

# Intersight Kubernetesサービスを使用したKubernetesクラスタの設定

## 内容

[概要](#)

[背景説明](#)

[ソリューションの概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[前提](#)

[コンフィギュレーション](#)

[ステップ1: ポリシーの設定](#)

[ステップ2: プロファイルの設定](#)

[確認](#)

[Kubernetesクラスタへの接続](#)

[CLIによる確認](#)

[トラブルシューティング](#)

[関連情報](#)

## 概要

このドキュメントでは、Cisco Intersight™ Kubernetes Service(IKS)を使用して、Cisco Intersight(SaaS)から実稼働グレードのKubernetesクラスタをプロビジョニングするための設定について説明します。

## 背景説明

最近では、Kubernetesは事実上のコンテナ管理ツールになっています。これは、組織がコンテナ化ソリューションを使用してアプリケーションの最新化に投資する傾向があるためです。Kubernetesを使用すると、開発チームはコンテナ化されたアプリケーションを簡単に導入、管理、拡張できるため、継続的な配送パイプラインに対するイノベーションへのアクセスが向上します。

しかし、Kubernetesは、インストールと設定に時間と技術的な専門知識を必要とするため、運用上の課題を抱えています。

Kubernetesと、クラスタの作成、ストレージ、ネットワーク、セキュリティの設定、およびオペレーション(重要なセキュリティのバグのアップグレード、更新、パッチ適用など)のインストールには、人的資本に対する多大な投資が必要です。

IKSを入力します。IKSは、場所を問わず一貫した実稼働グレードのKubernetesを管理するためのターンキーSaaSソリューションです。IKSの機能の詳細については、こちらのリンクをご覧ください。

## ソリューションの概要

このドキュメントでは、VMware ESXiおよびvCenterを実行するオンプレミスのインフラストラクチャとシームレスに統合できるIKSの機能を紹介します。数回のクリックで、VMwareインフラストラクチャに実稼働グレードのKubernetesクラスタを導入できます。

しかし、そのためには、オンプレミスのvCenterとIntersightを統合する必要があります。これは「ターゲットの請求」と呼ばれ、ここでvCenterがターゲットになります。エンドポイントのターゲットをCisco Intersightに追加するのに役立つCisco Intersight Assist仮想アプライアンスが必要です。Intersight Assistは、シスコの公式WebサイトにあるブートストラップOVAを使用してインストールできます。

このドキュメントの範囲を限定するために、Cisco Intersight Assist仮想アプライアンスのインストールに焦点を当てていません。しかし、このプロセスを見ることができます

## 前提条件

### 要件

次の項目に関する知識があることが推奨されます。

- Intersightアカウント：有効なCisco IDとIntersightアカウントが必要です。シスコIDがない場合は、シスコのWebサイトでシスコIDを作成できます。次に、[Intersight]の[Create an Account]リンクをクリックします。
- Cisco Intersight Assist: Cisco Intersight Assistは、vCenter/ESXiをエンドポイントターゲットとしてCisco Intersightに追加するのに役立ちます。
- 接続: 環境でHTTP/Sプロキシがサポートされている場合は、これを使用してCisco Intersight Assistアプライアンスをインターネットに接続できます。または、サイト間URLへのポートを開く必要があります。詳細なネットワーク接続要件については、次のリンクを確認してください。
- Intersightで要求するvCenterクレデンシャル。

### 使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

### 前提

Cisco Intersightアプライアンスの導入は、このドキュメントの範囲外です。

既に有効なIntersightアカウントを持っており、事前にvCenter/Esxiを申請していると仮定します。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

# コンフィギュレーション

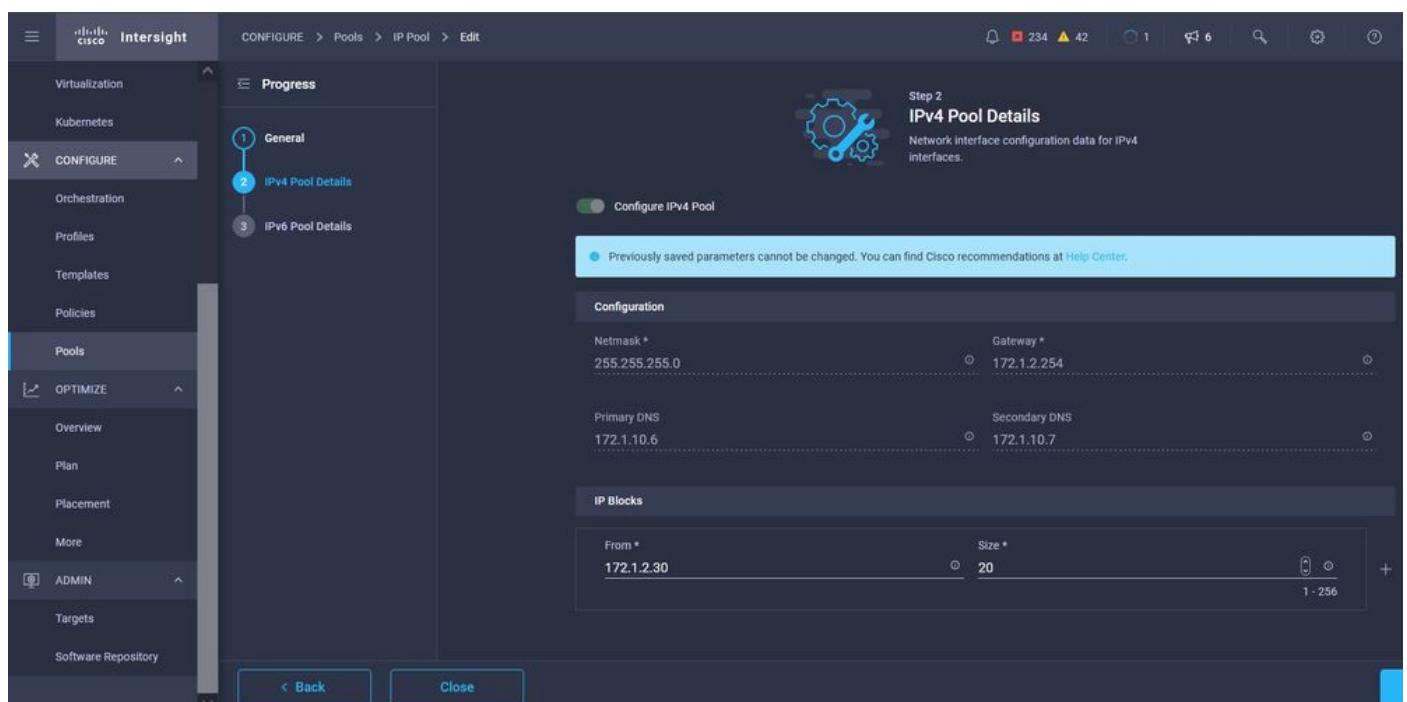
## ステップ1：ポリシーの設定

ポリシーにより、構成を再利用可能なテンプレートに抽象化できるため、管理が簡素化されます。

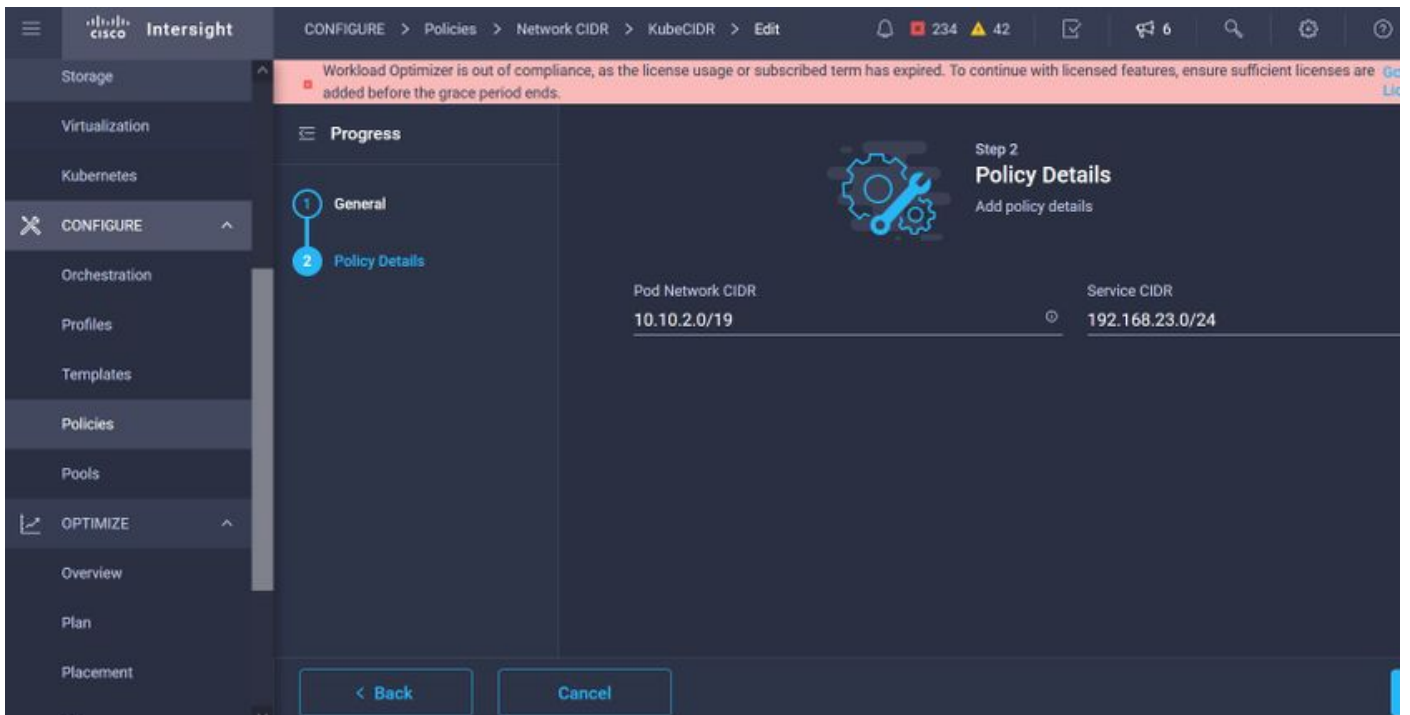
設定が必要なポリシーの一部を次に示します。これらのポリシーはすべて、Intersightの [Configure] > [Policies & Configure] >> [Pools]セクションで作成されます。

次に示すように、各スクリーンショットの上にポリシーのパスが表示されます。

このIPプールは、ESXiホストで起動したときに、制御およびワーカーノード仮想マシンのIPアドレスに使用されます。

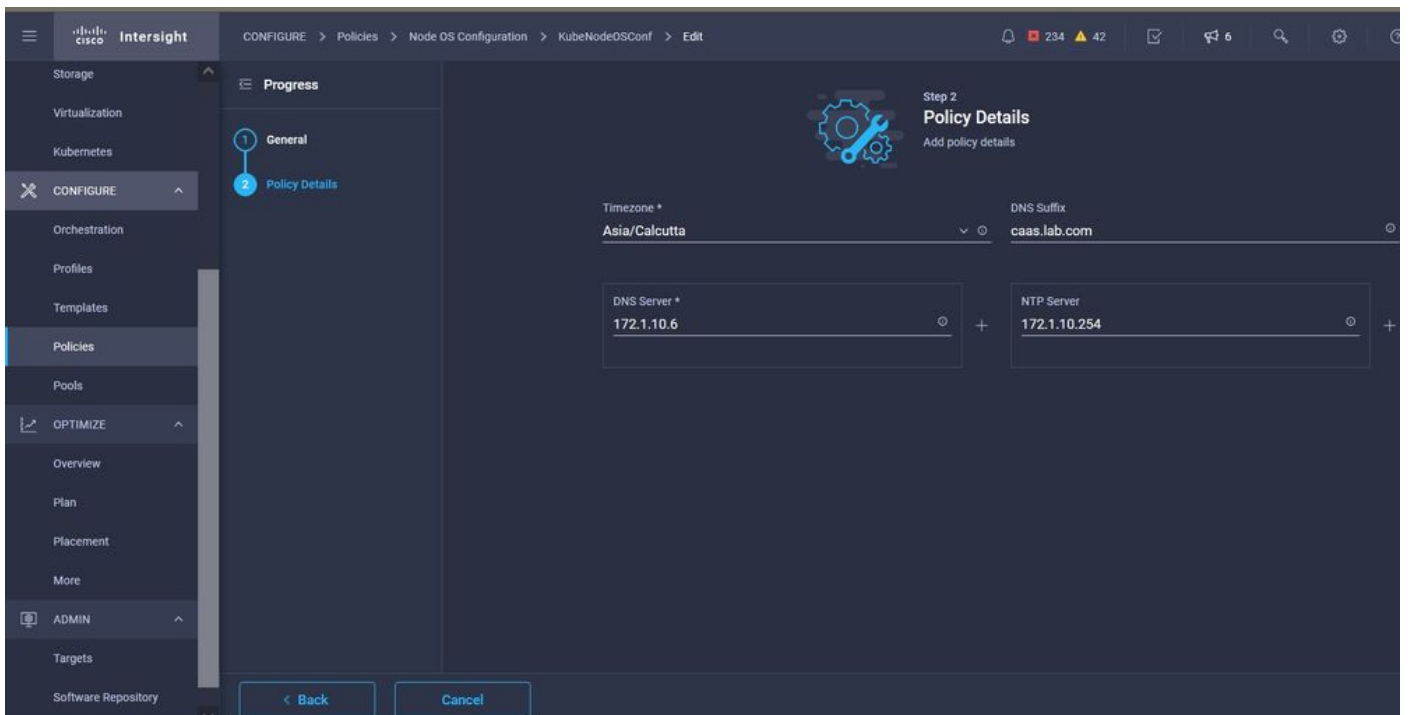


ここでは、Kubernetesクラスタ内の内部ネットワーキング用のポッドおよびサービスネットワークCIDRを定義します。



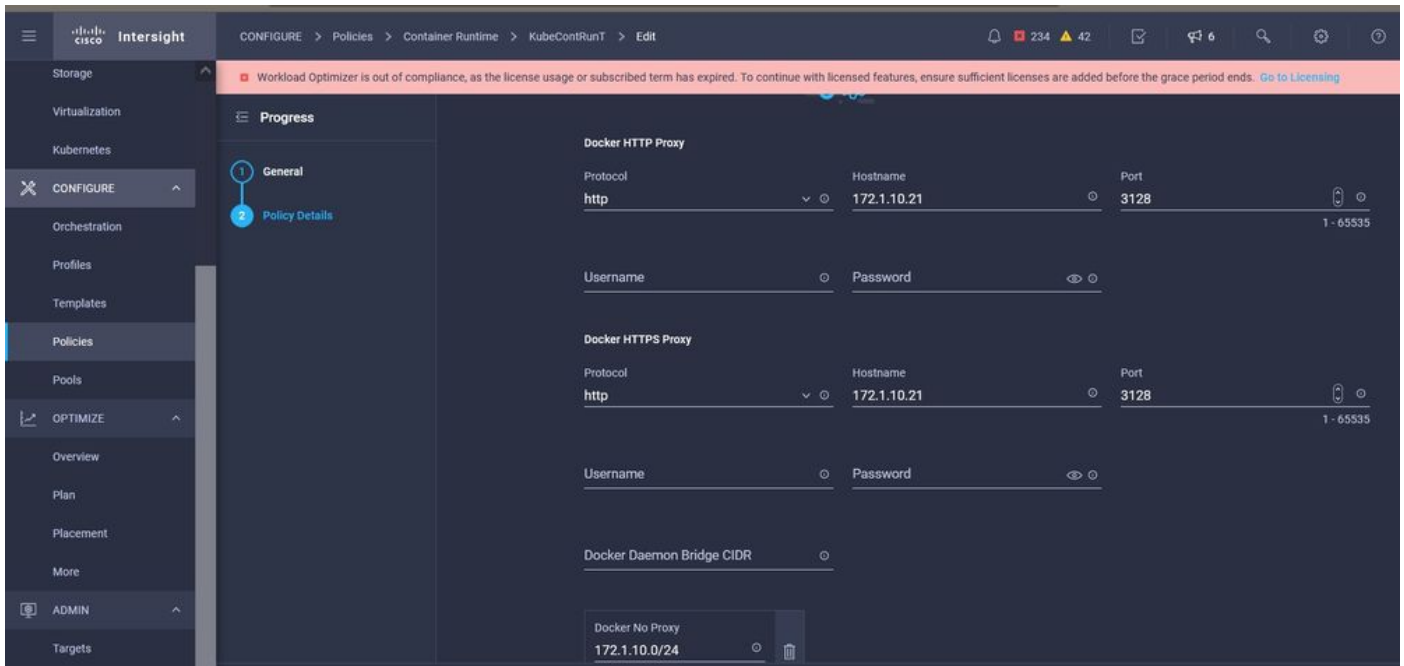
## サービスおよびネットワークCIDR

このポリシーは、NTPとDNSの設定を定義します。



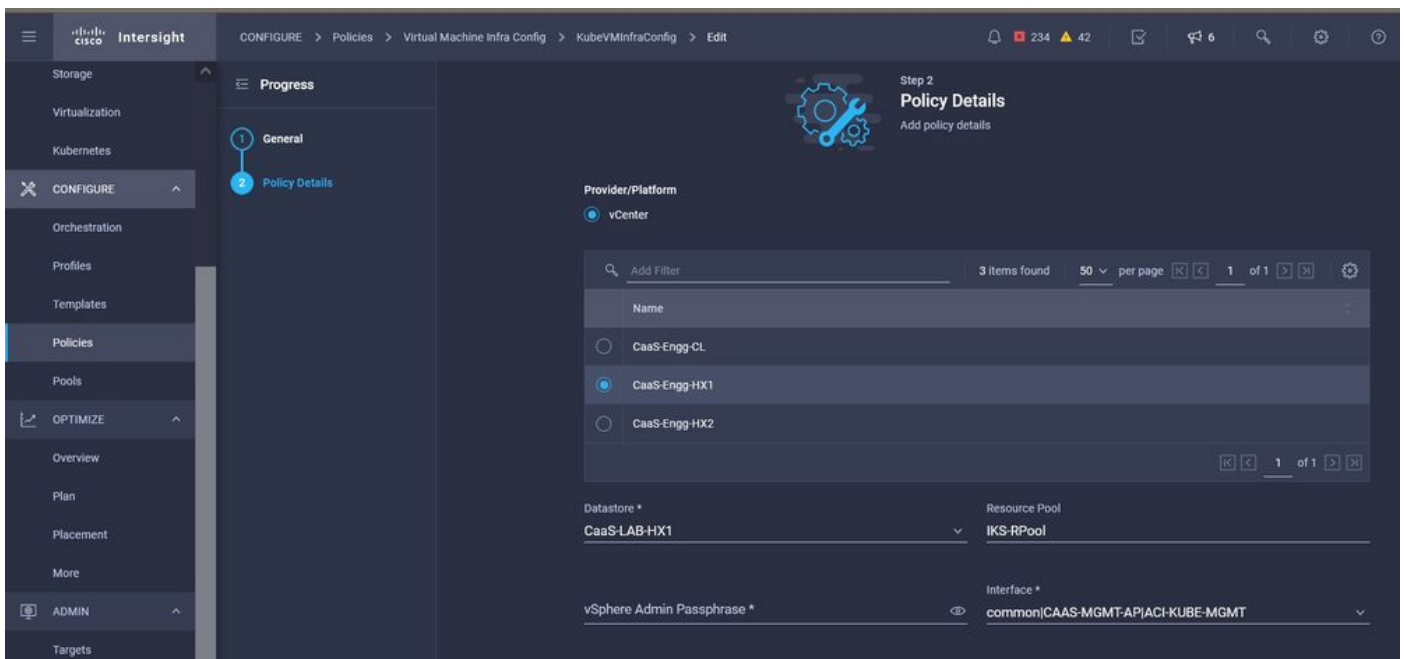
## NTPとDNSの設定

このポリシーを使用すると、dockerコンテナランタイムのプロキシ設定を定義できます。



## Dockerのプロキシ設定

このポリシーでは、マスターおよびワーカーノードとして展開された仮想マシンに必要な構成を定義します。



## 使用するVMの設定

## ステップ2: プロファイルの設定

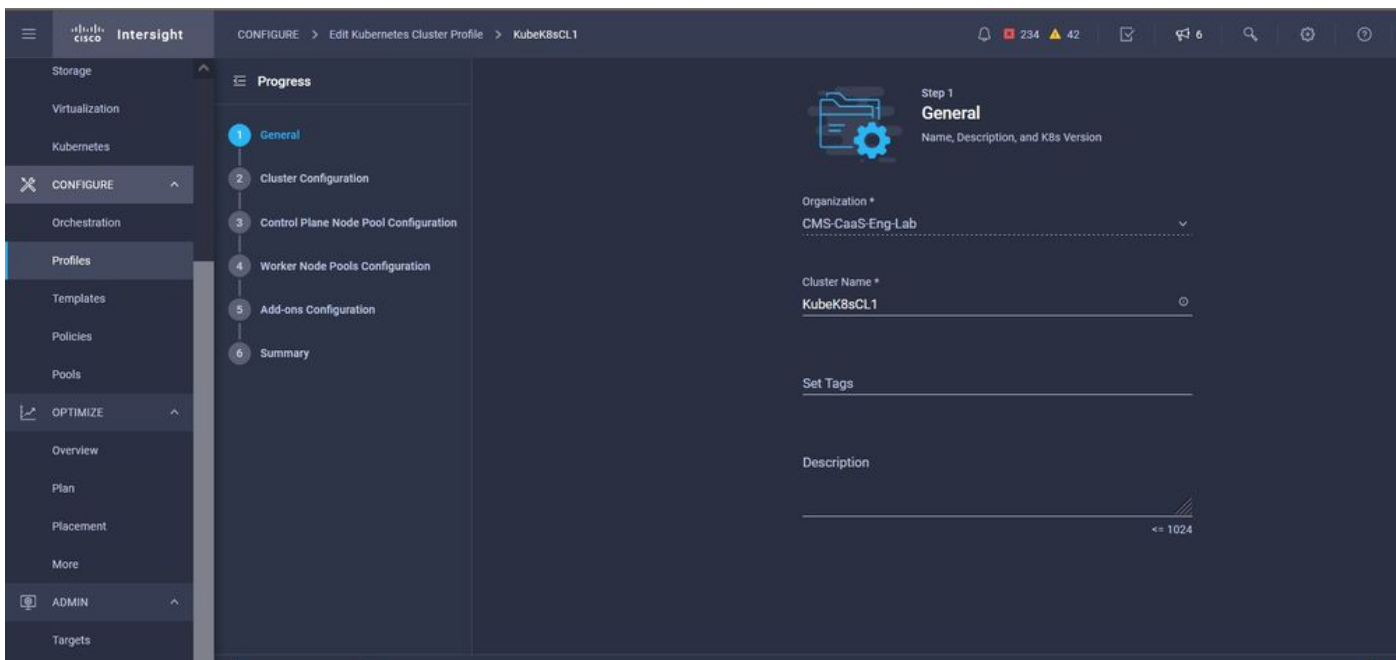
上記のポリシーを作成したら、それらをプロファイルにバインドし、展開します。

ポリシーとプロファイルを使用して設定を展開すると、設定レイヤが抽象化され、繰り返し迅速に展開できるようになります。

このプロファイルをコピーし、少なくともそれ以上の変更を基にした新しいプロファイルを数分で作成して、1つまたは複数のKubernetesクラスターを手動プロセスで必要な数分で作成できます。

。

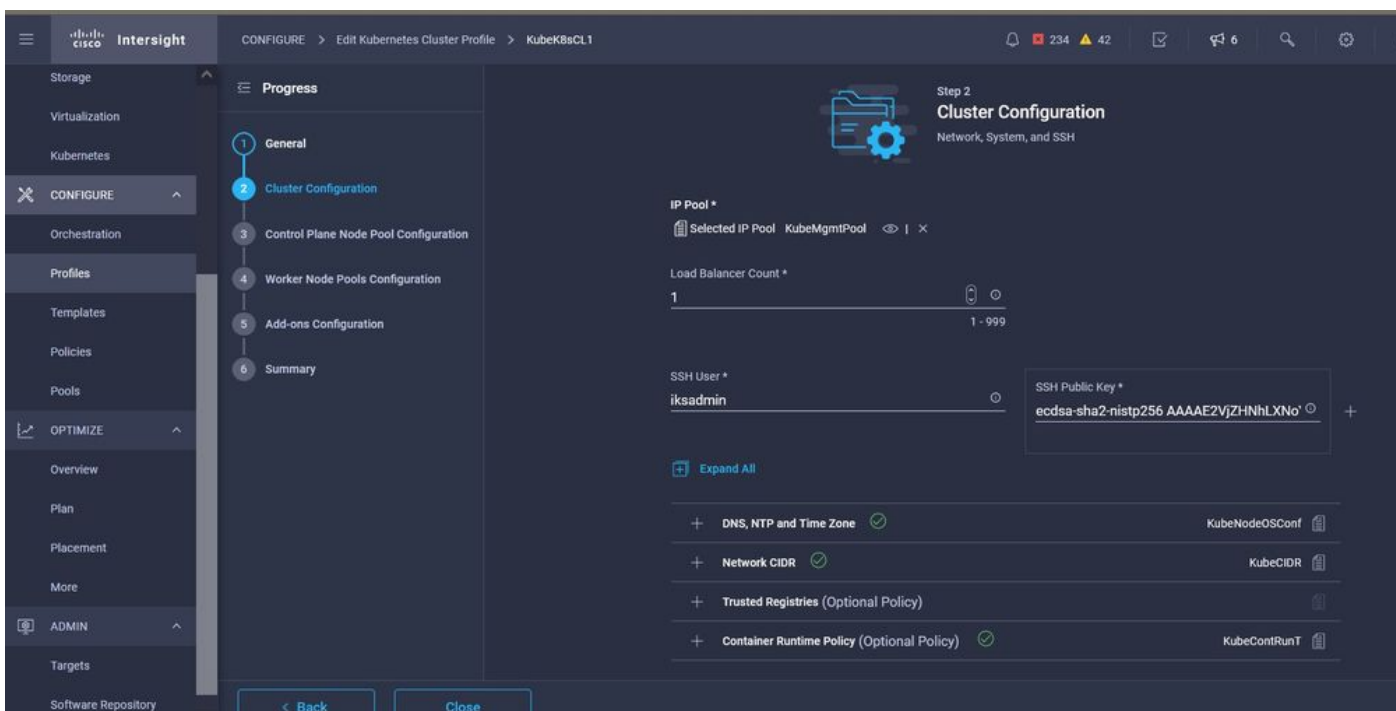
[名前]で[タグ]を設定します。



名前とタグのプロファイル設定

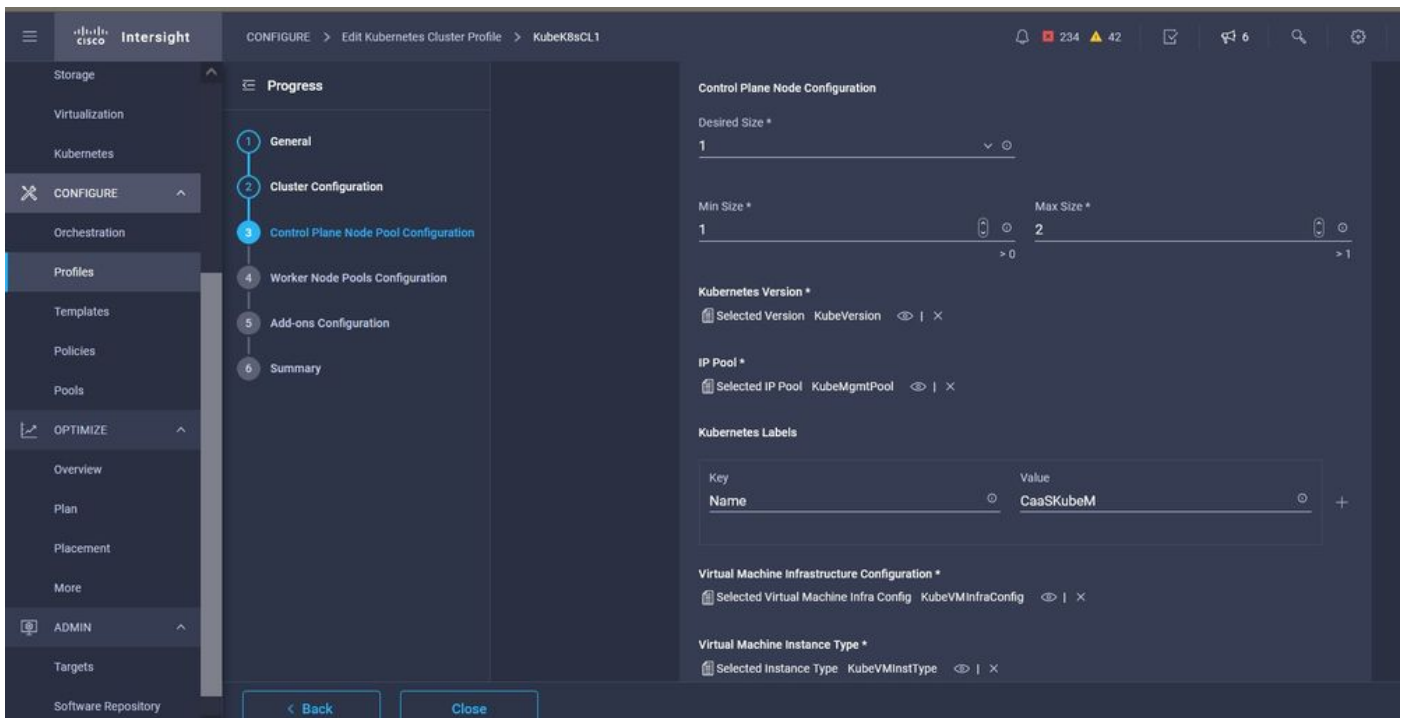
プール、ノードOS、ネットワークCIDRポリシーを設定します。ユーザIDとSSHキー(public)を設定する必要もあります。

対応する秘密キーは、マスターおよびワーカーノードへのsshに使用されます。



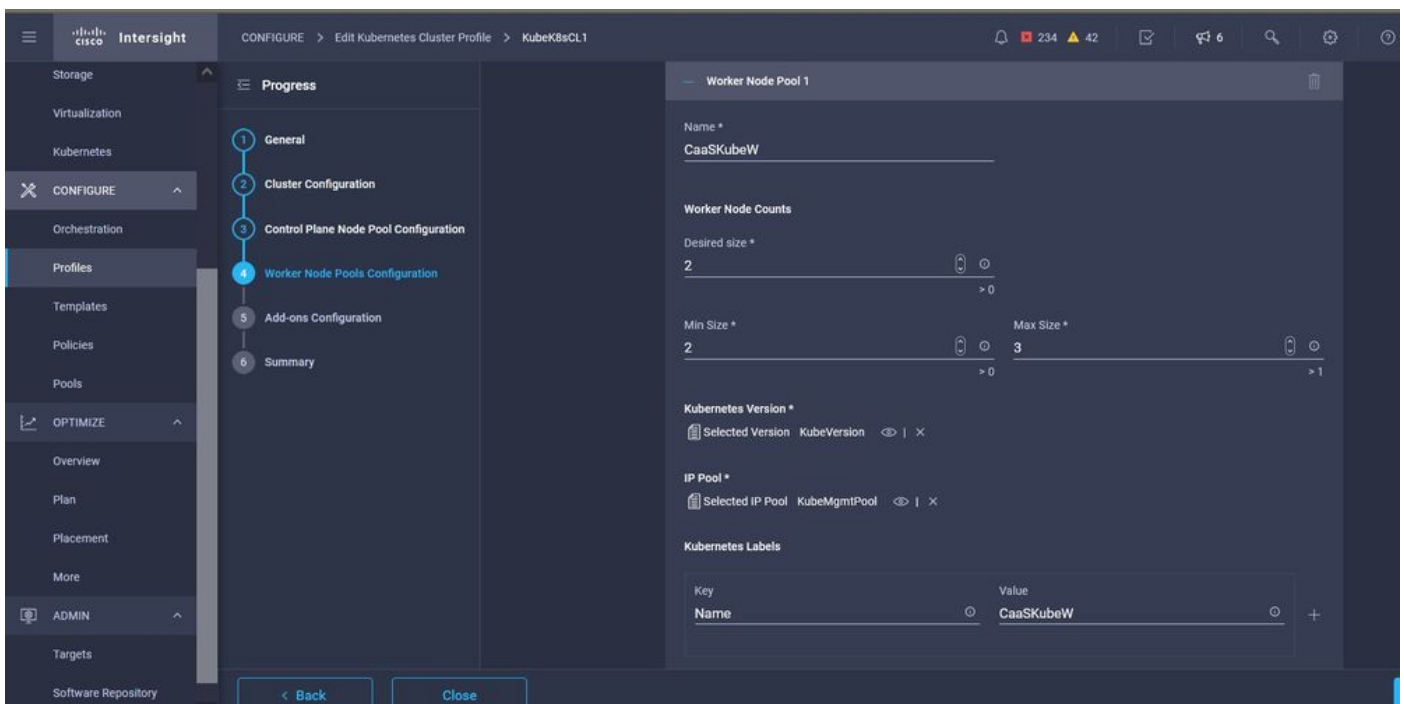
ポリシーが割り当てられたプロファイル設定

コントロールプレーンを設定します。コントロールプレーンで必要なマスターノードの数を定義できます。



## マスターノードの設定

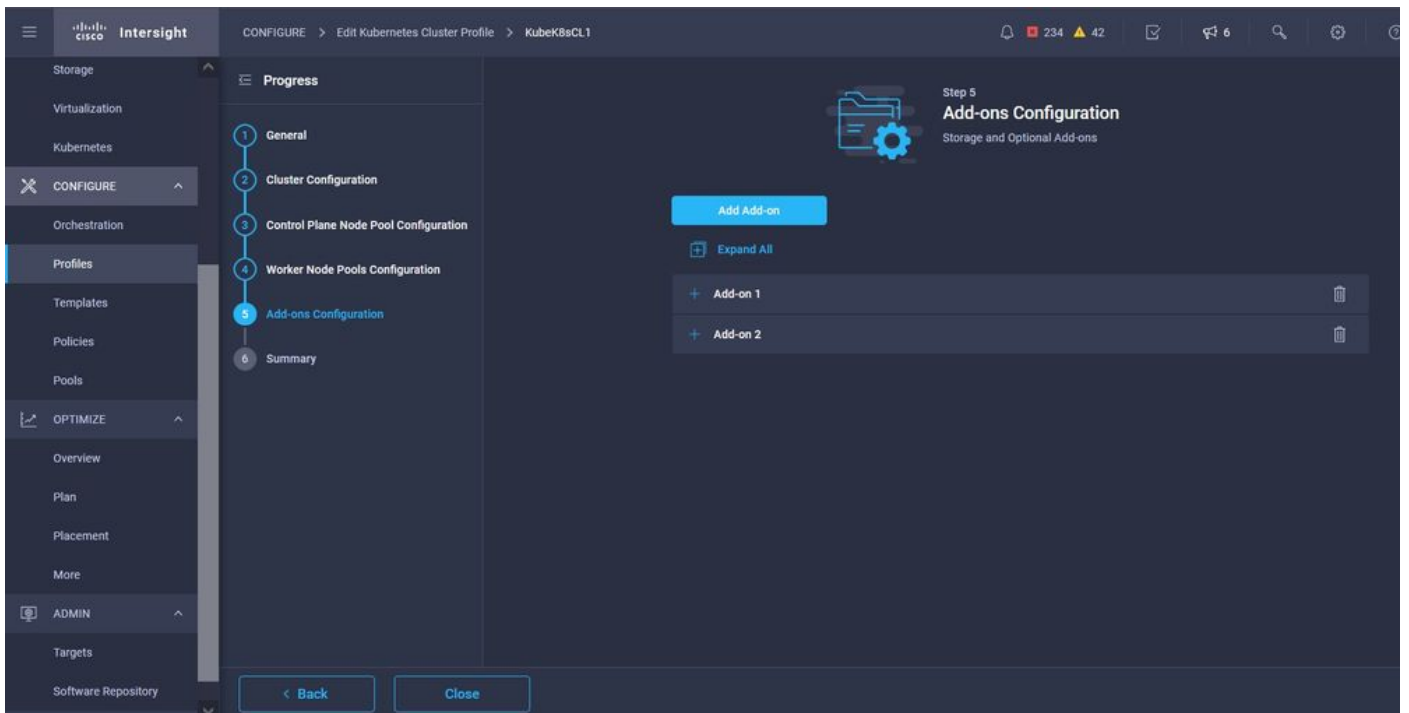
Workerノードを構成します。アプリケーション要件に応じて、ワーカーノードをスケールアップまたはスケールダウンできます。



## ワーカーノードの構成

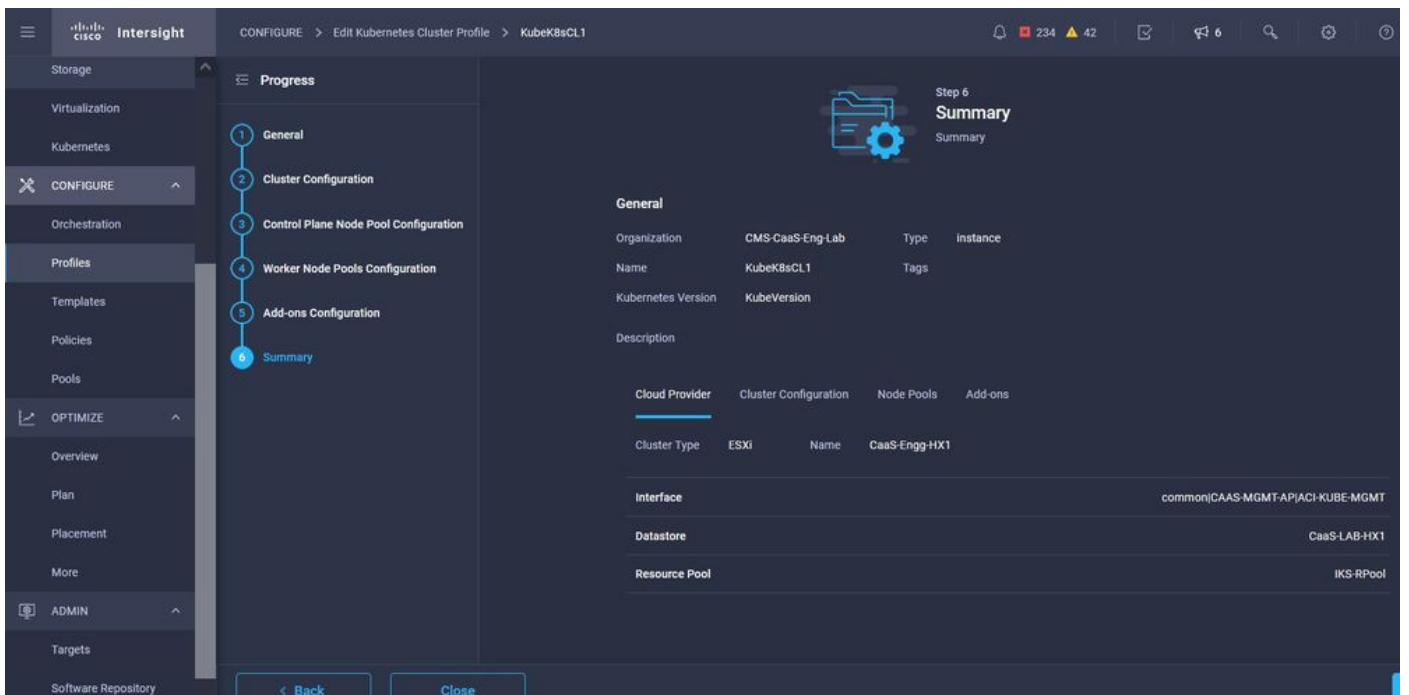
アドオンを設定します。現在、Kubernetes DashboardとGrafanaをPrometheusモニタリングで自動的に導入できます。

将来的には、IKSを使用して自動的に導入できるアドオンを追加できます。



アドオンがあれば追加

サマリーを確認し、[Deploy]をクリックします。



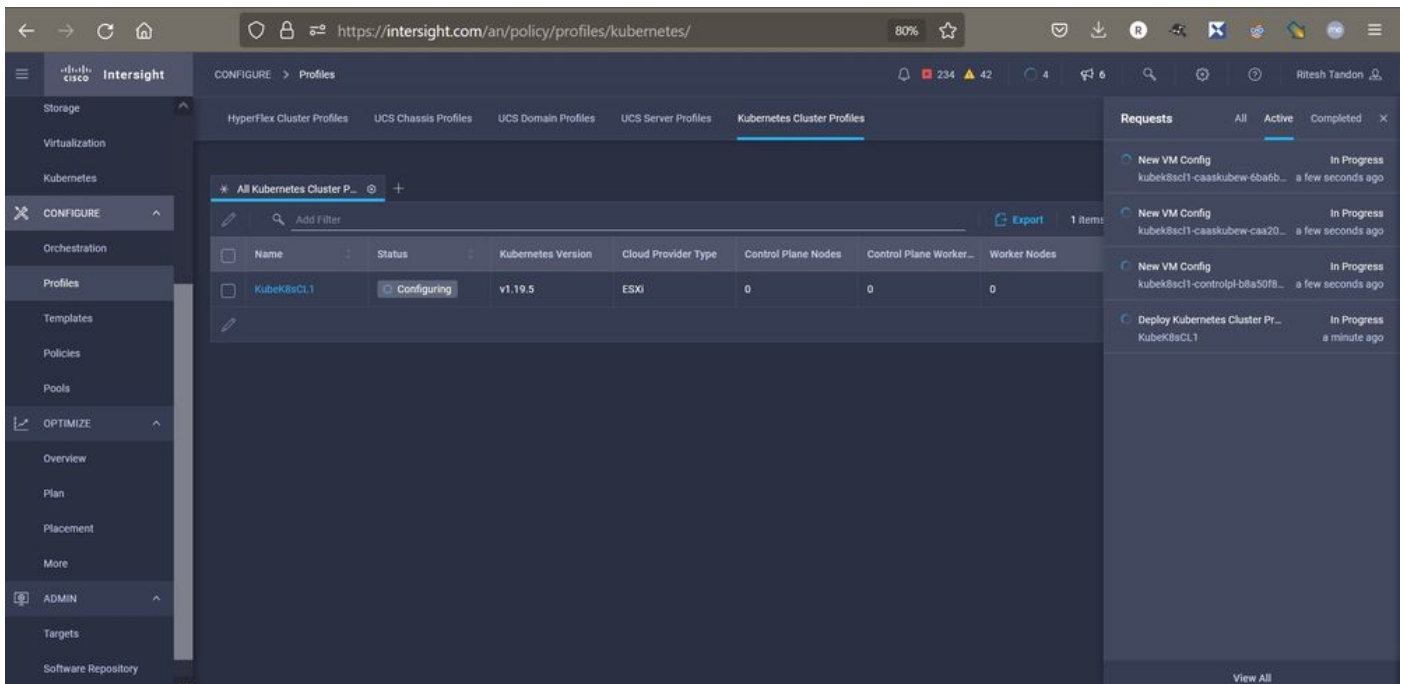
[Profile Creation Summary]画面

## 確認

ここでは、設定が正常に機能しているかどうかを確認します。

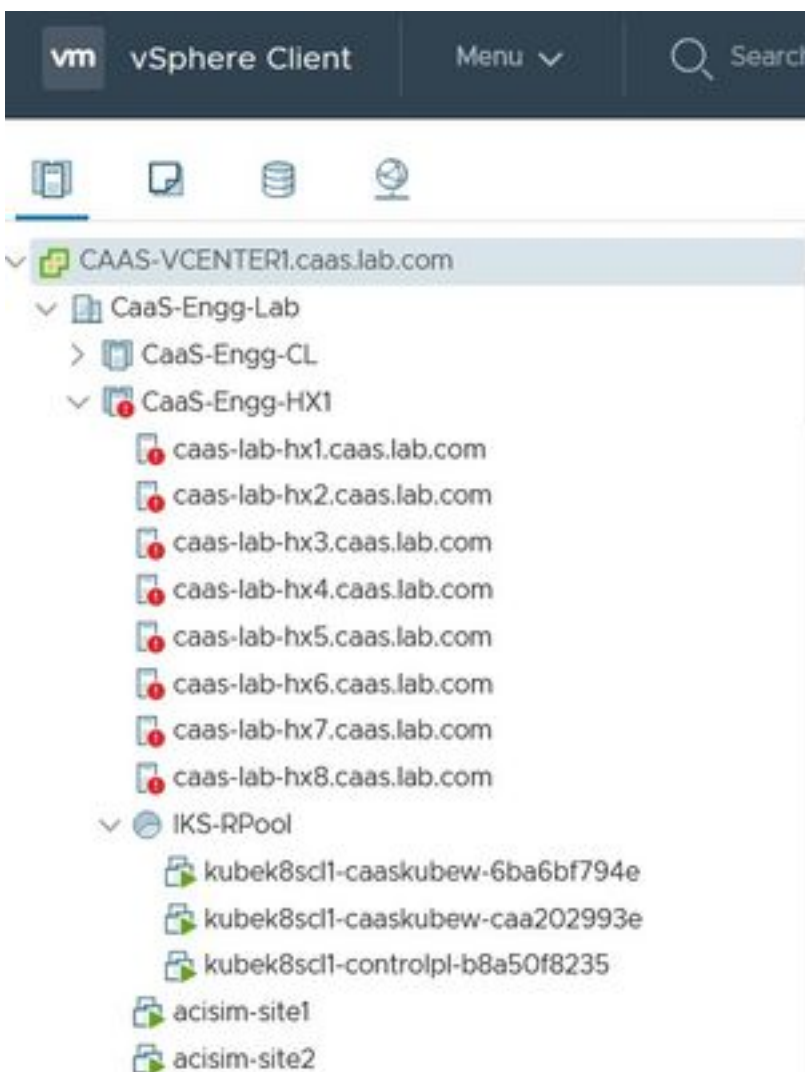
右上では、導入の進行状況を追跡できます。





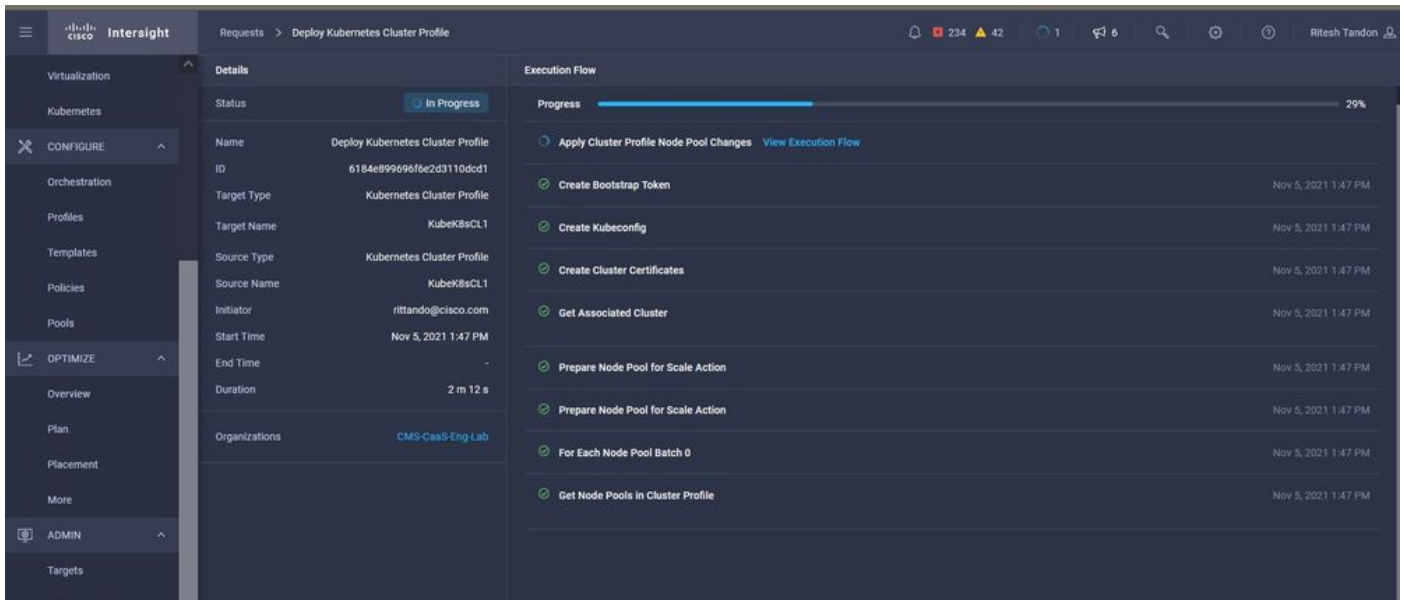
## IKS GUIを使用した確認

導入の進行に伴い、vCenterでKubernetes MasterノードとWorkerノードが表示されます。



vCenterで起動するIKSクラスタ

導入の詳細な手順を確認する必要がある場合は、実行をさらに詳しく説明できます。



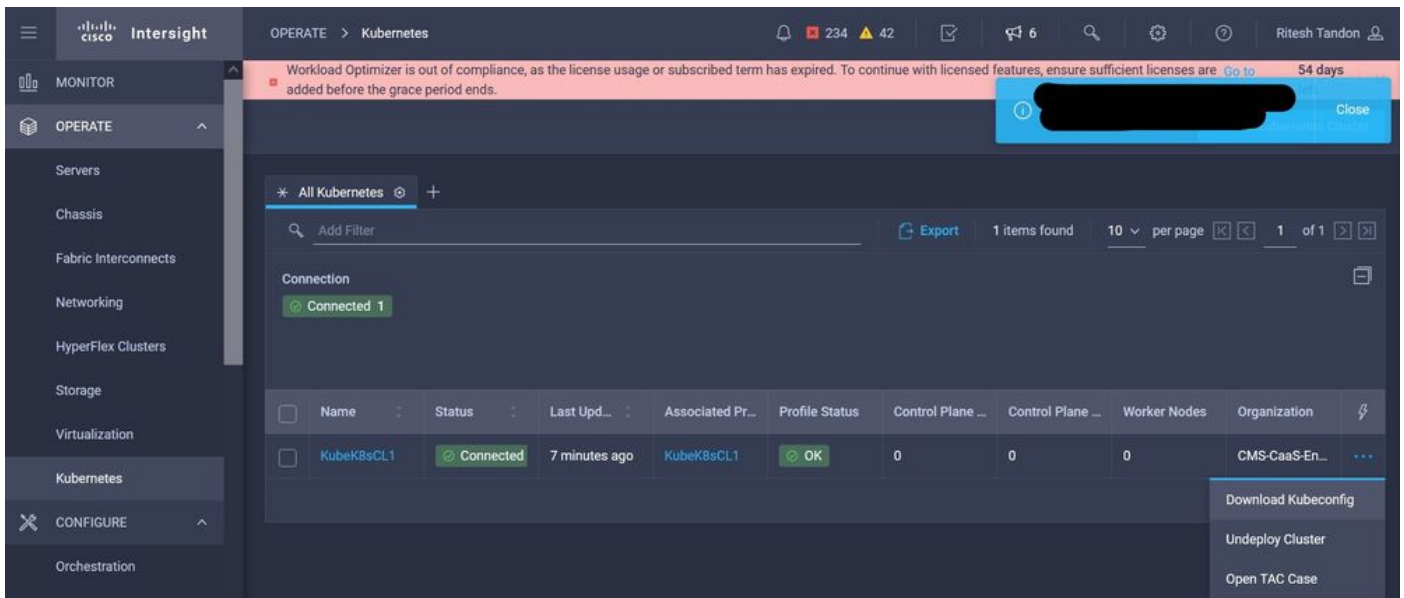
プロファイルの作成の実行

## Kubernetesクラスタへの接続

Kubernetesクラスタには、次の方法で接続できます。

KubeConfigファイルを使用します。Operate > Kubernetes > 右端のオプションを選択します。

このクラスタへのアクセス元の管理ワークステーションにKubeCtlがインストールされている必要があります。



IKSからKubeConfigファイルをダウンロードします

また、PuttyなどのSSHアプリケーションを使用して、導入時に設定されたクレデンシャルと秘密キーを使用して、マスターノードに直接SSH接続することもできます

アドオンとして「Kubernetes Dashboard」を導入すると、GUIを使用してアプリケーションを直接導入することもできます。

詳細については、「Kubernetes Clustersへのアクセス」セクションを参照して[ください](#)。

## CLIによる確認

kubectlを使用してKubernetesクラスタに接続できたら、次のコマンドを使用して、クラスタにすべてのコンポーネントがインストールされ、実行されているかどうかを確認できます。

クラスタ内のノードが「ready」状態であることを確認します。

```
iksadmin@kubek8scl1-controlpl-b8a50f8235:~$ kubectl get nodes NAME STATUS ROLES AGE VERSION
kubek8scl1-caaskubew-6ba6bf794e Ready
```

クラスタ上の必須コンポーネントのインストール時に作成されたポッドのステータスを確認します。

```
iksadmin@kubek8scl1-controlpl-b8a50f8235:~$ kubectl get pod -n iks | grep apply- apply-ccp-monitor-2b7tx 0/1 Completed 0 6d3h apply-cloud-provider-qczsj 0/1 Completed 0 6d3h apply-cni-g7dcc 0/1 Completed 0 6d3h apply-essential-cert-ca-jwdtk 0/1 Completed 0 6d3h apply-essential-cert-manager-bg5fj 0/1 Completed 0 6d3h apply-essential-metallb-nzj7h 0/1 Completed 0 6d3h apply-essential-nginx-ingress-8qrnq 0/1 Completed 0 6d3h apply-essential-registry-f5wn6 0/1 Completed 0 6d3h apply-essential-vsphere-csi-tjfnq 0/1 Completed 0 6d3h apply-kubernetes-dashboard-rslt4 0/1 Completed 0 6d3h
```

ローカルで実行されているヘルムを管理し、アドオンをインストールするccp-helm-operatorポッドのステータスを確認します。

```
iksadmin@kubek8scl1-controlpl-b8a50f8235:~$ kubectl get helmcharts.helm.ccp.----.com -A
NAMESPACE NAME STATUS VERSION INSTALLED VERSION SYNCED iks ccp-monitor INSTALLED 0.2.61-helm3
iks essential-cert-ca INSTALLED 0.1.1-helm3 iks essential-cert-manager INSTALLED v1.0.2-cisco-helm3
iks essential-metallb INSTALLED 0.12.0-cisco3-helm3 iks essential-nginx-ingress INSTALLED 2.10.0-cisco2-helm3
iks essential-registry INSTALLED 1.8.3-cisco10-helm3 iks essential-vsphere-csi INSTALLED 1.0.1-helm3
iks kubernetes-dashboard INSTALLED 3.0.2-cisco3-helm3 iks vsphere-cpi INSTALLED 0.1.3-helm3
iksadmin@kubek8scl1-controlpl-b8a50f8235:~$ helm ls -A WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /home/iksadmin/.kube/config
NAME NAMESPACE REVISION UPDATED STATUS CHART APP VERSION addon-operator iks 1 2021-11-05 07:45:15.44180913 +0000 UTC deployed ccp-helm-operator-9.1.0-alpha.44.g415a48c4bel.0 ccp-monitor
iks 1 2021-11-05 08:23:11.309694887 +0000 UTC deployed ccp-monitor-0.2.61-helm3 essential-cert-ca iks 1 2021-11-05 07:55:04.409542885 +0000 UTC deployed cert-ca-0.1.1-helm3 0.1.0 essential-cert-manager iks 1 2021-11-05 07:54:41.433212634 +0000 UTC deployed cert-manager-v1.0.2-cisco-helm3 v1.0.2 essential-metallb iks 1 2021-11-05 07:54:48.799226547 +0000 UTC deployed metallb-0.12.0-cisco3-helm3 0.8.1 essential-nginx-ingress iks 1 2021-11-05 07:54:46.762865131 +0000 UTC deployed ingress-nginx-2.10.0-cisco2-helm3 0.33.0 essential-registry iks 1 2021-11-05 07:54:36.734982103 +0000 UTC deployed docker-registry-1.8.3-cisco10-helm3 2.7.1 essential-vsphere-csi kube-system 1 2021-11-05 07:54:58.168305242 +0000 UTC deployed vsphere-csi-1.0.1-helm3 v2.0.0 kubernetes-dashboard iks 1 2021-11-05 07:55:10.197905183 +0000 UTC deployed kubernetes-dashboard-3.0.2-cisco3-helm3 2.1.0 vsphere-cpi kube-system 1 2021-11-05 07:54:38.292088943 +0000 UTC deployed vsphere-cpi-0.1.3-helm3 1.1.0
```

すべてのIKSテナントクラスタにインストールされているEssential(core)アドオンを管理するessential-\*ポッドのステータスを確認します。

```
iksadmin@kubek8scl1-controlpl-b8a50f8235:~$ kubectl get pod -n iks | grep ^essential- essential-cert-manager-6bb7d776d-tpkhj 1/1 Running 0 6d4h essential-cert-manager-cainjector-549c8f74c-x5sjp 1/1 Running 0 6d4h essential-cert-manager-webhook-76f596b686-drf79 1/1 Running 0 6d4h essential-metallb-controller-6557847d57-djs9b 1/1 Running 0 6d4h essential-metallb-speaker-7t54v 1/1 Running 0 6d4h essential-metallb-speaker-ggmbn 1/1 Running 0 6d4h essential-metallb-speaker-mwmfg 1/1 Running 0 6d4h essential-nginx-ingress-ingress-nginx-controller-k2hsw 1/1 Running 0 6d4h essential-nginx-ingress-ingress-nginx-controller-kfkm9 1/1 Running 0 6d4h essential-nginx-ingress-ingress-nginx-defaultbackend-695fbj4mnd 1/1 Running 0 6d4h essential-registry-docker-registry-75b84457f4-4fmlh 1/1 Running 0 6d4h
```

IKS名前空間に展開されているサービスとロードバランサの状態を確認します。

```
iksadmin@kubek8sctl1-controlpl-b8a50f8235:~$ kubectl get svc -n iks NAME TYPE CLUSTER-IP  
EXTERNAL-IP PORT(S) AGE ccp-monitor-grafana ClusterIP 192.168.23.161
```

## トラブルシューティング

ここでは、設定のトラブルシューティングに使用できる情報を示します。

特定のポッドが起動しない場合は、次のコマンドを使用して原因をドリルダウンできます。

Syntax : `kubectl describe pod`

## 関連情報

- [IKSサービス概要はこちらで確認してください。](#)
- [ユーザーガイドを確認してください。](#)
- [Intersight Kubernetesサービスのデモを確認してください。](#)
- [テクニカル サポートとドキュメント - Cisco Systems](#)