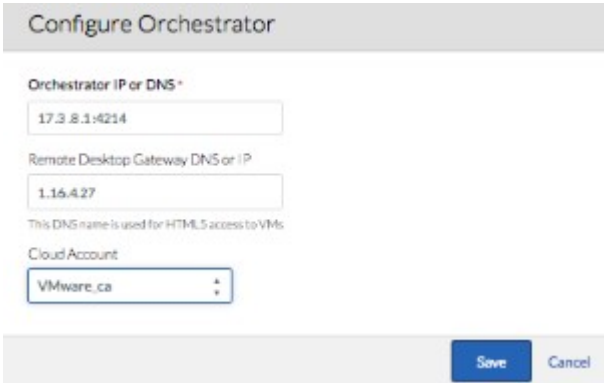
   a. このクラウド リージョンの [Region Name] およびオプションの [Display Name] を追加します。
   b. [Save] をクリックします。新しいリージョンが表示され、ここで、クラウド設定、オーケストレータ(CCO)、およびインスタンス タイプを設定できます。

7. [Edit Cloud Settings] をクリックして CloudCenter の初期設定に基づいてクラウド設定を設定し、VMware クラウドの設定を更新します。[Configure Cloud Settings] ポップアップに次の情報を入力します。



   a. **Instance Naming Strategy**：VM コールアウトのストラテジを指定します。追加情報については、「VM Name Config」を参照してください。
   b. **Node Name Config**：ノード参照を指定します。追加情報については、「VM Name Config」を参照してください。
   c. **Instance IPAM Strategy**：OS カスタマイズ仕様のストラテジを指定します。追加情報については、「VM Name Config」を参照してください。
   d. [Save] をクリックします。

8. [Configure Orchestrator] をクリックしてオーケストレータ（CCO）を設定し、このクラウド アカウントの CCO の詳細を追加します。[Configure Orchestrator] ポップアップに次の情報を入力します。

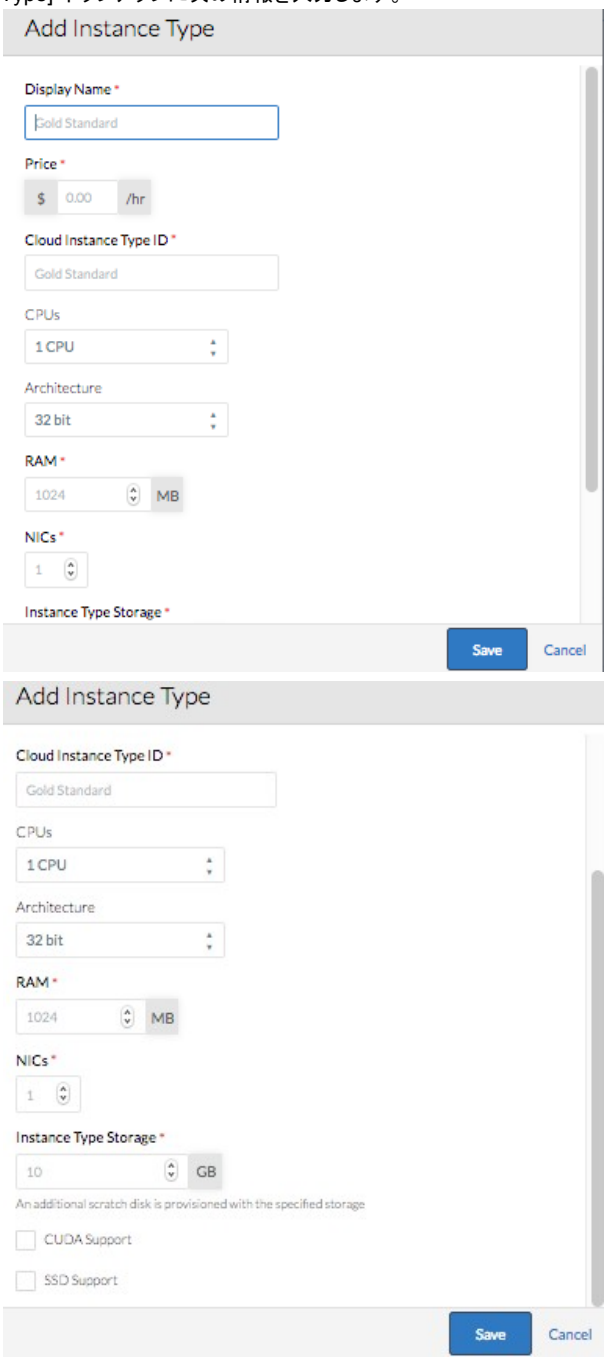


   a. **Orchestrator IP or DNS**：CCO サーバの IP または DNS アドレスを指定します。
   b. **Remote Desktop Gateway DNS or IP**：カスタム SSL 証明書をアップロードするための Guacamole（HTML5 リモート デスクトップ）ゲートウェイ。この IP アドレスは、理想的には、AMQP サーバの IP アドレスと同じにします。追加情報については、「Per CloudCenter Region Installation (Required)」>「CCO」を参照してください。

      [Appliance Setup] オプションを使用している場合は、すぐに使用できる Guacamole 自己署名証明書が作成されます。

   c. **Cloud Account**：[Add Cloud Account] の手順で設定したクラウド アカウントを選択します。
   d. [Save] をクリックします。

9. [Add Instance Type] をクリックして、クラウド リージョンで少なくとも 1 つのインスタンス タイプを設定します。[Add Instance Type] ポップアップに次の情報を入力します。



   a. パブリック クラウドでは、インスタンス タイプはクラウドの定義に基づいて自動的にインポートされます。
   b. プライベート クラウドでは、[Display Name]、[Price]（「price」を参照）、[Cloud Instance Type ID]、[RAM]、[NICs]、[Instance Type Storage]、およびオプションで [CPUs] と [Architecture] の詳細を手動で指定します。詳細については、「cloudInstanceTypes」を参照してください。CloudCenter プラットフォームで定義されているサンプル インスタンス タイプを次に示します。

Instance Type	Name	Hardware Spec	Cost Per Hour	Actions
small	small	2CPUs, 2048GB, 1000Mbps	0.2	Edit | Delete
medium	medium	2CPUs, 4096GB, 1000Mbps	0.3	Edit | Delete
large	large	4CPUs, 8192GB, 1000Mbps	0.6	Edit | Delete
xlarge	xlarge	4CPUs, 16384GB, 1000Mbps	1.2	Edit | Delete
2xlarge	2xlarge	8CPUs, 32768GB, 1000Mbps	2.4	Edit | Delete

c. [Save] をクリックします。導入のためにインスタンス タイプを追加すると、[Configure Instance Types] セクションに設定済みのインスタンスが表示されるようになります。

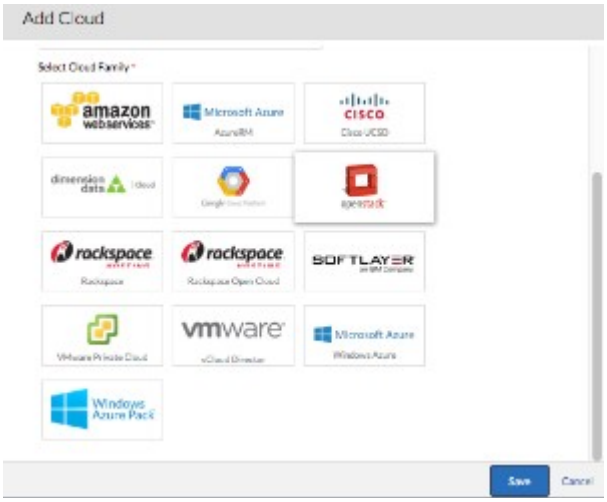
> **CloudCenter イメージで検出された変更**
> CCM により CCO を登録すると、検出された詳細が [Instance Types] セクションに***自動的に***表示されます。
>
> インスタンス タイプまたはイメージ マッピングのいずれかまたは両方で、CloudCenter ソフトウェアが***変更を検出する場合もあります***。必要な場合は次の手順を実行します。
>
> - イメージ マッピングの同期が必要になることがあります。詳細は、「イメージのマッピング」>「*イメージ マッピングの同期*」を参照してください。
> - インスタンス タイプの同期が必要になることがあります。詳細は、「インスタンス タイプの管理」>「*インスタンス タイプの同期*」を参照してください。

これで、VMware クラウドが設定されました。

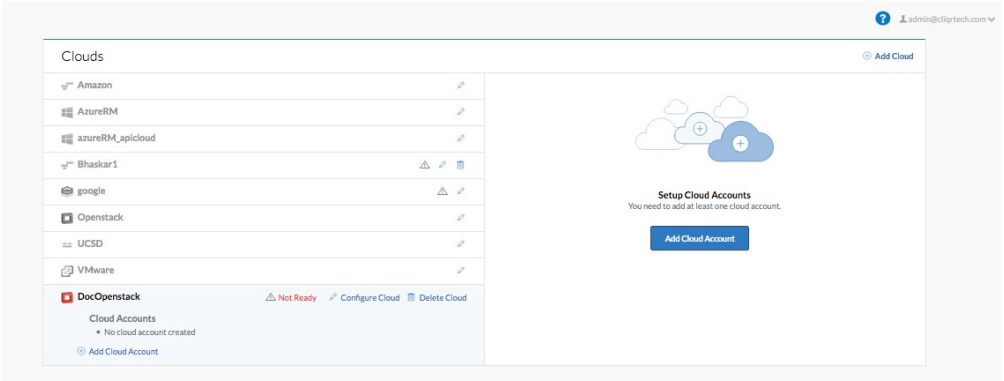
- OpenStack クラウドでの論理マッピング
  - OpenStack クラウドの設定

    OpenStack クラウドで論理マッピングを設定するには、次の手順に従います。

    1. 新しいクレデンシャルを使用して、CloudCenter UI にログインします。
    2. 左ペインに表示されている [Admin] リンクをクリックします。
    3. サイド パネルで [Clouds] をクリックし、[Clouds] ページを表示します。
       a. クラウドを追加するには、[Add Cloud] をクリックし、OpenStack クラウド ファミリを選択してから、このクラウドの名前と説明を入力し、[Save] をクリックします。

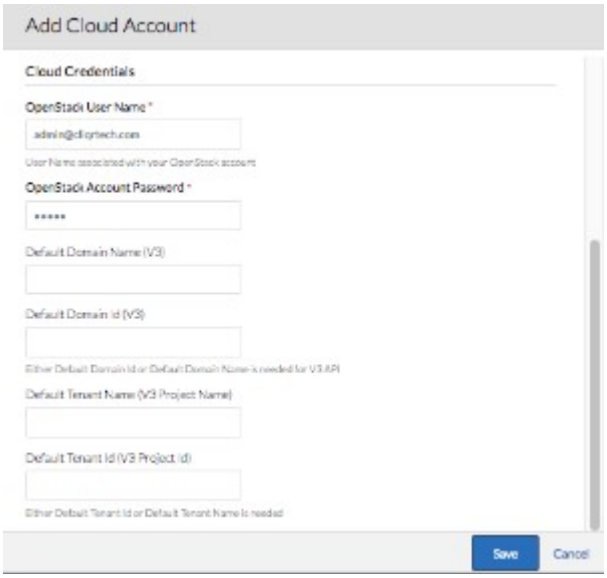


       b. 新しいクラウドを追加すると、**OpenStack** クラウド ファミリを含む行に鉛筆アイコン、または [Add Cloud Account] ボタンが自動的に表示され、これを使用してこのクラウド ファミリにアカウントを追加できます。



       c. [!] アイコンは、クラウド アカウントが設定されていないことを示します。[Add Cloud Account] をクリックして、このクラウド ファミリにアカウントを追加します。[Add Cloud Account] ポップアップが表示されます。

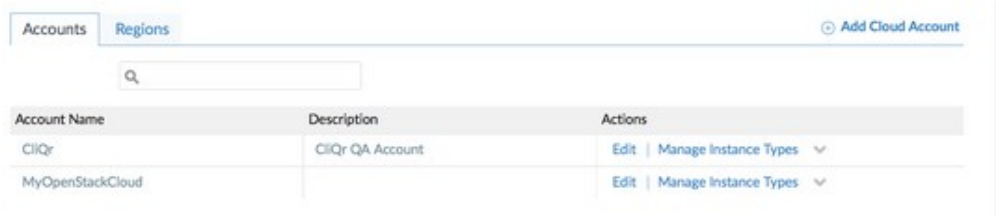
4. [Add Cloud Account] をクリックして、このクラウド ファミリにアカウントを追加します。[Add Cloud Account] ポップアップが表示されます。



> ⚠ OpenStack ユーザに完全な管理者権限がない場合は、(CloudCenter を使用する)各ユーザに少なくともアクセス権限があることを確認してください。

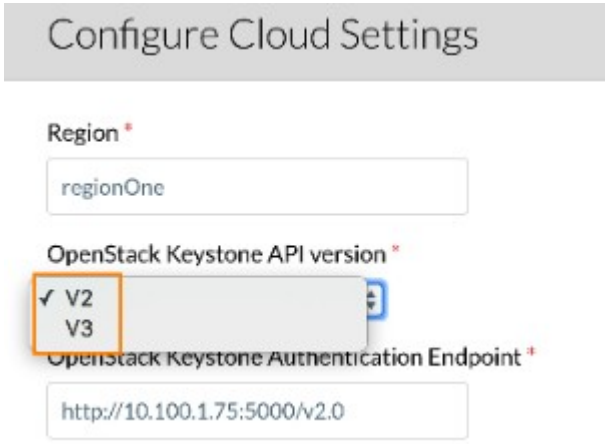
クラウド アカウントの詳細	説明
OpenStack User Name/Password	このクラウド アカウントにアクセスするためのクレデンシャル。
Default Domain Name (V3)	オプション。OpenStack クラウド アカウントの追加時には、V2 または V3 OpenStack エンドポイントを選択できます。 • V2 を使用する場合は必須ではありません。 • V3 を使用する場合は、デフォルトのドメイン ID またはデフォルトのドメイン名を入力します。 • クラウド リージョンの設定によりリージョンが検証されます。
Default Domain ID (V3)	
Default Tenant Name (V3 Project Name)	オプション。OpenStack のプロジェクト名。
Default Domain ID (V3 Project ID)	オプション。設定すると、[Default Tenant ID](CloudCenter での OpenStack 設定)が [Default Tenant Name] よりも優先されます。

   a. 新しいクラウドに**名前**を割り当てます。
   b. [Track Cloud Costs] チェックボックスをオンにして、クラウド コストを追跡します。
   c. OpenStack のユーザ クレデンシャルを入力します。
   d. オプション。V3 API を使用する OpenStack ドメインを設定するには、ドメイン ID とドメイン名を追加します。V3 API ドメインを選択する場合は、ドメイン ID またはドメイン名を入力します。
   e. [Save] をクリックし、新しく追加したクラウド アカウント(用語の詳細な説明は「Cloud」を参照)が [Cloud] ページに表示されていることを確認します。
   f. [Configure Cloud] リンクをクリックします。[Cloud Accounts] ページの [Accounts] タブに、設定したクラウド アカウントが表示されます。



5. [Regions] タブをクリックします。
6. [Add Regions] リンクをクリックします。
   a. OpenStack インフラストラクチャ環境での物理リソースの地理的グループ化を説明する [Region Name] を追加します。
   b. この地理的グループ化で展開する [Display Name] を追加します。

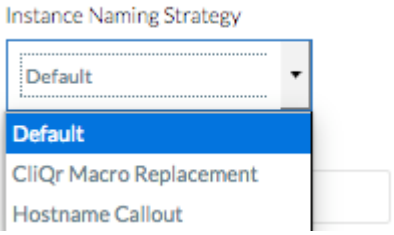
7. [Edit Cloud Settings] リンクをクリックし、このクラウド リージョンの設定を更新します。
   - 必須の設定は次のとおりです。
     - **Region**:(リージョンは有効な OpenStack リージョンにする必要があります)
     - **OpenStack Keystone API Version**:(ドロップダウン リストには、v2 と v3 の両方のオプションが含まれています。これらのオプションのいずれかを選択します)



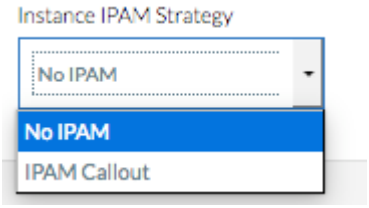
     - **OpenStack Keystone Authentication Endpoint**:(この設定は [Cloud Account] ポップアップでは使用できません。[Cloud Region] ポップアップでのみ使用できます)
   - オプションの設定

> ⚠ クラウド リージョンのコールアウトまたは属性に変更を加えた場合は、変更を有効にするために CCO を再起動する必要があります。

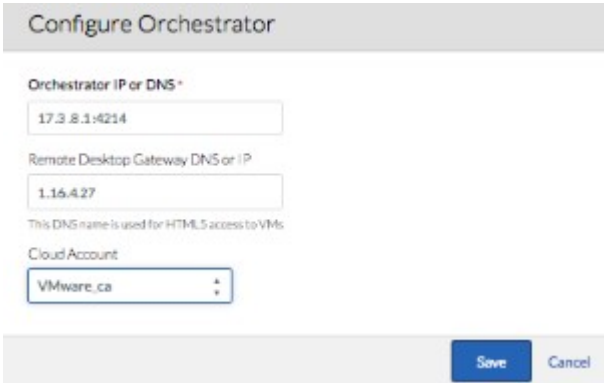
   a. **Instance Naming Strategy**:VM コールアウトのストラテジを指定します。追加情報については、「VM Name Config」を参照してください。



   b. **Node Name Config**:ノード参照を指定します。追加情報については、「VM Name Config」を参照してください。
   c. **Instance IPAM Strategy**:OS カスタマイズ仕様のストラテジを指定します。追加情報については、「VM Name Config」を参照してください。



   d. [Save] をクリックします。
8. [Configure Orchestrator] をクリックしてオーケストレータ(CCO)を設定し、このクラウド アカウントの CCO の詳細を追加します。[Configure Orchestrator] ポップアップに次の情報を入力します。

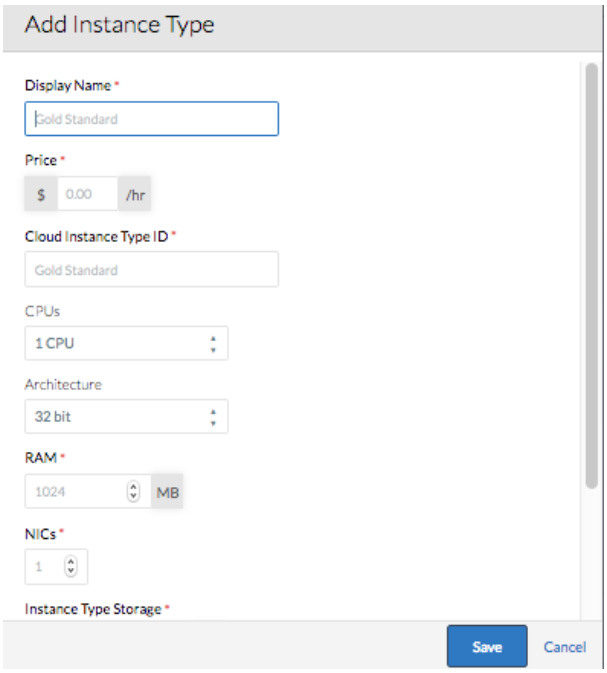


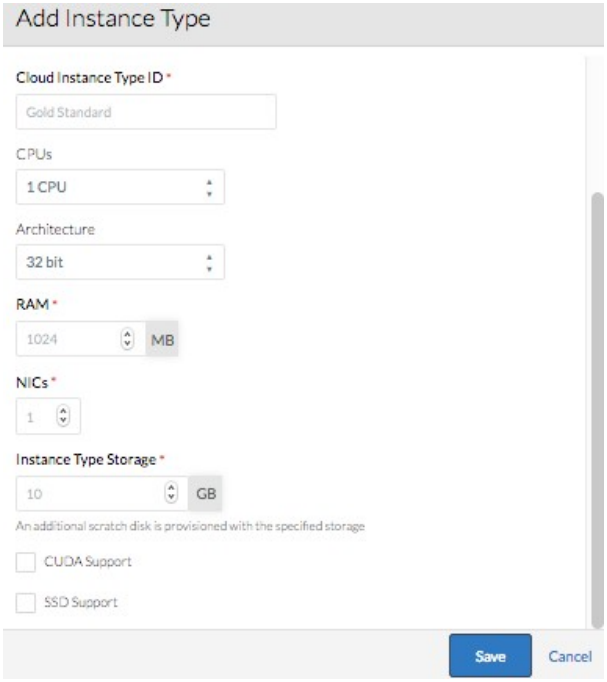
a. **Orchestrator IP or DNS**：CCO サーバの IP または DNS アドレスを指定します。
b. **Remote Desktop Gateway DNS or IP**：カスタム SSL 証明書をアップロードするための Guacamole（HTML5 リモート デスクトップ）ゲートウェイ。この IP アドレスは、理想的には、AMQP サーバの IP アドレスと同じにします。追加情報については、「Per CloudCenter Region Installation (Required)」>「CCO」を参照してください。

> [Appliance Setup] オプションを使用している場合は、すぐに使用できる Guacamole 自己署名証明書が作成されます。

c. **Cloud Account**：[Add Cloud Account] の手順で設定したクラウド アカウントを選択します。
d. [Save] をクリックします。

9. [Add Instance Type] をクリックして、クラウド リージョンで少なくとも 1 つのインスタンス タイプを設定します。[Add Instance Type] ポップアップに次の情報を入力します。





a. パブリック クラウドでは、インスタンス タイプはクラウドの定義に基づいて自動的にインポートされます。
b. プライベート クラウドでは、[Display Name]、[Price]（「price」を参照）、[Cloud Instance Type ID]、[RAM]、[NICs]、[Instance Type Storage]、およびオプションで [CPUs] と [Architecture] の詳細を手動で指定します。詳細については、「cloudInstanceTypes」を参照してください。OpenStack プライベート クラウドでは、コマンドライン（nova flavor-list）から nova クライアントを実行して、すべてのインスタンス タイプ（フレーバー）の情報を取得します。

```
root@test1:~# nova flavor-list
+--------------------------------------+-----------+-----------+------+-----------+--------+-------+-------------+-----------+
| ID                                   | Name      | Memory_MB | Disk | Ephemeral | Swap_MB | VCPUs | RXTX_Factor | Is_Public |
+--------------------------------------+-----------+-----------+------+-----------+--------+-------+-------------+-----------+
| 00487266-7ba9-4652-8ad9-f57c78c2b073 | m1.small  | 2048      | 20   | 0         |        | 1     | 1.0         | True      |
| 5a22cd16-c5c4-4fe1-b3e5-9e0e8b720dbd | m1.tiny   | 1024      | 10   | 0         |        | 1     | 1.0         | True      |
| 99ff730c-1161-4d70-92a2-d3dea3a39a88 | m1.medium | 4096      | 30   | 0         |        | 2     | 1.0         | True      |
| c7a73497-df9b-4a82-b433-88932d325e1a | m1.large  | 8192      | 30   | 0         |        | 4     | 1.0         | True      |
+--------------------------------------+-----------+-----------+------+-----------+--------+-------+-------------+-----------+
```

c. [Save] をクリックします。導入のためにインスタンス タイプを追加すると、[Configure Instance Types] セクションに設定済みのインスタンスが表示されるようになります。

> **CloudCenter イメージで検出された変更**
> CCM により CCO を登録すると、検出された詳細が [Instance Types] セクションに***自動的に***表示されます。
>
> インスタンス タイプまたはイメージ マッピングのいずれかまたは両方で、CloudCenter ソフトウェアが***変更を検出する場合もあります***。必要な場合は次の手順を実行します。
>
> - イメージ マッピングの同期が必要になることがあります。詳細は、「イメージのマッピング」>「*イメージ マッピングの同期*」を参照してください。
> - インスタンス タイプの同期が必要になることがあります。詳細は、「インスタンス タイプの管理」>「*インスタンス タイプの同期*」を参照してください。

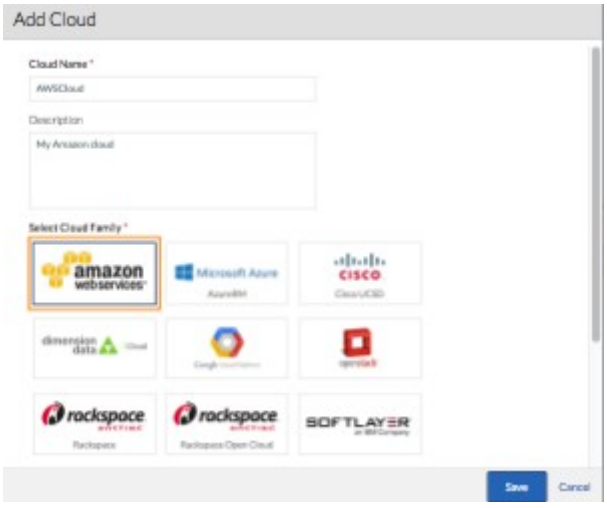
これで、OpenStack クラウドが設定されました。

AWS クラウドでの論理マッピング

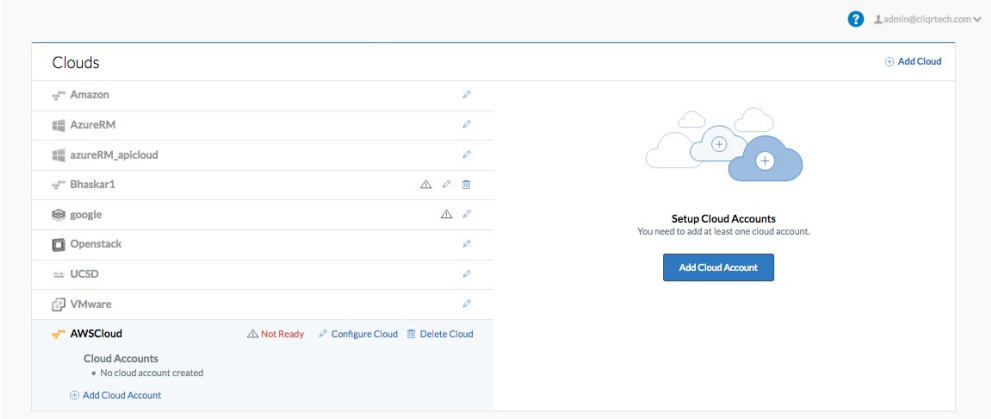
AWS クラウドの設定

Amazon クラウドで論理マッピングを設定するには、次のプロセスに従います。

1. 新しいクレデンシャルを使用して、CloudCenter UI にログインします。
2. 左ペインに表示されている [Admin] リンクをクリックします。
3. サイド パネルで [Clouds] をクリックし、[Clouds] ページを表示します。
   a. クラウドを追加するには、[Add Cloud] をクリックし、**Amazon Web Services**（AWS）クラウド ファミリを選択してから、このクラウドの名前と説明を入力し、[Save] をクリックします。

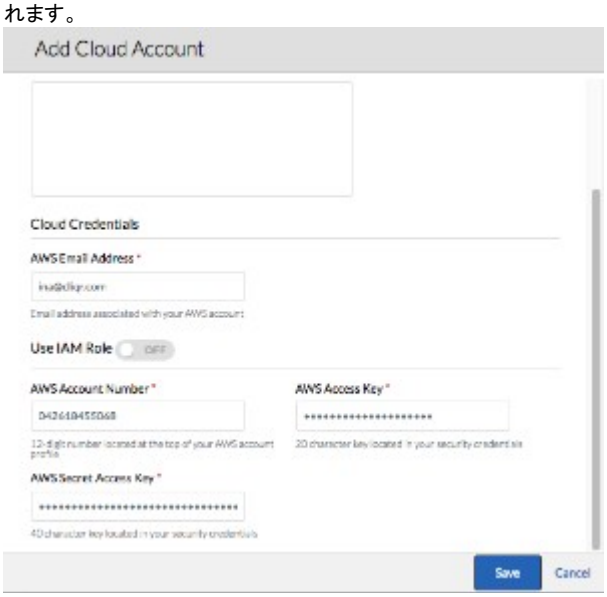


b. 新しいクラウドを追加すると、**AWS** クラウド ファミリを含む行に鉛筆アイコン、または [Add Cloud Account] ボタンが自動的に表示され、これを使用してこのクラウド ファミリにアカウントを追加できます。



c. [!] アイコンは、クラウド アカウントが設定されていないことを示します。

4. [Add Cloud Account] をクリックして、このクラウド ファミリにアカウントを追加します。[Add Cloud Account] ポップアップが表示されます。

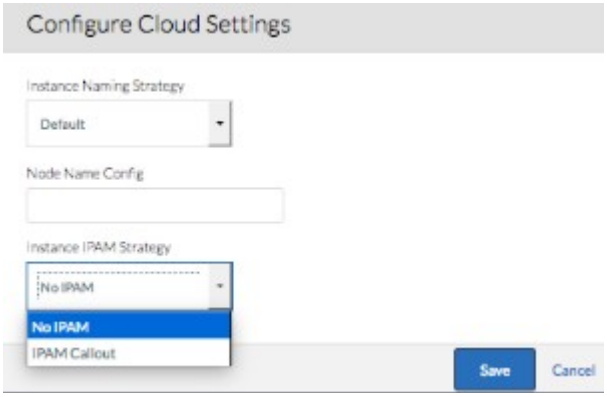


AWS ユーザに完全な管理者権限がない場合は、(CloudCenter を使用する)各ユーザに少なくともアクセス権限があることを確認してください。

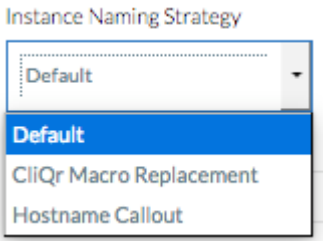
クラウド アカウントの詳細	説明
AWS Email Address	AWS クラウド ファミリのアカウントに関連付けられている電子メール アドレス。
Use IAM Role	[ON] または [OFF](デフォルト)を選択します。アクセス キーや秘密キーなどを指定する代わりに、IAM ロールを使用してインスタンス タイプを管理できます。追加情報については、「AWS Configurations」>「*IAM and STS*」を参照してください。
AWS Account Number	使用中の AWS アカウントのアカウント番号。
AWS Access Key	この AWS アカウントにアクセスするセキュリティ クレデンシャルは、たとえば `AKIAIOSFODNN7EXAMPLE` です。追加情報については、AWS マニュアルを参照してください。
AWS Secret Access Key	この AWS アカウントにアクセスするためのセキュリティ クレデンシャルは、たとえば `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY` です。追加情報については、AWS マニュアルを参照してください。

a. 新しいクラウドに**名前**を割り当てます。
b. [Track Cloud Costs] チェックボックスをオンにして、クラウド コストを追跡します。
c. vCenter ユーザの詳細を入力します。
d. [Save] をクリックし、新しく追加したクラウドが [Cloud Configurations] ページに表示されていることを確認します。

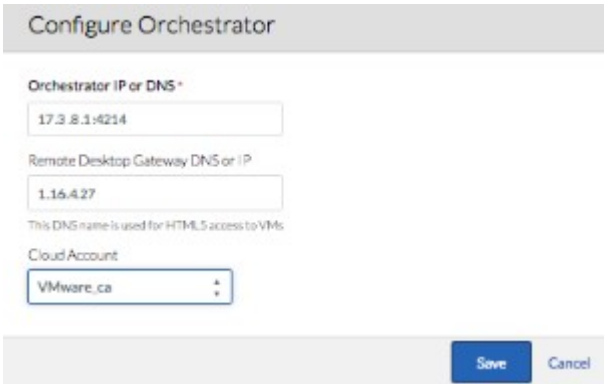
5. クラウド アカウントが正常に追加されたら、[Regions] タブをクリックして、クラウド リージョンを追加します。
6. [Add Region] をクリックして、クラウド リージョンを追加します。
   a. このクラウド リージョンの [Region Name] およびオプションの [Display Name] を追加します。
   b. [Save] をクリックします。新しいリージョンが表示され、ここで、クラウド設定、オーケストレータ（CCO）、およびインスタンス タイプを設定できます。
7. [Edit Cloud Settings] をクリックして CloudCenter の初期設定に基づいてクラウド設定を設定し、VMware クラウドの設定を更新します。[Configure Cloud Settings] ポップアップに次の情報を入力します。



   a. **Instance Naming Strategy**：VM コールアウトのストラテジを指定します。追加情報については、「VM Name Config」を参照してください。



   b. **Node Name Config**：ノード参照を指定します。追加情報については、「VM Name Config」を参照してください。
   c. **Instance IPAM Strategy**：OS カスタマイズ仕様のストラテジを指定します。追加情報については、「VM Name Config」を参照してください。
   d. [Save] をクリックします。
8. [Configure Orchestrator] をクリックしてオーケストレータ（CCO）を設定し、このクラウド アカウントの CCO の詳細を追加します。[Configure Orchestrator] ポップアップに次の情報を入力します。

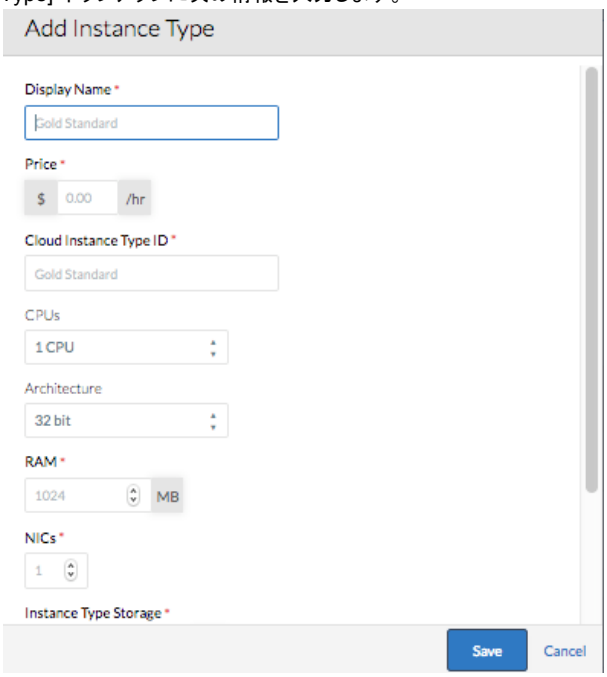


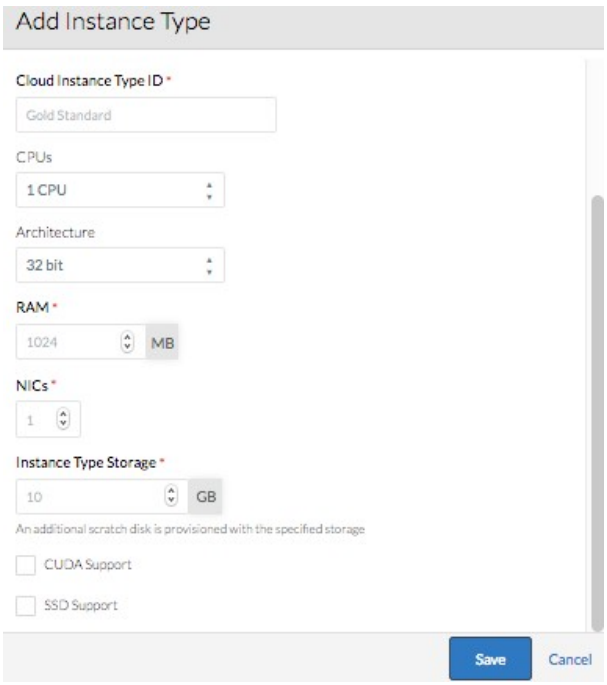
   a. **Orchestrator IP or DNS**：CCO サーバの IP または DNS アドレスを指定します。
   b. **Remote Desktop Gateway DNS or IP**：カスタム SSL 証明書をアップロードするための Guacamole（HTML5 リモート デスクトップ）ゲートウェイ。この IP アドレスは、理想的には、AMQP サーバの IP アドレスと同じにします。追加情報については、「Per CloudCenter Region Installation (Required)」>「CCO」を参照してください。

> [Appliance Setup] オプションを使用している場合は、すぐに使用できる Guacamole 自己署名証明書が作成されます。

c. **Cloud Account**: [Add Cloud Account] の手順で設定したクラウド アカウントを選択します。
d. [Save] をクリックします。

9. [Add Instance Type] をクリックして、クラウド リージョンで少なくとも 1 つのインスタンス タイプを設定します。[Add Instance Type] ポップアップに次の情報を入力します。





a. パブリック クラウドでは、インスタンス タイプはクラウドの定義に基づいて自動的にインポートされます。
b. プライベート クラウドでは、[Display Name]、[Price]（「price」を参照）、[Cloud Instance Type ID]、[RAM]、[NICs]、[Instance Type Storage]、およびオプションで [CPUs] と [Architecture] の詳細を手動で指定します。詳細については、「cloudInstanceTypes」を参照してください。CloudCenter プラットフォームで定義されているサンプル インスタンス タイプを次に示します。

Instance Type	Name	Hardware Spec	Cost Per Hour	Actions
small	small	2CPUs, 2048GB, 1000Mbps	0.2	Edit | Delete
medium	medium	2CPUs, 4096GB, 1000Mbps	0.3	Edit | Delete
large	large	4CPUs, 8192GB, 1000Mbps	0.6	Edit | Delete
xlarge	xlarge	4CPUs, 16384GB, 1000Mbps	1.2	Edit | Delete
2xlarge	2xlarge	8CPUs, 32768GB, 1000Mbps	2.4	Edit | Delete

c. [Save] をクリックします。導入のためにインスタンス タイプを追加すると、[Configure Instance Types] セクションに設定済みのインスタンスが表示されるようになります。

> **CloudCenter *イメージで検出された変更***
> CCM により CCO を登録すると、検出された詳細が [Instance Types] セクションに***自動的に***表示されます。
>
> インスタンス タイプまたはイメージ マッピングのいずれかまたは両方で、CloudCenter ソフトウェアが***変更を検出する場合もあります***。必要な場合は次の手順を実行します。
>
> - イメージ マッピングの同期が必要になることがあります。詳細は、「イメージのマッピング」>「*イメージ マッピングの同期*」を参照してください。
> - インスタンス タイプの同期が必要になることがあります。詳細は、「インスタンス タイプの管理」>「*インスタンス タイプの同期*」を参照してください。

これで、AWS クラウドが設定されました。

- Azure Classic クラウドでの論理マッピング
  - Azure Classic クラウドの設定

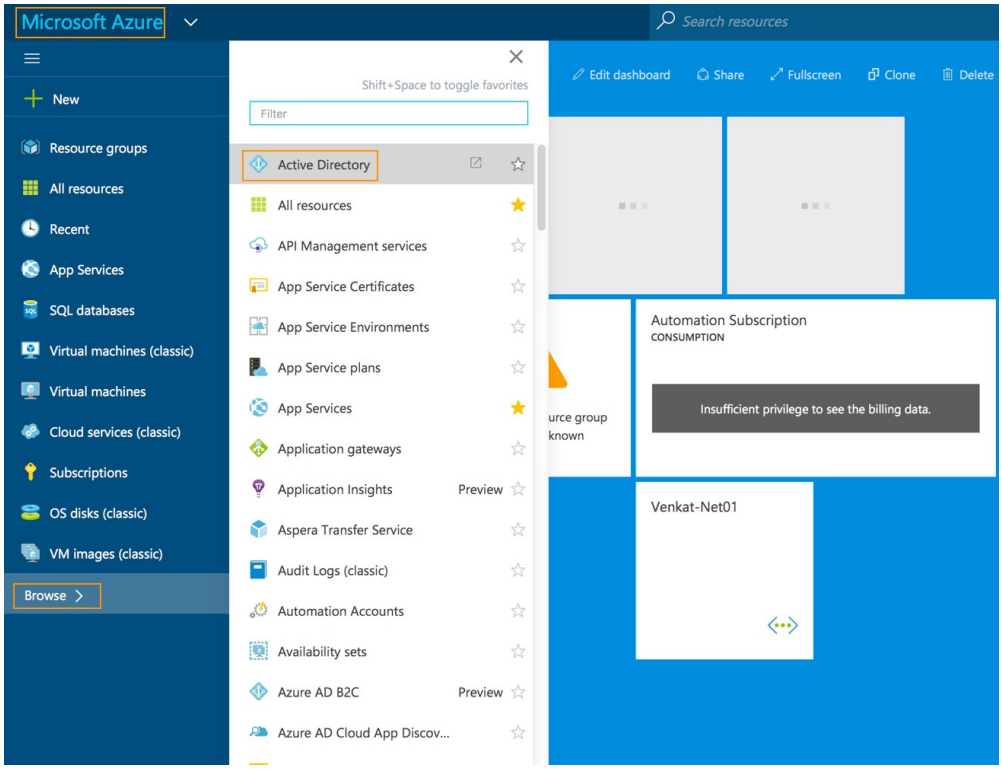
前提条件

> これらのの画面キャプチャは Azure ポータルの変更によって変わる可能性があることに注意してください。この項ではあくまでも参考として示しています。

Azure Classic クラウドをマッピングする前に、次の前提条件を確認してください。

- 有効な Windows Azure アカウントを持っている。
- 登録済みのプロバイダーを確認するのに ARM（本マニュアルでは Azure Resource Manager を意味する）モードを使用しているかどうかを確認します。この手順では、Azure CLI を使用する必要があります。
  1. Azure CLI をプラットフォームにインストールします。追加情報については、https://azure.microsoft.com/en-us/documentation/articles/xplat-cli-install/ を参照してください。
  2. Azure CLI にログインします。
     azure login
  3. ARM モードを使用していることを確認します。
     azure config mode arm
  4. プロバイダーを登録します。
     azure provider register Microsoft.Compute Microsoft.Storage Microsoft.Network Microsoft.Resources Microsoft.Authorization

     > PowerShell を使用している場合は、次のコマンドを使用してプロバイダーを登録します。
     >
     > **Register-AzureProvider**
  5. これらのプロバイダーが登録されているかどうかを確認します。
     azure provider list
- **Azure Classic ポータル**のインターフェイスは Azure Resource Manager のインターフェイスと異なります。左上隅にある [Microsoft Azure] ドロップダウンをクリックして、これらのインターフェイス間の切り替えをする必要があります。CloudCenter で必要な設定の値は、現在、Azure Classic ポータルのインターフェイスより取得できます。Azure Classic ポータルの Active Directory にアクセスするには、次の図を参照してください。



- このアカウントの SSH キーと管理証明書。

> ⚠ MacPort からのものでない OpenSSL、特に OpenSSL 1.0.1f 6 Jan 2014 は、有効な SSH キー ペアを作成しない傾向があることに注意してください。他のオプションとしては、FWIW、Ubuntu 12.04 LTS 上の OpenSSL 1.0.1c 10 May 2012 があります。

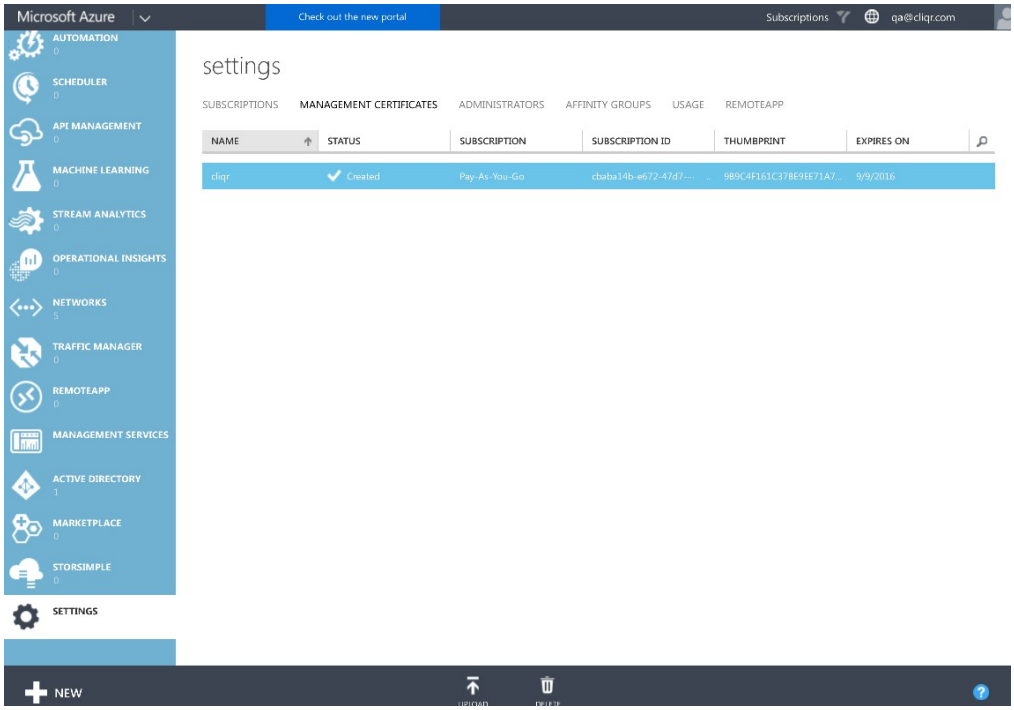
- OpenSSL を使用して SSH キーを生成します。

```
# You may want to use another dir than /tmp
cd /tmp
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout azure-
private.key -out azure-certificate.pem
chmod 600 azure-private.key azure-certificate.pem
openssl x509 -outform der -in azure-certificate.pem -out azure-
certificate.cer
```

- すべての Azure API コール用の証明書で、認証のためにプラグインによって使用されるキーストアを生成します。

```
# Generate a keystore (azurekeystore.pkcs12)
# Transform private key to PEM format
openssl pkcs8 -topk8 -nocrypt -in azure-private.key -inform PEM -out
azure-pk.pem -outform PEM
# Transform certificate to PEM format
openssl x509 -inform der -in azure-certificate.cer -out azure-cert.pem
cat azure-cert.pem azure-pk.pem > azure.pem.txt
# You MUST enter a password!
openssl pkcs12 -export -in azure.pem.txt -out azurekeystore.pkcs12 -name
azure -noiter -nomaciter
```

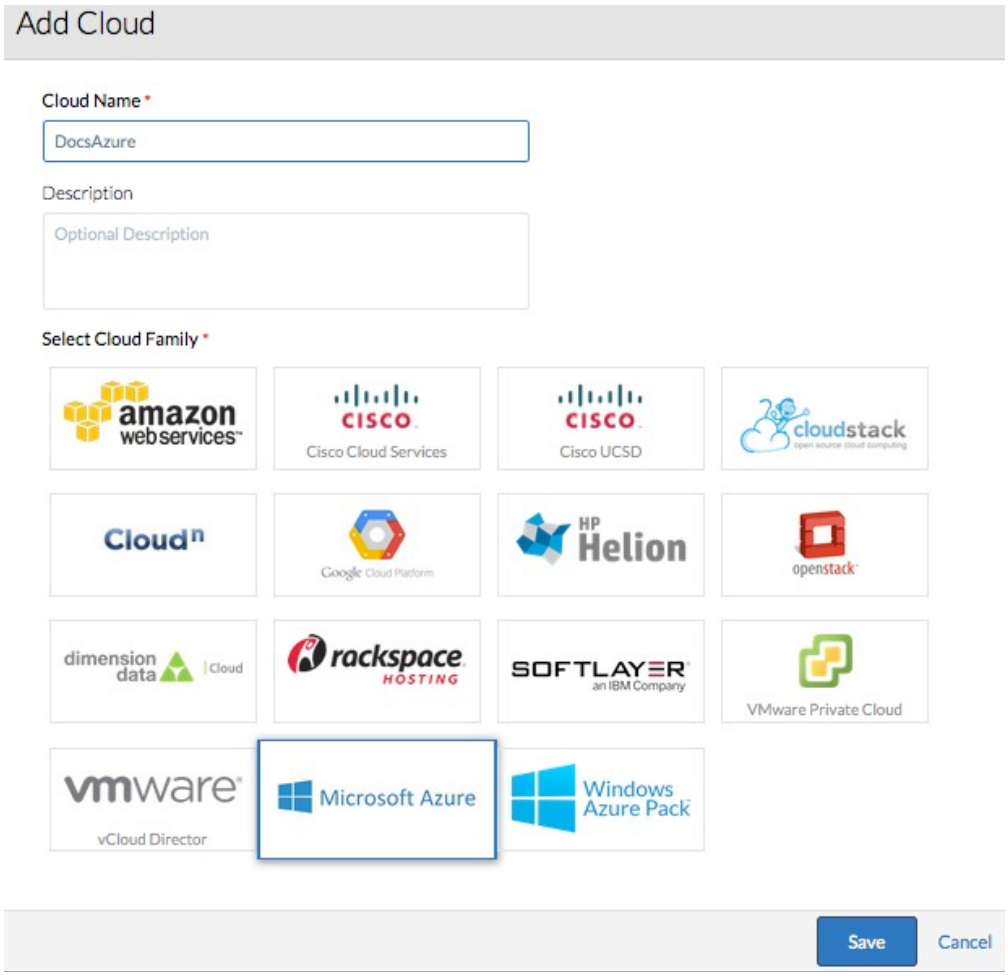
- [Azure Console] > [Settings] > [Manage Certificates] を選択して、azure-certificate.cer ファイル（Cisco 作成）をアップロードします。



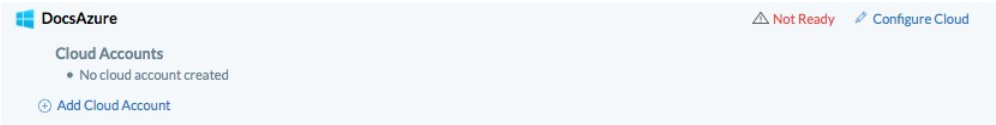
> CCM UI で Azure リージョンを作成する場合、これらのファイルを Azure にアップロードして、ローカル デバイスまたはそこから簡単にファイルをコピーできるデバイスでコマンドを実行できるようにする必要があります。

Azure クラウドで論理マッピングを設定するには、次のプロセスに従ってください。

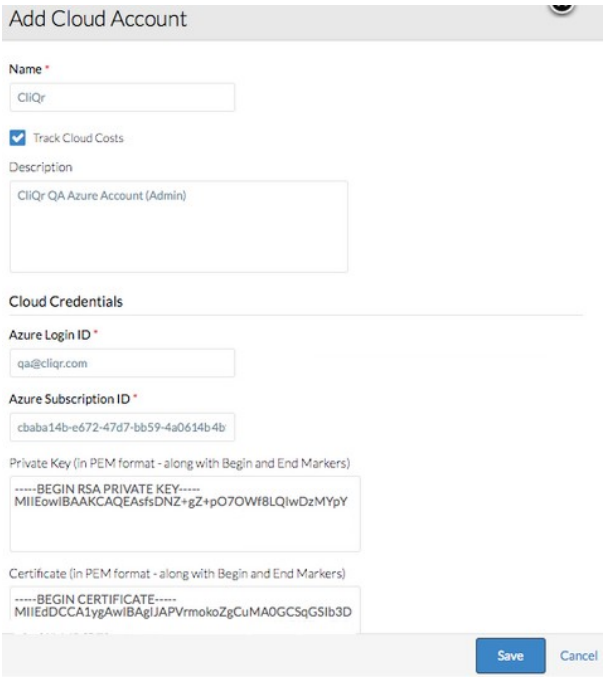
1. 新しいクレデンシャルを使用して、CloudCenter UI にログインします。
2. 左ペインに表示されている [Admin] リンクをクリックします。
3. サイド パネルで [Clouds] リンクをクリックし、[Clouds] ページを表示します。
4. [Add Cloud] をクリックして、選択したクラウドを追加します。



5. 新しく追加したクラウドを見つけるために（必要に応じて）スクロールします。



6. [Add Cloud Account] リンクをクリックします。[Add Cloud Account] ポップアップが表示されます。



a. 新しいクラウドに名前を割り当てます。

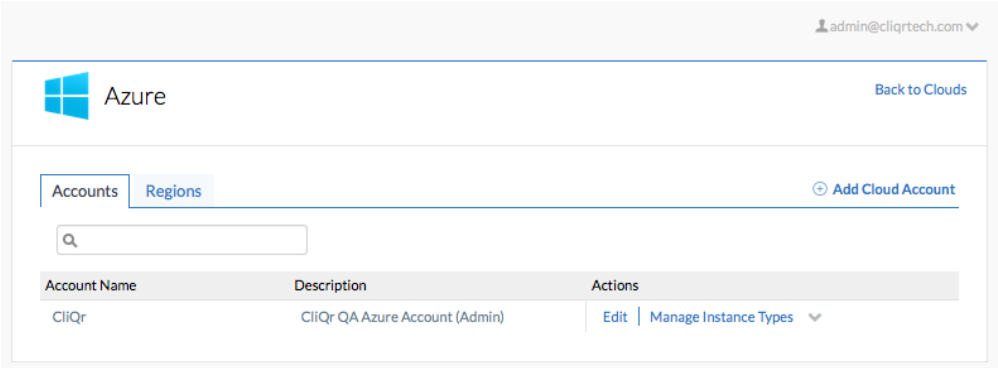
> **ヒント**
> 名前には、スペース、ダッシュ、特殊文字を含めることはできません。

b. Azure アカウントに関連付けられたクラウド クレデンシャル（ログイン ID、サブスクリプション ID、プライベート キー、および証明書）を追加します。この手順の開始時に、「*前提条件*」の項で指定した PEM および証明書のテキストをコピー アンド ペーストします。

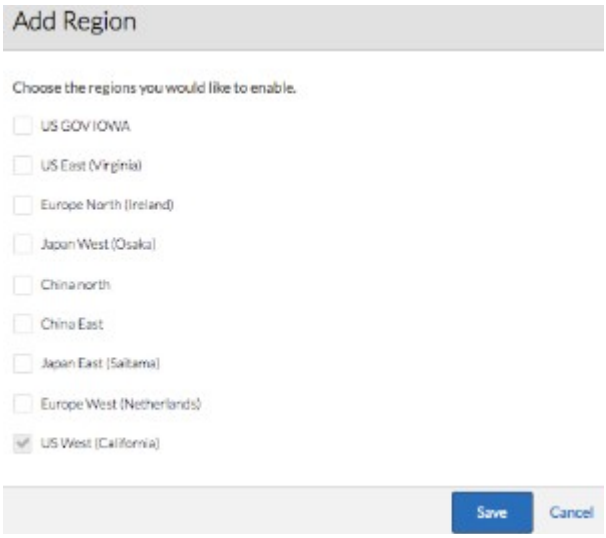
> **サブスクリプション ID** を取得するには、上記の「*前提条件*」の項の説明に従って **Azure Classic ポータル**のインターフェイスに切り替えて、[Settings] にアクセスします。
>
>

c. [Save] をクリックし、新しく追加したクラウド アカウント（用語の詳細な説明は「Cloud」を参照）が [Cloud] ページに表示されていることを確認します。

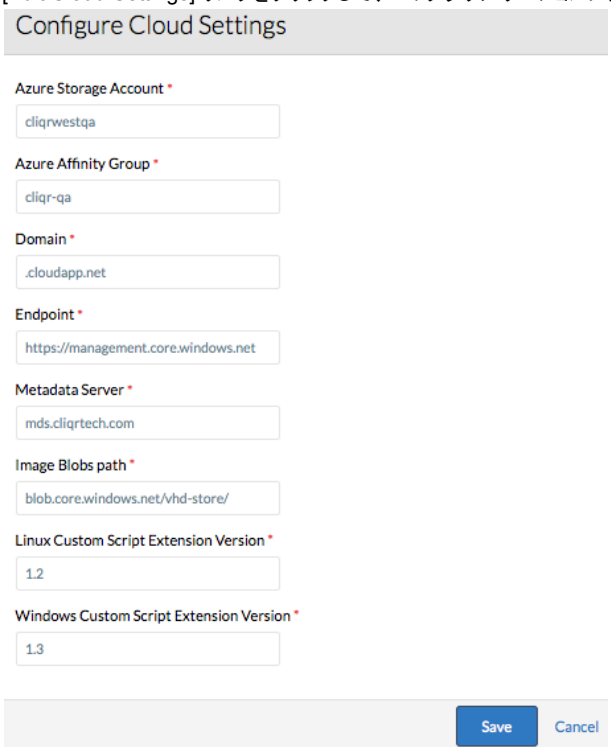
d. [Configure Cloud] リンクをクリックします。[Cloud Accounts] ページの [Accounts] タブに、設定したクラウド アカウントが表示されます。



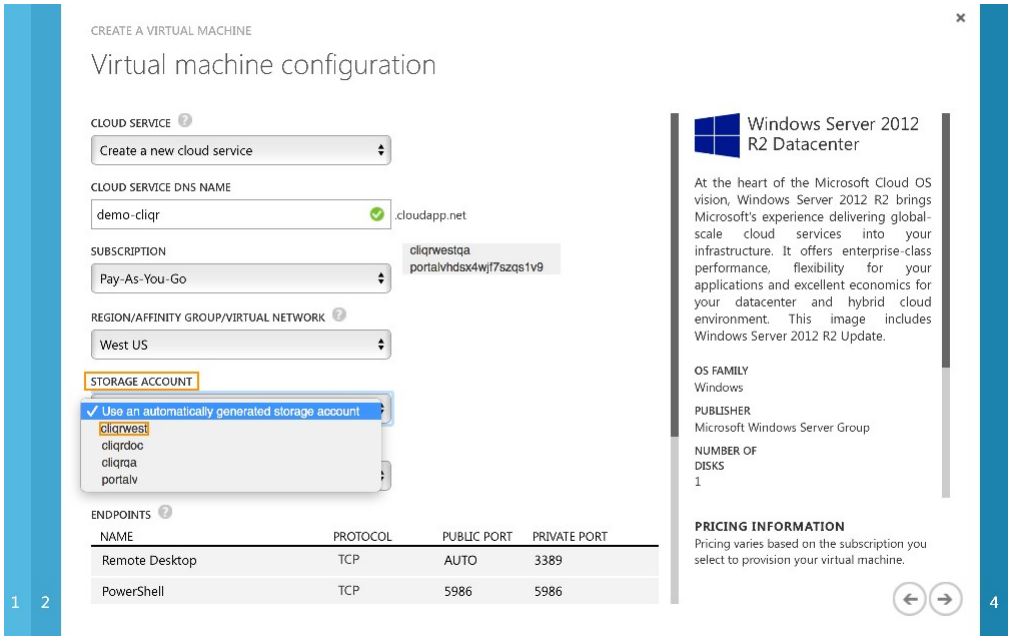
7. [Regions] タブをクリックします。
8. [Add Regions] リンクをクリックし、Azure インフラストラクチャ環境での物理リソースの地理的グループ化を説明する [Region Name] を追加します。



9. [Edit Cloud Settings] リンクをクリックして、このクラウド リージョンに必要な設定を更新します。



- **Azure Storage Account**: 次の Azure、および対応する CloudCenter のイメージの例で示すように、このフィールドには、対応する Azure アカウント(URL フィールド)をセットアップする際に使用された名前と同じ名前が含まれている必要があります。**Azure の設定例**を以下に示します。

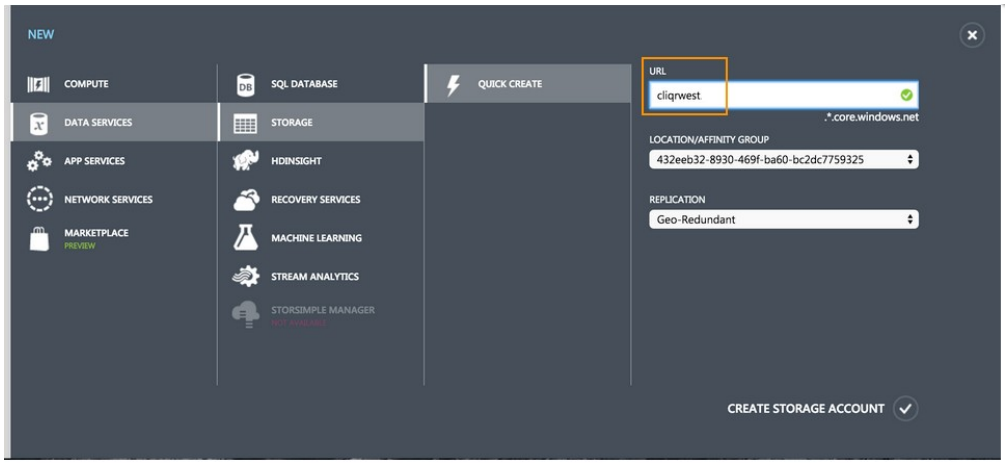


対応する CloudCenter の設定例:

US West (California)

Cloud Settings

Azure Storage Account	cliqrwest
Azure Affinity Group	Joey-Test
Domain	.cloudapp.net
Endpoint	https://management.core.windows.net
Metadata Server	mds.cliqrtech.com
Image Blobs path	blob.core.windows.net/vhd-store/
Linux Custom Script Extension Version	1.2
Windows Custom Script Extension Version	1.3



既存のストレージ アカウントを使用する場合は、**Azure Classic ポータル**の [STORAGE] タブで必要な値を見つけることができます。既存のストレージ アカウントの正確な名前で検索し、このアカウントが CCO と同じクラウド リージョンにあることを確認します。

- **Azure Affinity Group**:新しいアフィニティ グループを作成するか、既存のアフィニティ グループを使用することができます。

既存のアフィニティ グループを使用する場合は、**Azure Classic ポータル**の [SETTINGS] タブで必要な値を見つけることができます。既存のアフィニティ グループの正確な名前で検索し、このグループが CCO と同じクラウド リージョンにあることを確認します。

- **Azure Network**:このネットワークにわかりやすい名前を付けます。この値には、どのような文字列も入力できます。
- **Domain**:パブリック IP でインスタンスを起動すると、Azure はドメインの DNS 名を作成することを許可します。このフィールドに表示されるデフォルト値は .cloud.app.net です。CloudCenter プラットフォームで作成したホスト サービス名をこの値の前に付ける必要があります。設定した Azure ドメインが .cloud.app.net ではない場合、デフォルト値を適切に変更し、その値の前に有効なホスト サービス名を付けます。

既存のドメインを使用する場合は、**Azure Classic ポータル**の [CLOUD SERVICES] タブで必要な値を見つけることができます。必要なクラウド サービスにアクセスし、[Dashboard] をクリックして、DNS 名を検索します。

- **Endpoint**:コールの宛先の API エンドポイント。このコールを行うために使用される CloudCenter REST API の URL。ほとんどの場合(営利目的の設定、中国を除く設定の場合)、この値をそのまま使用できます。この値は、新しい政府リージョンや中国内のリージョンに必要な変更に対応するよう設定することができます。
- **Metadata Server**:このフィールドは、ノードに関する追加情報(CCO IP アドレス、その他の詳細)を提供します。このフィールドは、主にブートストラッピングのために使用します。企業は、アプリケーション VM と CCO にアウトバウンド インターネット アクセスがあることを確認する必要があります。または、企業は、この通信を行うためのサーバを社内に設定できます。

- **Image Blobs path**：Microsoft Storage のアカウント名（OS ドライブおよびデータ ディスクドライブの情報が保存されている場所）。このフィールドには、デフォルトで、デフォルトの URL が表示されます。URL がこのフィールドに表示されるデフォルト URL と異なる場合は、URL を blob.core.windows.net/<Azure_Storage_Account_name> に更新してください。ほとんどの場合（営利目的の設定、中国を除く設定の場合）、この値をそのまま使用できます。この値は、新しい政府リージョンや中国内のリージョンに必要な変更に対応するよう設定することができます。
- **Linux Custom Script Extension Version**：エージェントのないノードで自動ブートストラップを実行するために CloudCenter プラットフォームにより使用される Microsoft Azure プラグイン バージョン（追加情報については、「On a Windows Image」を参照）。ほとんどの場合（営利目的の設定、中国を除く設定の場合）、この値をそのまま使用できます。この値は、新しい政府リージョンや中国内のリージョンに必要な変更に対応するよう設定することができます。
- **Windows Custom Script Extension Version**：エージェントのないノードで自動ブートストラップを実行するために CloudCenter プラットフォームにより使用される Microsoft Windows プラグイン バージョン（追加情報については、「On a Windows Image」を参照）。ほとんどの場合（営利目的の設定、中国を除く設定の場合）、この値をそのまま使用できます。この値は、新しい政府リージョンや中国内のリージョンに必要な変更に対応するよう設定することができます。

10. [Save] をクリックし、変更内容をクラウド リージョンに保存します。
11. [Add Instance Type] リンクをクリックして、クラウド リージョンのインスタンス タイプを追加します。導入のための新しいインスタンス タイプを追加すると、[Configure Instance Types] セクションに設定済みのインスタンスが表示されるようになります。

> **注意**
> [Configure Image Mapping] セクションに、使用可能なイメージ名が自動的に入力されます。ただし、イメージ マッピングを設定する前に、まず、CCO VM を設定し、CCM により CCO を登録する必要があります。

これで、Azure クラウドが設定されました。

- Azure Resource Manager クラウドでの論理マッピング
  - Azure Resource Manager クラウドの設定

前提条件

> これらのの画面キャプチャは Azure ポータルの変更によって変わる可能性があることに注意してください。この項ではあくまでも参考として示しています。

Azure Resource Manager クラウドをマッピングする前に、次の要件を確認してください。

- 有効な Windows Azure Resource Manager アカウントを持っている。
- Azure CLI の ARM モードにログインし、必要な Azure プロバイダーを登録します。

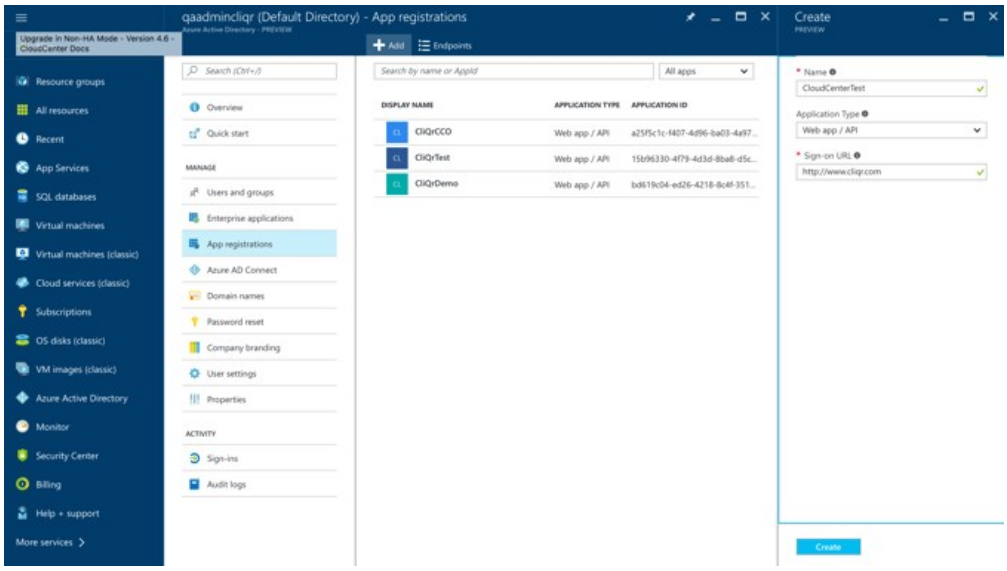
> この手順は、Azure CLI の使用によってのみ実行できます。

1. Azure CLI をプラットフォームにインストールします。追加情報については、https://azure.microsoft.com/en-us/documentation/articles/xplat-cli-install/ を参照してください。
2. Azure CLI にログインします。
   azure login
3. ARM モードを使用していることを確認します。
   azure config mode arm
4. 次のように、Azure プロバイダーを登録します。
   azure provider register Microsoft. Compute Microsoft.Storage Microsoft. Network Microsoft. Resources Microsoft. Authorization

   > PowerShell を使用している場合、次のコマンドを発行します。
   > **Register-AzureProvider**

5. Azure プロバイダーが登録されていることを確認します。
   azure provider list

- **Azure Resource Manager ポータル**で、[Azure Active Directory] に移動します。
  a. [App Registration] を選択し、[Add] をクリックします。
  b. [Name] および [Sign on URL] に入力し、アプリケーションを作成します。

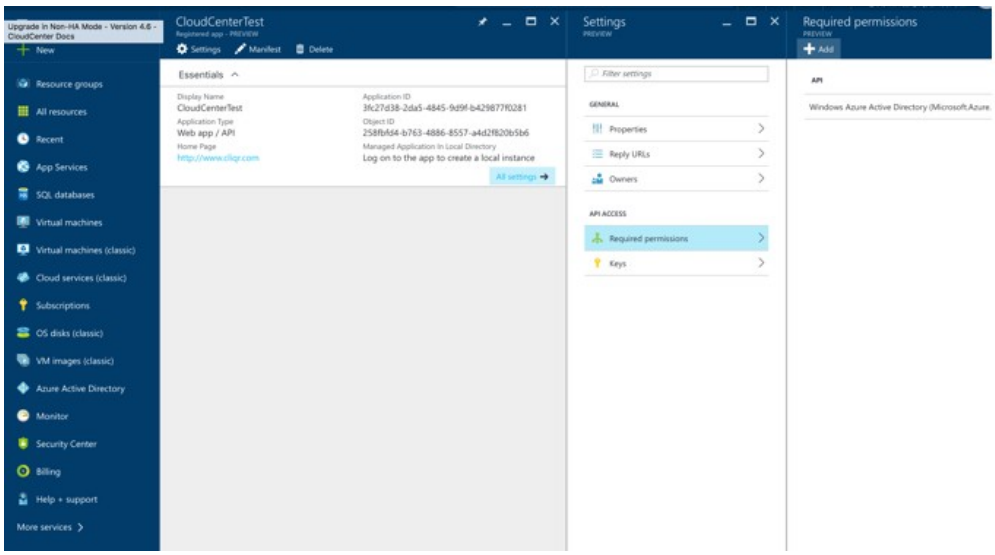


c. 新しく作成されたアプリケーションを選択します。

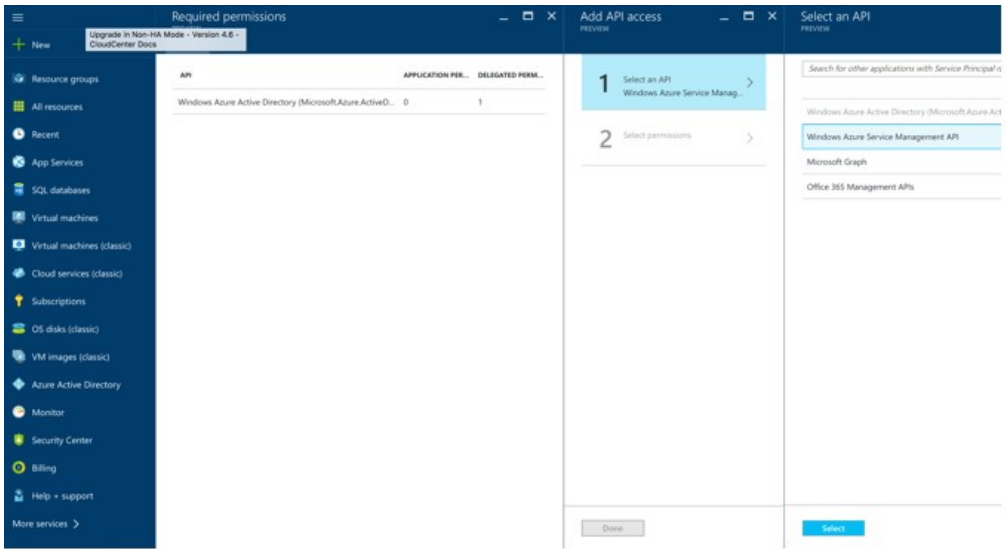
> **アプリケーション ID を書き留めてください。CloudCenter でクラウド アカウントを作成するときに必要です（クライアント ID になります）。**

d. [All Settings] をクリックします。

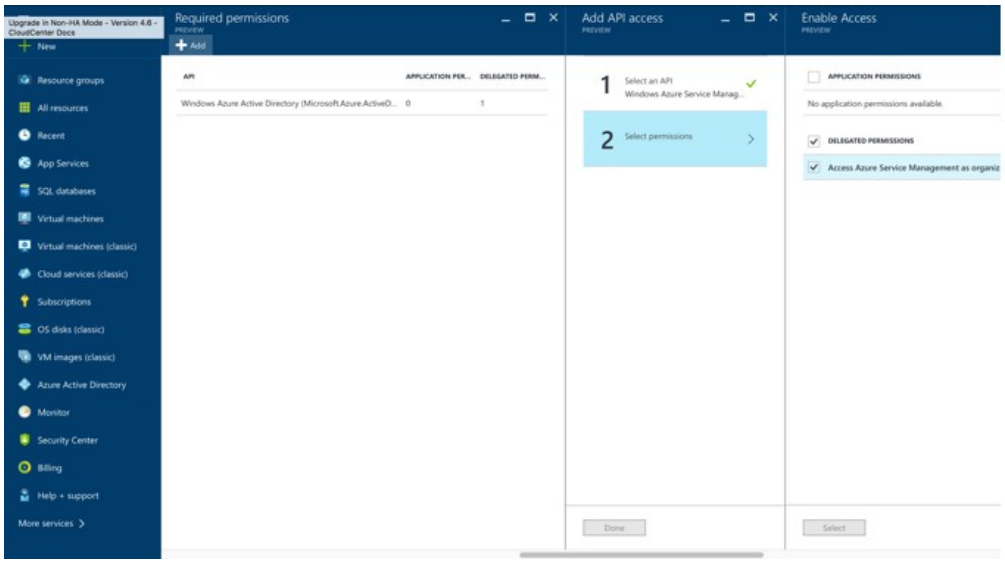
e. [API Access] の下にある [Required Permission] を選択し、[Add] をクリックします。

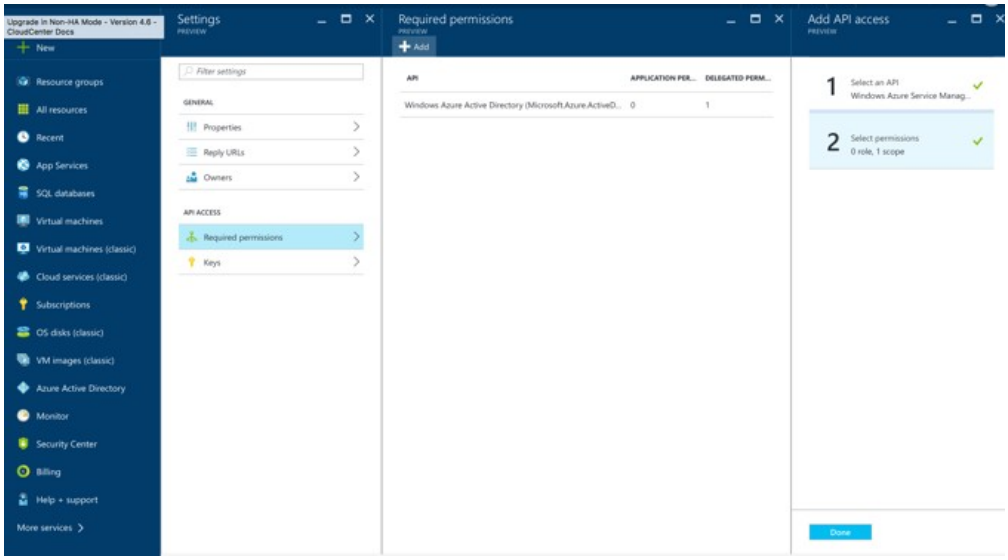


f. [Windows Azure Service Manager AP] を選択します。



g. 権限として [Delegated Permission] を選択し、[Done] をクリックします。

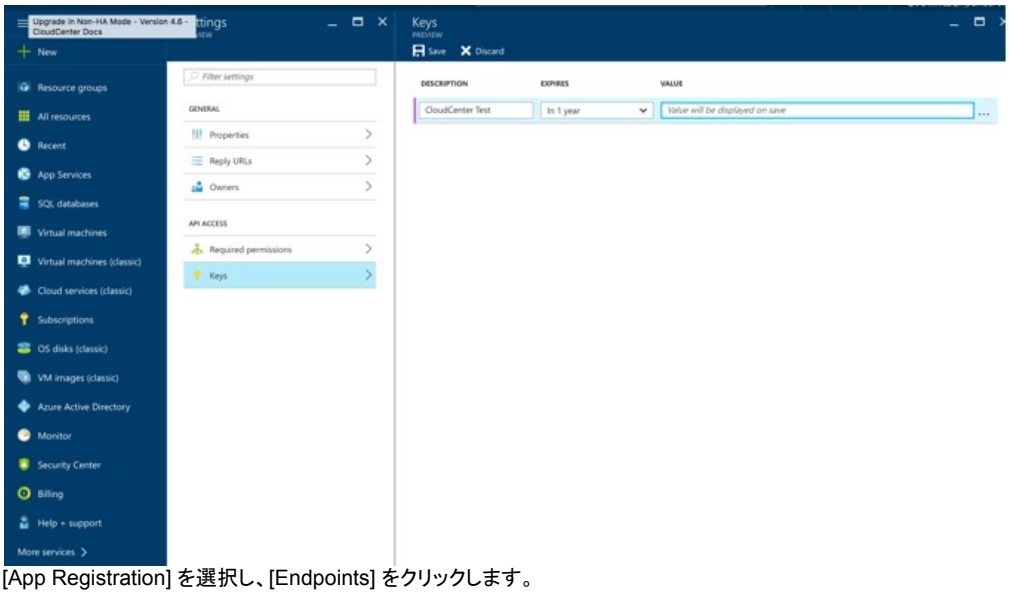




h. [API Access] の下にある [Keys] を選択します。

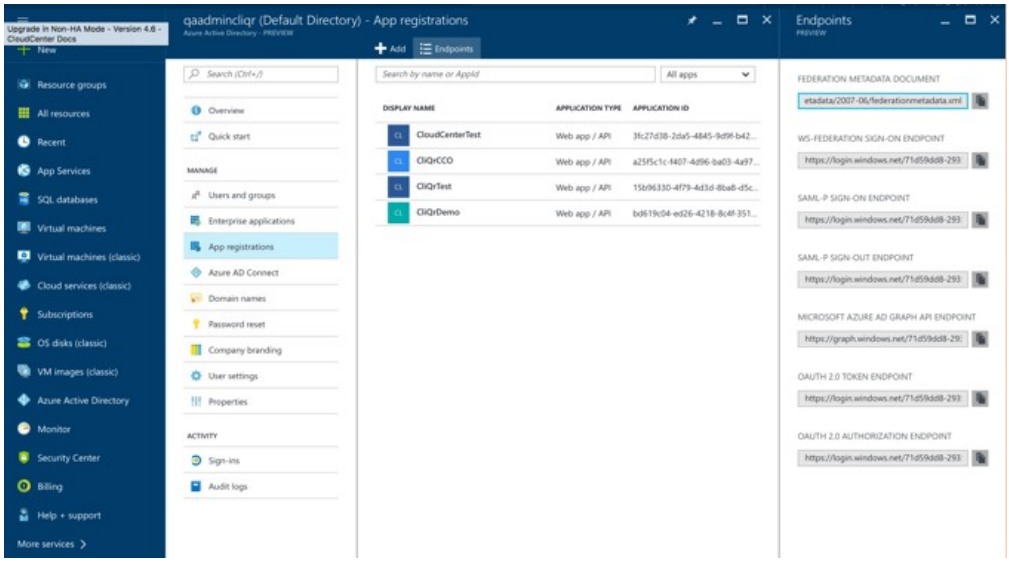
i. [Description]、[Expires] を指定し、[Save] クリックします。

> **[Save] をクリックした後、キーを書き留めてください。このキーを後でポータルから取得することはできません。このキーは、クラウド アカウントを作成する際に、クライアント キーとして CloudCenter ポータルで使用されます。**

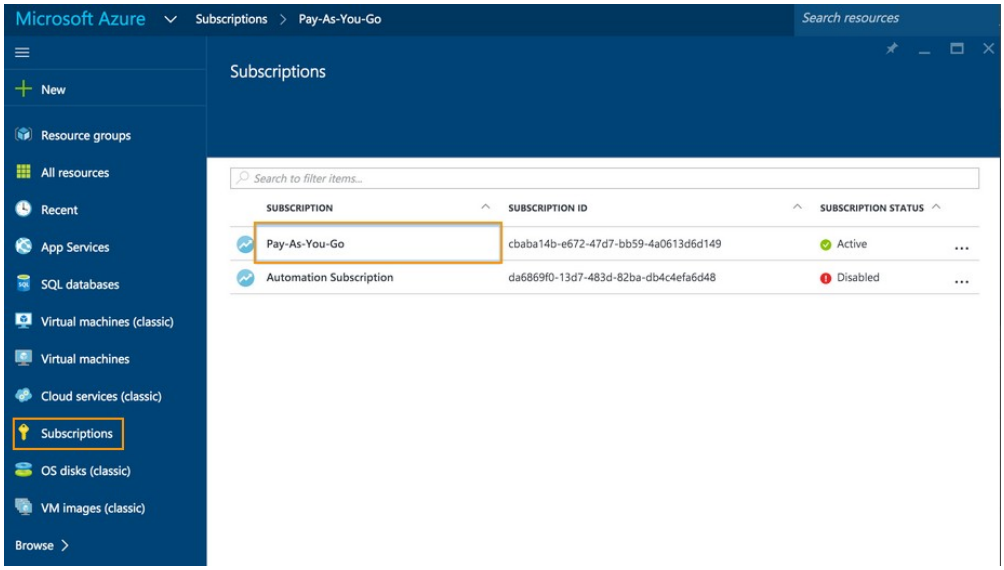


j. [App Registration] を選択し、[Endpoints] をクリックします。

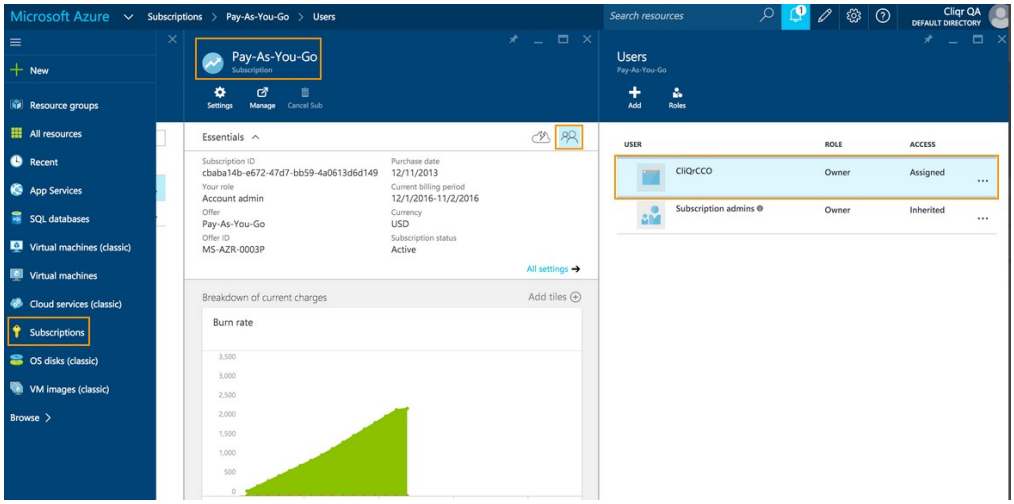
> **[OAuth 2.0 Authorization Endpoint] のテナント ID を書き留めてください。この ID は、クラウド アカウントを作成する際に、テナント ID として CloudCenter ポータルで使用されます。**



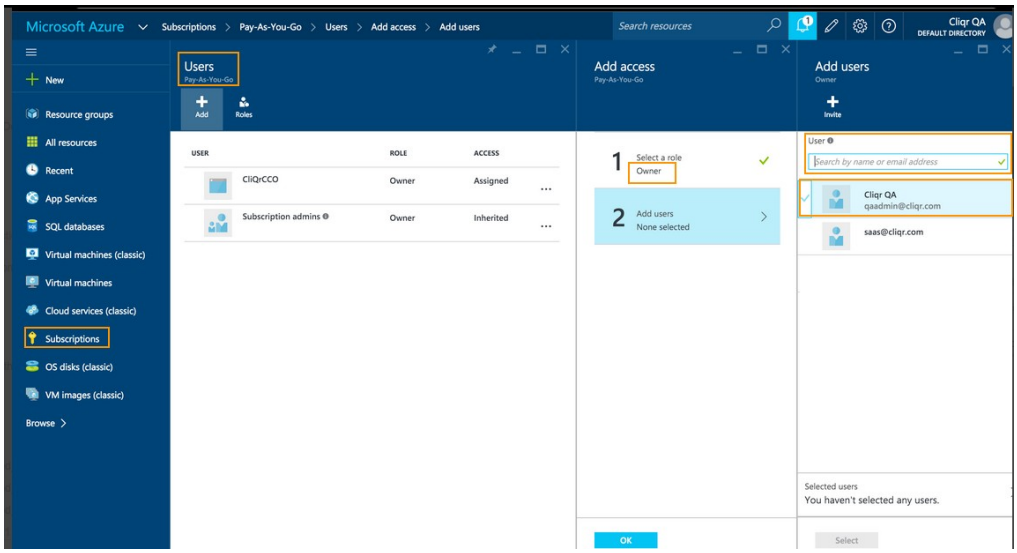
- **Azure Resource Manager ポータル**で、Web アプリケーション用のユーザ ロール設定を次のように設定します。
  1. [Subscription] > [Pay-As-You-Go] を選択します（次の図に、これは、管理する必要があるサブスクリプションとして表示されています）。



  2. [managed subscription] ペインの右上隅にある [users] アイコンをクリックします。



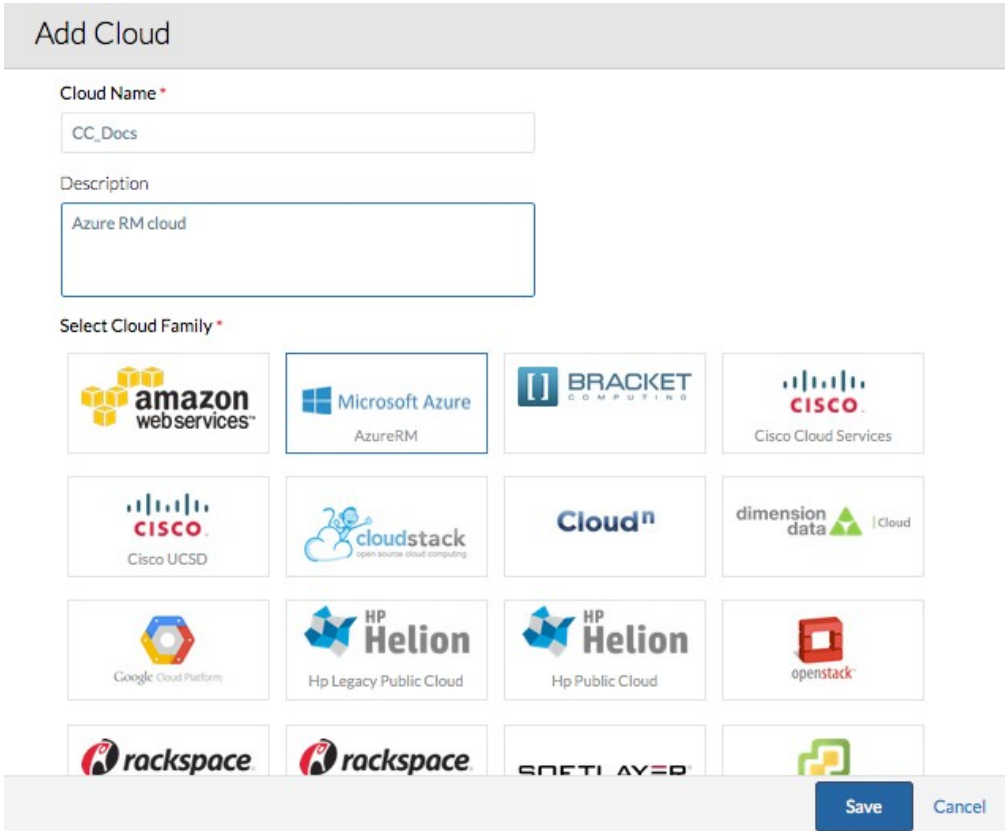
  3. [Add User] をクリックし、[OWNER] ロールを選択します。より詳細な管理をするために、他のロールを選択することもできます。
  4. [User] 検索ボックスに、あらかじめ定義している Web アプリケーションの名前を入力します。この例では、**CliQr QA** です。



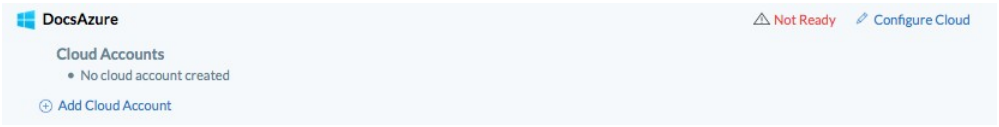
  5. [OK] をクリックして設定内容を保存します。

Azure Resource Manager クラウドで論理マッピングを設定するには、次の手順に従います。

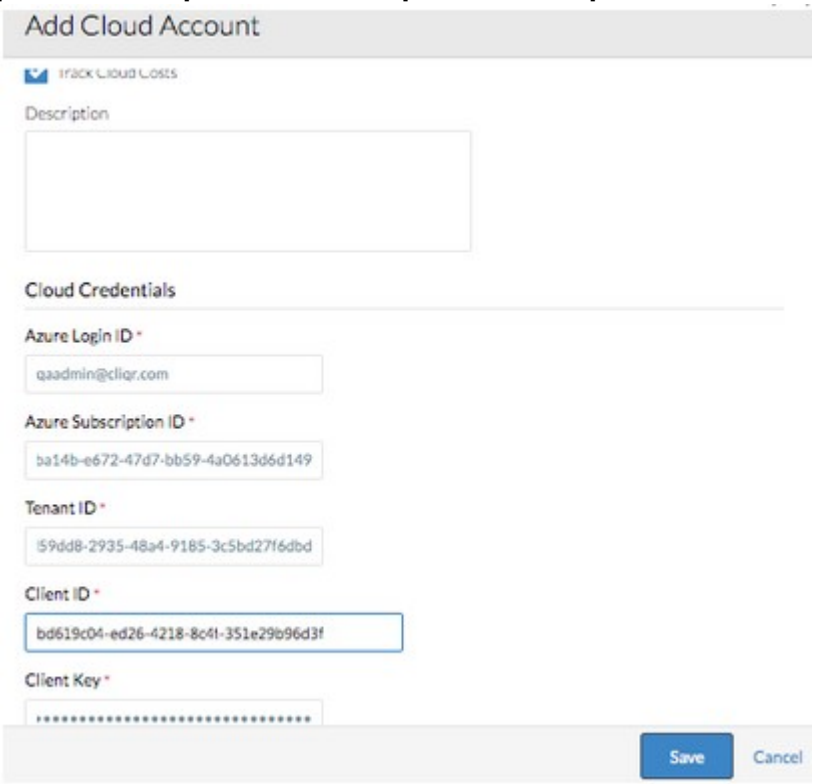
1. 新しいクレデンシャルを使用して、CloudCenter UI にログインします。
2. 左ペインに表示されている [Admin] リンクをクリックします。
3. サイド パネルで [Clouds] リンクをクリックし、[Clouds] ページを表示します。
4. [Add Cloud] をクリックして、選択したクラウドを追加します。



5. 新しく追加したクラウドを見つけるために（必要に応じて）スクロールします。



6. [Add Cloud Account] リンクをクリックします。[Add Cloud Account] ポップアップが表示されます。



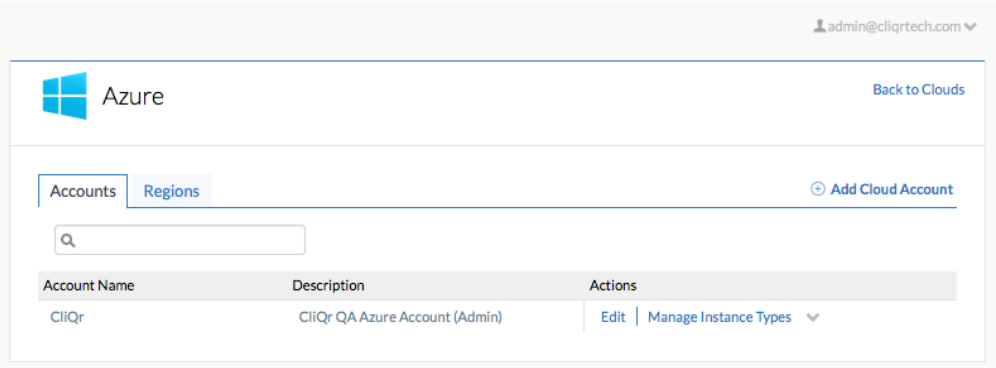
a. 新しいクラウドに名前を割り当てます。

> **ヒント**
> 名前には、スペース、ダッシュ、特殊文字を含めることはできません。

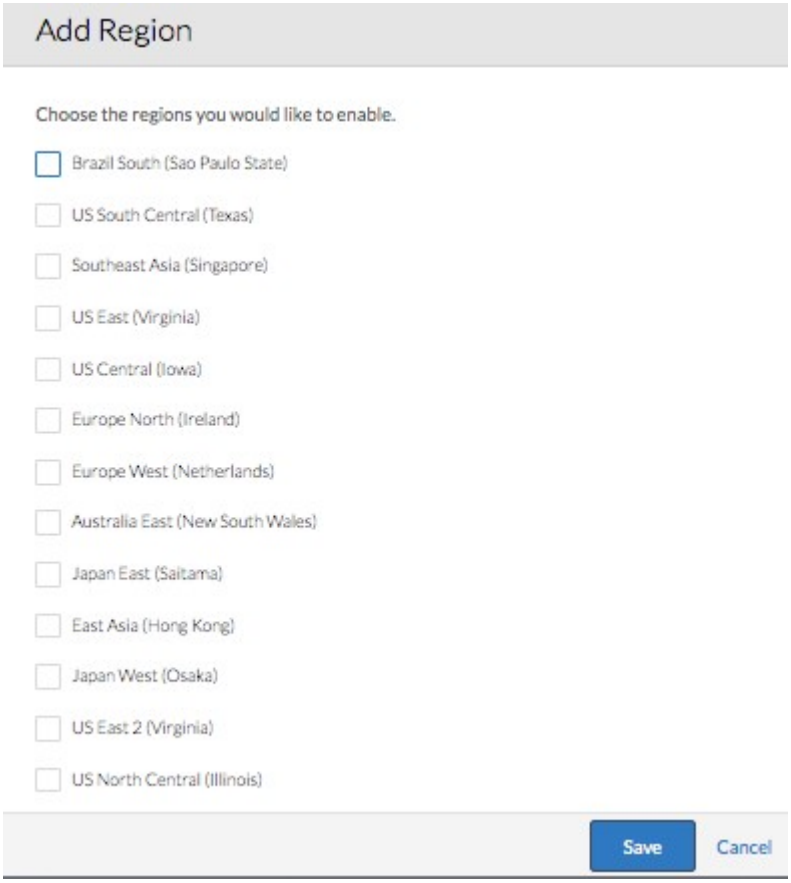
b. Azure アカウントに関連付けられている次のクラウド クレデンシャルを追加します。
   i. **Azure Login ID**：Azure Resource Manager クラウド アカウントへのログインに使用される電子メール アドレス。
   ii. **Azure Subscription ID**：**サブスクリプション ID** を取得するには、上記の「*前提条件*」の項の説明に従って **Azure Classic ポータル**のインターフェイスに切り替えて、[Settings] にアクセスします。

   iii. **Tenant ID**：上記の「*前提条件*」の項の、*エンドポイントの表示*に関する箇条書きで指定されている UUID。
   iv. **Client ID**：上記の「*前提条件*」の項の、青色のアイコンに関する箇条書きで指定されている UUID。
   v. **Client Key**：上記の「*前提条件*」の項の、*キー*に関する箇条書きで指定されています。

c. [Save] をクリックし、新しく追加したクラウド アカウント（用語の詳細な説明は「Cloud」を参照）が [Cloud] ページに表示されていることを確認します。
d. [Configure Cloud] リンクをクリックします。[Cloud Accounts] ページの [Accounts] タブに、設定したクラウド アカウントが表示されます。



7. [Regions] タブをクリックします。
8. [Add Regions] リンクをクリックし、Azure インフラストラクチャ環境での物理リソースの地理的グループ化を説明する [Region Name] を追加します。



9. [Edit Cloud Settings] リンクをクリックして、このクラウド リージョンに必要な設定を更新します（後述）。

CloudCenter の熟練者のアドバイスによる場合を除き、デフォルト値を使用し、変更を行うことは避けてください。

- **Azure Resource Manager Url**：AzureRM のサービス エンドポイント。非政府向けクラウドおよび非中国向けクラウドのリージョン、つまり URL は、https://management.azure.com/ です。
- **Azure Active Directory Url**：Microsoft により提供され、クレデンシャルを認証するために使用される Azure のアクティブ ディレクトリ URL。非政府向けクラウド、または非中国向けのリージョン、つまり URL は、https://login.microsoftonline.com/ です。
- **Linux Custom Script Extension Version**：Microsoft により提供され、ダイナミック ブートストラップを実行するためのカスタム スクリプトの拡張。使用するバージョンを指定します。
- **Windows Custom Script Extension Version**：Microsoft により提供され、ダイナミック ブートストラップを実行するためのカスタム スクリプトの拡張。使用するバージョンを指定します。
- **Linux Diagnostics Extension Version**：Microsoft により提供され、メトリック モニタリングを実行するための診断の拡張。使用するバージョンを指定します。
- **Windows Diagnostics Extension Version**：Microsoft により提供され、メトリック モニタリングを実行するための診断の拡張。使用するバージョンを指定します。

- **Instance Naming Strategy**：AWS、OpenStack、および AzureRM でサポートされるネーミング ストラテジ。デフォルトは **cqjw-** が前に付くランダムな文字列です。このストラテジを選択する場合は、次のフィールドで [Node Name Config] の詳細を指定する必要があります。追加情報については、「VM Name Config」を参照してください。
- **Node Name Config**：追加情報については、「VM Name Config」を参照してください。

10. [Save] をクリックし、変更内容をクラウド リージョンに保存します。
11. [Add Instance Type] リンクをクリックして、クラウド リージョンのインスタンス タイプを追加します。導入のための新しいインスタンス タイプを追加すると、[Configure Instance Types] セクションに設定済みのインスタンスが表示されるようになります。

> **注意**
> [Configure Image Mapping] セクションに、使用可能なイメージ名が自動的に入力されます。ただし、イメージ マッピングを設定する前に、まず、CCO VM を設定し、CCM により CCO を登録する必要があります。

これで、Azure Resource Manager クラウドが設定されました。

追加情報については、「Azure Configurations」を参照してください。

> **注意**
> [Cloud Region] ページ（[CCM UI] > [Admin] >[Clouds] > [Configure Region]）の [Edit Cloud Settings] で設定を編集したら、[Configure Orchestrator]、次に [Save] をクリックして、変更内容を CCO に保存し、これらの変更がファイアウォール 設定内の CCO にもプロパゲートされていることを確認する必要があります。

インスタンス タイプの管理

- 概要
- インスタンス タイプの例
- インスタンス タイプの追加
- インスタンス タイプの同期
- 課金管理
- スクラッチ ディスク ストレージ（ローカル ストレージ）
- ルート ディスク サイズ
- インスタンス タイプの非表示

概要

CloudCenter は、イメージ マップ プロセス中に管理者によって選択されたアプリケーション VM インスタンス タイプのみを表示します。

- CloudCenter は、イメージ マップ プロセス中に提供されたインスタンスだけを自動的にリストするよう、[Instance Type] ドロップダウンをフィルタリングします。
- 選択したクラウドにイメージ マッピングがない場合や、イメージと選択したクラウドとに互換性がない場合、CloudCenter はインスタンス タイプを表示しません。
- インスタンス タイプは、パブリック クラウドでは動的に設定されます。導入のためにインスタンス タイプを追加すると、[Configure Instance Types] セクションに設定済みのインスタンスが表示されるようになります。
- 管理者は、マッピングされたイメージを編集して、追加のインスタンス タイプをサポートできます。
- クラウド リージョンにインスタンス タイプを追加すると、ユーザはローカル ストレージとしてゼロ ディスク サイズを指定できます。0（ゼロ）の場合は、イメージ内のディスクが使用され、追加のディスクは VM に導入されません。
- インスタンス タイプおよびイメージの同期へのサポーは、AWS SoftLayer、Azure、AzureRM、Google、Dimension Data で提供されます。

インスタンス タイプの例

企業用のインスタンス タイプを入力します。CloudCenter プラットフォームで定義されているサンプル インスタンス タイプを次に示します。

Instance Type	Name	Hardware Spec	Cost Per Hour	Actions
small	small	2CPUs, 2048GB, 1000Mbps	0.2	Edit | Delete
medium	medium	2CPUs, 4096GB, 1000Mbps	0.3	Edit | Delete
large	large	4CPUs, 8192GB, 1000Mbps	0.6	Edit | Delete
xlarge	xlarge	4CPUs, 16384GB, 1000Mbps	1.2	Edit | Delete
2xlarge	2xlarge	8CPUs, 32768GB, 1000Mbps	2.4	Edit | Delete

インスタンス タイプの追加

インスタンス タイプを追加するには、次の手順に従います。

1. 左ペインに表示されている [Admin] リンクをクリックします。
2. サイド パネルで [Cloud Configurations] リンクをクリックし、[Cloud Configurations] ページを表示します。
3. 必要なクラウドに対し [Configure] リンク（[Actions] 列）をクリックします。
4. [Cloud Regions] セクションまでスクロールし、必要なリージョンに対し [Configure] リンクをクリックします。
5. [Configure Image Mapping] セクションまでスクロールし、必要なイメージについて [Add Mapping] をクリックします。
6. [Add Instance Type] をクリックして新しいインスタンス タイプ マッピングを追加するか、既存のマッピングを上書きします。
7. [Save] をクリックし、新しく追加されたインスタンス タイプを保存します。
8. 導入の必要に応じて、他のイメージに対してこのプロセスを繰り返します。

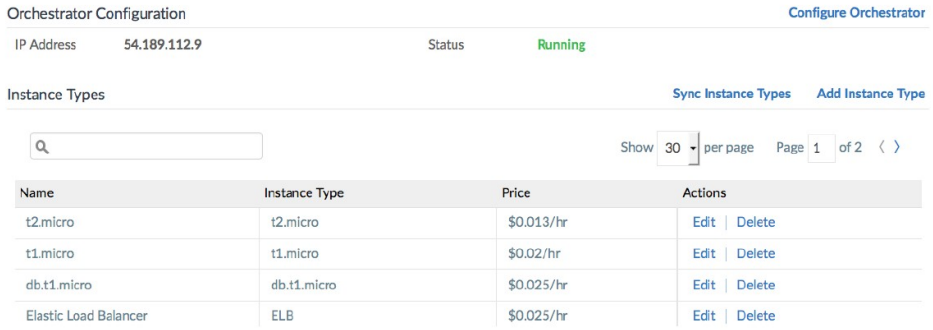
インスタンス タイプの同期

> この手順を実行するには、パッケージ ストアにアクセスできる必要があります。

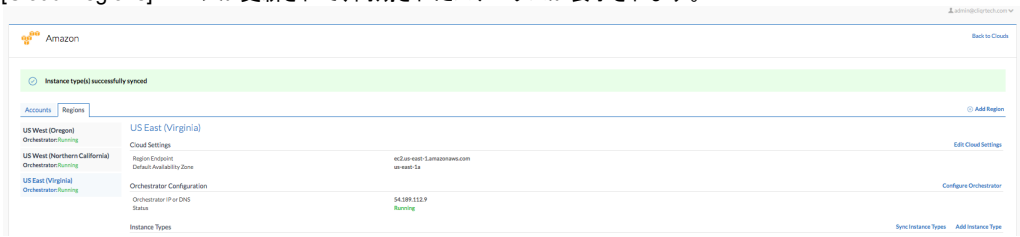
管理者は、パッケージ ストアからインスタンス タイプおよび価格情報を同期できます。この機能により、管理者は、クラウド プロバイダーのインスタンス タイプ定義および価格情報に変更を見つけたときに、情報を同期できます（他の課金情報については、「Tenant Billing」を参照してください）。

インスタンス タイプを同期するには、次の手順に従います。

1. [Configure Cloud(s)] ページで指定したとおりにクラウド リージョンを設定します。
2. CCM により CCO を登録します。
3. [CCO (Orchestrator) Configuration] セクションに [Running] ステータスが表示されたら、[Sync Instance Types] をクリックします。



4. [Cloud Regions] ページが更新されて、同期されたステータスが表示されます。



課金管理

ユーザへの課金の追加情報については、「Tenant Billing」>「*Granular Billing Control* 」を参照してください。

スクラッチ ディスク ストレージ（ローカル ストレージ）

通常、ルート ディスクはクラウド イメージによって制御され、CloudCenter プラットフォームによりサイズ変更されません。ほとんどのクラウドには自動化されたサイズ変更オプションはありません。アプリケーション VM のルート ディスクは導入に使用されるイメージと必ず同じサイズです。

CloudCenter プラットフォームでは、クラウド イメージ ルート ディスクのサイズに関係なく、インスタンス タイプの追加的（2 次的）サイズが提供されます。このサイズは、エクストラ ストレージ用に VM に追加されている追加のディスク（インストール ベースの OS イメージ）用です。

⚠ このローカル ストレージは、イメージ テンプレートにあるドライブに加えて、2 番目のハード ドライブとして追加されます。

導入時に CloudCenter プラットフォームから VM に割り当てられる追加ディスク サイズには独自の要件があります。ユーザが、インスタンス タイプによってサポートされるディスク ボリューム サイズを超えて追加した場合、導入は失敗し、関連付けられているクラウドから対応するエラー メッセージが発行されます。

インスタンス タイプ定義中のストレージ サイズ設定は、ユーザがインスタンス タイプを選択する際にどのようなハードウェア仕様を採用するかを理解するための（単に表示目的の）ものです。AWS や Azure などのパブリック クラウドでは、インスタンス タイプとそのハードウェア仕様はすべてクラウドによって事前定義されます。この設定を 2000 GB に変更しても、2000 GB の一時的ディスクが作成されるわけではありません。

ルート ディスク サイズ

ルート ディスク サイズの動作は、次の表に示すクラウドに依存します。

クラウド	クラウドごとのイメージ ディスク サイズの特徴
AWS	インスタンス タイプ ディスク サイズは一時的（一時的スクラッチ）ディスク用のサイズであり、ルート ディスク サイズは VM イメージで指定されるサイズです。一部のインスタンス タイプは複数の一時的ディスクを含む場合があります。
Azure	動作は AWS と類似しています。インスタンス タイプ ディスク サイズは一時的ディスク用で、ルート ディスク サイズは VM イメージで指定されるサイズです。インスタンス タイプによって、最大ルート ディスクのサイズに制限があります。 Azure インスタンス タイプのストレージ サイズは、一時的な（temp）ディスク サイズを意味します。A4\ExtraLarge を選択した場合、一時的ディスクは 650 GB になります。
Openstack	インスタンス タイプ ディスク サイズは、実際には、ルート ディスクのサイズです。インスタンス タイプが 40 GB のディスクを指定し、ベース イメージは 20 GB のルート ディスク サイズの場合、このイメージとインスタンス タイプを使用して VM が起動すると、ルート ディスクは自動的に 40 GB にサイズ変更します。ただし、ベース イメージが 80 GB のルート ディスクで、ユーザが 40 GB のディスク サイズを指定するものと同じインスタンス タイプを選択した場合、VM の起動は失敗します。
VMware	VMware にはインスタンス タイプの概念はありません。CloudCenter プラットフォームの定義では、インスタンス タイプ ディスク サイズは一時的ディスク用です。インスタンス タイプでディスク サイズが指定されている場合、このインスタンス タイプを使用する VM が起動するとき、CloudCenter プラットフォームは、指定のサイズの別のディスクを VM にアタッチします。追加情報については、「VMware Configurations」>「Resize VM Root Disk」を参照してください。

インスタンス タイプの非表示

⚠ インスタンス タイプは、各リージョンに固有ではありません。一部のインスタンスは一部のリージョンで使用できない場合があることに注意してください。

指定した ベース OS イメージの [Instance Type] ドロップダウンに使用可能なインスタンス タイプが表示されない場合、次の設定を確認してください。

- ベース イメージのマッピング：イメージをマッピングするときに、次の情報を確認してください（「イメージのマッピング」を参照）。
  - サービスが必要なベース OS イメージを使用していること、およびイメージ マッピングが定義済みであること。
  - （クラウド マッピングを追加するか、またはクラウド マッピングを編集する際に）イメージのインスタンス タイプを選択するとき、必要なインスタンス タイプがマッピングされていることを確認してください。
- ハードウェア仕様：使用する仕様が、ターゲットのインスタンス タイプの設定よりも低いことを確認します。上記の「サンプル インスタンス タイプ」の項に挙げたいくつかの例を参照してください。
- インスタンス タイプのアーキテクチャ設定：32 ビットおよび 64 ビット アーキテクチャのサポートを確認するため、[Both] オプションの選択を試行してから、導入を再実行します。

イメージ ID の設定

- 概要
- イメージ ID のクラウドごとの特徴
- ベース OS イメージをダウンロードおよびインポートして、イメージ ID を取得します

概要

CloudCenter プラットフォームは、CCM および CCO アプライアンスに加え、Ubuntu12.04、Ubuntu14.04、CentOS6、および Ubuntu Web アプライアンス用のベース OS イメージをサポートします。CloudCenter がサポートするイメージを使用するか、選択したインポート アプライアンスを使用するかを選択できます（追加情報については、「Image Administration」を参照）。いずれにしても、アプライアンスをダウンロードし、各アプライアンスに対し、イメージ ID を取得する必要があります。イメージ ID は、イメージのマッピングをする際に、クラウドでアプライアンスを参照するために必要です。

イメージ ID のクラウドごとの特徴

イメージ ID の用語は、クラウド間の異なります。

- VMware のイメージ ID

  <VM name >/<snapshot name>

  スナップショットを使用する場合は、vSphere にフォルダを追加し（CloudCenter のスナップショットを保存するため）、***CliqrTemplates*** という名前を付けて、このスナップショット ***CliqrTemplates*** フォルダに追加します。

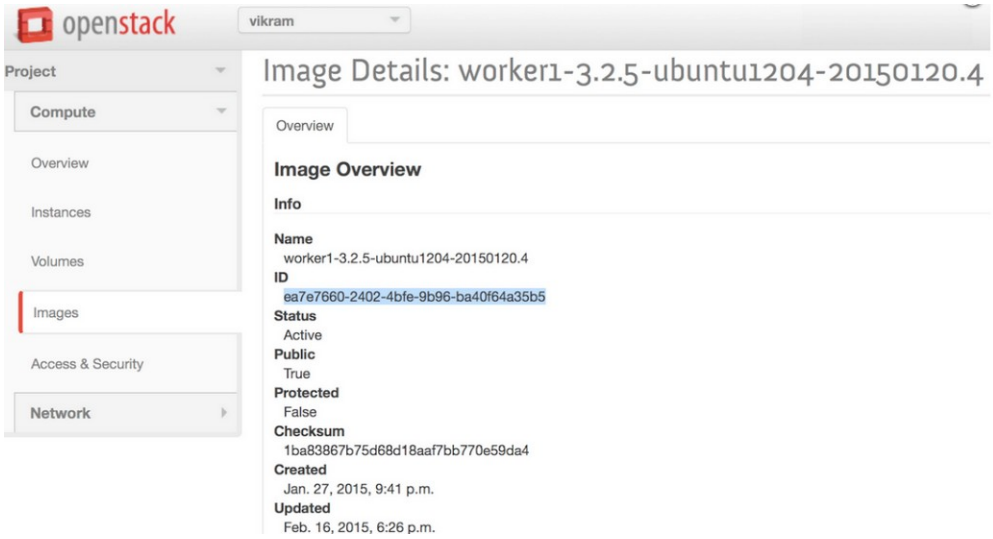
  または、VM テンプレート名を使用してクラウド イメージを設定することもできます。この場合、VMware コンソールおよびシステム上のイメージ ID は常に指定されたデータストアまたはデータストア クラスタに対して完全な複製を実行するため、VM または VM テンプレートの名前を指定します。

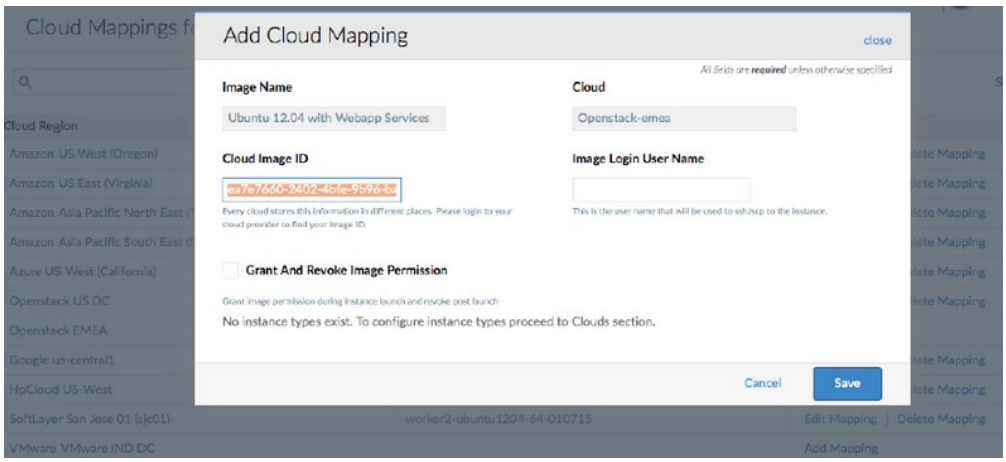
  完全な複製は送信元 VM または VM テンプレートで実行され、複製された VM はユーザが指定したデータストアまたはデータストア クラスタのいずれかに配置できます。

  追加情報については、「VMware Configurations」を参照してください。

- OpenStack のイメージ ID

  QCOW2 のイメージ ID（次の例で協調表示されたサンプル ID のマッピング）



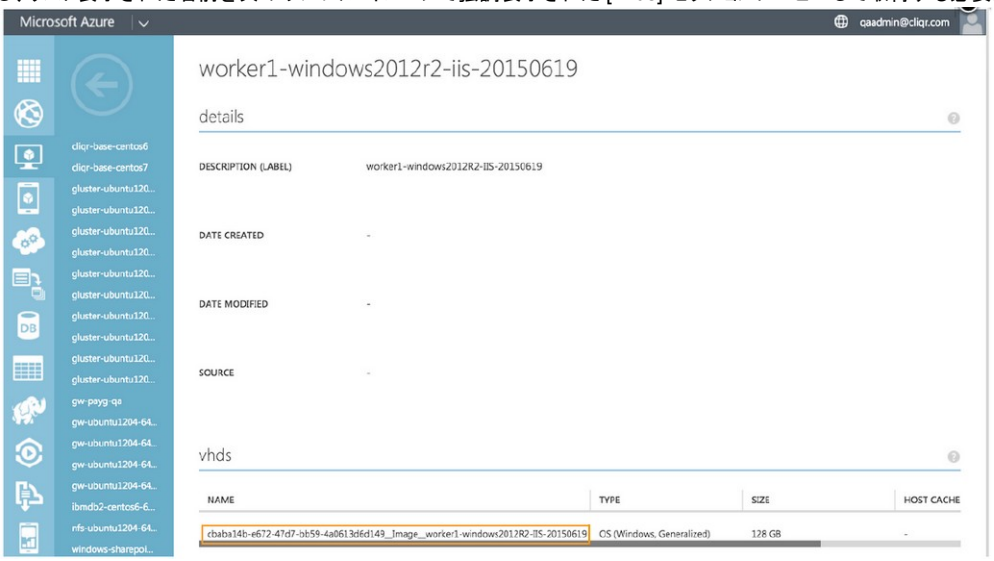


- Amazon のイメージ ID

  CloudCenter により自動的に処理されます。

- Azure のイメージ ID

  Azure では、アプリケーション VM イメージのマッピング名はディスク名です。ディスク名は、[Virtual instances] > [Image] タブをクリックし、リスト表示された名前を次のサンプル イメージで強調表示された [vhds] セクションにコピーして取得する必要があります。



- Google のイメージ ID

  GCE では、CloudCenter インスタンス用の VM を起動し、ページの一番下にある [REST] をクリックします。[REST] の出力に **sourceImage** の値を確認できます。

```
{
    "disks": [{
        "type": "PERSISTENT",
        "boot": true,
        "mode": "READ_WRITE",
        "autoDelete": true,
        "deviceName": "instance-1",
        "initializeParams": {
            "sourceImage":
"https://www.googleapis.com/compute/v1/projects/centos-cloud/global/images/
centos-7-v20160418",
            "diskType":
"projects/x-signifier-537/zones/us-central1-f/diskTypes/pd-ssd",
            "diskSizeGb": "10"
        }
    }]
}
```

ベース OS イメージをダウンロードおよびインポートして、イメージ ID を取得します

ワーカー アプライアンス用のイメージ ID を取得するには、次の手順に従います。

1. ワーカー アプライアンス（OVA または QCOW2 ファイル）を、ワーカー VM の /tmp フォルダにダウンロードします。
2. ワーカー アプライアンス用のイメージ ID を取得します。このイメージ ID は、CCM UI でイメージをマッピングするために必要です。

イメージのマッピング

- 概要
- [Cloud Configuration] タブでの方法
- [Images] タブでの方法
- サービス マッピングの変更
- イメージ マッピングの同期
- インスタンス タイプ
- ルート ディスク サイズ

概要

アプリケーション VM イメージをインストールした後は、CloudCenter VA ベース イメージのイメージ ID を、このクラウド リージョンでの CloudCenter の論理イメージ名にマッピングします。

イメージ ID はクラウド間で異なるため、イメージ マッピングはクラウドによって異なります。詳細については、「イメージ ID の設定」の項を参照してください。

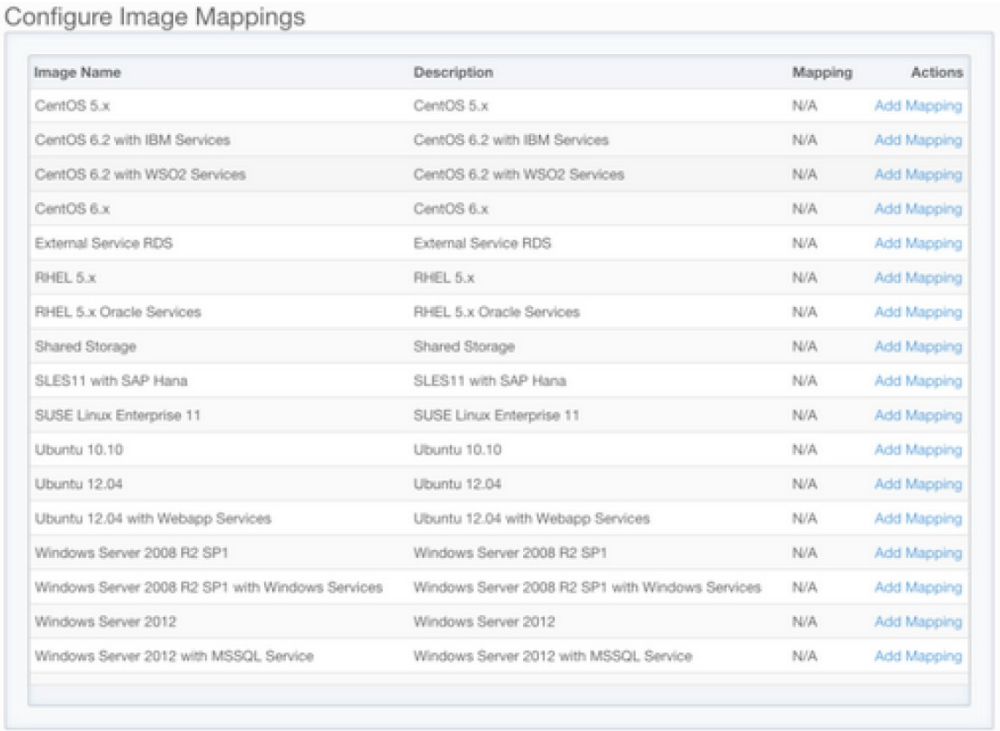
CloudCenter VA ベース イメージ ID を CloudCenter の論理イメージ名にマッピングするには、[Cloud Configuration] タブでの方法、または [Images] タブでの方法を使用できます。

> シスコでは、データセンターとプライベート クラウドをイメージ マッピングするための設定済み機能を提供していません。アプリケーション VM のイメージ マッピングを設定すると、利用可能なサービスが表示されます。

[Cloud Configuration] タブでの方法

CloudCenter VA ベース イメージ ID を CloudCenter プラットフォームの論理イメージ名にマッピングするには、次の手順に従います。

1. 左ペインに表示されている [Admin] リンクをクリックします。
2. サイド パネルで [Cloud Configurations] リンクをクリックし、[Cloud Configurations] ページを表示します。
3. 必要なクラウドに対し [Configure] リンク（[Actions] 列）をクリックします。
4. [Cloud Regions] セクションまでスクロールし、必要なリージョンに対し [Configure] リンクをクリックします。
5. [Configure Image Mapping] セクションまでスクロールし、必要な CloudCenter ベース OS イメージについて [Add Mapping] をクリックします。次に例を示します。
    - CentOS 7
    - RHEL 7
    - Ubuntu12.04



a. [Add Instance Type] をクリックして新しいインスタンス タイプ（「インスタンス タイプの管理」を参照）マッピングを追加するか、既存のマッピングを上書きします。
b. [Save] をクリックし、新しく追加されたインスタンス タイプを保存します。
c. 導入の必要に応じて、他のイメージに対してこのプロセスを繰り返します。

[Images] タブでの方法

「Manage Cloud Mapping」を参照してください。

サービス マッピングの変更

デフォルトでは、[Services] タブ内のすべてのサービスが、CentOS 6 または RHEL 6 イメージにマッピングされます。

> **（注）**
> CloudCenter の導入で他のイメージを使用する場合は、CCM UI で適用可能なイメージを明示的に指定する必要があります。

サービス変更をマッピングするには、次の手順に従います。

1. 左ペインで、[Admin] リンクをクリックします。
2. [Services] リンクをクリックします。
3. [Edit] をクリックし、導入に適用可能なサービスを編集します。
4. [Edit Service] ページで、[Supporting Images] フィールドの設定を確認します。
5. 選択内容を、使用中の環境で使用するイメージに変更します。
6. [Save] をクリックします。

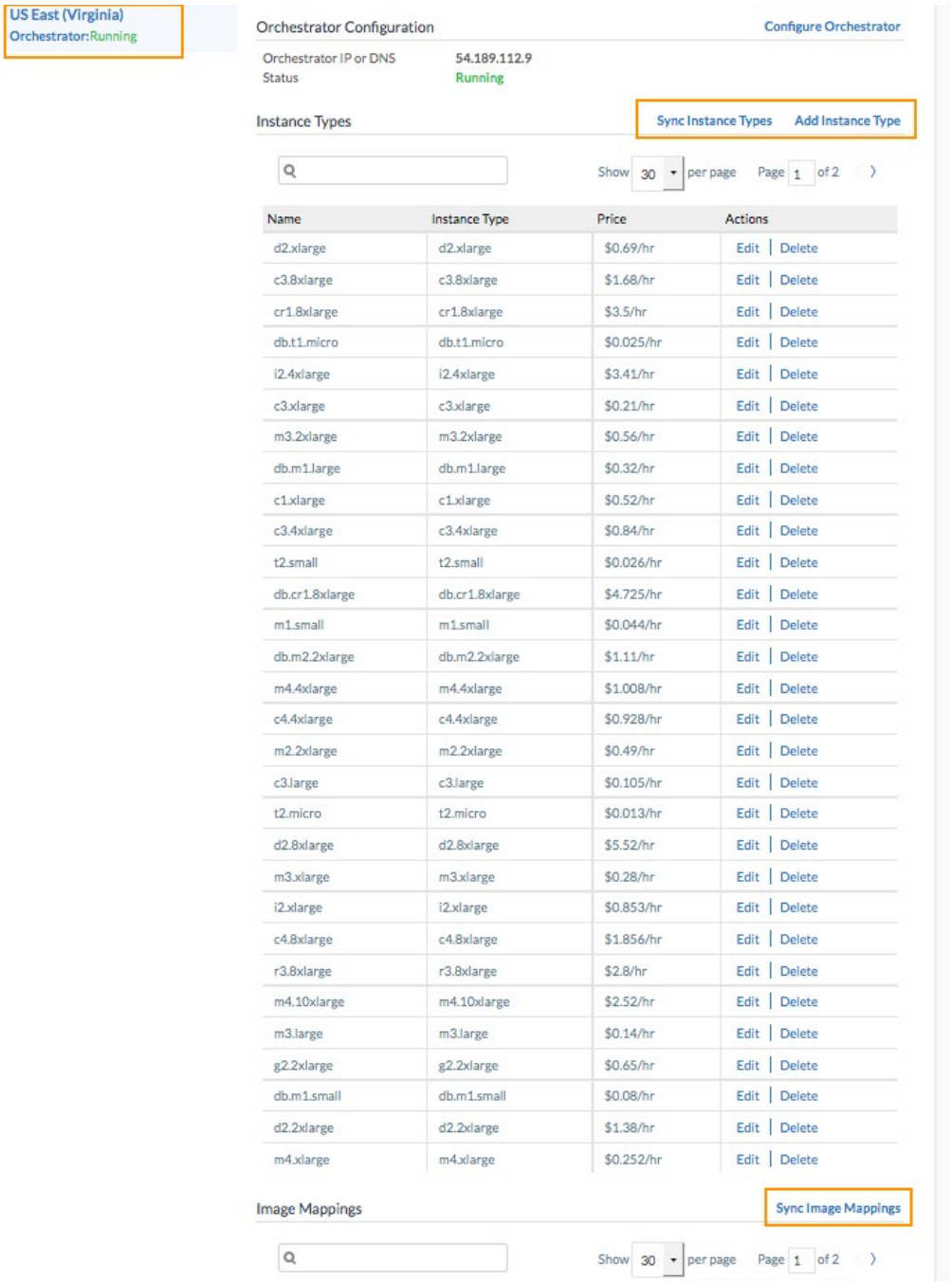
イメージ マッピングの同期

> この手順を実行するには、パッケージ ストアにアクセスできる必要があります。

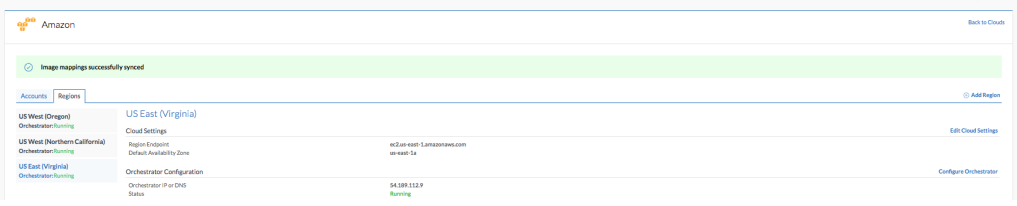
パッケージ ストアのクラウド イメージ マッピングに同期することができます。この機能により、管理者は、クラウド プロバイダーのイメージ マッピング定義に変更が見つかったときに情報を同期することができます。一度に 1 つのクラウド リージョン（1 つのクラウドに属するすべてのクラウド リージョンでなく）の情報のみ同期できます。

イメージ マッピングを同期するには、次の手順に従います。

1. [Configure Cloud(s)] ページで指定したとおりにクラウド リージョンを設定します。
2. CCM により CCO を登録します。
3. [CCO (Orchestrator) Configuration] セクションに [Running] ステータスが表示されたら、[Sync Image Mappings] をクリックします。



4. [Cloud Regions] ページが更新されて、同期されたステータスが表示されます。



インスタンス タイプ

追加情報については、「インスタンス タイプの管理」を参照してください。

ルート ディスク サイズ

追加情報については、「インスタンス タイプの管理」を参照してください。

CCM による CCO の登録

> この手順を開始する***前***に、必ず「Phase 2: Configure Network Rules」の確認を実行してください。

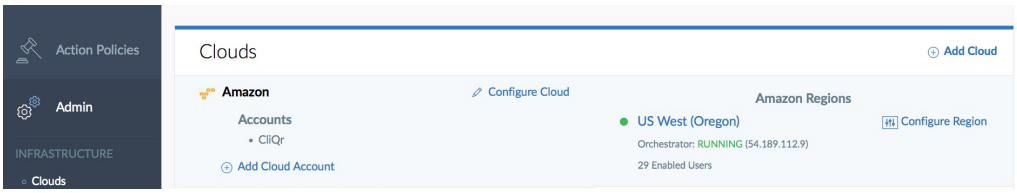
- CCM により CCO を登録すると、CloudCenter プラットフォームは、このクラウド リージョンがアクティブであるとみなし、CloudCenter プラットフォームからこのクラウド リージョンを削除できなくなります。
- 常に、1 つのクラウド リージョンには 1 つの CCO だけを登録できます。以前に登録した CCO を新しいクラウド リージョンに登録すると、CCO は新しいクラウド リージョンに登録を切り替え、元のクラウド リージョンでは機能しなくなります。

CCM VM および CCO VM が実行中のときは、CCM VM にログインして両方の VM 間の相互信頼関係を確立します。

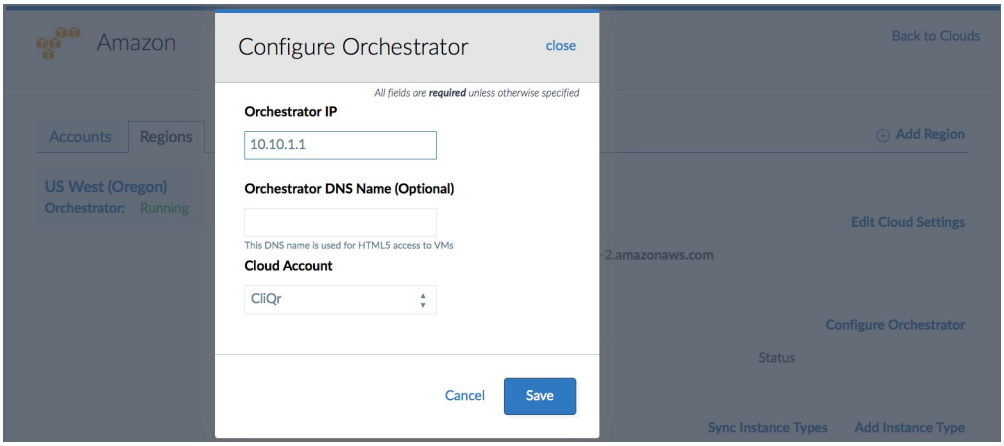
> クラウド リージョンに対して CCO の登録をしていない場合、ユーザアクティブ化できません。

CCM により CCO を登録するには、次の手順に従います。

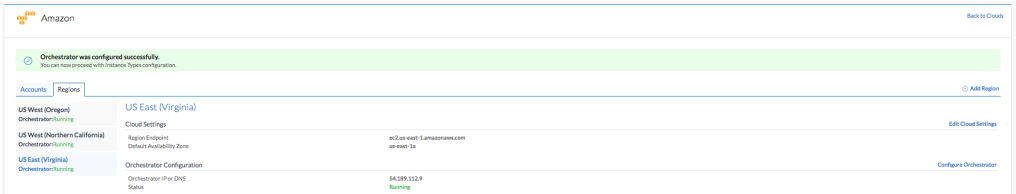
1. 左ペインで、[Admin] リンクをクリックします。
2. [Cloud] リンクをクリックし、[Cloud] ページにアクセスします。
3. CCO のクラウド名に対応する [Configure Region] をクリックします。



4. [Configure Orchestrator] リンクをクリックし、CCM からアクセス可能な CCO の IP アドレスを入力し、CCO をホストするために使用されるクラウド アカウントを選択します。



5. [Save] をクリックします。これで、CCM と CCO の相互信頼関係が確立されました。CloudCenter プラットフォームは現在、導入された CCO によりクラウド リージョンを管理しています。



6. 他のクラウド アカウントの CCM と CCO の相互信頼関係を確立するために、ステップ 3 からステップ 5 を繰り返します。

CCO VM が登録されました。

設置のベスト プラクティス

- HA のベスト プラクティス
- モニタリングのベスト プラクティス
- CloudCenter の外部 URL
- バックアップと復元

ハイ アベイラビリティのベスト プラクティス

- データベース ハイ アベイラビリティの考慮事項
- CCM ハイ アベイラビリティの考慮事項
- CCO ハイ アベイラビリティ
- AMQP ハイ アベイラビリティ
- Guacamole

データベース ハイ アベイラビリティの考慮事項

CloudCenter プラットフォームでは、導入ごとに 1 つの CCM と 1 つのデータベースのみを必要とします。CCM ハイ アベイラビリティ（HA）およびデータ複製を企業が必要とする場合は、次の手順を実行します。

- 同一のクラウドまたはデータセンターに 2 つの CCM サーバと 2 つのデータベース サーバを配置する必要があります。
- CCM サーバは、設定および実行されるデータベース サーバによって異なります。必ず、CCM サーバを設定する前にデータベース サーバを設定してください。
- トラフィックを CCM インスタンスの 1 つに送信するためにロード バランサを設定し、セッションのスティッキ性ポリシーをイネーブルにします。
- どこかの時点でプライマリ サーバに障害が発生すると、セカンダリ サーバがシームレスに引き継ぎます。
- 障害が発生したサーバがオンラインに戻ると、データは新たなプライマリ サーバにより自動的に同期されます。このとき、オフラインだったサーバはセカンダリ サーバになります。

CloudCenter のコンテキストで使用される場合、PostgreSQL データベースはマスタースレーブ ハイ アベイラビリティ（HA）設定に従います。

- マスター サーバは非同期的にスレーブ サーバにデータの変更を送信します。
- マスター サーバは実行中、読み取り専用のクエリーに応答します。

マスター サーバに障害が発生すると、次の 2 つの PostgreSQL データベース サーバが連携して、2 台目のサーバに引き継ぎを許可します。

- **マスター** サーバ：データを変更するサーバ。このサーバが負荷を負います。
- **スレーブ** サーバ：マスター サーバで行われた変更に応答するか、または変更を複製するサーバ。

CCM ハイ アベイラビリティの考慮事項

CloudCenter のハイ アベイラビリティ（HA）ソリューションには、次の特徴があります。

- 同期的です。すべてのサーバがトランザクションを完了するまで、トランザクションは確定したものと見なされません。
- フェールオーバーによりデータが損失することはありません。CCM サーバの 1 つがオフラインになると、別のサーバがプライマリ サーバとして引き継ぎ、必要なすべての作業を処理し続けます。
- データベースとして、両方のサーバに一貫性のある結果を返し、サーバの設定変更は常に同期が保たれます。

ハイ アベイラビリティを確保するために、CloudCenter CCM および PostgreSQL データベース サーバが連携して、2 台目のサーバに引き継ぎを許可します。2 つのサーバ（CCM セットアップまたはデータベース セットアップ）を連携させるために、CloudCenter プラットフォームは、一方のサーバだけにデータの変更を許可します。

- **プライマリ** サーバ：データを変更するサーバ。このサーバが負荷を負います。
- **セカンダリ** サーバ：プライマリ サーバで行われた変更に応答するか、または変更を複製するサーバ。

CloudCenter 導入で CCM サーバに HA を提供するには、次のサーバをインストールし、それらのサーバにプライマリとセカンダリの複製セットアップを設定する必要があります。

- **2 つのデータベース サーバ**：マスター データベースとスレーブのデータベースに複製セットアップを設定。
- **2 つの CCM サーバ**：プライマリ CCM とセカンダリ CCM にハイ アベイラビリティ セットアップを設定。

障害が発生したサーバがオンラインに戻ると、データは新たなプライマリ サーバにより自動的に同期されます。このとき、オフラインだったサーバはセカンダリ サーバになります。

CCO ハイ アベイラビリティ

CCO の HA 手順は複雑で、DevOps についての深い理解が必要です。まず最初に、会社の DevOps チームと相談し、この項に示した各要件に対応していることを確認します。

ハイ アベイラビリティを確保するには、CloudCenter 導入で、各クラウド(リージョン)またはデータセンターに少なくとも 1 つの CCO がインストールされ、設定されている必要があります。CCO サーバはロード バランサの背後で同時に動作します。各サーバはアクティブで、クラスタ内のすべてのサーバはオーケストレーション タスクを並行して実行します。クラスタ内のサーバの 1 つがオフラインになっても、他のアクティブなサーバはオーケストレーション タスクの処理を続行します。オフラインになったサーバがオンラインに戻ると、データはアクティブなサーバにより自動的に同期されます。

CCO HA セットアップは、リージョンごとに行います。

CloudCenter は、クロス リージョンでの CCO HA の設定をサポートしません。

CCO HA セットアップでは、次の 3 台の CCO サーバが必要です。

- CCO1:CCO HA および MongoDB HA を設定
- CCO2:CCO HA および MongoDB HA を設定
- CCO3: MongoDB HA(MongoDB の最小要件)

HA を確実にするには、3 つの CCO サーバを、それぞれ異なるアベイラビリティ ゾーンで起動します。追加情報については、「Availability Sets and Zones」を参照してください。

CCO ロード バランサを使用するかどうか

- CloudCenter UI を使用して、ロード バランサの IP アドレスにより、[CCO IP] フィールドを設定します。
- 既存の導入に新しい CCO を追加する場合は、以前に設定した CCO IP アドレスをロード バランサの IP アドレスに変更します。

AMQP ハイ アベイラビリティ

CloudCenter 導入に真の HA サポートを提供するには、ロード バランサを RabbitMQ クラスタの最上位に配置する必要があります。

CloudCenter の管理エージェントは、AMQP サーバからメッセージを取得します。これは、即時プロセスではありません。メッセージ取得の進行中に AMPQ サーバで障害が発生すると、メッセージが失われる可能性があります。

CloudCenter には、クラスタリングや HA の既定済みソリューションを提供していません。そのため、真の HA を提供するためには、CloudCenter 導入用のクラスタ環境に、手動でロード バランサを追加する必要があります。クラスタリングを実装せずにロード バランサを使用するだけでは、複数の AMQP サーバ間のキュー交換が保証されません。

(ロード バランサを使用せずに)CCO 側に複数のクラスタ化された AMQP サーバを使用した場合、クラスタ環境は保証されますが、既定済みの CloudCenter の導入において複製は確立されません。ディザスタ発生時に必要な場合に、手動で管理されるフェイル オーバー メカニズムが提供されるだけです。

現在、CloudCenter プラットフォーム、Guacamole 用 HA をサポートしていません。Guacamole は、各 AMQP サーバ内に常駐する別個のコンポーネントです。追加情報については、「Component Modes and Roles」を参照してください。

各 AMQP サーバを個別のリージョンに配置する場合は、HA ソリューションがリージョンをまたがって機能しないことに注意してください。

エンドツーエンドの AMQP HA を実現するには、CloudCenter 導入で以下を設定する必要があります。

- ミラーリングによるクラスタ化された AMQP サーバの設定:サーバのクラスタ化に関する追加情報は、https://www.rabbitmq.com/clustering.html を参照してください。
- すべての AMQP サーバを管理するロード バランサ:使用環境でこの設定をするには、それぞれのロード バランサ関連のマニュアルを参照してください。

ハイ アベイラビリティのプロセスを開始する前に、次の要件を確認してください。

- データベースが設定済みである。
- IP アドレスではく DNS 名を使用して AMQP クラスタが設定されている。
- AMQP サーバを管理するようにロード バランサが設定されている。
- すべての CCO キューに適用されるミラーリング ポリシーによって、クラスタ化された AMQP サーバが設定されている。
- RabbitMQ クラスタの最上位にロード バランサを設定した後、AMQP サーバの IP アドレスを読み取るように CCO サーバを設定している。

Guacamole

Guacamole は個別のコンポーネントであることに注意してください。現在、CloudCenter プラットフォーム、Guacamole 用 HA をサポートしていません。追加情報については、「Component Modes and Roles」を参照してください。

モニタリングのベスト プラクティス

- 概要
- CloudCenter コンポーネントのバージョン
- VM モニタリング
- ディスク領域のモニタリング
- モニタする必要のある主要なサービスの例
- 追加のシステム モニタリング ポインタ

概要

この項では、CloudCenter プラットフォームを設定し、プロアクティブにモニタして、正常かつ継続的なパフォーマンスを確保するシステム管理者（「People」を参照）へのヒントを提供します。

CloudCenter コンポーネントのバージョン

CCO、CCM、および他の CloudCenter コンポーネントはすべて、同じ CloudCenter バージョンで実行される必要があります。異なる CloudCenter バージョンで実行することはできません。

VM モニタリング

システム レベルで VM をモニタして、メモリ使用率、ディスク使用率、および CPU 使用率を収集することができます。サードパーティ製モニタリング ツール（たとえば、Sensu オープン ソース モニタリング フレームワーク）を使用して、クライアントからシステム レベルのデータを収集し、VM をモニタし、使用に関する統計情報を提供することができます。このモニタリング フレームワークを使用するか、または企業用に合った他のフレームワークを使用するかを選択できます。

CloudCenter プラットフォームのモニタ中に収集された統計情報に基づき、管理者は、プラットフォームを効率的に機能させるための正しい方法を把握できます。

システム パフォーマンス	推奨モニタリングしきい値
メモリ	空きメモリが常に 10% 未満である。
CPU	CPU 使用量が一貫して 50% ～ 100% を超えている。
ディスク	ディスク使用量が常に 90% を超えている。

ディスク領域のモニタリング

複数のアプリケーション（データベース アプリケーション、OS アプリケーションなど）によるデータ ダンプが原因で、CCM VM のディスクスペースが不足する場合があります。このようなディスク スペース問題に対処するために、次の保護対策を実施することができます。

- ルート パーティションからディスク領域を解放する。
- CCM の容量を追加し、追加で設定したボリュームにデータを書き込むよう、データベース アプリケーション（またはデータ ダンプを起こしたほかのアプリケーション）を設定します。
- 特定のしきい値に達したときに自動アラートを送信するよう、Nagios モニタリングを設定します。

モニタする必要のある主要なサービスの例

- CCM
  - Tomcat 8
  - PostgreSQL
  - MongoDB
- CCO
  - Tomcat 8：2 つのインスタンス
  - MongoDB

追加のシステム モニタリング ポインタ

CloudCenter が正常にインストールされたら、CloudCenter プラットフォームおよびデータのセキュリティを確認します。ローカルでの安全性関連の問題が原因で CloudCenter プラットフォームが使用できなくなった場合は、CloudCenter プラットフォームおよびデータをモニタ、バックアップ、および復元する計画を整備ください。

大規模の企業では必ず、事業継続性計画の実行を確保するためにシステム モニタリング ツールを使用しています。CloudCenter プラットフォームのインスタンスで次のタスクを確保することは、企業の責任です。

- CloudCenter データベースのインスタンスを定期的にバックアップする。
- 企業にとって重要な時点で、CloudCenter のインスタンスをモニタする。

CloudCenter の外部 URL

- 概要
- 要件
- 使用法
- ロード バランシングとポート使用

概要

CloudCenter 外部 URL（*outfaceDnsName または outface DNS* とも呼ぶ）は、CloudCenter のユーザが接続する外部ポイントの DNS 名または IP アドレスを意味します。ユーザが CloudCenter にアクセスする方法に応じて、次のように実行します。

- **ロード バランサを使用**：ロード バランサの入力 DNS アドレスを入力します。
- **直接**（HA 設定を使用していない場合）：マシン の DNS を使用します。システムは、常に CloudCenter 外部 URL を使用して、ユーザがどのようにシステムにアクセスするかを参照します。

要件

1. CloudCenter 外部 URL は、HA モードでの CCM の状態に関係なく、電子メールに表示されます。
2. セットアップに HA モードが含まれていない場合でも、これは記入する必要があります。

セットアップ ウィザード に表示される CCM URL は、CloudCenter サービスを参照する別の設定です。

- HA セットアップでは、CloudCenter 外部 URL としてロード バランサ アドレスを使用します。
- HA セットアップ以外では、CloudCenter サービスがホストされている CCM（サーバの）URL を使用します。
- CCM インストール オプションの詳細については、「Component Modes and Roles」を参照してください。

使用法

この値は、多くの場合、CCM URL がメール コンテンツ中で送信される 電子メール通知などの機能や、CCM へのストレージ アクセスをリダイレクトする SaaS SSO の場合に使用されます。場合によっては、顧客が CCM で 2 つの NIC を使用することがあります。たとえば、1 つの NIC をパブリック ネットワークに接続してフロントエンド トラフィックに使用し、もう 1 つの NIC を（たとえば）プライベート ネットワークに接続して CCM と CCO 間のすべてのトラフィックに使用する場合があります。

CloudCenter 外部 URL 名の最も一般的な使用は、CCM HA セットアップでの使用です。この場合、CloudCenter 外部 URL とパブリック DNS 名の区別に注意してください。

- **CloudCenter 外部 URL**：CCM フロントエンド（UI または API）に到達するために、*outfaceDnsName* を使用して DNS または IP アドレスを指定します。つまり、インターネットに outface する DNS 名です。デフォルトで、outfaceDnsName は publicDnsName の値を使用するように設定されています。たとえば、2 つのロード バランサを（次の図に示すように）使用する場合、outfaceDnsName は CCM の左にある青いロード バランサのアドレスです。
- **パブリック DNS**：publicDnsName は、CCM により CCO を登録する際に、CCM が CCO に付与する DNS または IP アドレスであり、CCM と CCO 間の通信に使用されるアドレスです。

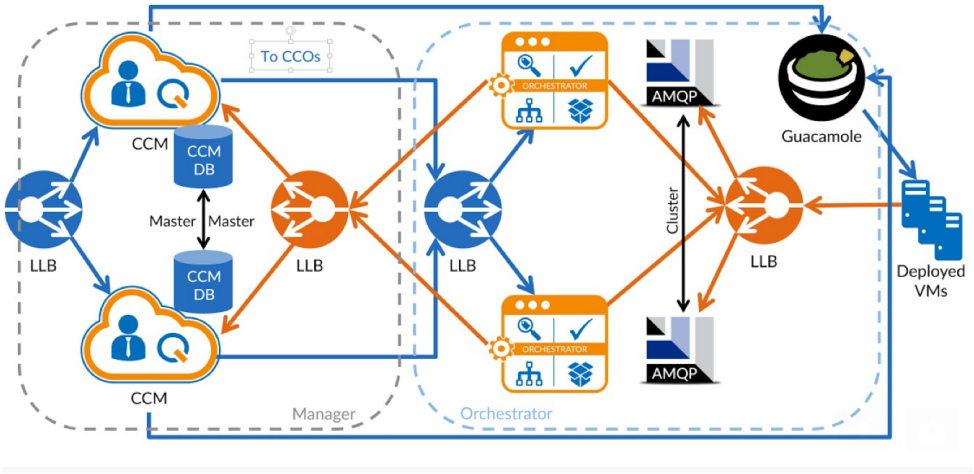
ロード バランシングとポート使用

ポート 441 経由の UI および API へのフロントエンド CCM トラフィックは、ロード バランサ向けの外部インターネットによりロード バランスされるため、CloudCenter 外部 URL 名は HA セットアップで重要になります。ポート 8443 経由の CCO からポート CCM へのバックエンド トラフィックでは、内部ロード バランサが使用されます。

> publicDnsName のエントリでは、パブリック ドメインの名前を使用する必要があります。CCO 側では、/etc/hosts は内部 IP へのエントリ ポインタを含む必要があります。

次の HA の図で、以下を示します。

- outfaceDnsName：CCM の左にある青いロード バランサのアドレス。
- publicDnsName：CCM の右にあるオレンジ色のロード バランサ。



毎日のバックアップと復元

- 推奨事項
- CCM データベースのバックアップ
- CCM データベースの復元
- CCO データベースのバックアップ
- CCO データベースの復元

推奨事項

CloudCenter 導入を効率的に管理するため、日単位で導入をバックアップします。cronjob を設定して、このバックアップを自動的に実行させることができます。

> CloudCenter 導入をアップグレードするときの手順が異なる可能性があることに注意してください。追加情報については、「Upgrade in Non-HA Mode」または「Upgrade in HA Mode」を参照してください。

CCM データベースのバックアップ

1. CCM サーバに SSH で接続します。
2. 既存の CCM データベースをバックアップします。
   a. **mkdir /root/backup_*date***
   b. **cd backup_*date***
   c. **pg_dump -U cliqr -d cliqrdb > pg_dump.sql**

> プロンプトで PostgreSQL のユーザ名とパスワードを入力します。
>
> - *Username = cliqr*
> - *Password = cliqr*

CCM データベースの復元

1. CCM サーバに SSH で接続します。
2. CCM データベースを復元します。

   psql -U cliqr -d cliqrdb < pg_dump.sql

> プロンプトで PostgreSQL のユーザ名とパスワードを入力します。
>
> - *Username = cliqr*
> - *Password = cliqr*
>
> cliqrdb および cliqr ユーザが存在***しない***場合は、ルート ユーザとして PostgreSQL にログインし、次のコマンドを実行します。
>
> a. **createdb -U postgres -E UNICODE cliqrdb**
> （プロンプトで PostgreSQL のユーザ パスワードを入力します）
> b. **psql -U postgres -d cliqrdb**
> （プロンプトで PostgreSQL のユーザ パスワードを入力します）

```
cliqrdb=# CREATE USER cliqr WITH PASSWORD 'cliqr' NOCREATEDB
NOCREATEUSER;
cliqrdb=# CREATE schema cliqr authorization cliqr;
cliqrdb=# GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN
schema cliqr to cliqr;
cliqrdb=# ALTER ROLE cliqr set search_path to cliqr;
cliqrdb=# \q
```

CCO データベースのバックアップ

1. CCO サーバに SSH で接続します。
2. 既存の CCO データベースをバックアップします。
   a. **mkdir /root/backup_*date***
   b. **cd backup_*date***
   c. **mongodump**

CCO データベースの復元

1. CCO サーバに SSH で接続します。
2. 現在のディレクトリから CCO データベースを復元します。
   mongorestore
3. 復元後に Tomcat サービスを再起動します。
   /etc/init.d/tomcat restart