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

内容

はじめに

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

Kubernetesは最近、事実上のコンテナ管理ツールになっています。組織は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より多くの投資を行う傾向があるためです。
Kubernetesを使用することで、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デプロイ、管理、およびスケーリングでき、継続的なデリバリーパイプラインにイノベーションをアクセスしやすくします。

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、運用上の課題があります。
Kubernetesと必要なさまざまなソフトウェアコンポーネントのインストール、クラスタの作成、ストレージ、ネットワーキング、セキュリティの設定、および運用（重要なセキュリティバグのアップグレード、更新、パッチ適用など）には、多大な人的投資を継続的に行う必要があります。

IKSを導入します。これは、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、ここを参照してください。

ソリューションの概要

このドキュメントの目的は、VMware ESXiおよびvCenterを実行するオンプレミスインフラストラクチャとシームレスに統合できるIKSの機能を紹介することです。
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。

ただし、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「ターゲットの要求」と呼ばれ、vCenterがターゲットになります。
Cisco Intersightにエンドポイントターゲットを追加するのに役立つCisco Intersight Assist Virtual Applianceが必要です。シスコの公式Webサイトで入手できるブートストラップOVAを使用して、Intersight Assistをインストールできます。

このドキュメントの対象範囲を限定するため、Cisco Intersight Assist仮想アプライアンスのインストールについては説明しません。ただし、ここでプロセスを確認できます。

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：有効なシスコIDとIntersightアカウントが必要です。
  シスコのWebサイトにシスコIDがない場合は、作成できます。次に、IntersightでCreate an Accountリンクをクリックします。
- Cisco Intersight Assist:Cisco Intersight Assistは、vCenter/ESXiをエンドポイントのターゲットとしてCisco Intersightに追加するのに役立ちます。
- 接続：ご使用の環境でHTTP/Sプロキシがサポートされている場合は、それを使用して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、URLをインターサイトするためにポートを開く必要があります。ネットワーク接続要件の詳細については、次のリンクを確認してください。
- Intersightで請求するための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については、このドキュメントでは取り扱いません。

すでにIntersightアカウントを使用しており、オンプレミスのvCenter/Esxiを正常に要求している

と想定しています。

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、クリアな（デフォルト）設定で作業を開始しています。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

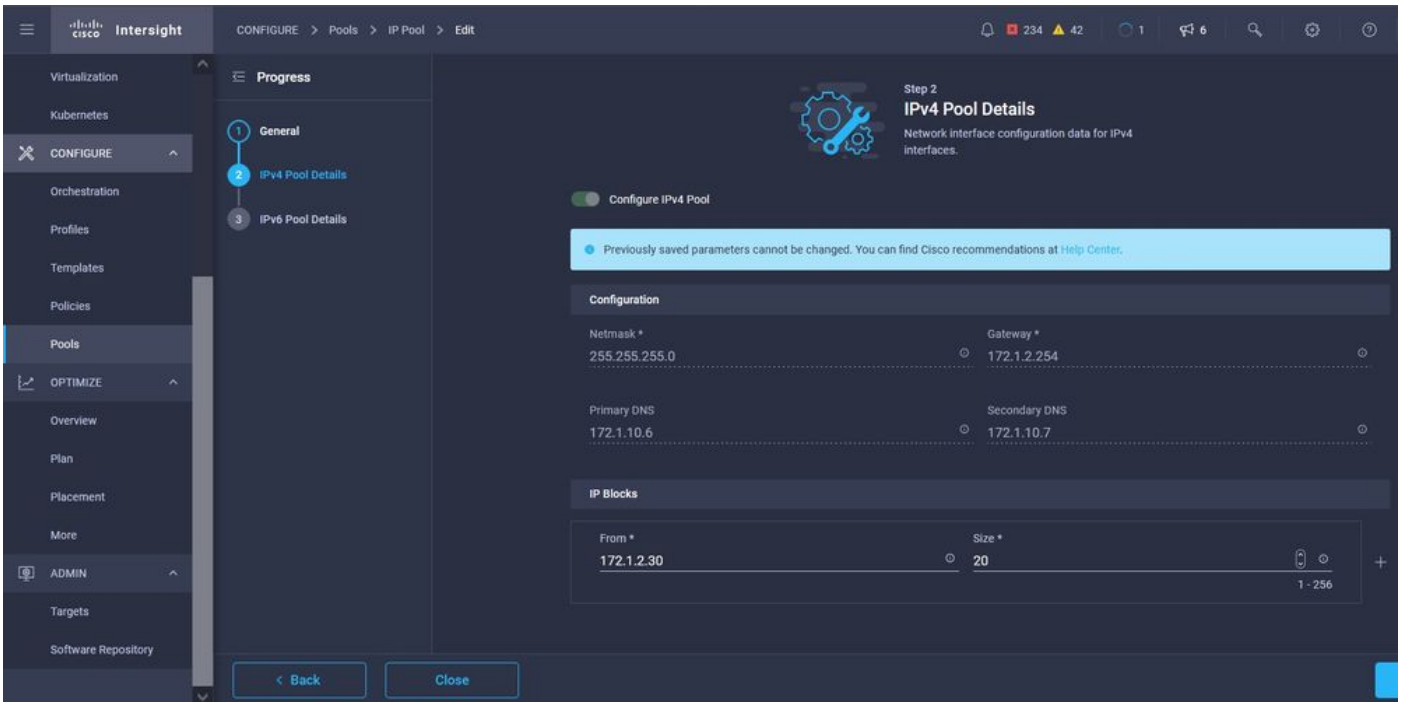
ステップ 1：ポリシーの設定

ポリシーを使用すると、構成が再利用可能なテンプレートに抽象化されるため、管理が簡素化されます。

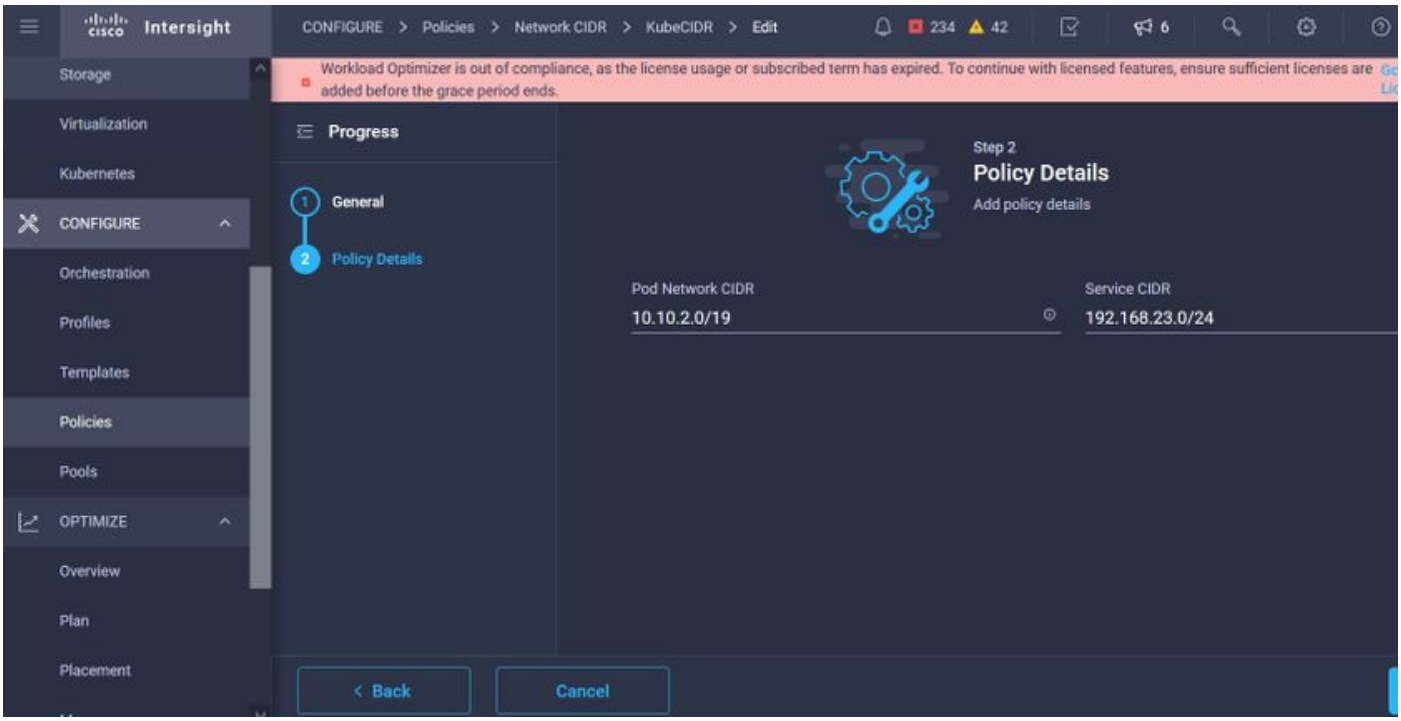
設定する必要があるポリシーの一部を次に示します。これらすべてのポリシーは、IntersightのConfigure >> Policies & Configure >> Poolsセクションに作成されることに注意してください。

各スクリーンショットの上部にポリシーのパスが表示されます（下図を参照）。

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

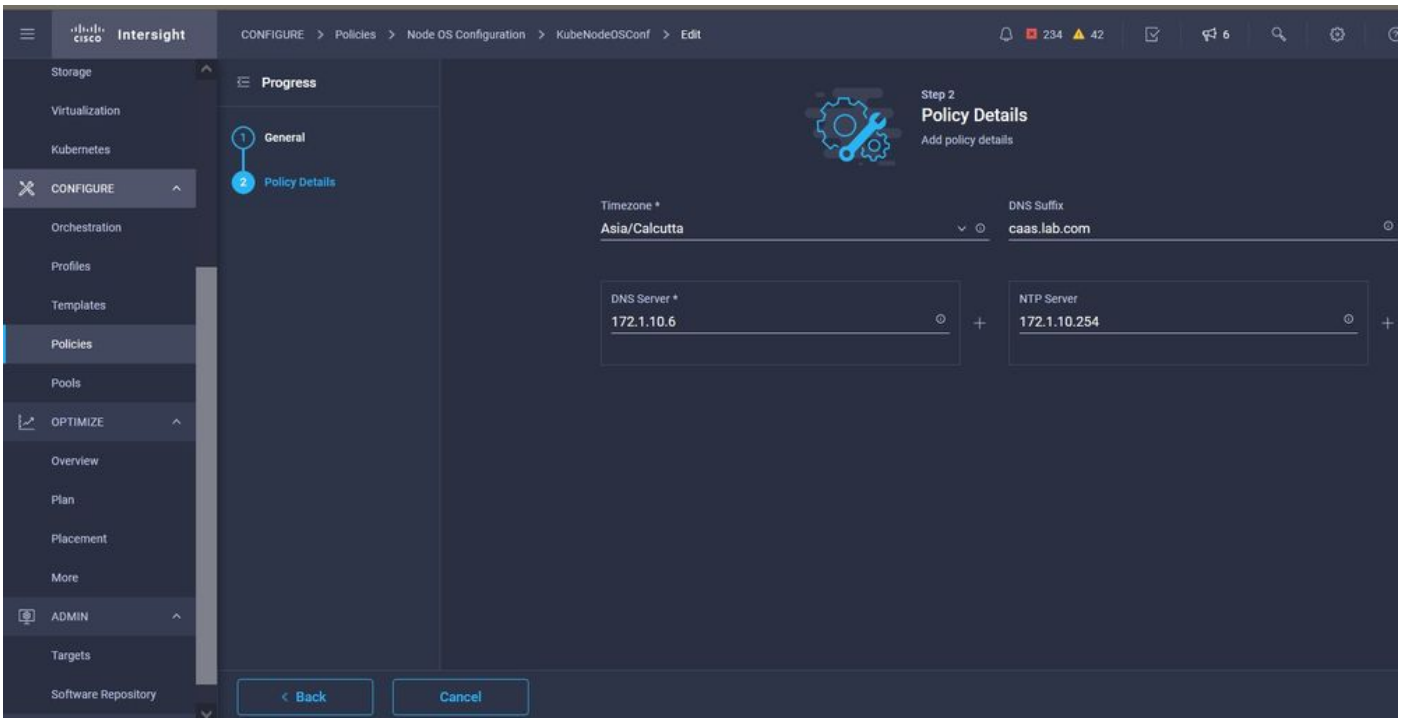


ここでは、Kubernetesクラスタ内の内部ネットワーキング用のPod and Services Network CIDRを定義します。



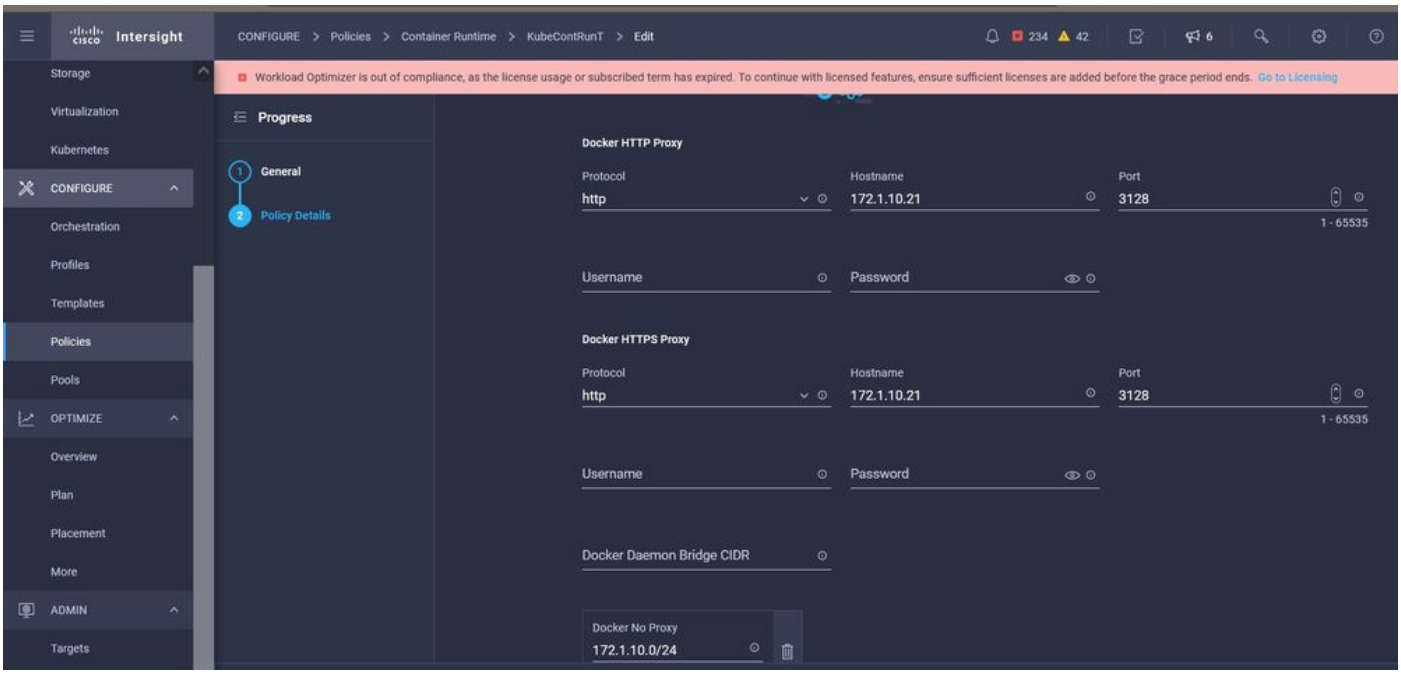
サービスおよびネットワークCIDR

このポリシーは、NTPとDNSの設定を定義します。



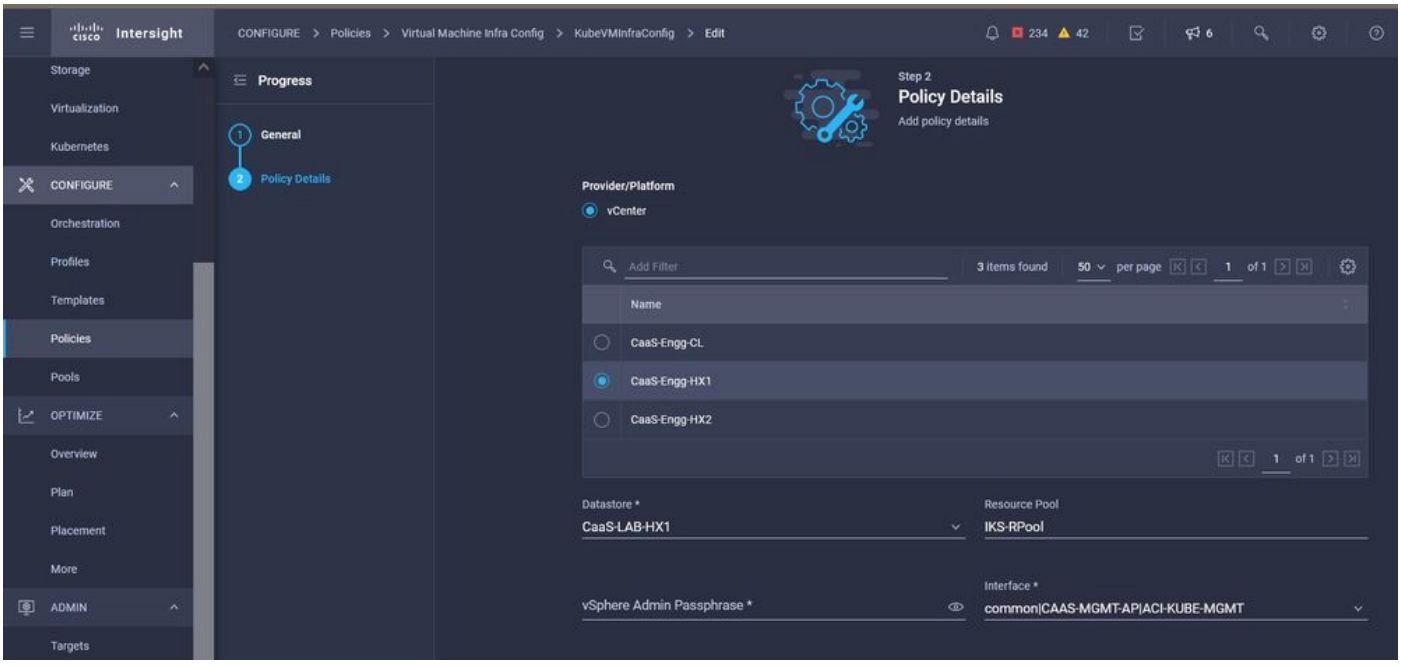
NTPとDNSの設定

このポリシーを使用して、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、マスターノードおよびワーカーノードとして展開された仮想マシンで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

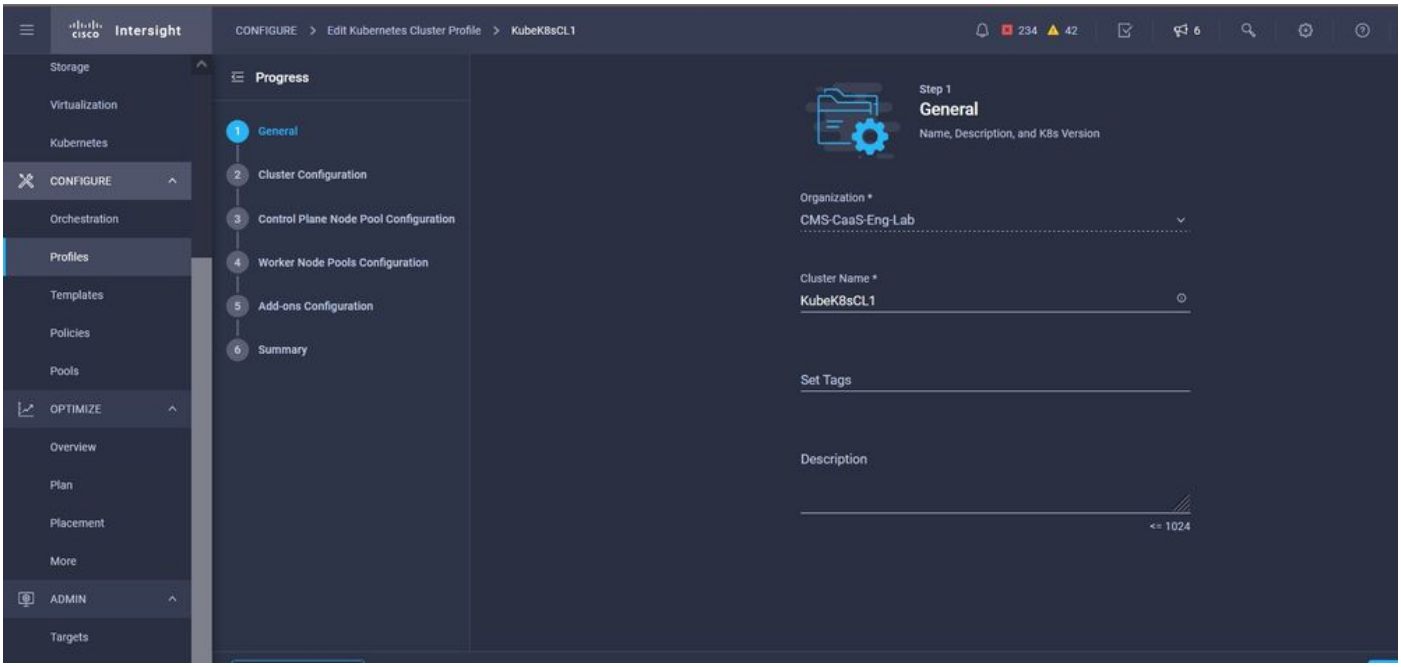
上記のポリシーを作成したら、それらをプロファイルにバインドして展開します。

ポリシーとプロファイルを使用して構成を展開すると、構成層が抽象化されるため、繰り返し迅速に展開できます。

このプロファイルをコピーして、基盤となるポリシーの変更をほとんど伴わない新しいプロファイルを数分で作成し、手動プロセスで必要な数分で1つ以上のKubernetesクラスターを作成でき

ます。

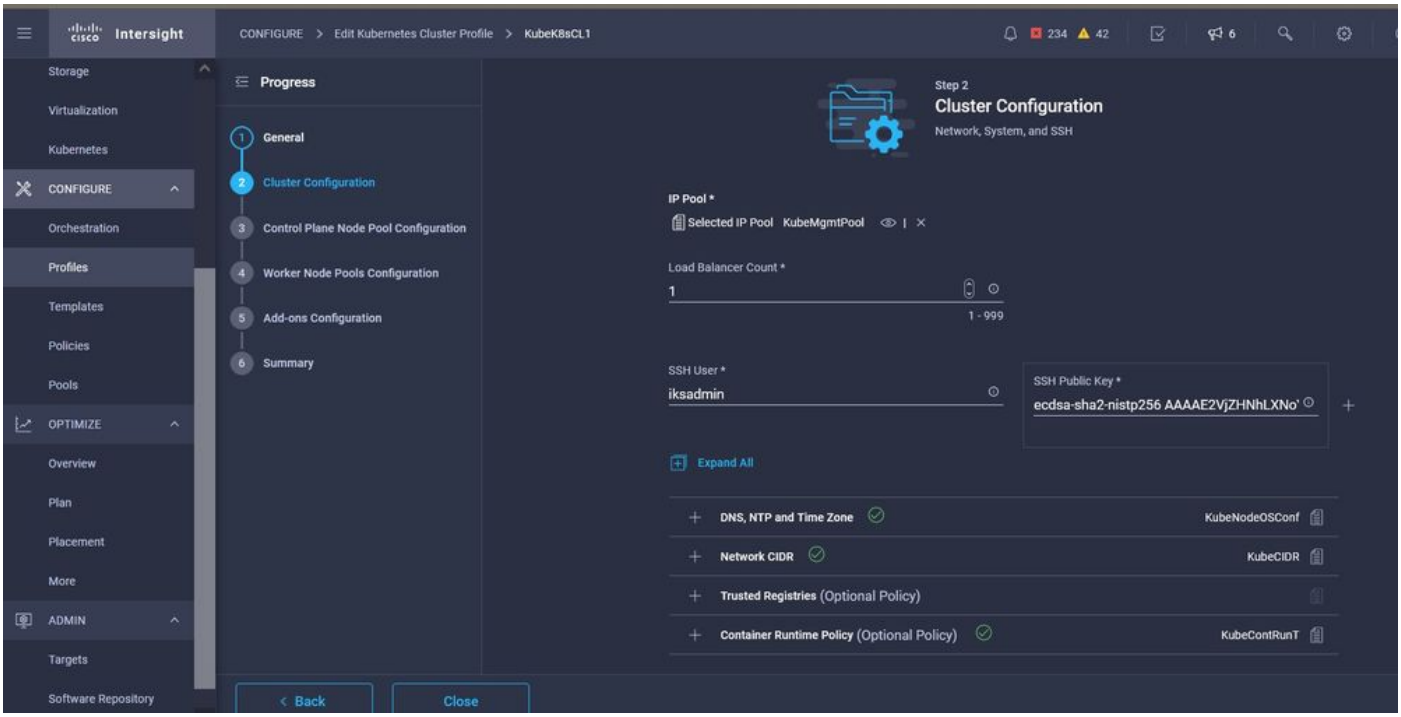
名前を入力し、タグを設定します。



名前とタグを使用したプロファイル設定

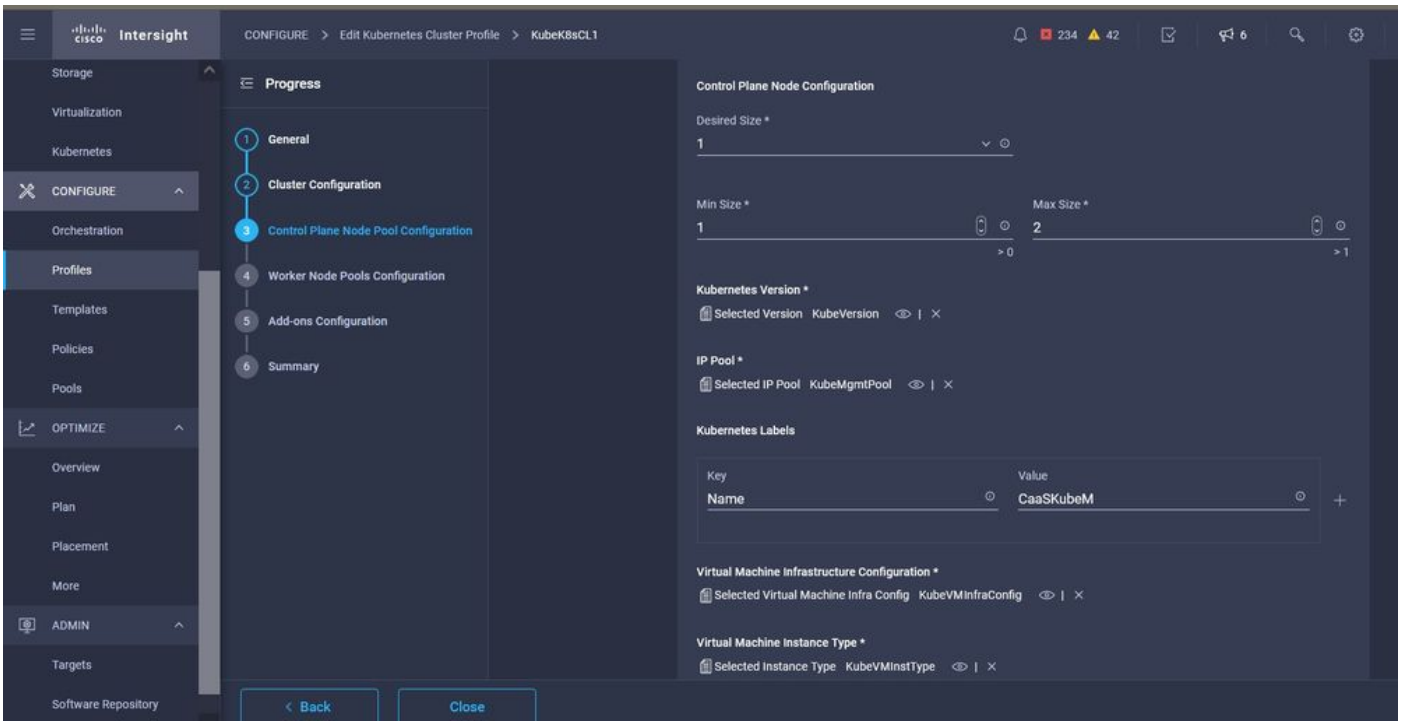
プール、ノードOS、ネットワークCIDRポリシーを設定します。また、ユーザIDとSSHキー(public)を設定する必要もあります。

対応する秘密キーを使用して、MasterおよびWorkerノードにssh接続します。



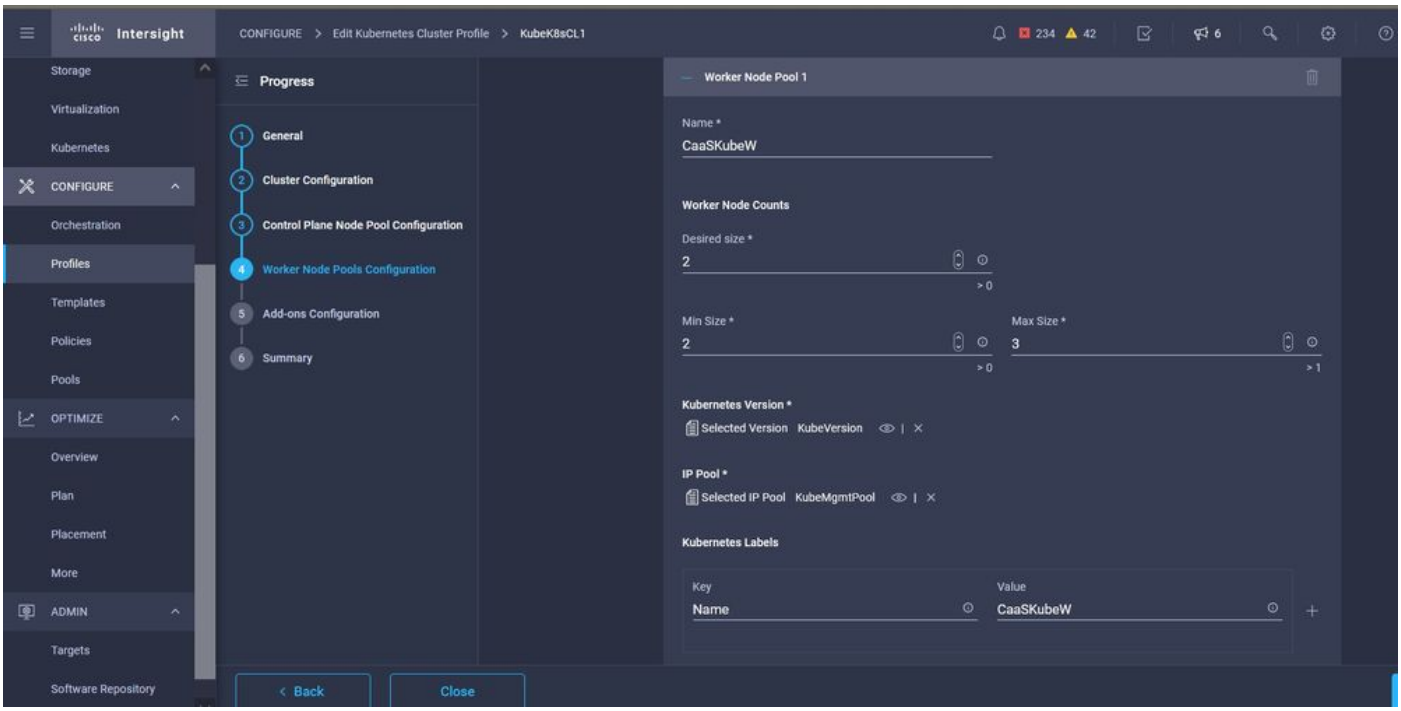
ポリシーが割り当てられたプロファイル構成

コントロールプレーンの設定：コントロールプレーンに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

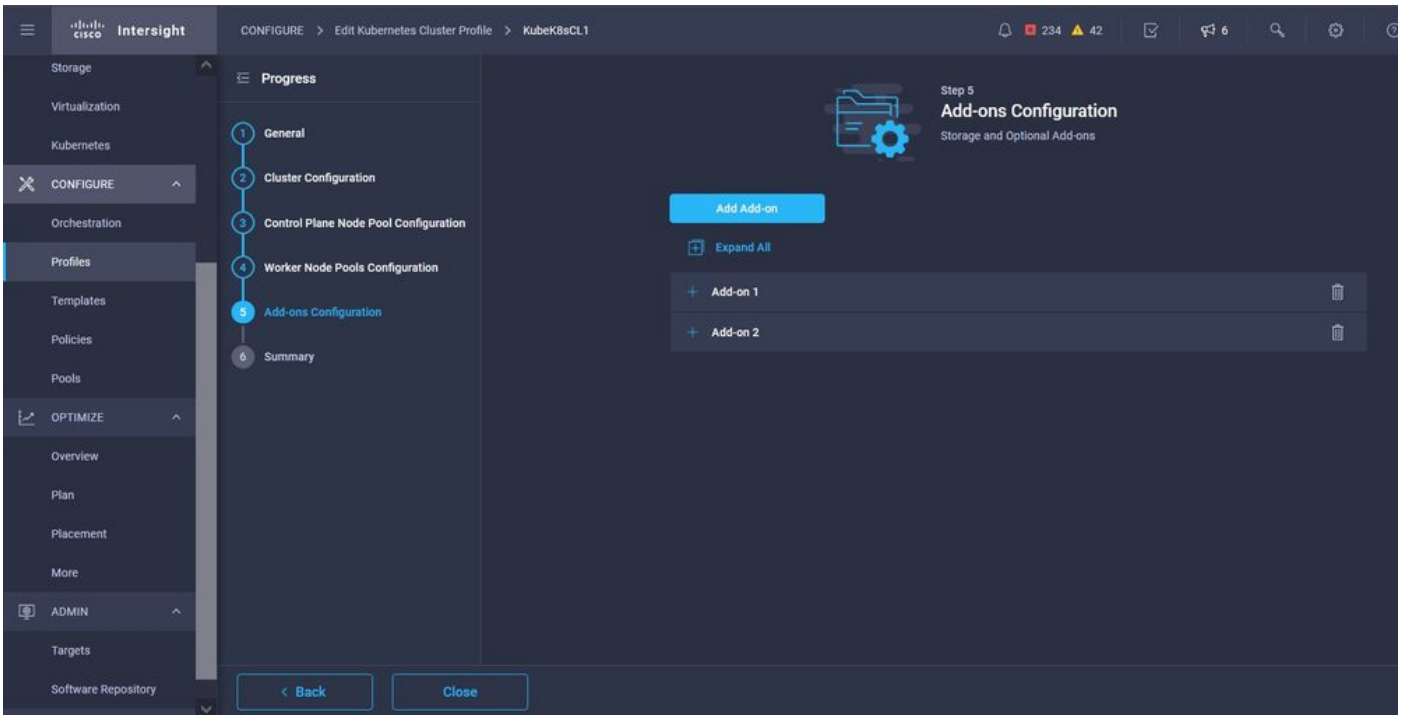
ワーカーノードの構成：アプリケーションの要件に応じて、ワーカーノードをスケールアップまたはスケールダウンできます。



ワーカーノードの構成

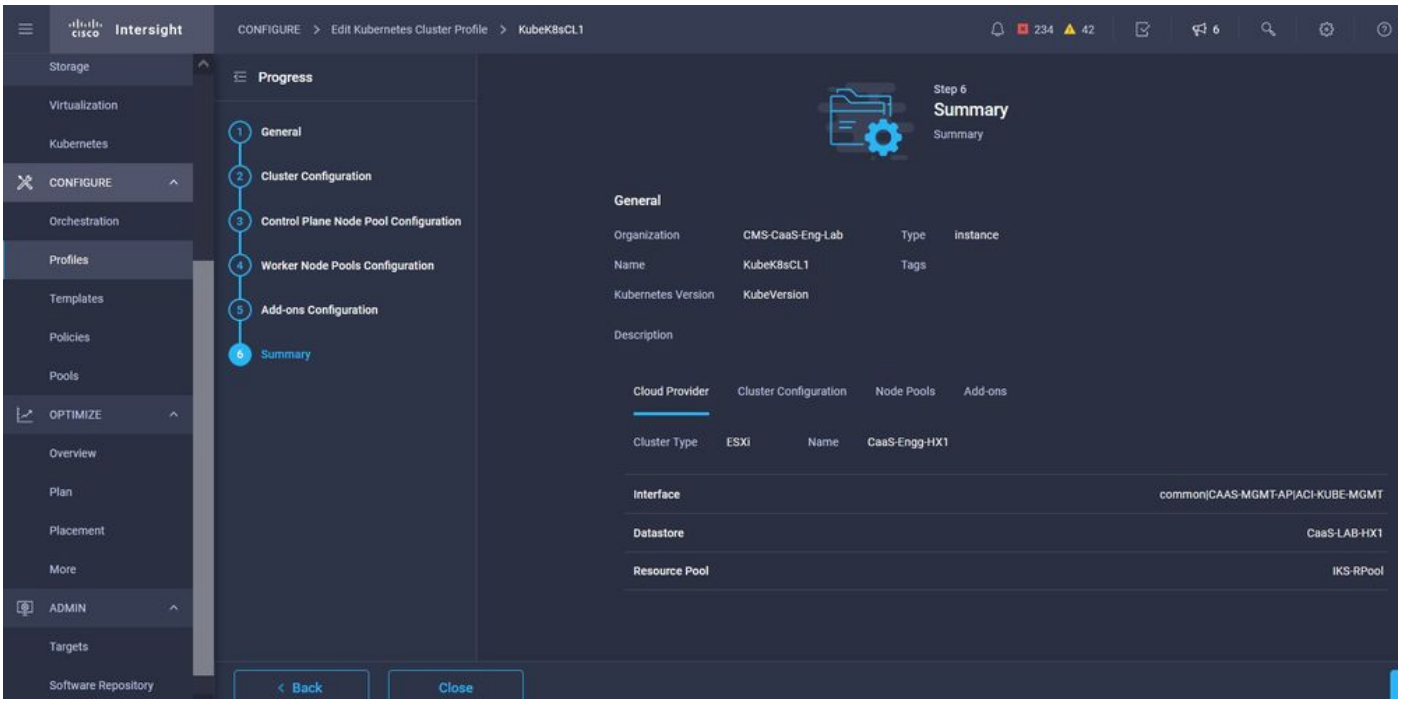
アドオンを構成します。現時点では、Prometheusモニタリングを使用して、KubernetesダッシュボードとGraffanaを自動的に展開できます。

今後、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。



アドオンがある場合は追加する

Summaryにチェックマークを入れて、Deployをクリックする。

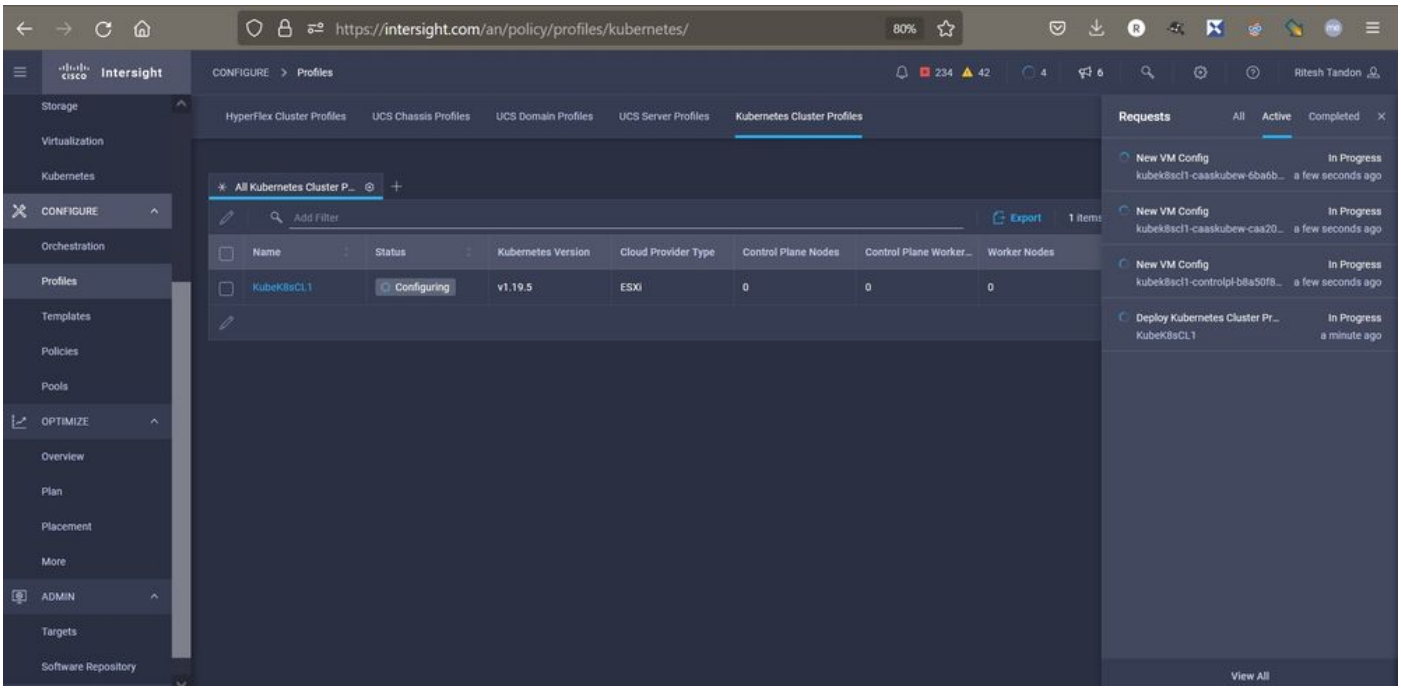


Profile creation Summary画面

確認

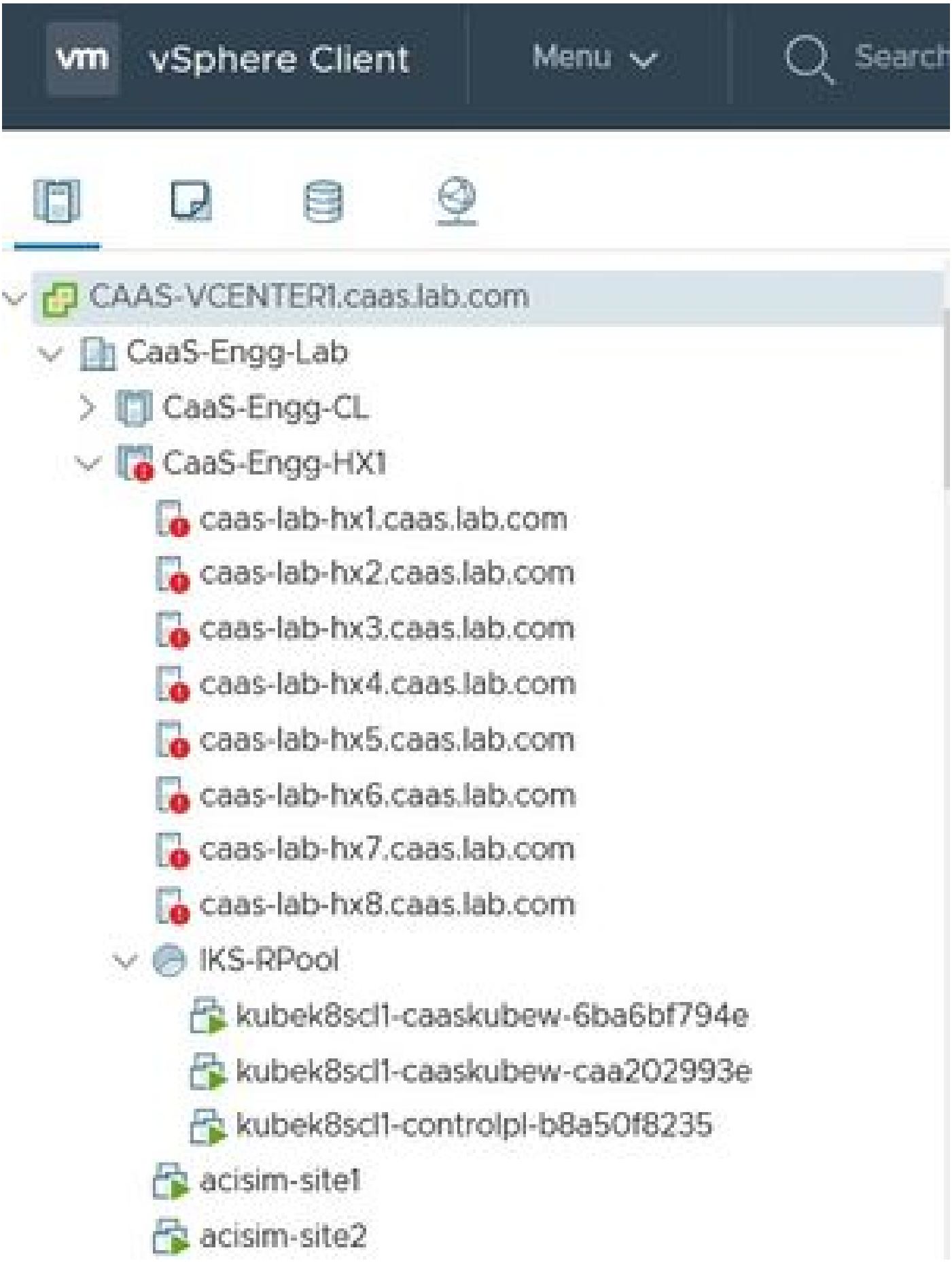
このセクションでは、設定が正常に動作していることを確認します。

右上で、導入の進捗状況を追跡できます。



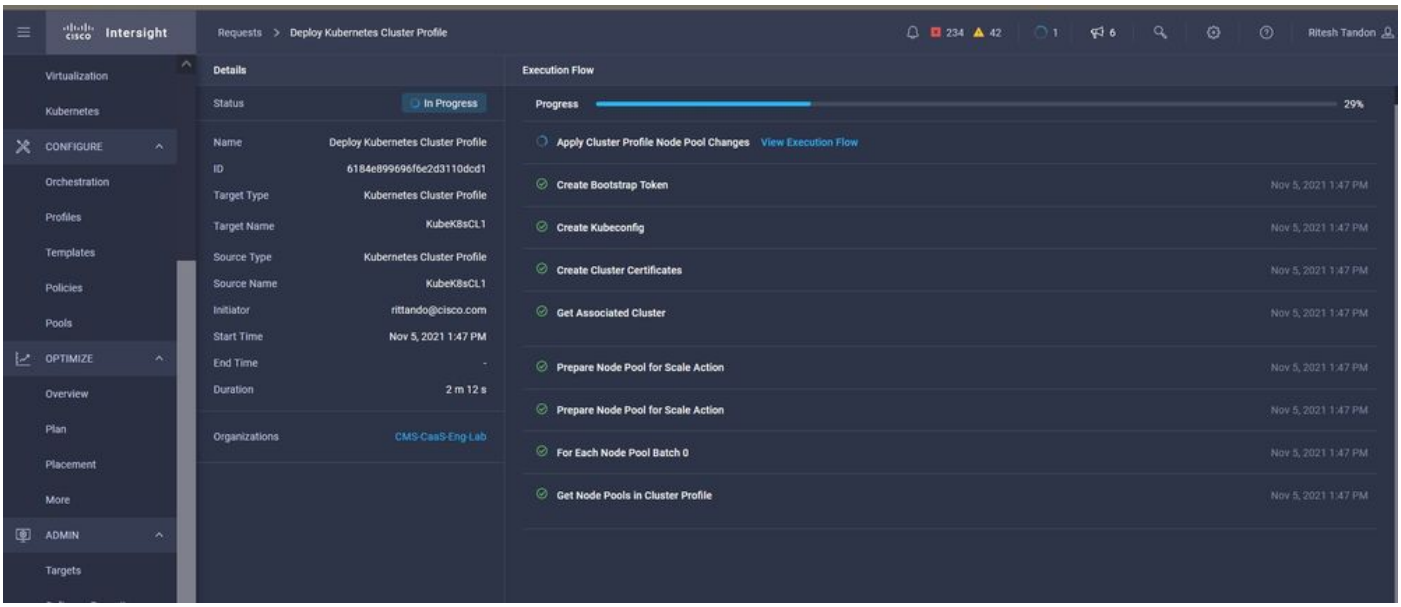
IKS GUIを使用した検証

導入の進捗に応じて、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

展開の詳細な手順を確認する必要がある場合は、さらに詳しく実行できます。



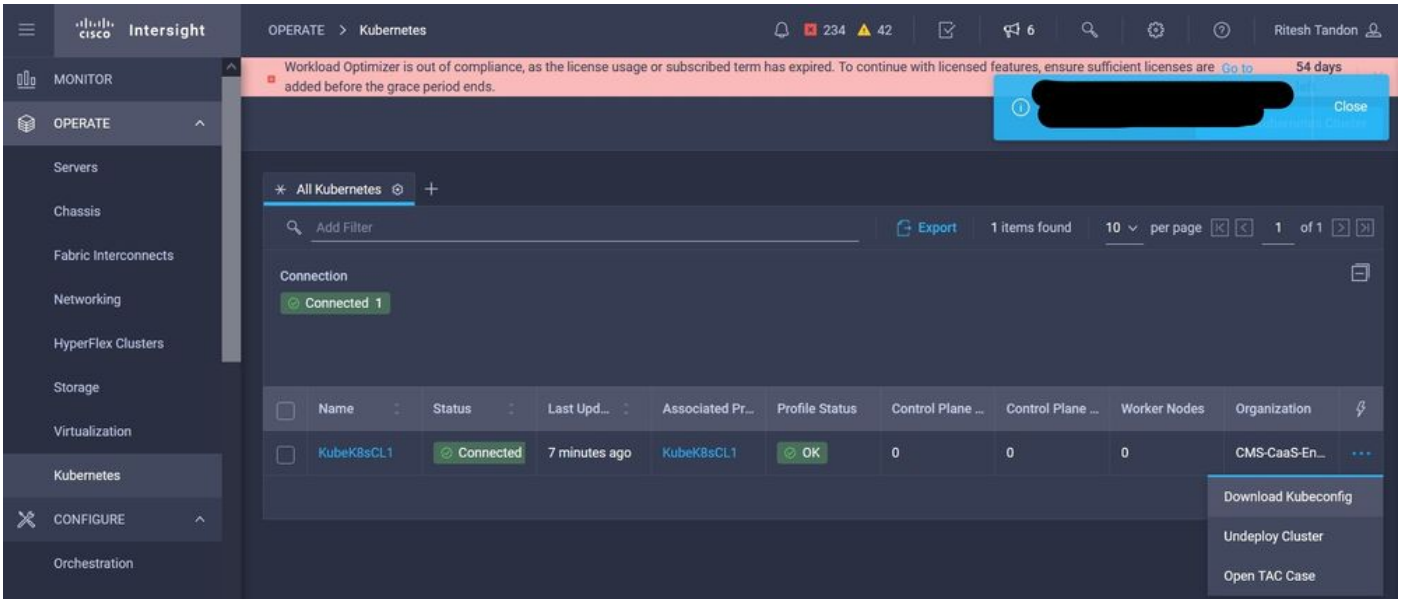
プロファイル作成の実行

Kubernetesクラスタへの接続

次の方法でKubernetesクラスタに接続できます。

KubeConfigファイルを使用します。このファイルは、Operate > Kubernetes >右端のオプションを選択するとダウンロードできます。

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をインストールする必要があります。



IKSからのKubeConfigファイルのダウンロード

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

「Kubernetes Dashboard」をアドオンとして展開する場合も、GUIを使用してアプリケーションを直接展開できます。

詳細については、ここにある「Kubernetesクラスタへのアクセス」を参照してください。

CLIを使用した確認

kubernetesクラスタに接続できたら、kubeCtlを使用して、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

           6d4h   v1.19.5
kubek8scl1-caaskubew-caa202993e   Ready

            6d4h   v1.19.5
kubek8scl1-controlpl-b8a50f8235   Ready    master   6d4h   v1.19.5
```

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helm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-cisco1-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/.k
NAME                    NAMESPACE       REVISION        UPDATED                                 STATUS
addon-operator          iks             1               2021-11-05 07:45:15.44180913 +0000 UTC  deploye
ccp-monitor             iks             1               2021-11-05 08:23:11.309694887 +0000 UTC deploye
essential-cert-ca       iks             1               2021-11-05 07:55:04.409542885 +0000 UTC deploye
essential-cert-manager  iks             1               2021-11-05 07:54:41.433212634 +0000 UTC deploye
essential-metallb       iks             1               2021-11-05 07:54:48.799226547 +0000 UTC deploye
essential-nginx-ingress iks             1               2021-11-05 07:54:46.762865131 +0000 UTC deploye
essential-registry      iks             1               2021-11-05 07:54:36.734982103 +0000 UTC deploye
essential-vsphere-csi   kube-system     1               2021-11-05 07:54:58.168305242 +0000 UTC deploye
kubernetes-dashboard    iks             1               2021-11-05 07:55:10.197905183 +0000 UTC deploye
vsphere-cpi             kube-system     1               2021-11-05 07:54:38.292088943 +0000 UTC deploye
```

すべてのIKSテナントクラスタにデフォルトでインストールされるEssential（コア）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@kubek8scl1-controlpl-b8a50f8235:~$ kubectl get svc -n iks
NAME                                            TYPE           CLUSTER-IP       EXTERNAL-IP   PO
ccp-monitor-grafana                             ClusterIP      192.168.23.161


                80/TCP                      6d3h
ccp-monitor-prometheus-alertmanager                     ClusterIP      192.168.23.70
```

```
                  80/TCP                          6d3h
ccp-monitor-prometheus-kube-state-metrics              ClusterIP      None

                  80/TCP                          6d3h
ccp-monitor-prometheus-node-exporter                   ClusterIP      None

                 9100/TCP                         6d3h
ccp-monitor-prometheus-pushgateway                     ClusterIP      192.168.23.130

                 9091/TCP                         6d3h
ccp-monitor-prometheus-server                          ClusterIP      192.168.23.95

                  443/TCP                         6d3h
essential-cert-manager                                 ClusterIP      192.168.23.178

                 9402/TCP                         6d4h
essential-cert-manager-webhook                         ClusterIP      192.168.23.121

                  443/TCP                         6d4h
essential-nginx-ingress-ingress-nginx-controller       LoadBalancer   192.168.23.26    192.168.10.11
essential-nginx-ingress-ingress-nginx-defaultbackend   ClusterIP      192.168.23.205

                   80/TCP                         6d4h
essential-registry-docker-registry                     ClusterIP      192.168.23.12

                  443/TCP                         6d4h
kubernetes-dashboard                                   ClusterIP      192.168.23.203

                  443/TCP                         6d4h
```

トラブルシュート

このセクションでは、設定のトラブルシューティングに役立つ情報を紹介します。

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


         -n




Example :
kubectl describe pod vsphere-csi-controller-7d56dc7c8-qgbhw -n kube-system
Name:         vsphere-csi-controller-7d56dc7c8-qgbhw
Namespace:    kube-system
Priority:     0
Node:         kubek8scl1-controlpl-eb44cf1bf3/192.168.58.11
Start Time:   Tue, 28 Sep 2021 02:39:41 +0000
Labels:       app=vsphere-csi-controller
              pod-template-hash=7d56dc7c8
              role=vsphere-csi
Annotations:




Status:       Running
IP:           192.168.58.11
IPs:
  IP:           192.168.58.11
Controlled By:  ReplicaSet/vsphere-csi-controller-7d56dc7c8
Containers:
  csi-attacher:
    Container ID:  docker://60002693136d00f3b61237304a1fbc033df92f86dc1352965328fe3c4d264fdb
    Image:         registry.ci.x----x.com/cpsg_kaas-images/quay.io/k8scsi/csi-attacher:v2.0.0
    Image ID:      docker-pullable://registry.ci.x------x.com/cpsg_kaas-images/quay.io/k8scsi/csi-attac
    Port:




    Host Port:
```

```
    Args:
      --v=4
      --timeout=300s
      --csi-address=$(ADDRESS)
      --leader-election
    State:          Running
      Started:      Thu, 30 Sep 2021 05:44:11 +0000
    Last State:     Terminated
      Reason:       Error
      Message:      Lost connection to CSI driver, exiting
      Exit Code:    255
      Started:      Thu, 30 Sep 2021 05:38:20 +0000
      Finished:     Thu, 30 Sep 2021 05:39:06 +0000
    Ready:          True
    Restart Count:  531

X---------------------- Log Text Omitted ---------X----------------------X--------------------X
```

関連情報

- IKSサービスの概要をここで確認してください。
- ユーザガイドを確認してください。
- Intersight Kubernetesサービスのデモはこちら
- テクニカル サポートとドキュメント - Cisco Systems

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。