Configure Kubernetes Cluster using Intersight Kubernetes Service

Contents

Introduction

This document describes the configuration to provision a production-grade Kubernetes cluster from Cisco Intersight (SaaS) with the use of Cisco Intersight™ Kubernetes Service (IKS).

Background Information

Kubernetes, in recent times, has become a de-facto container management tool, as organizations tend to invest more in application modernization with Containerized solutions. With Kubernetes, development teams can deploy, manage, and scale their containerized applications with ease, making innovations more accessible to their continuous delivery pipelines.

However, Kubernetes comes with operational challenges, because it requires time and technical expertise to install and configure.
Installing Kubernetes and the different software components required, creating clusters, configuring storage, networking and security, along with operations (e.g. upgrading, updating and patching critical security bugs) require ongoing significant human capital investment.

Enter IKS, a turn-key SaaS solution for managing consistent, production-grade Kubernetes anywhere. To read more on IKS's capabilities, check this link here.

Solution Overview

For this document, the idea is to want to showcase IKS's ability to integrate seamlessly with your on-prem infrastructure, running VMware ESXi and vCenter.
With a few clicks, you can deploy a production-grade Kubernetes cluster on your VMware infrastructure.

But, to do that you have to integrate your on-prem vCenter with Intersight, which is known as 'claiming a target', vCenter being the target here.
You would need a Cisco Intersight Assist Virtual Appliance, which helps to add endpoint targets to Cisco Intersight. You can install Intersight Assist using the bootstrap OVA that is available on Cisco's official website.

To limit the scope of this document, we would not focus on Cisco Intersight Assist Virtual Appliance installation. But, you can have a look at the process here

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Intersight Account: You need a valid Cisco ID and an Intersight account.
  You can create a Cisco ID on Cisco's website if you don't have one. And then, click the Create an Account link on Intersight.
- Cisco Intersight Assist: Cisco Intersight Assist helps you add vCenter/ESXi as an endpoint target to Cisco Intersight.
- Connectivity: If your environment supports an HTTP/S proxy, you can use that to connect your Cisco Intersight Assist Appliance to the internet. Alternatively, you have to open ports to intersight URLs. Please check this link for detailed network connectivity requirements :
- vCenter credentials to claim it on Intersight.

Components Used

This document is not restricted to specific software and hardware versions.

Assumptions

Since deploying a Cisco Intersight Appliance is out of the scope of this document.

We assume that you already have a working Intersight account, and have successfully claimed an on-prem vCenter/Esxi to it.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Configuration

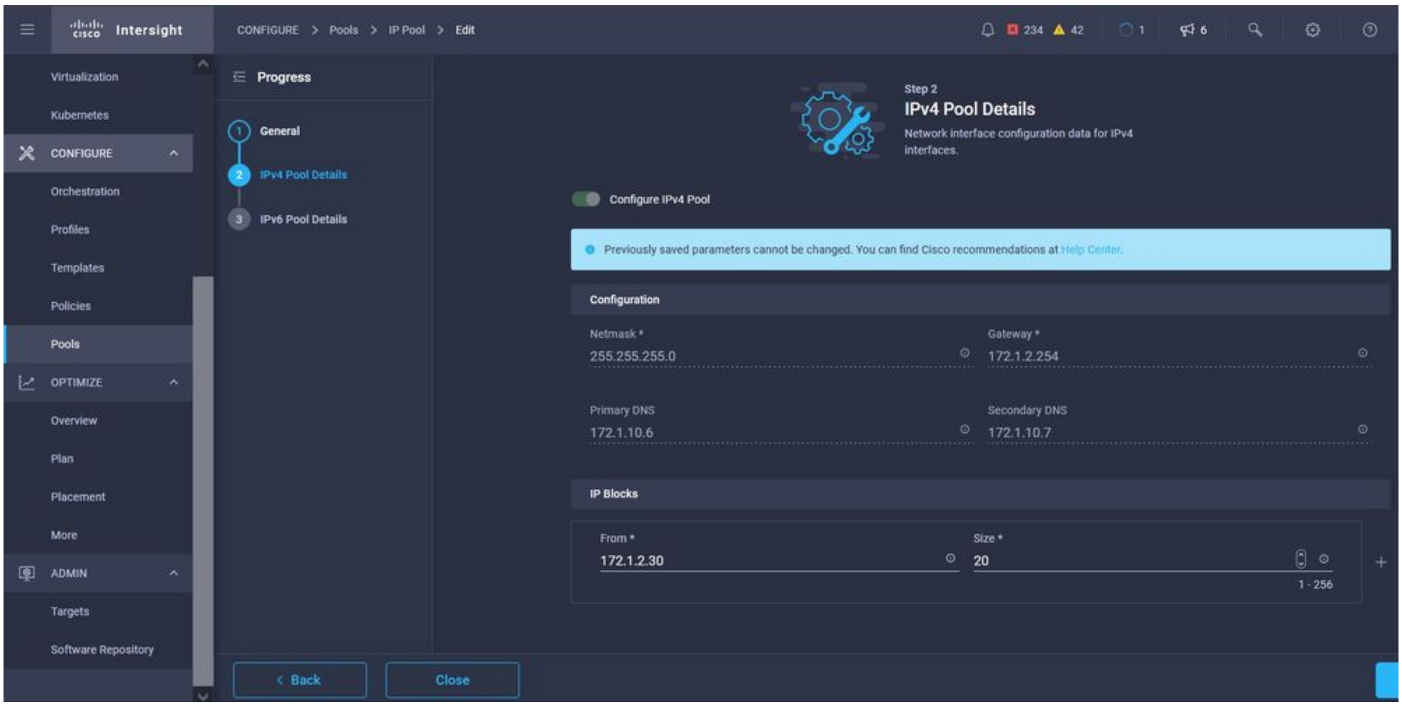
Step 1. Configure Policies

Policies allow simplified management as they abstract the configuration into re-usable templates.

Some of the policies which we need to configure are listed below. Please note all these policies would be created under Configure >> Polices & Configure >> Pools section on Intersight.

You can see the path of the policy on top of each screenshot too, given below.

This IP Pool will be used for IP Addresses on your Control and Worker Nodes Virtual Machines, when launched on the ESXi host.



Step 2
IPv4 Pool Details
Network interface configuration data for IPv4 interfaces.

Configure IPv4 Pool

Previously saved parameters cannot be changed. You can find Cisco recommendations at Help Center.

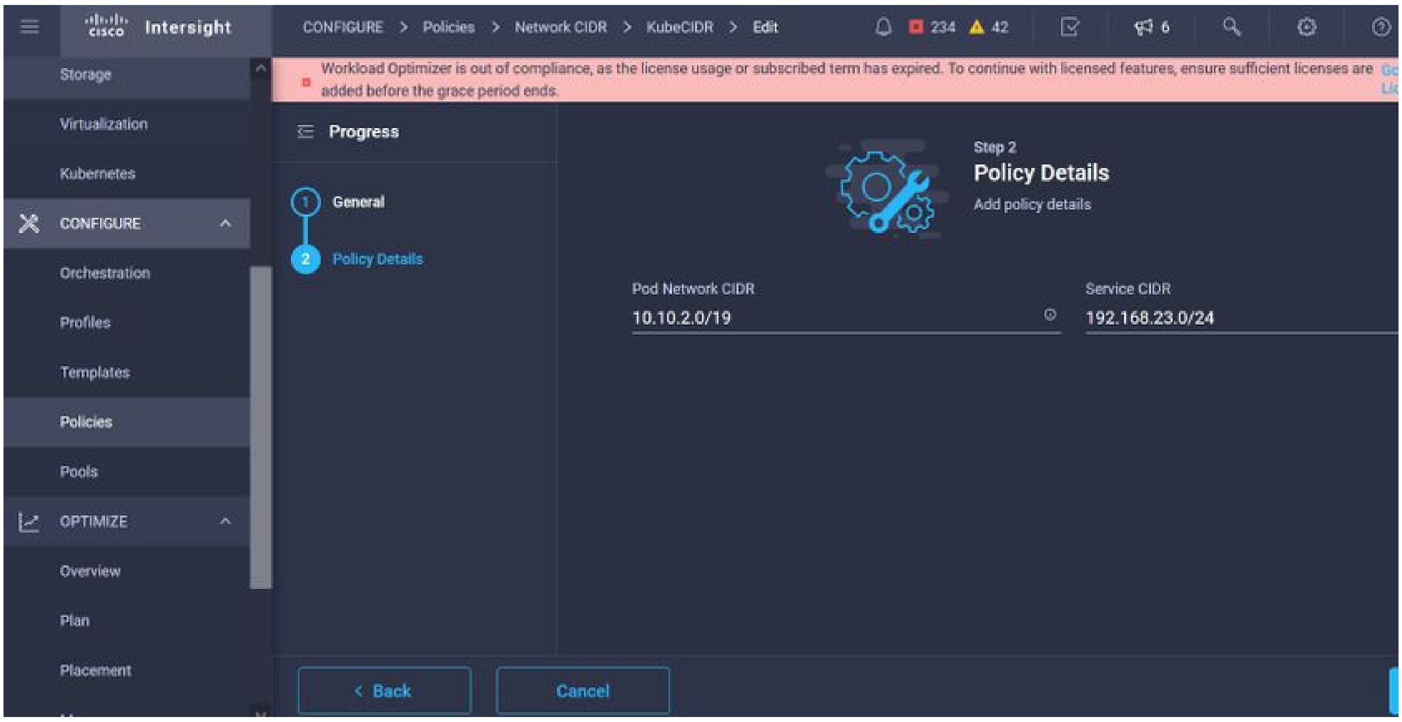
Configuration

Netmask *	Gateway *
255.255.255.0	172.1.2.254
Primary DNS	Secondary DNS
172.1.10.6	172.1.10.7

IP Blocks

From *	Size *
172.1.2.30	20
	1 - 256

Here you define the Pod and Services Network CIDR, for internal networking within the Kubernetes cluster.



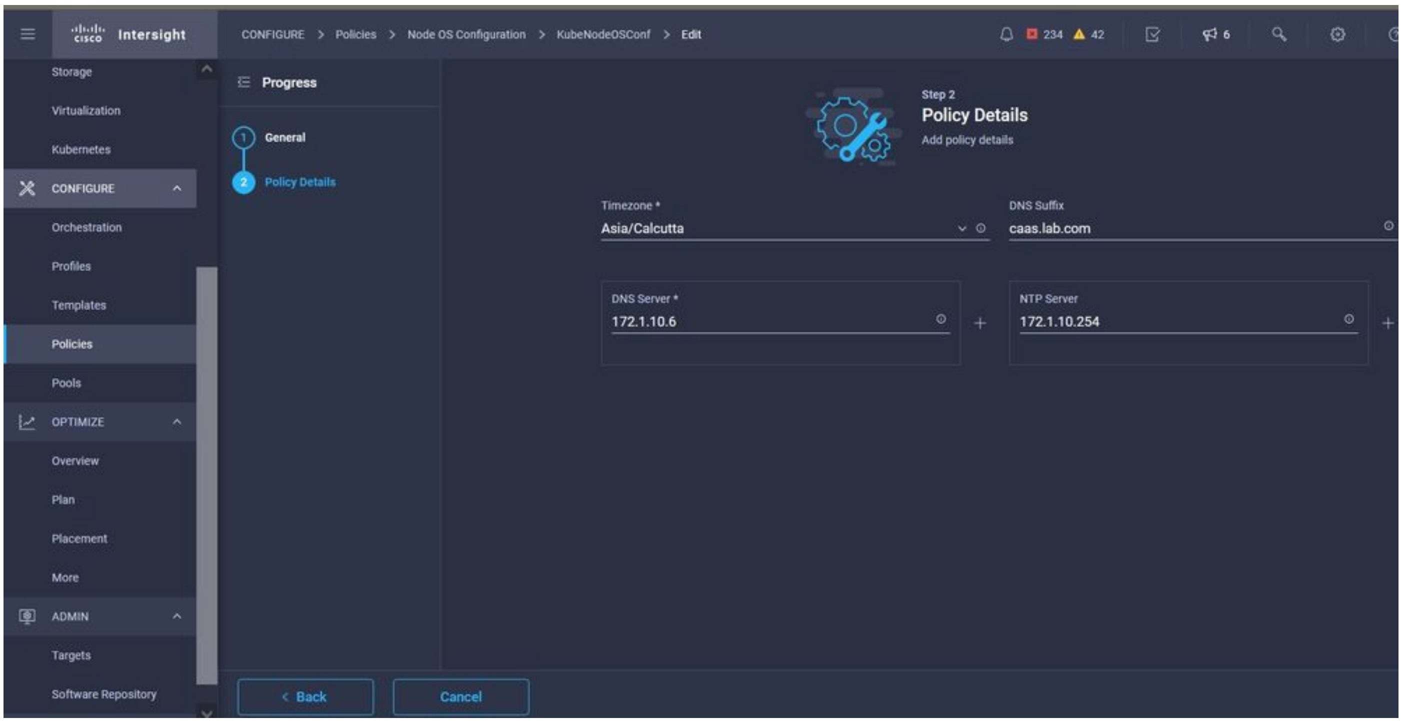
Workload Optimizer is out of compliance, as the license usage or subscribed term has expired. To continue with licensed features, ensure sufficient licenses are added before the grace period ends.

Step 2
Policy Details
Add policy details

Pod Network CIDR	Service CIDR
10.10.2.0/19	192.168.23.0/24

Services and Network CIDR

This Policy defines your NTP and DNS configuration.

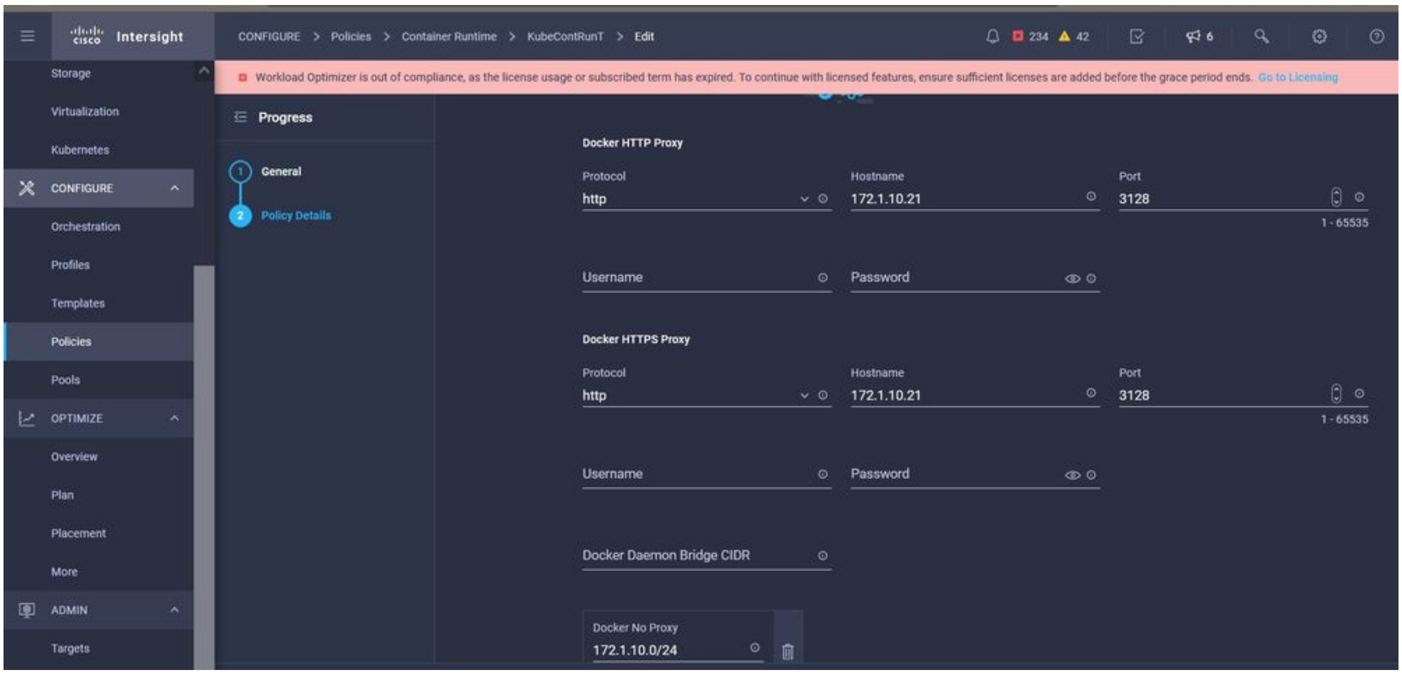


Step 2
Policy Details
Add policy details

Timezone * Asia/Calcutta
DNS Suffix caas.lab.com
DNS Server * 172.1.10.6
NTP Server 172.1.10.254

NTP and DNS configuration

With this policy, you can define the proxy configuration for your docker container runtime.



Workload Optimizer is out of compliance, as the license usage or subscribed term has expired. To continue with licensed features, ensure sufficient licenses are added before the grace period ends. Go to Licensing

Docker HTTP Proxy
Protocol http
Hostname 172.1.10.21
Port 3128
1 - 65535
Username
Password

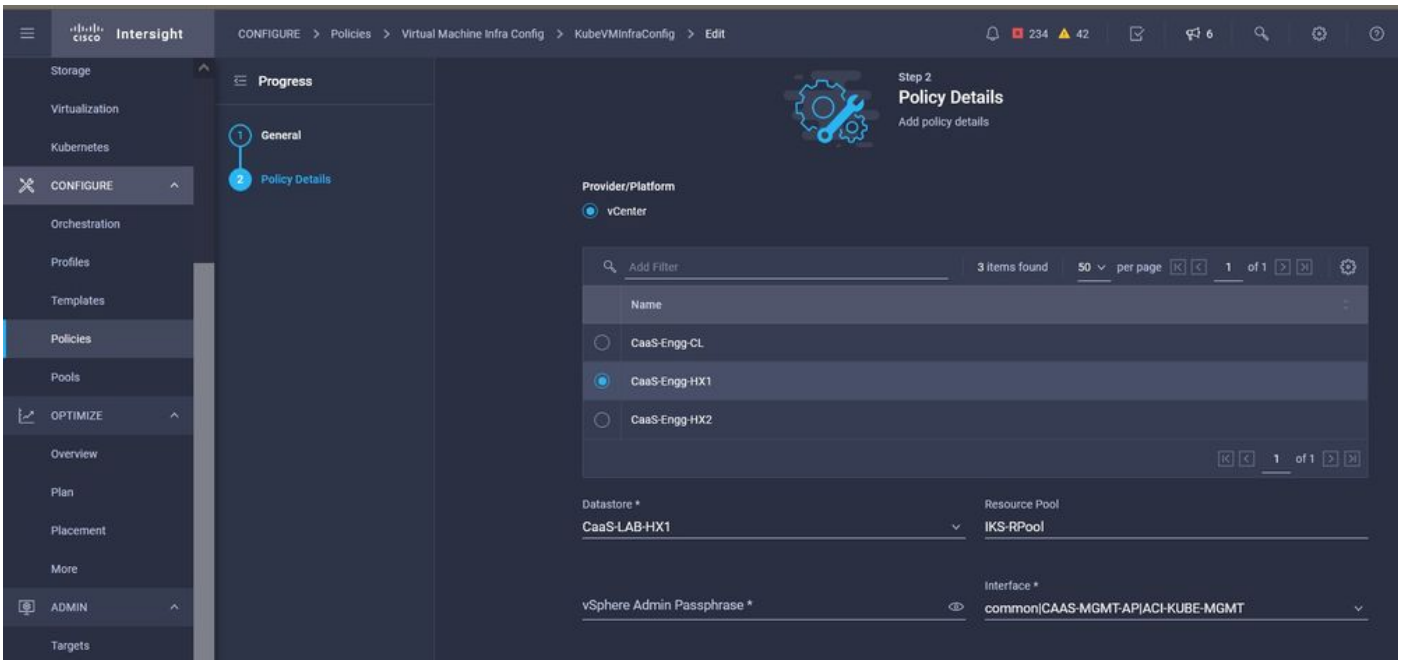
Docker HTTPS Proxy
Protocol http
Hostname 172.1.10.21
Port 3128
1 - 65535
Username
Password

Docker Daemon Bridge CIDR

Docker No Proxy 172.1.10.0/24

Proxy configuration for Docker

In this policy, you will define the configuration needed on the Virtual Machines deployed as Master and Worker nodes.



Provider/Platform
vCenter

Name
CaaS-Engg-CL
CaaS-Engg-HX1
CaaS-Engg-HX2

Datastore * CaaS-LAB-HX1
Resource Pool IKS-RPool
vSphere Admin Passphrase *
Interface * common|CAAS-MGMT-AP|ACI-KUBE-MGMT

Configuration of VMs used

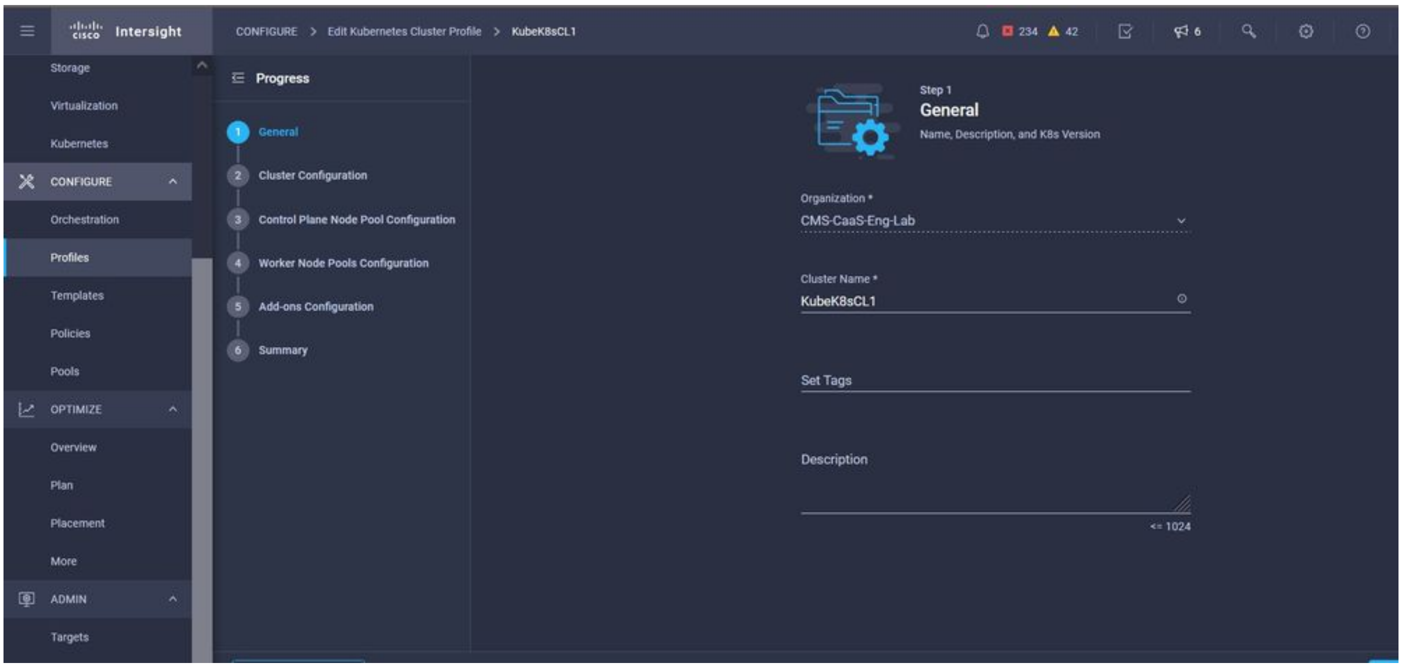
Step 2. Configure Profile

Once we have created the above policies, we would then bind them into a profile which we can then deploy.

Deploying configuration using policies and profiles abstracts the configuration layer so that it can be repeatedly deployed quickly.

You can copy this profile and create a new one with little or more modifications on the underlying policies within minutes, to one or more Kubernetes cluster up in a fraction of time needed with a manual process.

GIve in the Name and set Tags.

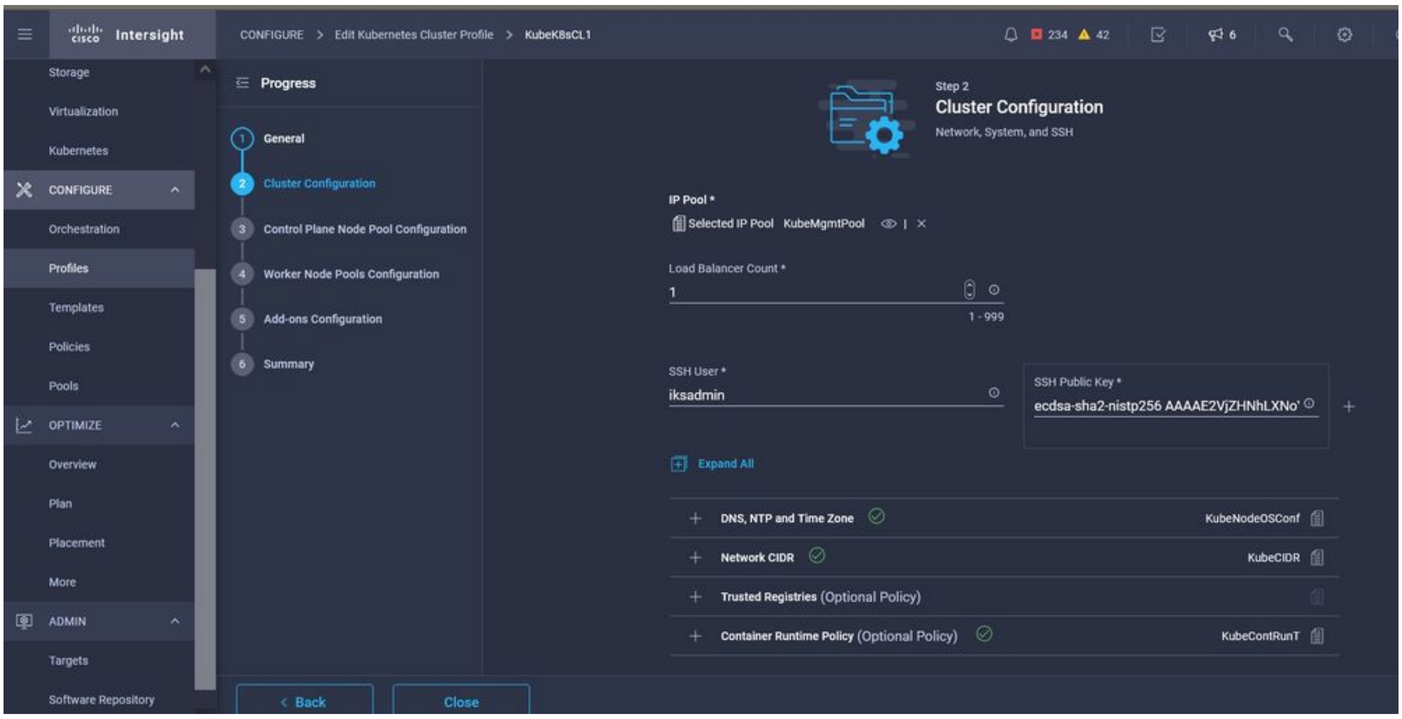


Organization * CMS-CaaS-Eng-Lab
Cluster Name * KubeK8sCL1
Set Tags
Description
<= 1024

Profile Config with Name and Tags

Set the Pool, Node OS, Network CIDR Policies. You also need to configure a userid and SSH key (public).

Its corresponding private key would be used to ssh into Master & Worker nodes.



IP Pool *
Selected IP Pool KubeMgmtPool

Load Balancer Count *
1
1 - 999

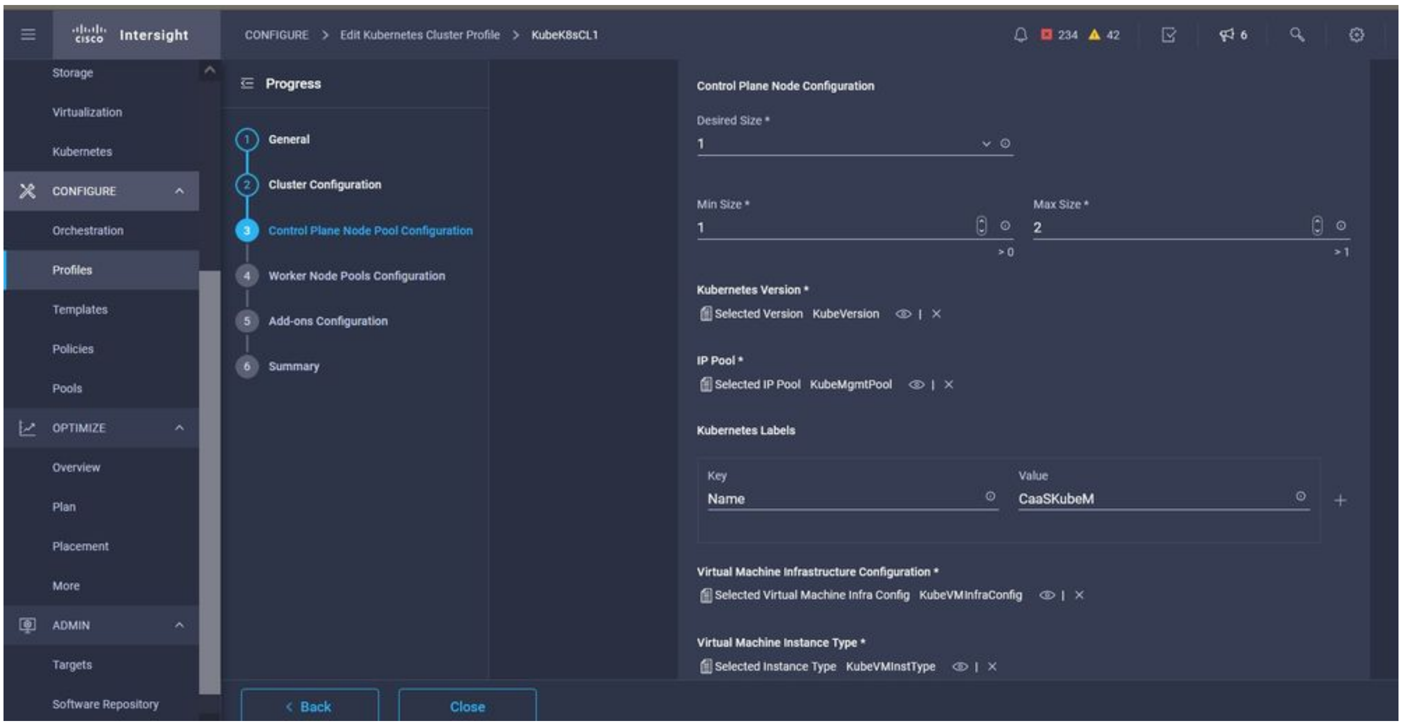
SSH User *
iksadmin

SSH Public Key *
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNo'

DNS, NTP and Time Zone	KubeNodeOSConf
Network CIDR	KubeCIDR
Trusted Registries (Optional Policy)	
Container Runtime Policy (Optional Policy)	KubeContRunT

Profile Config with policies assigned

Configure the Control plane: You can define how many Master nodes you would need on the control plane.



Control Plane Node Configuration

Desired Size *
1

Min Size *
1
> 0

Max Size *
2
> 1

Kubernetes Version *
Selected Version KubeVersion

IP Pool *
Selected IP Pool KubeMgmtPool

Kubernetes Labels

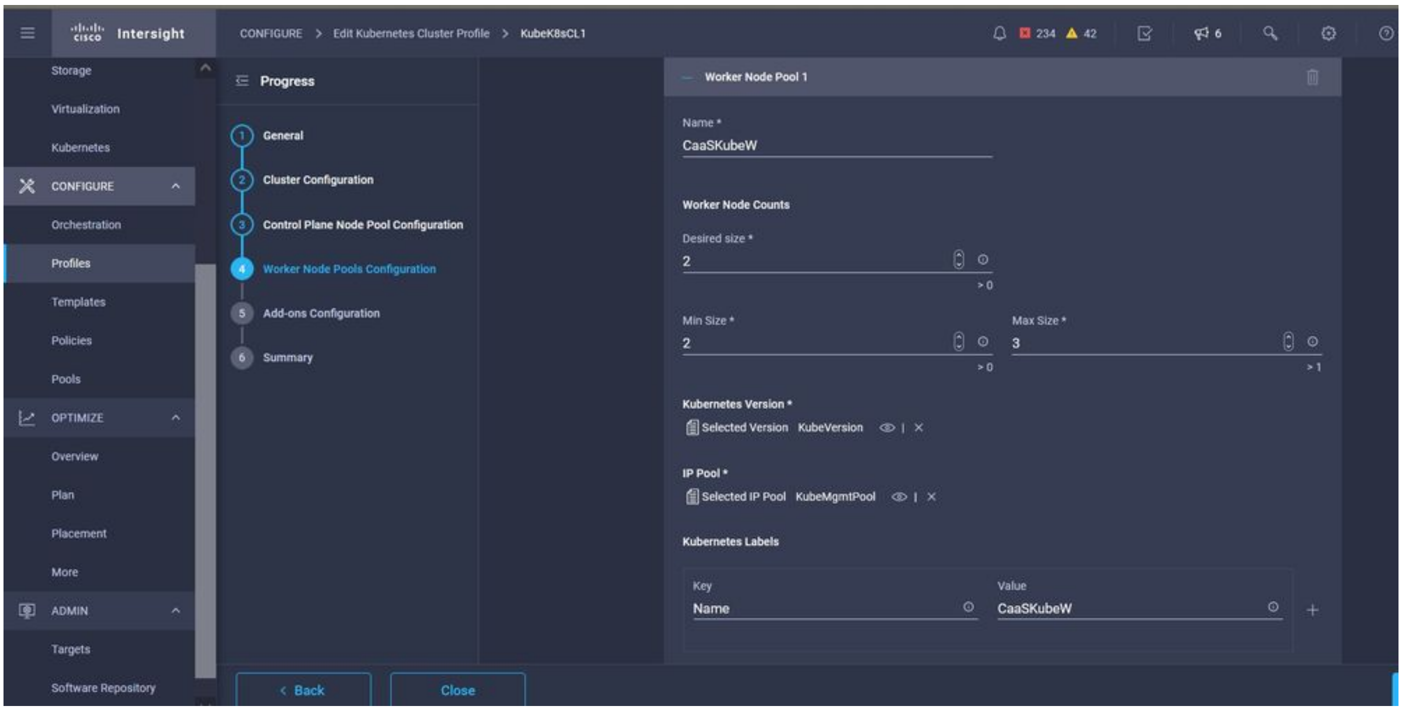
Key	Value
Name	CaaSKubeM

Virtual Machine Infrastructure Configuration *
Selected Virtual Machine Infra Config KubeVMInfraConfig

Virtual Machine Instance Type *
Selected Instance Type KubeVMInstType

Master node configuration

Configure the Worker nodes: Depending on the application requirements, you can scale up or scale down your worker nodes.



Worker Node Pool 1

Name * CaaSKubeW

Worker Node Counts

Desired size * 2 (> 0)

Min Size * 2 (> 0)

Max Size * 3 (> 1)

Kubernetes Version * Selected Version KubeVersion

IP Pool * Selected IP Pool KubeMgmtPool

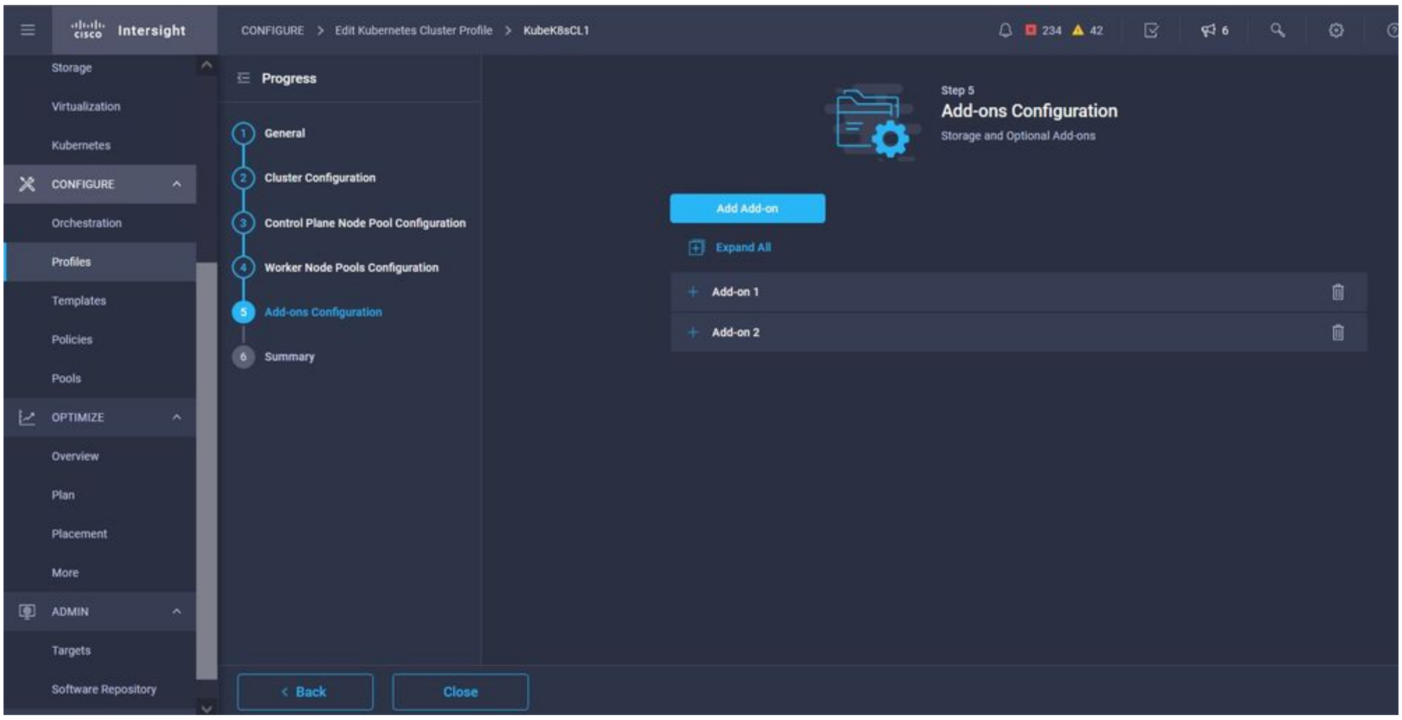
Kubernetes Labels

Key	Value
Name	CaaSKubeW

Worker Nodes configuration

Configure Add-on. As of now, you can automatically deploy, Kubernetes Dashboard and Graffana with Prometheus monitoring.

In future, you can add more add-on's which you can automatically deploy using IKS.



Step 5
Add-ons Configuration
Storage and Optional Add-ons

Add Add-on

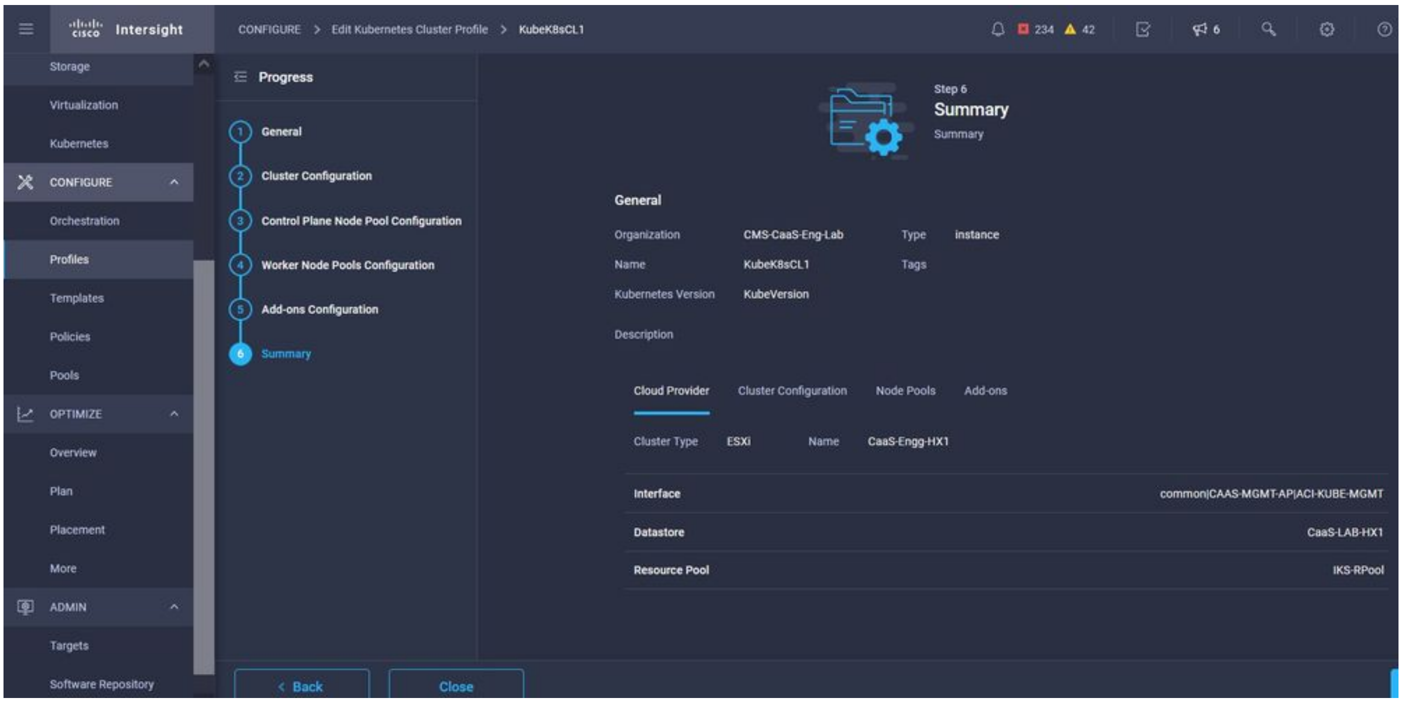
Expand All

Add-on 1

Add-on 2

Add Add-ONs if any

Check the Summary, and click **Deploy**.


CONFIGURE > Edit Kubernetes Cluster Profile > KubeK8sCL1
Progress
1 General
2 Cluster Configuration
3 Control Plane Node Pool Configuration
4 Worker Node Pools Configuration
5 Add-ons Configuration
6 Summary
Back
Close

Step 6
Summary
Summary

General

Organization	CMS-CaaS-Eng-Lab	Type	instance
Name	KubeK8sCL1	Tags	
Kubernetes Version	KubeVersion		
Description			

Cloud Provider | Cluster Configuration | Node Pools | Add-ons

Cluster Type	ESXi	Name	CaaS-Engg-HX1

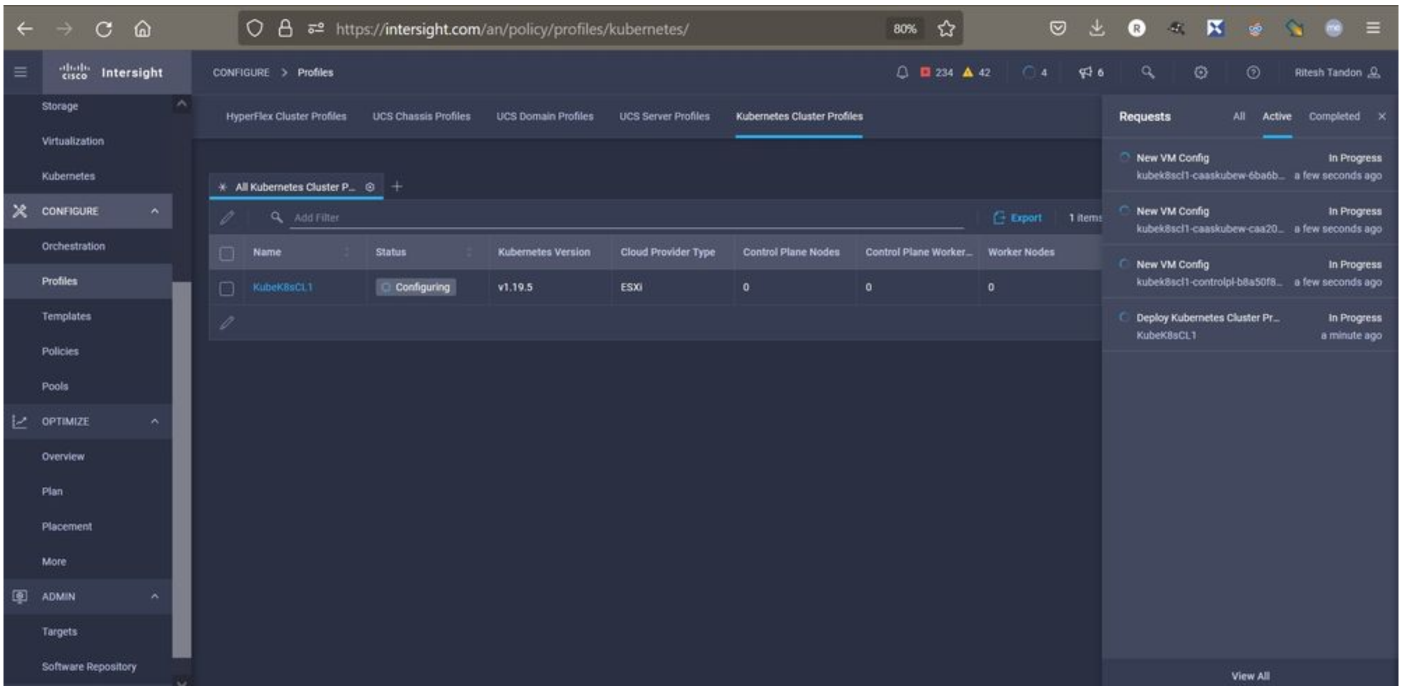
Interface	common|CAAS-MGMT-AP|ACI-KUBE-MGMT
Datastore	CaaS-LAB-HX1
Resource Pool	IKS-RPool

Profile creation Summary screen

Verify

Use this section to confirm that your configuration works properly.

On the upper right-hand side, you can track the progress of the deployment.


https://intersight.com/an/policy/profiles/kubernetes/
CONFIGURE > Profiles
HyperFlex Cluster Profiles
UCS Chassis Profiles
UCS Domain Profiles
UCS Server Profiles
Kubernetes Cluster Profiles
All Kubernetes Cluster P...
Add Filter
Export
1 items
View All

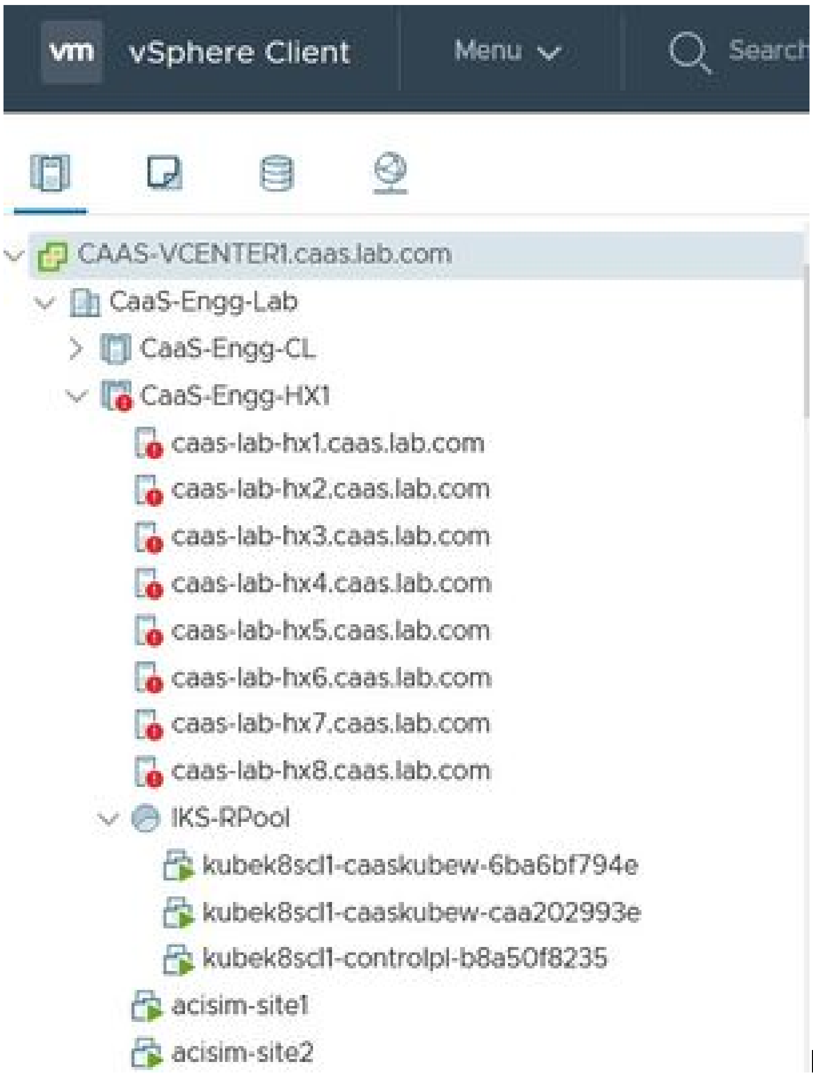
Name	Status	Kubernetes Version	Cloud Provider Type	Control Plane Nodes	Control Plane Worker...	Worker Nodes
KubeK8sCL1	Configuring	v1.19.5	ESXi	0	0	0

Requests (All | Active | Completed)

New VM Config kubek8scl1-caaskubew-6ba6b...	In Progress a few seconds ago
New VM Config kubek8scl1-caaskubew-caa20...	In Progress a few seconds ago
New VM Config kubek8scl1-controlpl-b8a50f8...	In Progress a few seconds ago
Deploy Kubernetes Cluster Pr... KubeK8sCL1	In Progress a minute ago

Verify using IKS GUI

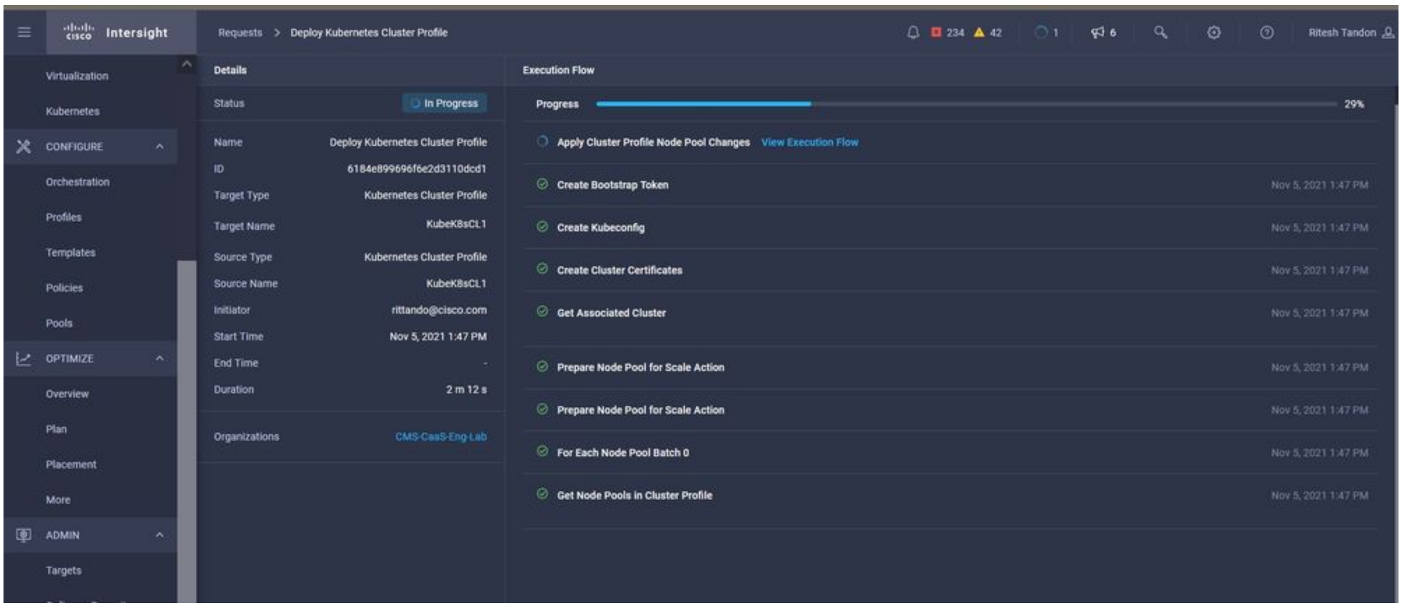
As the deployment progresses, you can see your Kubernetes Master and Worker Nodes coming up on the vCenter.



CAAS-VCENTER1.caas.lab.com
CaaS-Engg-Lab
CaaS-Engg-CL
CaaS-Engg-HX1
caas-lab-hx1.caas.lab.com
caas-lab-hx2.caas.lab.com
caas-lab-hx3.caas.lab.com
caas-lab-hx4.caas.lab.com
caas-lab-hx5.caas.lab.com
caas-lab-hx6.caas.lab.com
caas-lab-hx7.caas.lab.com
caas-lab-hx8.caas.lab.com
IKS-RPool
kubek8scl1-caaskubew-6ba6bf794e
kubek8scl1-caaskubew-caa202993e
kubek8scl1-controlpl-b8a50f8235
acisim-site1
acisim-site2

IKS cluster coming up in vCenter

In case you need to see detailed steps for the deployment, you can drill further into the execution.



Details

Status	In Progress
Name	Deploy Kubernetes Cluster Profile
ID	6184e899696f6e2d3110dcd1
Target Type	Kubernetes Cluster Profile
Target Name	KubeK8sCL1
Source Type	Kubernetes Cluster Profile
Source Name	KubeK8sCL1
Initiator	rittando@cisco.com
Start Time	Nov 5, 2021 1:47 PM
End Time	-
Duration	2 m 12 s
Organizations	CMS-CaaS-Eng-Lab

Execution Flow

Progress 29%

Apply Cluster Profile Node Pool Changes View Execution Flow	
Create Bootstrap Token	Nov 5, 2021 1:47 PM
Create Kubeconfig	Nov 5, 2021 1:47 PM
Create Cluster Certificates	Nov 5, 2021 1:47 PM
Get Associated Cluster	Nov 5, 2021 1:47 PM
Prepare Node Pool for Scale Action	Nov 5, 2021 1:47 PM
Prepare Node Pool for Scale Action	Nov 5, 2021 1:47 PM
For Each Node Pool Batch 0	Nov 5, 2021 1:47 PM
Get Node Pools in Cluster Profile	Nov 5, 2021 1:47 PM

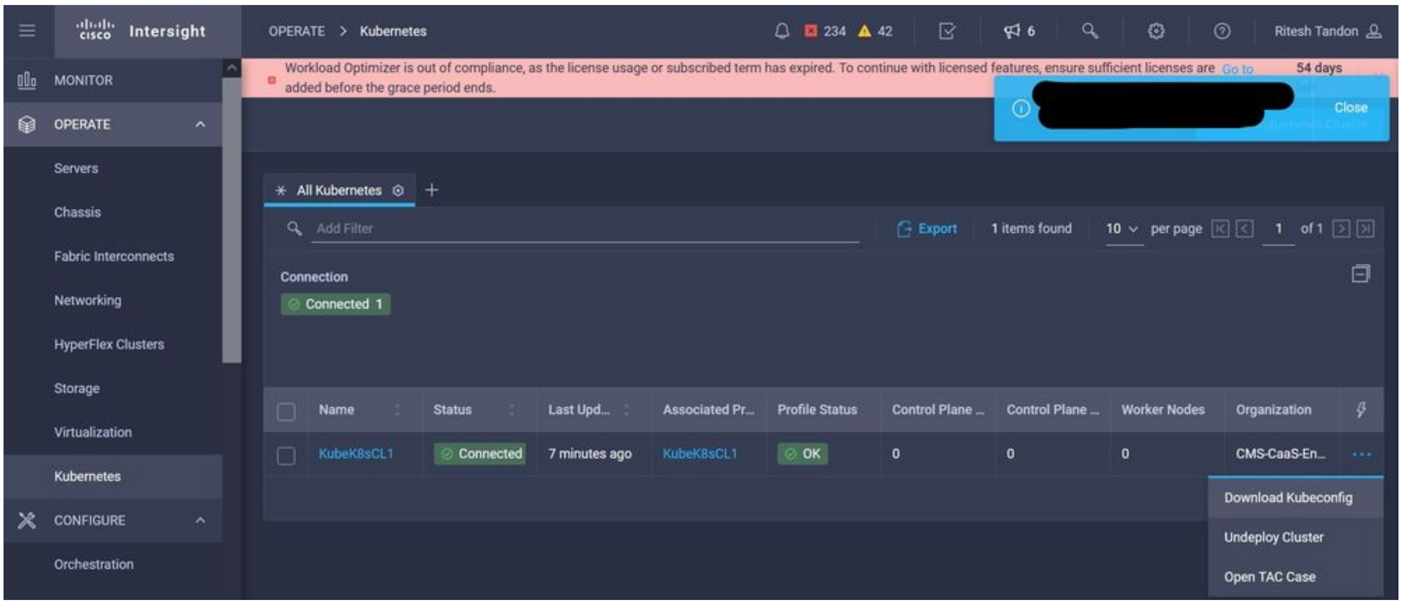
Profile creation execution

Connect to the Kubernetes Cluster

You can connect to the Kubernetes cluster in the following ways:

Using the KubeConfig file, which you can download from **Operate > Kubernetes > Select the options on the far right**.

You need to have KubeCtl installed on the Management workstation, from where you want to access this cluster from.



Workload Optimizer is out of compliance, as the license usage or subscribed term has expired. To continue with licensed features, ensure sufficient licenses are added before the grace period ends. Go to 54 days

[redacted]

Name	Status	Last Upd...	Associated Pr...	Profile Status	Control Plane ...	Control Plane ...	Worker Nodes	Organization
KubeK8sCL1	Connected	7 minutes ago	KubeK8sCL1	OK	0	0	0	CMS-CaaS-En...

Download KubeConfig file from IKS

You can also directly SSH into the master node, using SSH applications like Putty with the credentials and Private Key configured at the time of deployment

If you deploy 'Kubernetes Dashboard' as an Add-on you can use that too, to deploy applications directly using GUI.

To check further details, please check out the 'Accessing Kubernetes Clusters' section, here :

Verify with CLI

Once you are able to connect to the Kubernetes cluster, using kubeCtl, you can use the following commands to verify if the cluster has all components installed and running.

Verify nodes in the cluster are in a 'ready' state.

```
iksadmin@kubek8scl1-controlpl-b8a50f8235:~$ kubectl get nodes NAME STATUS ROLES AGE VERSION
kubek8scl1-caaskubew-6ba6bf794e Ready <none> 6d4h v1.19.5 kubek8scl1-caaskubew-caa202993e Ready
<none> 6d4h v1.19.5 kubek8scl1-controlpl-b8a50f8235 Ready master 6d4h v1.19.5
```

Verify the status of the pods that were created at the time of the installation of the essential components on the cluster.

```
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
```

Verify the status of the ccp-helm-operator pod that manages the locally running helm and installs add-ons.

```
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/.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.g415a48c4be1.0 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-cisco1-
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
```

Verify the status of the essential-* pods that manage the Essential (core) add-ons, installed by default, on every IKS tenant cluster.

```
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
```

Verify the status of the services and the loadbalancer deployed in the IKS namespace.

```
iksadmin@kubek8scl1-controlpl-b8a50f8235:~$ kubectl get svc -n iks NAME TYPE CLUSTER-IP
EXTERNAL-IP PORT(S) AGE ccp-monitor-grafana ClusterIP 192.168.23.161 <none> 80/TCP 6d3h ccp-
monitor-prometheus-alertmanager ClusterIP 192.168.23.70 <none> 80/TCP 6d3h ccp-monitor-
prometheus-kube-state-metrics ClusterIP None <none> 80/TCP 6d3h ccp-monitor-prometheus-node-
exporter ClusterIP None <none> 9100/TCP 6d3h ccp-monitor-prometheus-pushgateway ClusterIP
192.168.23.130 <none> 9091/TCP 6d3h ccp-monitor-prometheus-server ClusterIP 192.168.23.95 <none>
443/TCP 6d3h essential-cert-manager ClusterIP 192.168.23.178 <none> 9402/TCP 6d4h essential-
cert-manager-webhook ClusterIP 192.168.23.121 <none> 443/TCP 6d4h essential-nginx-ingress-
ingress-nginx-controller LoadBalancer 192.168.23.26 192.168.10.11 80:31121/TCP,443:31753/TCP
6d4h essential-nginx-ingress-ingress-nginx-defaultbackend ClusterIP 192.168.23.205 <none> 80/TCP
6d4h essential-registry-docker-registry ClusterIP 192.168.23.12 <none> 443/TCP 6d4h kubernetes-
dashboard ClusterIP 192.168.23.203 <none> 443/TCP 6d4h
```

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

In case a particular pod is not coming up, you can use these commands to drill down on the cause.

```
Syntax : kubectl describe pod <POD_NAME> -n <NAMESPACE> 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: <none> 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-attacher@sha256:71e2b9b5b8c52d789ef89ba901c8fba270fa623789a988c77c52ebb0257bf723 Port: <none> Host Port: <none> 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
```

Related Information

- Check IKS service brief here.
- Check User Guide here.
- Check Intersight Kubernetes Service Demo here.
- **Technical Support & Documentation - Cisco Systems**