

CN-Seriesファイアウォールのクラウド およびオンプレミスへのデプロイメン ト

Contact Information

Corporate Headquarters:

Palo Alto Networks

3000 Tannery Way

Santa Clara, CA 95054

www.paloaltonetworks.com/company/contact-support

About the Documentation

- For the most recent version of this guide or for access to related documentation, visit the Technical Documentation portal docs.paloaltonetworks.com.
- To search for a specific topic, go to our search page docs.paloaltonetworks.com/search.html.
- Have feedback or questions for us? Leave a comment on any page in the portal, or write to us at documentation@paloaltonetworks.com.

Copyright

Palo Alto Networks, Inc.

www.paloaltonetworks.com

© 2021-2021 Palo Alto Networks, Inc. Palo Alto Networks is a registered trademark of Palo Alto Networks. A list of our trademarks can be found at www.paloaltonetworks.com/company/trademarks.html. All other marks mentioned herein may be trademarks of their respective companies.

Last Revised

December 13, 2021

Table of Contents

CN-Series ファイアウォールを GKE にデプロイする.....	5
CN-Series ファイアウォールを Kubernetes サービスとして GKE に デプロイする.....	7
CN-Series ファイアウォールを GKE に DaemonSet としてデプロイする.....	20
CN-Series ファイアウォールを OKE にデプロイする.....	33
CN-Series ファイアウォールを OKE に Kubernetes サービスとしてデプロイする.....	35
CN-Series ファイアウォールを OKE に DaemonSet としてデプロイする.....	48
CN-Series ファイアウォールを EKS にデプロイする.....	61
CN-Series ファイアウォールを AWS EKS に Kubernetes サービスとしてデプロイする.....	63
CN-Series ファイアウォールを AWS EKS に DaemonSet としてデプロイする.....	72
AWS Marketplace から CN-Seriesをデプロイする.....	82
CN-Series ファイアウォールを AliCloud (ACK) に Kubernetes サービスとしてデプロイする.....	89
CN-Series を OpenShift にデプロイする.....	111
CN-Series を OpenShift にデプロイする.....	113

CN-Series ファイアウォールを GKE にデプロイする

どこで使用できますか？	何が必要ですか？
<ul style="list-style-type: none"> CN-Seriesデプロイメント 	<ul style="list-style-type: none"> CN-Series 10.1.x or above Container Images PanoramaPAN-OS 10.1.x以降のバージョンを実行している Helm 3.6 or above version clientHelmを使用したCNシリーズのデプロイメント用

CNシリーズ ビルディング ブロックとCNシリーズ ファイアウォールによるKubernetesの安全な環境内のワークフローの概要を確認したら、同じクラスタ内のコンテナ間やコンテナと他のワークロード タイプ(仮想マシンやベアメタル サーバーなど)間のトラフィックを保護するためのGKE プラットフォームにおけるCN-Seriesファイアウォールのデプロイから始めることができます。



Kubernetes クラスタ、アプリケーション、およびファイアウォール サービスをデプロイして管理するためには、*kubectl* や *Helm* などの標準の *Kubernetes* ツールが必要です。

詳細については、「*Helm* チャートとテンプレートを使用した [CN-Seriesファイアウォールのデプロイ](#)」を参照してください。*Panorama* は、*Kubernetes* クラスタのデプロイメントと管理用のオーケストレーターになるようには設計されていません。クラスタ管理用のテンプレートがマネージド *Kubernetes* プロバイダから提供されています。*Palo Alto Networks* は、[Helm](#) および [Terraform](#) で *CN-Series* をデプロイするためのコミュニティサポートのテンプレートを提供しています。

- [CN-Series ファイアウォールを Kubernetes サービスとして GKE に デプロイする](#)
- [CN-Series ファイアウォールを GKE に DaemonSet としてデプロイする](#)



CN-Series を *DaemonSet* としての *CN-Series* からサービスとしての *CN-Series*、またはその逆に移行する前に、*plugin-serviceaccount.yaml* を削除して再適用する必要があります。詳細については、「[クラスタ認証用サービスアカウントの作成](#)」を参照してください。

- *CN-Series* を *DaemonSet* として *GKE* にデプロイする場合、*pan-plugin-cluster-mode-secret* が存在してはいけません。
- *CN-Series* を *Kubernetes* サービスとして *GKE* にデプロイする場合、*pan-plugin-cluster-mode-secret* が存在する必要があります。

CN-Series ファイアウォールを Kubernetes サービスとして GKE に デプロイする

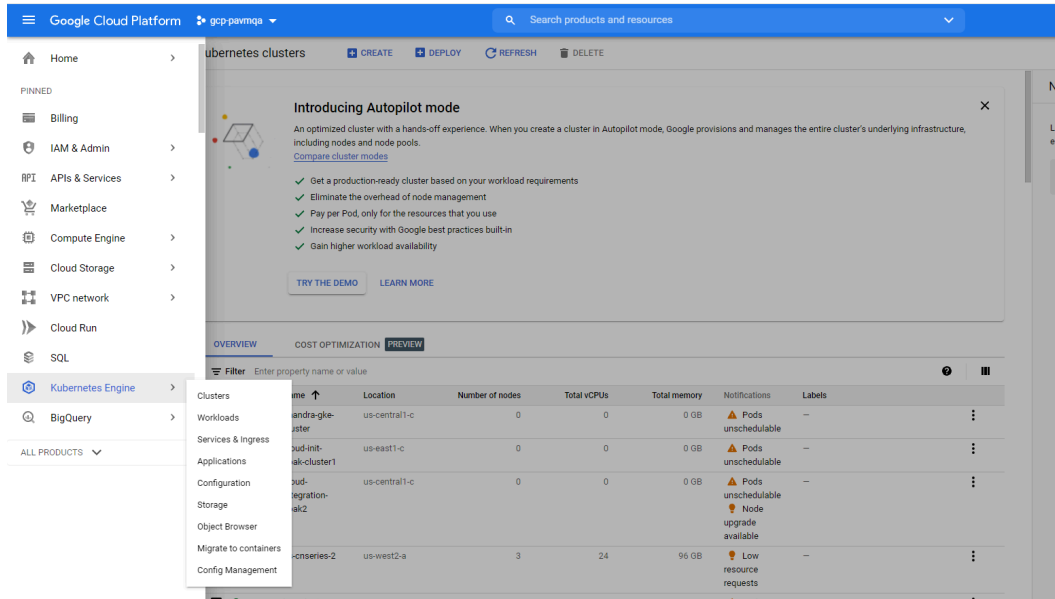
どこで使用できますか?	何が必要ですか?
<ul style="list-style-type: none">• CN-Seriesデプロイメント	<ul style="list-style-type: none">• CN-Series 10.1.x or above Container Images• PanoramaPAN-OS 10.1.x以降のバージョンを実行している• Helm 3.6 or above version clientHelmを使用したCNシリーズのデプロイメント用

以下の手順を実行して、CN-Series ファイアウォールを Kubernetes サービスとして GKE プラットフォームにデプロイします。

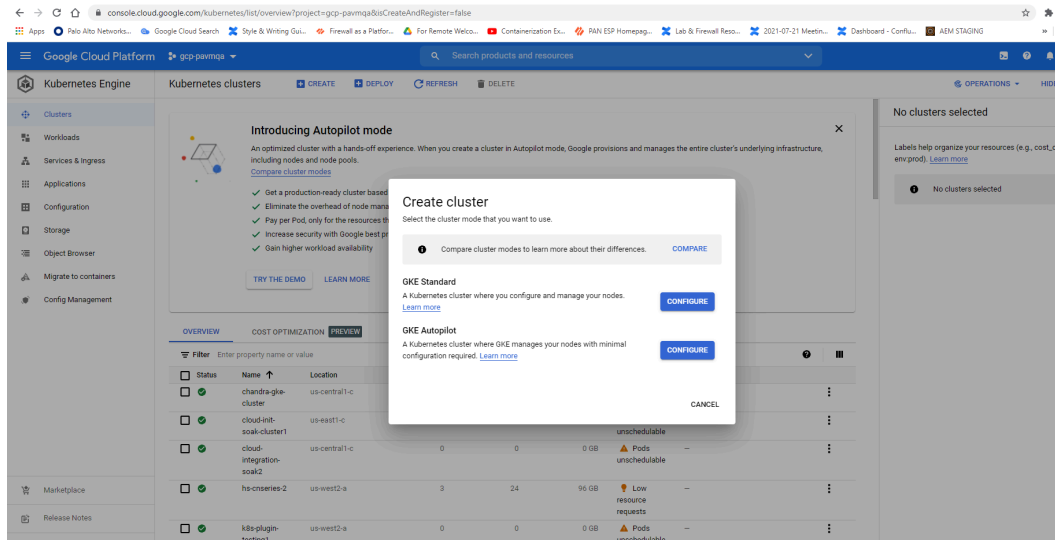
STEP 1 | Kubernetes クラスタをセットアップします。

GKEでクラスタを作成するには、次のようにします。

1. ナビゲーションメニューをクリックし、**Kubernetes Engine**に移動してから、クラスターを選択します。



2. 作成をクリックします。
3. 使用するクラスタモードとして **GKE 標準** を選択し、設定をクリックします。



4. 名前、バージョン、場所、ノードサブネットなどのクラスタの基本情報を入力し、作成をクリックします。



クラスタが *GKE* 上に存在する場合は、*Kubernetes* ネットワーク ポリシー *API* を有効にして、クラスタ管理者が相互に通信可能なポッドを指定できることを確認してください。この *API* は、*CN-NGFW* ポッドと *CN-MGMT* ポッドが通信するために必要です。

1. クラスタのリソースが適切であることを確認します。デフォルトのGKE ノードプール仕様は、CN-Series ファイアウォールには適していません。クラスタにファイアウォー

ルをサポートするための[CN-Series前提条件](#)リソースがあることを確認する必要があります。

kubectl get nodes

kubectl describe node <node-name>

コマンド出力の[容量]見出しの下にある情報を表示して、指定したノードで使用可能な CPU とメモリーを確認します。

CPU、メモリー、およびディスク ストレージの割り当てはニーズによって異なります。[CNシリーズのパフォーマンスとスケーリング](#)を参照してください。

以下の情報があることを確認してください。

- PanoramaにAPI サーバーを設定するためのエンドポイント IP アドレスを収集します。

Cluster Definition

Name: on_prem-clstr

Description:

API server address: 10.2...

Type: Native-Kubernetes

Credentials

Label Selector | Label Filter | Custom Certificate

0 items

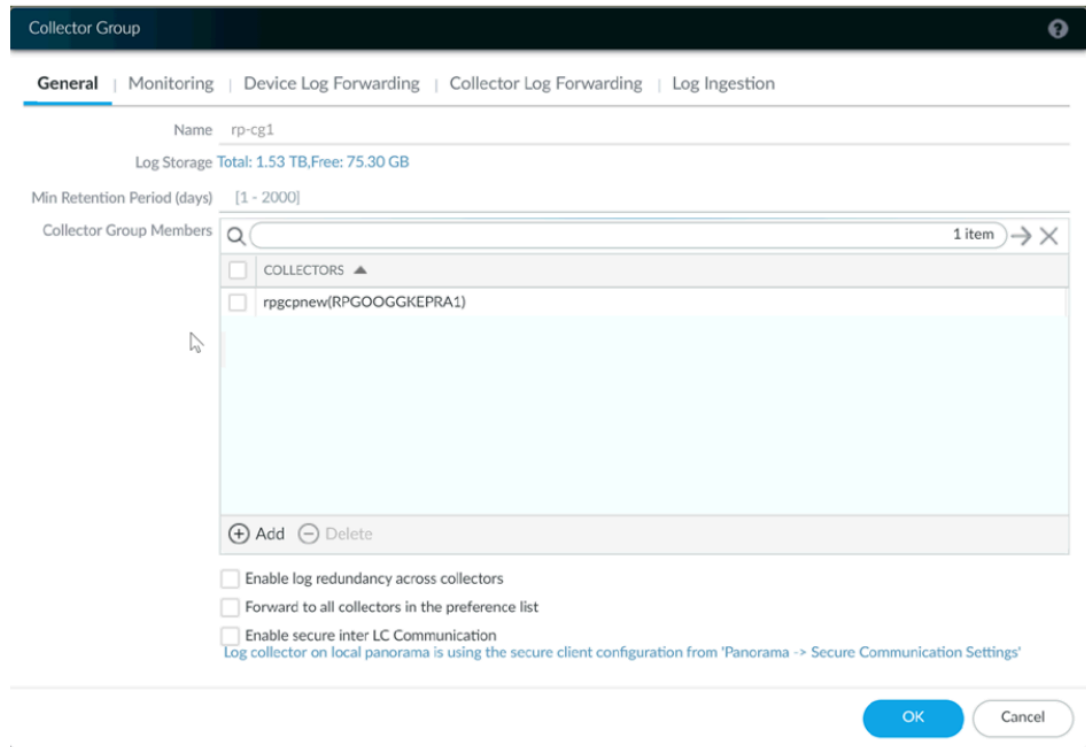
TAG PREFIX	NAMESPACE	LABEL SELECTOR FILTER	APPLY ON
------------	-----------	-----------------------	----------

+ Add - Delete

Validate OK Cancel

Panorama は、この IP アドレスを使用して、Kubernetes クラスターに接続します。

- テンプレートスタック名、デバイス グループ名、Panorama IP アドレス、およびオプションで ログ コレクタグループ名をPanorama から収集します。



詳細については、[親 デバイス グループとテンプレート スタックの作成](#)を参照してください。

- [認証コードと自動登録の PIN ID と値](#)を収集します。
- イメージをダウンロードしたコンテナ イメージ リポジトリの場所。

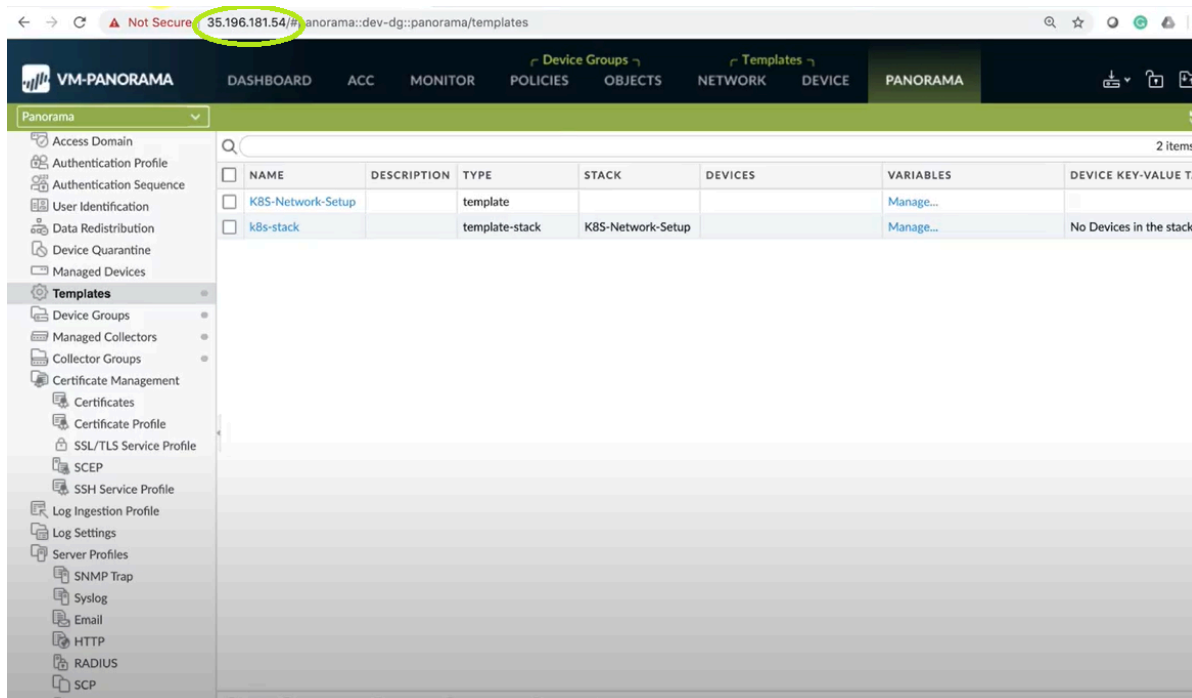
STEP 2 | (任意) Panorama の Kubernetes プラグインでカスタム証明書を設定した場合は、次のコマンドを実行して証明書シークレットを作成する必要があります。ファイル名を `ca.crt` から変更しないでください。 `pan-cn-mgmt.yaml` および `pan-cn-ngfw.yaml` のカスタム証明書のボリュームはオプションです。

```
kubectl -n kube-system create secret generic custom-ca --from-file = ca.crt
```

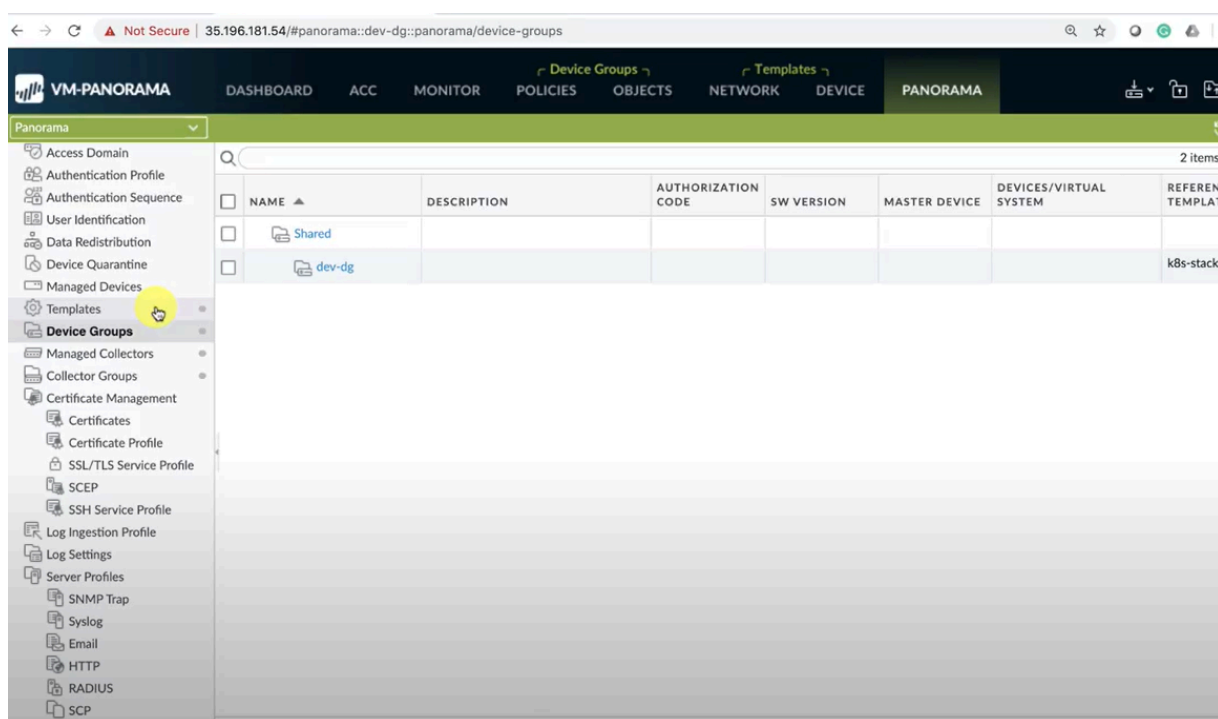
STEP 3 | YAML ファイルを編集して、CN-Series ファイアウォールをデプロイするために必要な詳細を記入します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: pan-mgmt-config
  namespace: kube-system
data:
  PAN_OPERATION_MODE: "daemonset"
  PAN_SERVICE_NAME: "pan-mgmt"
  # Panorama settings
  PAN_PANORAMA_IP: "35.196.181.54"
  PAN_PANORAMA_AUTH_KEY: 
  PAN_DEVICE_GROUP: "dev-dg"
  PAN_TEMPLATE: "k8s-stack"
#Non-mandatory parameters
  PAN_PANORAMA_CGNAME: "rp-cg1"
  #PAN_CERTIFICATE: ""
  #PAN_CERTKEYFILE: ""
  #PAN_CERTPASSPHRASE: ""
```

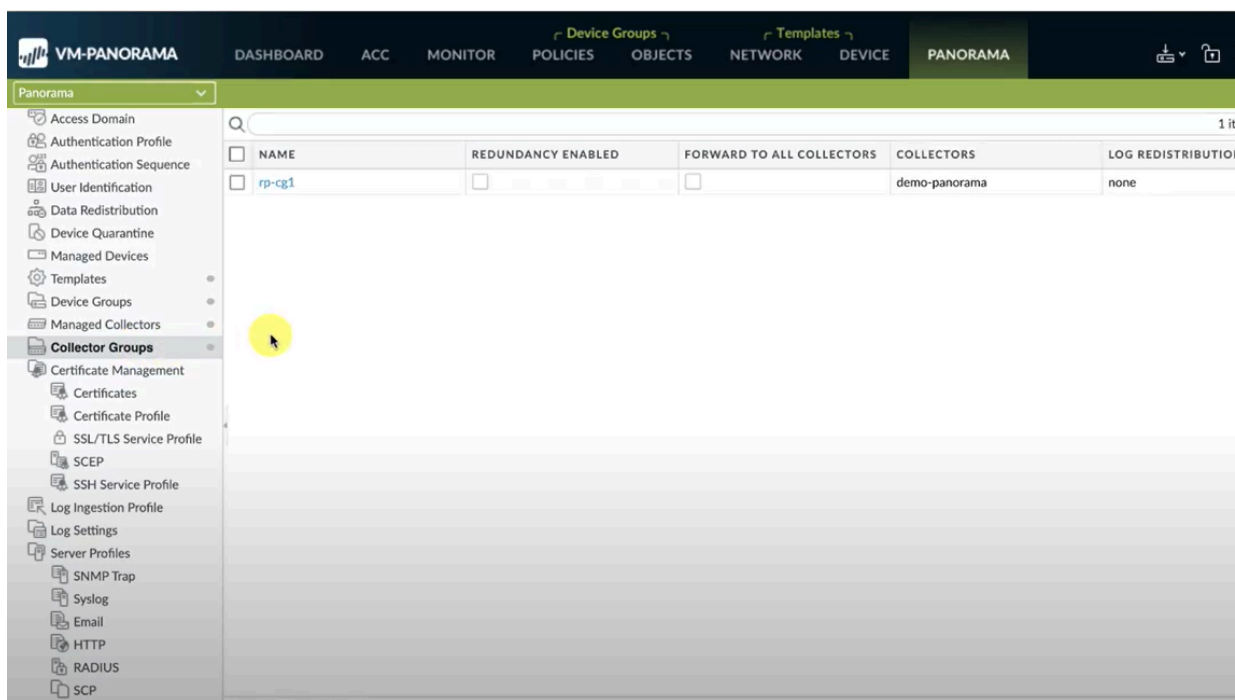
以下の図に示すように、YAML ファイルの PAN_PANORAMA_IP パラメータの値が実際の Panorama IP アドレスと一致していることを確認する必要があります。



以下の図に示すように、YAML ファイルの PAN_DEVICE_GROUP と PAN_TEMPLATE のパラメータ値が、Panorama で作成したデバイスグループとテンプレートスタックの名前と一致していることを確認する必要があります。



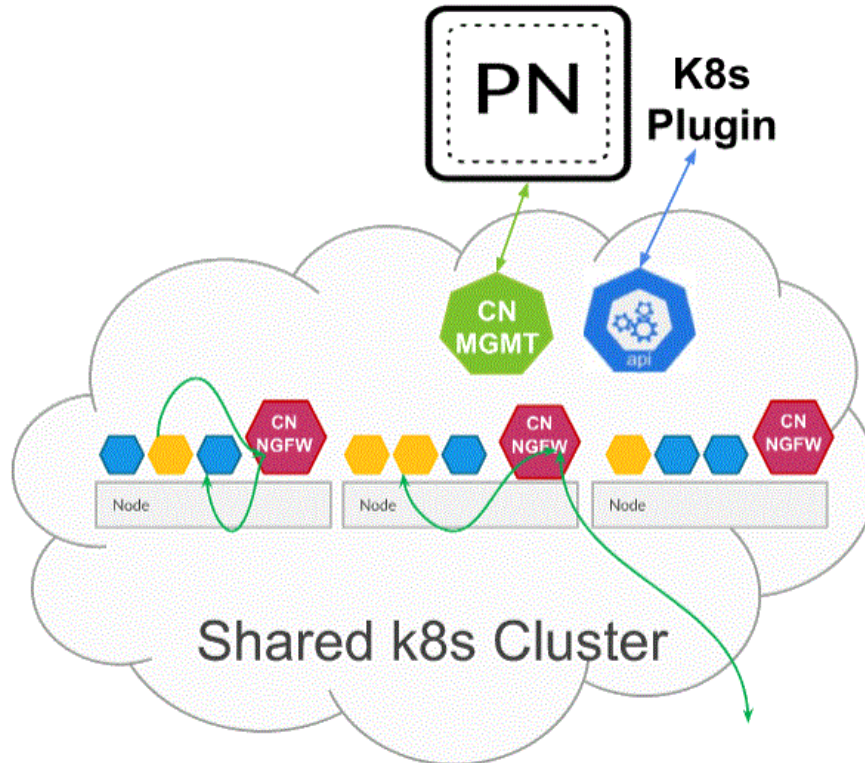
PAN_PANORAMA_CG_NAME のパラメータ値が、作成したログコレクター名と同じであることを確認する必要があります。



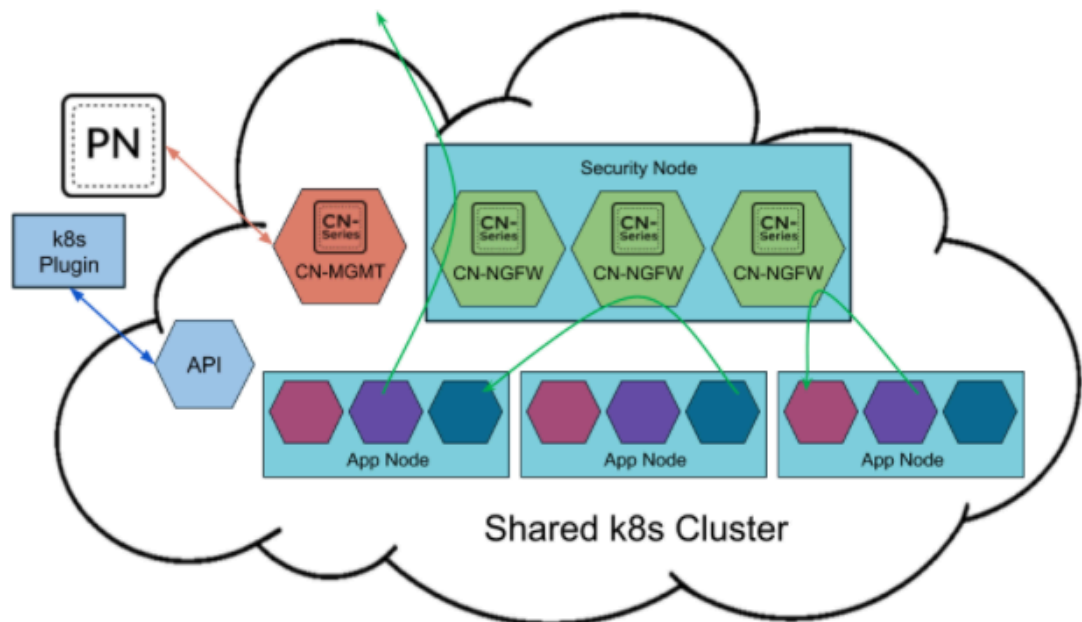
詳細については、[Editable parameters in CN-Series deployment yaml files \(CN-Series デプロイメント yaml ファイルの編集可能なパラメーター\)](#)を参照してください。

STEP 4 | Kubernetes 環境でオートスケーリングを使用している場合は、[Enable Horizontal Pod scaling \(Horizontal Pod のスケーリング\)](#)を参照してください。

STEP 5 | CN-NGFW サービスをデプロイします。次の手順を実行します。



Kubernetesサービスとしてデプロイすると、CN-NGFW のインスタンスをセキュリティノードにデプロイでき、アプリケーションポッドトラフィックは、検査と執行のために利用可能な CN-NGFW インスタンスにリダイレクトされます。



1. pan-cni-serviceaccount.yaml を使用してサービス アカウントが作成されたことを確認します。

クラスタ認証用のサービス アカウントの作成を参照してください。

2. Kubectl を使用して pan-cni-configmap.yaml を実行します。

```
kubectl apply -f pan-cni-configmap.yaml
```

3. kubectl を使用して pan-cn-ngfw-svc.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-svc.yaml
```



この yaml は *pan-cni.yaml* の前にデプロイする必要があります。

4. Kubectl を使用して pan-cni.yaml を実行します。

```
kubectl apply -f pan-cni.yaml
```

5. pan-cni-configmap YAML ファイルと pan-cni YAML ファイルが変更されたことを確認します。
6. 以下のコマンドを実行して、出力が以下の例のようになっていることを確認します。

```
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $ kubectl get pods -n kube-system | grep pan-cni
pan-cni-nmqkf          Running    0          2m11s
pan-cni-wjrkq          Running    0          2m11s
pan-cni-xrc2z          Running    0          2m12s
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $
```

STEP 6 | CN-MGMT StatefulSet をデプロイします。

デフォルトで、管理プレーンは耐障害性を提供する StatefulSet としてデプロイされます。1 つの CN-MGMT StatefulSet に最大 30 個のファイアウォール CN-NGFW ポッドを接続できます。

1. (静的にプロビジョニングされた PV のみに必要) CN-MGMT StatefulSet 用の永続ボリューム (PV) をデプロイします。

1. pan-cn-pv-local.yaml で定義されたローカル ボリューム名と一致するディレクトリを作成します。

少なくとも 2 つのワーカー ノード上に 6 つのディレクトリが必要です。CN-MGMT StatefulSet をデプロイする各ワーカー ノードにログインして、ディレクトリを作成

します。たとえば、/mnt/pan-local1 から /mnt/pan-local6 という名前のディレクトリを作成するには、次のコマンドを使用します。

```
mkdir -p /mnt/pan-local1 /mnt/pan-local2 /mnt/pan-local3 /
mnt/pan-local4 /mnt/pan-local5 /mnt/pan-local6
```

2. pan-cn-pv-local.yaml を変更します。

nodeaffinity の下でホスト名を一致させ、上記で spec.local.path に作成したディレクトリが変更されたことを確認してから、そのファイルをデプロイして、新しいストレージクラス pan-local-storage とローカル PV を作成します。

2. pan-cn-mgmt-configmap YAML ファイルと pan-cn-mgmt YAML ファイルが変更されたことを確認します。

EKS から pan-cn-mgmt-configmap をサンプリングします。

```
apiVersion: v1 kind:ConfigMap metadata: name: pan-mgmt-
config namespace: kube-system data: PAN_SERVICE_NAME: pan-
mgmt-svc PAN_MGMT_SECRET: pan-mgmt-secret # Panorama 設
定 PAN_PANORAMA_IP: "<panorama-IP>" PAN_DEVICE_GROUP:
"<panorama-device-group>" PAN_TEMPLATE_STACK: "<panorama-
template-stack>" PAN_CGNAME: "<panorama-collector-
group>" # ctnr mode: "k8s-service", "k8s-ilb-service"
PAN_CTNR_MODE_TYPE: "k8s-service" #必須でないパラメータ #
Panorama Kubernetes プラグインで提供されるクラスタ名と同じ名前
を持つことを推奨 - 同じ Panorama で複数のクラスタを管理する場合、
ポッドの識別が容易になります #CLUSTER_NAME: "<Cluster name>"
#PAN_PANORAMA_IP2: "" # CERT を使用する場合はコメントアウト
します。それ以外の場合は、pan-mgmt と pan-ngfw 間の IPsec 用に
PSK を使用します #IPSEC_CERT_BYPASS: "" # 値は不要 # jumbo-
frame モードの自動検出をオーバーライドし、システム全体を強制的に有
効にします #PAN_JUMBO_FRAME_ENABLED: "true" # GTP を有効にし
て MGMT を起動します。完全な機能を実現するには、Panorama でも GTP
# を有効にする必要があります。#PAN_GTP_ENABLED: 「true」# 高い
機能容量を有効にします。これらは MGMT ポッドには高いメモリを必要と
し、NGFW ポッドには以下の指定以上のメモリが必要です。#PAN_NGFW_MEMORY =
"6Gi" #PAN_NGFW_MEMORY = "40Gi" # より高速なデータパス-AF_XDP を有
効にするには、デフォルトは AF_PACKETV2 です。これにはカーネルのサポート
が必要です。#PAN_DATA_MODE: "次世代" #HPA params #PAN_CLOUD: "EKS"
#PAN_NAMESPACE_EKS: "EKSNamespace" #PUSH_INTERVAL: "15" #AWS
cloudwatch にメトリクスを発行する間隔
```

pan-cn-mgmt.yaml のサンプル

```
initContainers: - name: pan-mgmt-init image: <your-private-
registry-image-path>
```

```
containers: - name: pan-mgmt image: <your-private-registry-
image-path> terminationMessagePolicy:FallbackToLogsOnError
```

3. Kubectl を使用して yaml ファイルを実行します。

```
kubectl apply -f pan-cn-mgmt-configmap.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-crd.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-cr.yaml
```

```
kubectl apply -f pan-cn-mgmt-secret.yaml
```

```
kubectl apply -f pan-cn-mgmt.yaml
```

pan-mgmt-serviceaccount.yaml は、[クラスター認証用のサービスアカウントの作成](#)を以前に完了していない場合にのみ実行する必要があります。

4. 次のコマンドを実行して、CN-MGMT ポッドが起動していることを確認します。

```
kubectl get pods -l app = pan-mgmt -n kube-system
```

これには、5～6分かかります。

STEP 7 | CN-NGFW ポッドをデプロイします。

1. PAN-CN-NGFW-CONFIGMAP と PAN-CN-NGFW に詳述されているように YAML ファイルが変更されたことを確認します。

```
containers: - name: pan-ngfw-container image: <your-private-registry-image-path>
```

2. Kubectl apply を使用して pan-cn-ngfw-configmap.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-configmap.yaml
```

3. Kubectl apply を使用して pan-cn-ngfw.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw.yaml
```

4. CN-NGFW ポッドがデプロイされたことを確認します。

```
kubectl get pods -n kube-system -l app=pan-ngfw -o wide
```

STEP 8 | 次の手順を実行して、Horizontal podのオートスケーリングを有効にします。

1. CN-Series クラスターに [カスタム メトリック スタック ドライバー アダプター](#) をデプロイします。クラスター名は K8s シークレットを通じて指定する必要があります。
2. GKE 固有の HPA yaml ファイルを [Palo Alto Networks GitHub リポジトリ](#) からダウンロードします。
3. CN-MGMT がカスタム名前空間にデプロイされている場合は、カスタム名前空間を使用して pan-cn-adapater.yaml を更新します。デフォルトの名前空間は **kube-system** です。
4. GKE-specific の pan-cn-mgmt-configmap.yaml で HPA パラメータを更新します。

```
#PAN_CLOUD:"GKE"
```

```
#HPA_NAME: 「<name>」 #ネームスペースごと、またはテナントごとに hpa リソースを識別するための固有名称
```

```
#PUSH_INTERVAL: 「15」 #Stackdriver にメトリクスを発行するための時間間隔
```

5. 上記の pan-cn-mgmt-configmap.yaml ファイルで更新された HPA_NAME で **pan-cn-hpa-dp.yaml** と **pan-cn-hpa-mp.yaml** を変更し (name に置き換え)、トリガーする HPA に基づいてメトリックを更新します。
 1. レプリカの最小数と最大数を入力します。
 2. (任意) スケールダウンを変更し、デプロイメントに合わせて頻度値をスケールアップします。これらの値を変更しない場合は、デフォルト値が使用されます。
 3. (任意) スケーリングに使用する各メトリックのしきい値を変更します。これらの値を変更しない場合は、デフォルト値が使用されます。
 4. 変更を保存します。
6. HPA yaml ファイルをデプロイします。ファイルは、以下に説明する順序でデプロイする必要があります。
 1. Kubectl を使用して pan-cn-adapter.yaml を実行します
kubectl apply -f pan-cn-adapter.yaml
 2. Kubectl を使用して pan-cn-crole.yaml を実行します
kubectl apply -f pan-cn-crole.yaml
 3. Kubectl を使用して pan-cn-hpa-dp.yaml を実行します
kubectl apply -f pan-cn-hpa-dp.yaml
 4. Kubectl を使用して pan-cn-hpa-mp.yaml を実行します
kubectl apply -f pan-cn-hpa-mp.yaml
7. デプロイメントを確認します。

- kubectl を使用して、カスタム メトリックス名前空間内のカスタム メトリック アダプターポッドを確認します。

```
kubectl get pods -n custom-metrics
```

- kubectl を使用して、HPA リソースを確認します。

```
kubectl get hpa -n kube-system
```

```
kubectl describe hpa <hpa-name> -n kube-system
```

詳細については、「[Enable Horizontal Pod Autoscaling on the CN-Series \(CN-Seriesでの水平ポッド自動スケーリングの有効化\)](#)」を参照してください。

STEP 9 | Kubernetes クラスタ上の CN-MGMT、CN-NGFW、および PAN-CNI が表示されていることを確認します。

```
kubectl -n kube-system get pods
```

STEP 10 | 新しいポッドからのトラフィックがファイアウォールにリダイレクトされるようにアプリケーション yaml または名前空間に注釈を付けます。

検査のためにトラフィックを CN-NGFW にリダイレクトするには、以下のアノテーションを追加する必要があります：

```
annotations: paloaltonetworks.com/firewall: pan-fw
```

たとえば、「default」名前空間のすべての新しいポッドの場合：

```
kubectl annotate namespace default paloaltonetworks.com/  
firewall=pan-fw
```



一部のプラットフォームでは、CNI プラグイン チェーン内で *pan-cni* がアクティブになっていない状態でアプリケーション ポッドが開始する可能性があります。このようなシナリオを回避するには、アプリケーションポッド YAML にここに示すようにボリュームを指定する必要があります。

```
volumes: - name: pan-cni-ready hostPath: path: /var/log/  
pan-appinfo/pan-cni-ready type: ディレクトリ
```

STEP 11 | クラスタでアプリケーションをデプロイします。

CN-Series ファイアウォールを GKE に DaemonSet としてデプロイする

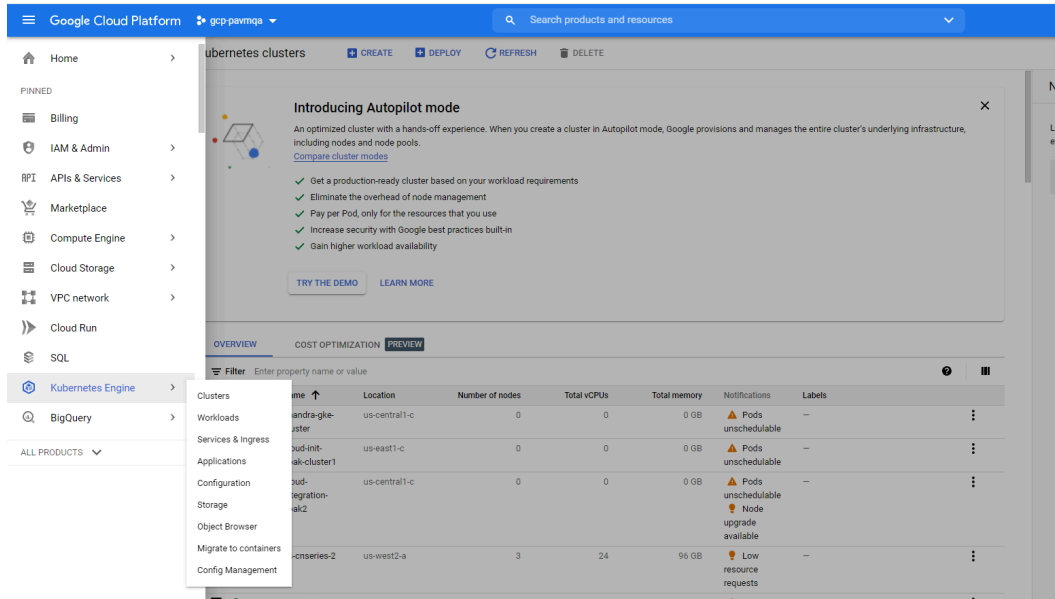
どこで使用できますか？	何が必要ですか？
<ul style="list-style-type: none">• CN-Seriesデプロイメント	<ul style="list-style-type: none">• CN-Series 10.1.x or above Container Images• PanoramaPAN-OS 10.1.x以降のバージョンを実行している• Helm 3.6 or above version clientHelmを使用したCNシリーズのデプロイメント用

CN-Series ファイアウォールを GKE プラットフォームに Daemonset としてデプロイするには、次の手順を実行します。

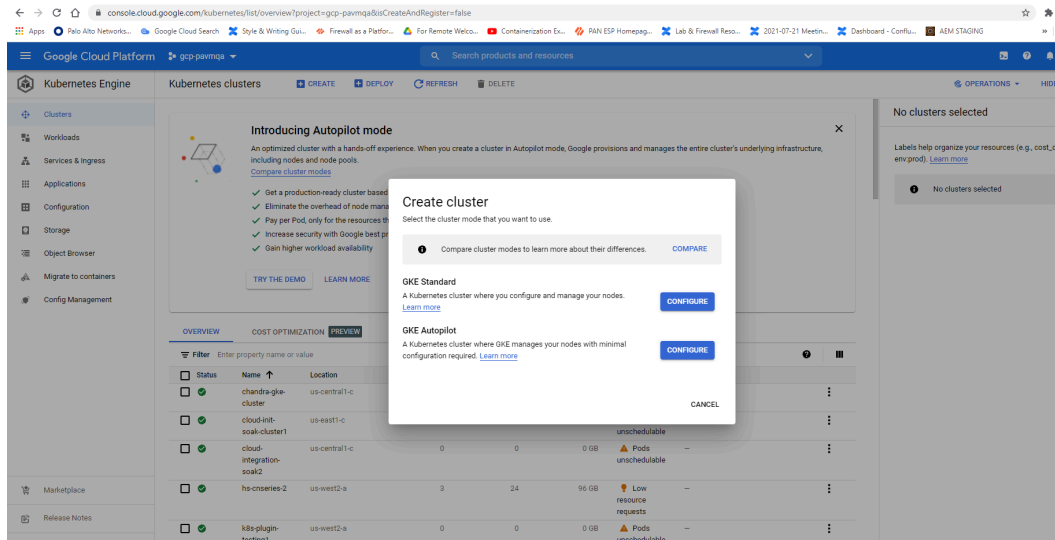
STEP 1 | Kubernetes クラスタをセットアップします。

GKEでクラスタを作成するには、次のようにします。

1. ナビゲーションメニューをクリックし、**Kubernetes Engine**に移動してから、クラスターを選択します。



2. 作成をクリックします。
3. 使用するクラスタモードとして **GKE 標準** を選択し、設定をクリックします。



4. 名前、バージョン、場所、ノードサブネットなどのクラスタの基本情報を入力し、作成をクリックします。



クラスタが GKE 上に存在する場合は、*Kubernetes* ネットワーク ポリシー API を有効にして、クラスタ管理者が相互に通信可能なポッドを指定できることを確認してください。この API は、*CN-NGFW* ポッドと *CN-MGMT* ポッドが通信するために必要です。

クラスタのリソースが適切であることを確認します。クラスタがファイアウォールをサポートするための *CN-Series システム要件* を持っていることを確認します。

kubectl get nodes

kubectl describe node <node-name>

コマンド出力の[容量]見出しの下にある情報を表示して、指定したノードで使用可能な CPU とメモリーを確認します。

CPU、メモリー、およびディスク ストレージの割り当てはニーズによって異なります。「[CN シリーズのパフォーマンスとスケーラビリティ](#)」を参照してください。

以下の情報があることを確認してください。

- Panorama 上で API サーバーをセットアップするためのエンドポイント IP アドレスを収集します。

Cluster Definition

Name: on_prem-clstr

Description:

API server address: 10.2...

Type: Native-Kubernetes

Credentials

Label Selector | Label Filter | Custom Certificate

0 items

TAG PREFIX	NAMESPACE	LABEL SELECTOR FILTER	APPLY ON
------------	-----------	-----------------------	----------

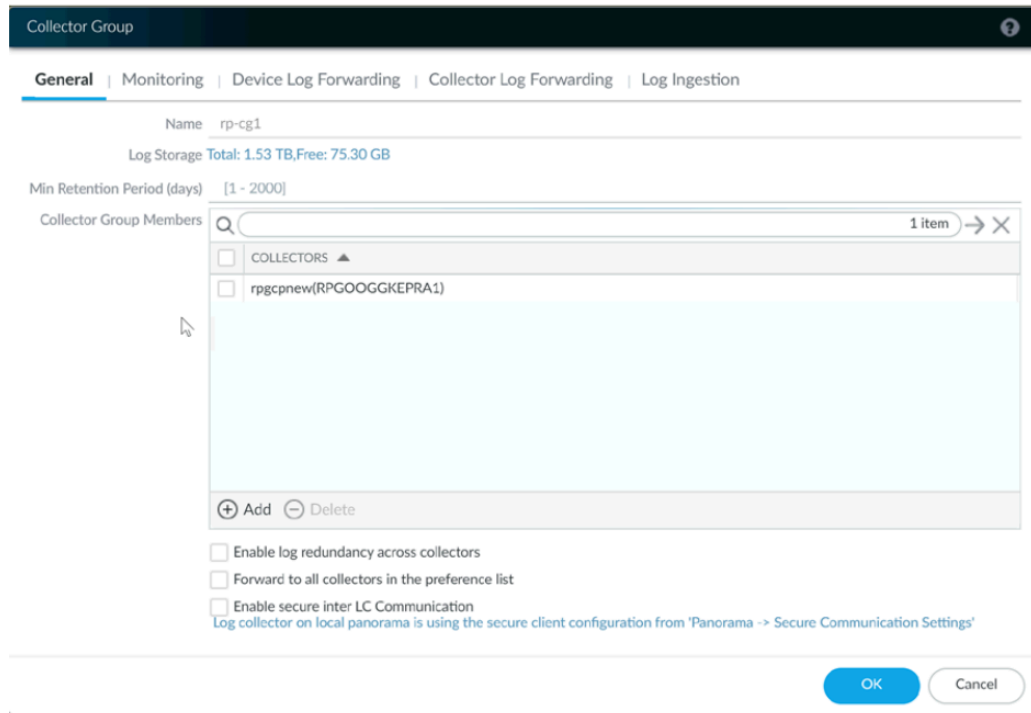
+ Add - Delete

Validate OK Cancel

Panorama は、この IP アドレスを使用して、Kubernetes クラスタに接続します。

詳細については、[クラスタをモニタリングするための Kubernetes プラグインの設定](#)をご覧ください。

- テンプレート スタック名、デバイス グループ名、Panorama IP アドレス、およびオプションで Panorama からログ コレクタ グループ名を収集します。



詳細については、[親 デバイス グループとテンプレート スタックの作成](#)を参照してください。

- [認証コードと自動登録の PIN ID と値](#)を収集します。
- イメージをダウンロードしたコンテナ イメージ リポジトリの場所。

STEP 2 | (任意) Panorama の Kubernetes プラグインでカスタム証明書を設定した場合は、次のコマンドを実行して証明書シークレットを作成する必要があります。ファイル名を `ca.crt` から変更しないでください。 `pan-cn-mgmt.yaml` および `pan-cn-ngfw.yaml` のカスタム証明書のボリュームはオプションです。

```
kubectl -n kube-system create secret generic custom-ca --from-file = ca.crt
```

STEP 3 | YAML ファイルを編集して、CN-Series ファイアウォールをデプロイするために必要な詳細を記入します。

非公開の Google Container レジストリへのパスを含め、必要なパラメータを提供するには、YAML ファイル内のイメージパスを置き換える必要があります。詳細は、「[CN-Series 配置の yaml ファイルで編集可能なパラメータ](#)」を参照してください。

STEP 4 | CNI DaemonSet をデプロイします。

CNI コンテナは、DaemonSet (ノードあたり 1 つのポッド) としてデプロイされ、ノード上にデプロイされたアプリケーションごとに 2 つずつのインターフェースを CN-NGFW ポッド上

に作成します。kubectl コマンドを使用して pan-cni YAML ファイルを実行すると、それが各ノード上のサービス チェーンに組み込まれます。

1. CN-Series ファイアウォールには、Kubernetes クラスター リソースと通信することを許可する最小権限を備えた 3 つのサービス アカウントが必要です。[CNシリーズクラスター認証用サービスアカウントを作成](#)し、pan-cni-serviceaccount.yaml を使用してサービス アカウントを作成したことを確認する必要があります。

2. Kubectl を使用して pan-cni-configmap.yaml を実行します。

```
kubectl apply -f pan-cni-configmap.yaml
```

3. Kubectl を使用して pan-cni.yaml を実行します。

```
kubectl apply -f pan-cni.yaml
```

4. pan-cni-configmap YAML ファイルと pan-cni YAML ファイルが変更されたことを確認します。
5. 以下のコマンドを実行して、出力が以下の例のようになっていることを確認します。

```
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $ kubectl get pods -n kube-system | grep pan-cni
pan-cni-nmqkf          Running    0          2m11s
pan-cni-wjtkq          Running    0          2m11s
pan-cni-xrc2z          Running    0          2m12s
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $
```

STEP 5 | CN-MGMT StatefulSet をデプロイします。

デフォルトで、管理プレーンは耐障害性を提供する StatefulSet としてデプロイされます。1 つの CN-MGMT StatefulSet に最大 30 個のファイアウォール CN-NGFW ポッドを接続できます。

1. pan-cn-mgmt-configmap YAML ファイルと pan-cn-mgmt YAML ファイルが変更されたことを確認します。

pan-cn-mgmt-configmap のサンプル

```
name: pan-mgmt-config
```

```
metadata:
```

```
namespace: kube-system
```

```
data:
```

```
PAN_SERVICE_NAME: pan-mgmt-svc
```

```
PAN_MGMT_SECRET: pan-mgmt-secret
```

```
# Panorama の設定
```

```
PAN_PANORAMA_IP: "x.y.z.a"
```

```
PAN_DEVICE_GROUP: "dg-1"
```

```
PAN_TEMPLATE_STACK: "temp-stack-1"
```

```
PAN_CGNAME: "CG-GKE"
```

```
必須ではないパラメータ
```

```
#Panorama Kubernetes プラグインで提供されるクラスタ名と同じ名前にすることを推奨。同じ Panorama で複数のクラスタを管理する場合、ポッドの識別が容易になります
```

```
#CLUSTER_NAME: "<Cluster name>"
```

```
#PAN_PANORAMA_IP2: ""
```

```
#Pan-mgmt と pan-ngfw の間で IPsec の PSK を使用しない場合は、CERT を使用するようにコメントアウトします
```

```
#IPSEC_CERT_BYPASS: ""
```

```
#値は不要です
```

```
#ジャンボフレームモードの自動検出をオーバーライドし、システム全体で強制的に有効化#PAN_JUMBO_FRAME_ENABLED: "true"
```

```
#GTPを有効にして MGMT ポッドを開始します。完全な機能を得るには、Panorama でも GTP の有効化が必要
```

```
#PAN_GTP_ENABLED: "true"
```

pan-cn-mgmt.yaml のサンプル

```
initContainers:  
  
  - name: pan-mgmt-init  
  
  image: <your-private-registry-image-path>  
  
containers: - name: pan-mgmt  
  
  image: <your-private-registry-image-path>  
  
terminationMessagePolicy:FallbackToLogsOnError
```

2. Kubectl を使用して yaml ファイルを実行します。

```
kubectl apply -f pan-cn-mgmt-configmap.yaml
```

```
kubectl apply -f pan-cn-mgmt-secret.yaml
```

```
kubectl apply -f pan-cn-mgmt.yaml
```

pan-mgmt-serviceaccount.yaml は、[クラスター認証用のサービスアカウントの作成](#)を以前に完了していない場合にのみ実行する必要があります。

3. CN-MGMT ポッドが起動していることを確認します。

これには、5 ～ 6 分かかります。

kubectl get pods -l app=pan-mgmt -n kube-system を使用します。

```
NAME READY STATUS RESTARTS AGEpan-mgmt-sts-0 1/1
```

```
Running 0 27hpan-mgmt-sts-1 1/1 Running 0 27h
```

STEP 6 | CN-NGFW ポッドをデプロイします。

デフォルトで、ファイアウォール データプレーン CN-NGFW ポッドは DaemonSet としてデプロイされます。CN-NGFW ポッドのインスタンスは、1 つのノード上で最大 30 個のアプリケーション ポッドのトラフィックを保護することができます。

1. PAN-CN-NGFW-CONFIGMAP と PAN-CN-NGFW に詳述されているように YAML ファイルが変更されたことを確認します。

```
containers: - name: pan-ngfw-container image: <your-private-registry-image-path>
```

2. Kubectl apply を使用して pan-cn-ngfw-configmap.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-configmap.yaml
```

3. Kubectl apply を使用して pan-cn-ngfw.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw.yaml
```

4. すべての CN-NGFW ポッド (クラスタ内のノードあたり 1 つずつ) が実行していることを確認します。

これは、4 ノード オンプレミス クラスタからの出力例です。

```
kubectl get pods -n kube-system -l app=pan-ngfw -o wide
```

```
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
```

```
pan-ngfw-ds-8g5xb 1/1 Running 0 27h 10.233.71.113 rk-k8-node-1 <none> <none>
```

```
pan-ngfw-ds-qsr6 1/1 Running 0 27h 10.233.115.189 rk-k8-vm-worker-1 <none> <none>
```

```
pan-ngfw-ds-vqk7z 1/1 Running 0 27h 10.233.118.208 rk-k8-vm-worker-3 <none> <none>
```

```
pan-ngfw-ds-zncqg 1/1 Running 0 27h 10.233.91.210 rk-k8-vm-worker-2 <none> <none>
```


STEP 7 | Kubernetes クラスタ上の CN-MGMT、CN-NGFW、および PAN-CNI が表示されていることを確認します。

```
kubectl -n kube-system get pods
```

```
0 27hpan-cni-5fhhg 1/1 Running
0 27hpan-cni-9j4rs 1/1 Running
0 27hpan-cni-ddwb4 1/1 Running
0 27hpan-cni-fwfrk 1/1 Running
0 27hpan-cni-h57lm 1/1 Running
0 27hpan-cni-h57lm 1/1 Running
0 27hpan-cni-j62rk 1/1 Running
0 27hpan-cni-lmxdz 1/1 Running
0 27hpan-mgmt-sts-0 1/1 Running
0 27hpan-mgmt-sts-1 1/1 Running
0 27hpan-ngfw-ds-8g5xb 1/1 Running
27hpan-ngfw-ds-qsr6 1/1 Running
0 27hpan-ngfw-ds-vqk7z 1/1 Running
0 27hpan-ngfw-ds-zncqg 1/1 Running
```

STEP 8 | 新しいポッドからのトラフィックがファイアウォールにリダイレクトされるようにアプリケーション yaml または名前空間に注釈を付けます。

検査のためにトラフィックを CN-NGFW にリダイレクトするには、以下のアノテーションを追加する必要があります：

```
annotations: paloaltonetworks.com/firewall: pan-fw
```

たとえば、「default」名前空間のすべての新しいポッドの場合：

```
kubectl annotate namespace default paloaltonetworks.com/
firewall=pan-fw
```



一部のプラットフォームでは、CNI プラグイン チェーン内で *pan-cni* がアクティブになっていない状態でアプリケーション ポッドが開始する可能性があります。このようなシナリオを回避するには、アプリケーションポッド YAML にここに示すようにボリュームを指定する必要があります。

```
volumes: - name: pan-cni-ready hostPath: path: /var/log/
pan-appinfo/pan-cni-ready type: ディレクトリ
```

STEP 9 | クラスタでアプリケーションをデプロイします。

CN-Series ファイアウォールを OKE にデプロイする

どこで使用できますか？	何が必要ですか？
<ul style="list-style-type: none"> • CN-Seriesデプロイメント 	<ul style="list-style-type: none"> • CN-Series 10.1.x or above Container Images • PanoramaPAN-OS 10.1.x以降のバージョンを実行している • Helm 3.6 or above version clientHelmを使用したCNシリーズのデプロイメント用

Oracle Kubernetes Engine (OKE) は、kubernetes クラスタをデプロイできる OCI サービスです。CN-Series ファイアウォールを Daemonset または Kubernetes サービスとして OKE クラスタにデプロイできるようになりました。

CNシリーズビルディングブロックとCNシリーズファイアウォールによるKubernetesの安全な環境内のワークフローの概要を確認したら、同じクラスタ内のコンテナ間やコンテナと他のワークロードタイプ(仮想マシンやベアメタルサーバーなど)間のトラフィックを保護するためのOKEプラットフォームにおけるCN-Seriesファイアウォールのデプロイから始めることができます。



Kubernetes クラスタ、アプリケーション、およびファイアウォールサービスをデプロイして管理するためには、*kubectl* や *Helm* などの標準の *Kubernetes* ツールが必要です。

詳細については、「*Helm* チャートとテンプレートを使用した **CN-Seriesファイアウォールのデプロイ**」を参照してください。*Panorama* は、*Kubernetes* クラスタのデプロイメントと管理用のオーケストレーターになるようには設計されていません。クラスタ管理用のテンプレートがマネージド *Kubernetes* プロバイダから提供されています。*Palo Alto Networks* は、*Helm* および *Terraform* で *CN-Series* をデプロイするためのコミュニティサポートのテンプレートを提供しています。

- CN-Series ファイアウォールを OKE に Kubernetes サービスとしてデプロイする
- CN-Series ファイアウォールを OKE に DaemonSet としてデプロイする



CN-Series を *DaemonSet* としての *CN-Series* からサービスとしての *CN-Series*、またはその逆に移行する前に、*plugin-serviceaccount.yaml* を削除して再適用する必要があります。詳細については、「[クラスタ認証用サービスアカウントの作成](#)」を参照してください。

- *CN-Series* を *DaemonSet* として *OKE* にデプロイする場合、*pan-plugin-cluster-mode-secret* が存在してはいけません。
- *CN-Series* を *Kubernetes* サービスとして *OKE* にデプロイする場合、*pan-plugin-cluster-mode-secret* が存在する必要があります。

CN-Series ファイアウォールを OKE に Kubernetes サービスとしてデプロイする

どこで使用できますか?	何が必要ですか?
<ul style="list-style-type: none">• CN-Seriesデプロイメント	<ul style="list-style-type: none">• CN-Series 10.1.x or above Container Images• PanoramaPAN-OS 10.1.x以降のバージョンを実行している• Helm 3.6 or above version clientHelmを使用したCNシリーズのデプロイメント用

以下の手順を実行して、CN-Series ファイアウォールを OKE プラットフォームに Kubernetes サービスとしてデプロイします。



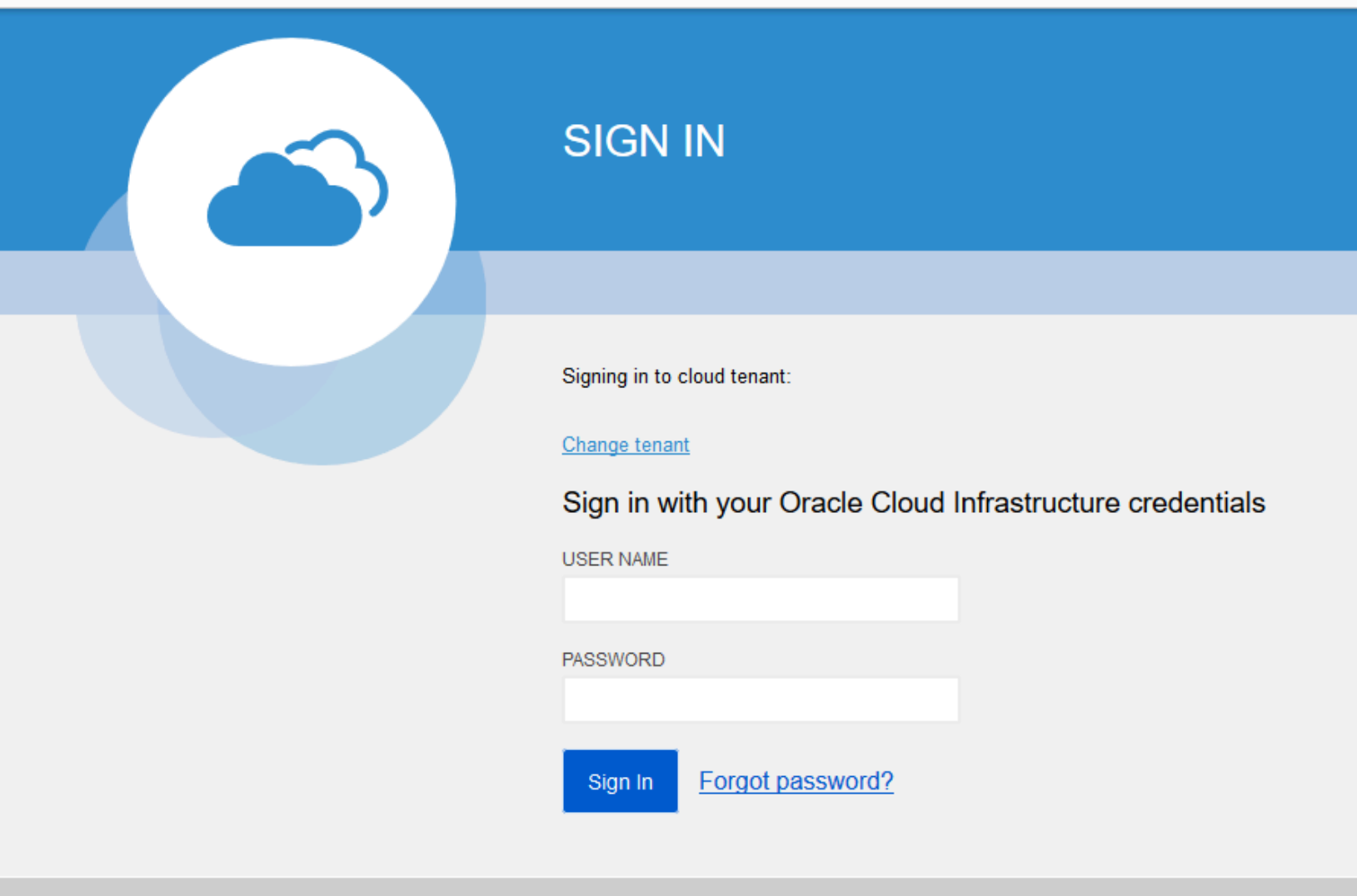
Oracle Linux 8.5 OSは、OKE に CN-Series ファイアウォールをデプロイするための唯一認定された環境です。

STEP 1 | Kubernetes クラスタをセットアップします。

OKE でクラスタを作成するには、次のように実施します。

1. Oracle Cloud Infrastructure にログインします。

ORACLE Cloud Infrastructure

The image shows the Oracle Cloud Infrastructure (OCI) Sign In page. It features a blue header with the OCI logo and the text "SIGN IN". Below the header is a large white circle containing a blue cloud icon. To the right of the icon, the text "SIGN IN" is displayed. Below this, the text "Signing in to cloud tenant:" is shown, followed by a link "Change tenant". The main section is titled "Sign in with your Oracle Cloud Infrastructure credentials". It contains two input fields: "USER NAME" and "PASSWORD". Below the "PASSWORD" field is a blue "Sign In" button and a link "Forgot password?".

Signing in to cloud tenant:

[Change tenant](#)

Sign in with your Oracle Cloud Infrastructure credentials

USER NAME

PASSWORD

[Sign In](#) [Forgot password?](#)

2. ナビゲーションメニューをクリックし、**Under Solutions and Platform**に移動して、**Developer Services**をクリックします。
3. **Kubernetes Clusters**をクリックします。
4. コンパートメントを選択し、クラスタの作成をクリックします。



Clusters *in* Tutorial2 Compartment



Clusters Requirements: [Preparing for Container Engine for Kubernetes](#)

[Show more information](#)

Create Cluster

Name

Status

Node Pools

VCN

Version

Cre

No clusters exist. Create one to get started.

5. [クラスタを作成] ダイアログ ボックスにて、カスタム作成をクリックし、ワークフローを起動をクリックします。
6. **Create Cluster** ページで、クラスタ名 とその他の詳細を入力します。
7. 次へをクリックして、新しいクラスタについて入力した詳細を確認します。
8. 確認ページで、**Create Cluster** をクリックします。



Cluster Creation

Cluster

NEW

Resources to be created

Basic Information

Cluster Name: cluster1**Compartment:** Tutorial2**Version:** v1.18.10

Network

Compartment: Tutorial2**VCN Name:** oke-vcn-quick-
cluster1-4baf5729a**Network Security Groups:** Not Enabled**Kubernetes API Private Endpoint:** Auto
Assigned**Kubernetes API Public Endpoint:** Auto
Assigned**Kubernetes CIDR Block:** 10.96.0.0/16[Create Cluster](#)[Cancel](#)

1. クラスタにファイアウォールをサポートするためのCN-Series前提条件リソースがあることを確認する必要があります。

kubectl get nodes**kubectl describe node <node-name>**

コマンド出力の[容量]見出しの下にある情報を表示して、指定したノードで使用可能なCPU とメモリーを確認します。

CPU、メモリー、およびディスク ストレージの割り当てはニーズによって異なります。CNシリーズのパフォーマンスとスケーリングを参照してください。

以下の情報があることを確認してください。

- PanoramaにAPI サーバーを設定するためのエンドポイント IP アドレスを収集します。

Cluster Definition ⓘ

Name:

Description:

API server address:

Type:

Credentials:

Label Selector | Label Filter | Custom Certificate

0 Items → ×

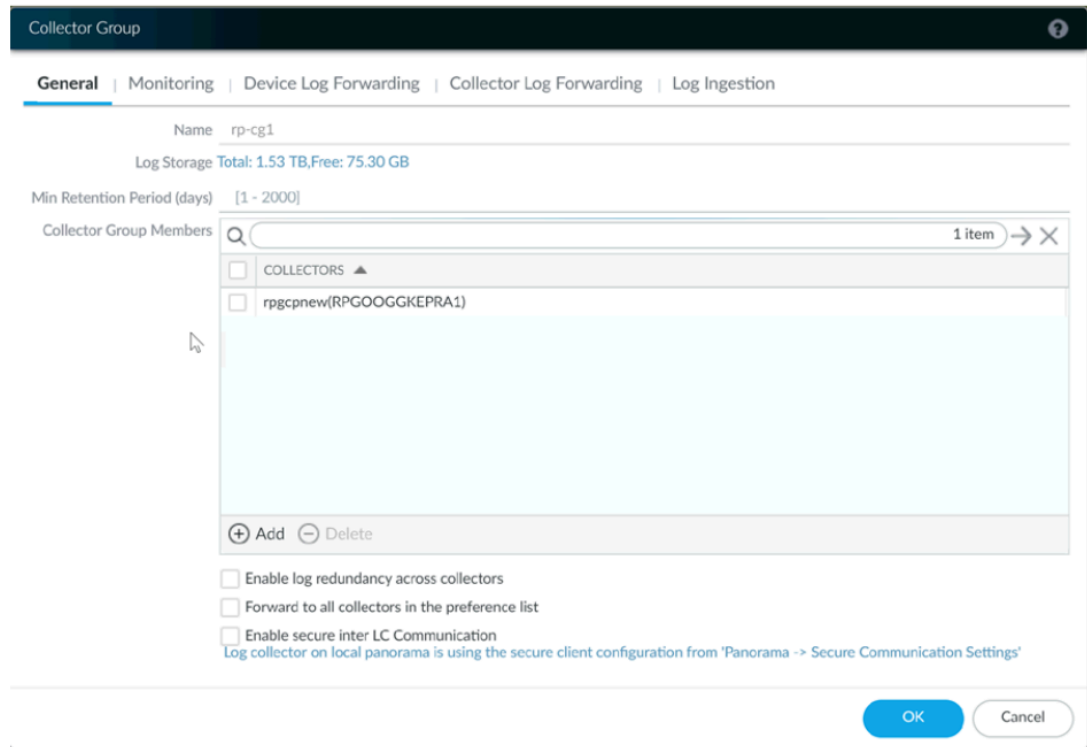
TAG PREFIX	NAMESPACE	LABEL SELECTOR FILTER	APPLY ON
------------	-----------	-----------------------	----------

+ Add - Delete

Validate OK Cancel

Panorama は、この IP アドレスを使用して、Kubernetes クラスタに接続します。

- テンプレートスタック名、デバイス グループ名、Panorama IP アドレス、およびオプションで ログ コレクタグループ名をPanorama から収集します。



詳細については、[親 デバイス グループとテンプレート スタックの作成](#)を参照してください。

- [認証コードと自動登録の PIN ID と値](#)を収集します。
- イメージをダウンロードしたコンテナ イメージリポジトリの場所を用意します。

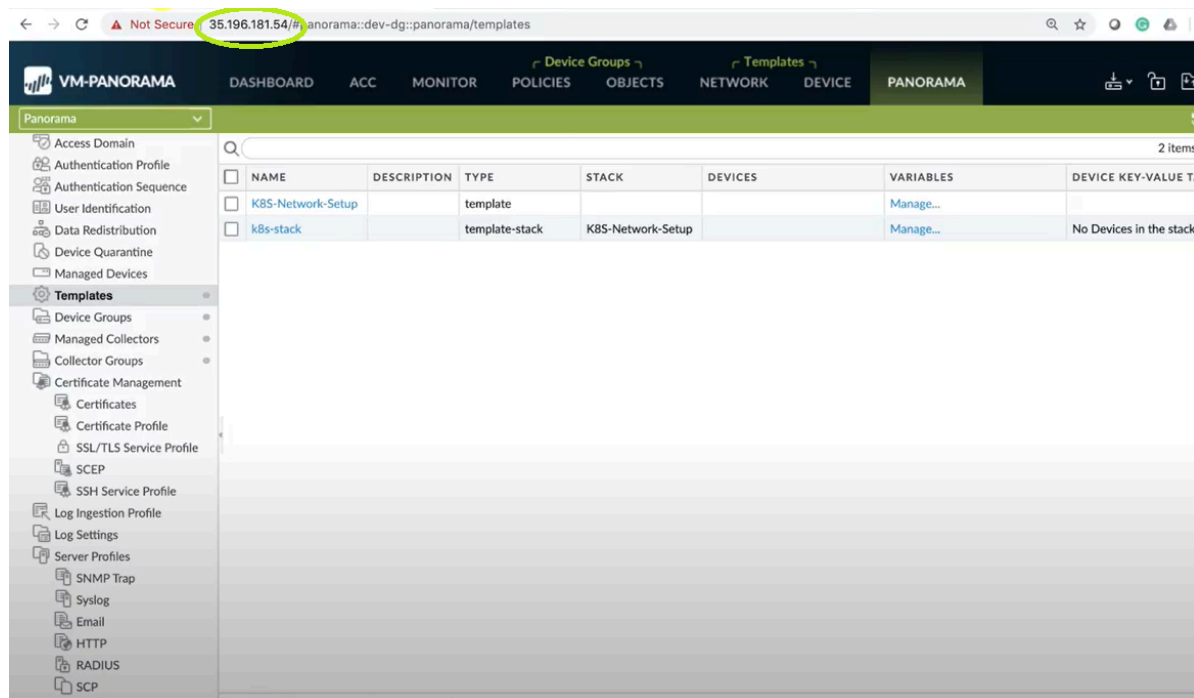
STEP 2 | (任意) Panorama の Kubernetes プラグインでカスタム証明書を設定した場合は、次のコマンドを実行して証明書シークレットを作成する必要があります。ファイル名を `ca.crt` から変更しないでください。 `pan-cn-mgmt-dynamic-pv.yaml` および `pan-cn-ngfw.yaml` のカスタム証明書のボリュームはオプションです。

```
kubectl -n kube-system create secret generic custom-ca --from-file = ca.crt
```

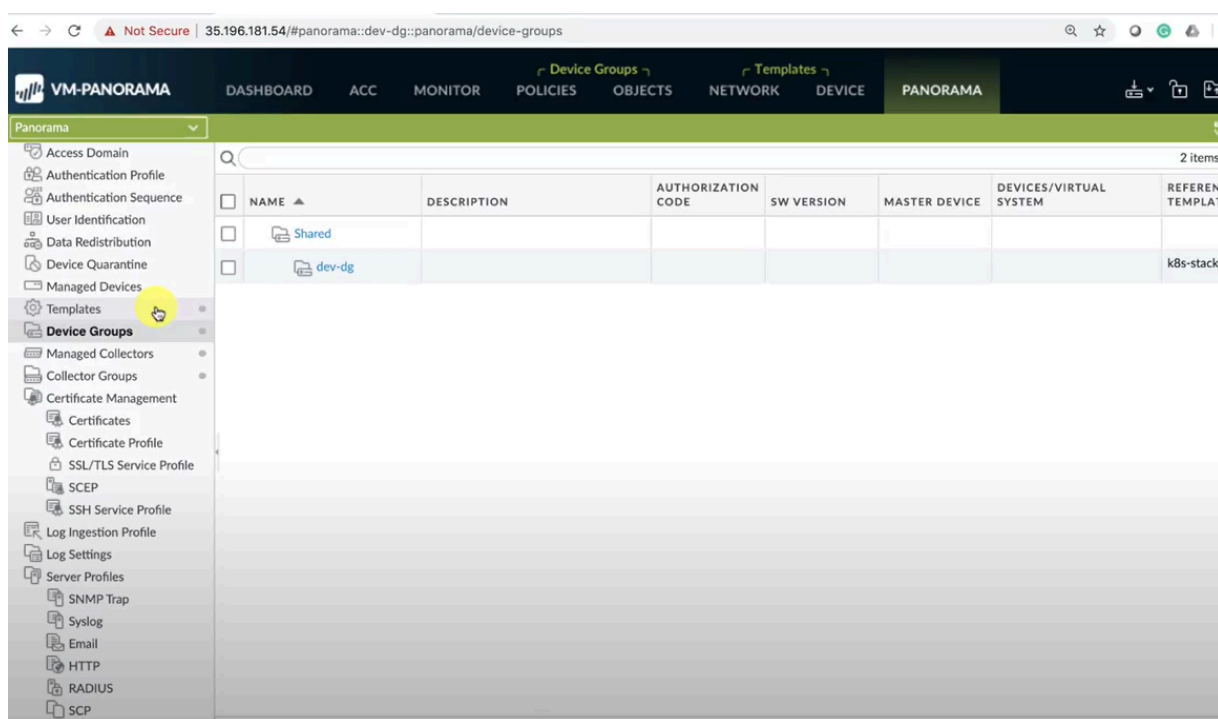
STEP 3 | YAML ファイルを編集して、CN-Series ファイアウォールをデプロイするために必要な詳細を記入します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: pan-mgmt-config
  namespace: kube-system
data:
  PAN_OPERATION_MODE: "daemonset"
  PAN_SERVICE_NAME: "pan-mgmt"
  # Panorama settings
  PAN_PANORAMA_IP: "35.196.181.54"
  PAN_PANORAMA_AUTH_KEY: 
  PAN_DEVICE_GROUP: "dev-dg"
  PAN_TEMPLATE: "k8s-stack"
#Non-mandatory parameters
  PAN_PANORAMA_CGNAME: "rp-cg1"
  #PAN_CERTIFICATE: ""
  #PAN_CERTKEYFILE: ""
  #PAN_CERTPASSPHRASE: ""
```

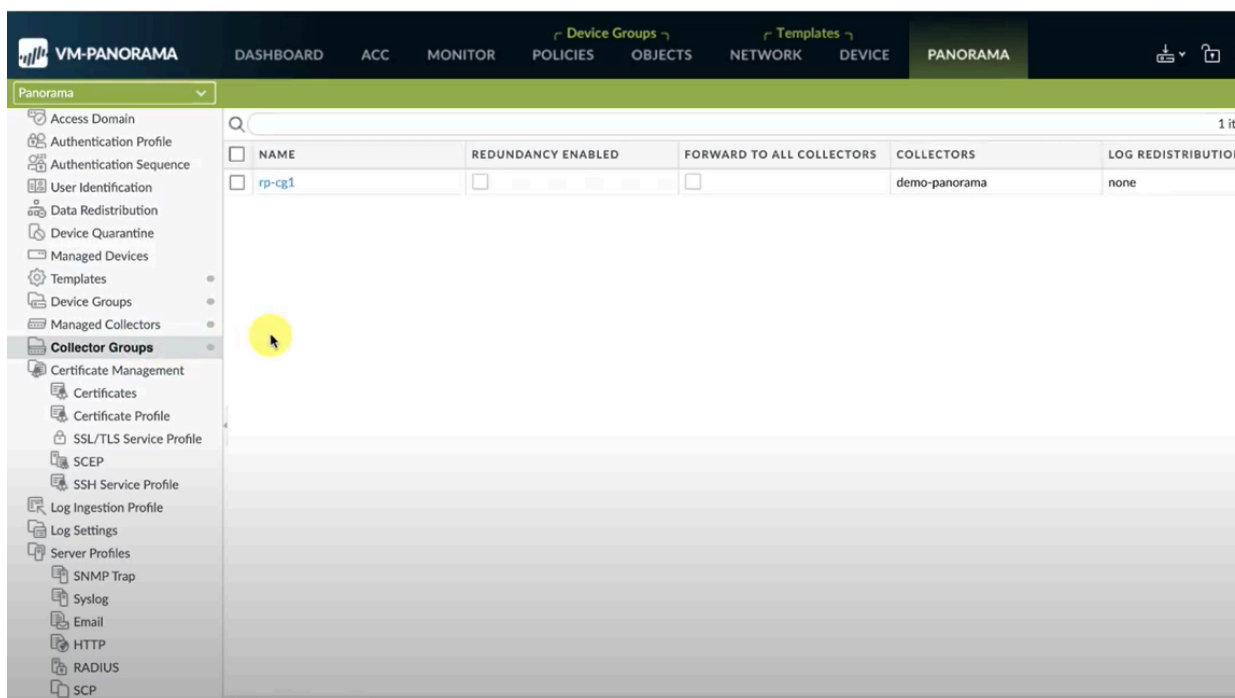
以下の図に示すように、YAML ファイルの PAN_PANORAMA_IP パラメータの値が実際の Panorama IP アドレスと一致していることを確認する必要があります。



以下の図に示すように、YAML ファイルの PAN_DEVICE_GROUP と PAN_TEMPLATE のパラメータ値が、Panorama で作成したデバイスグループとテンプレートスタックの名前と一致していることを確認する必要があります。

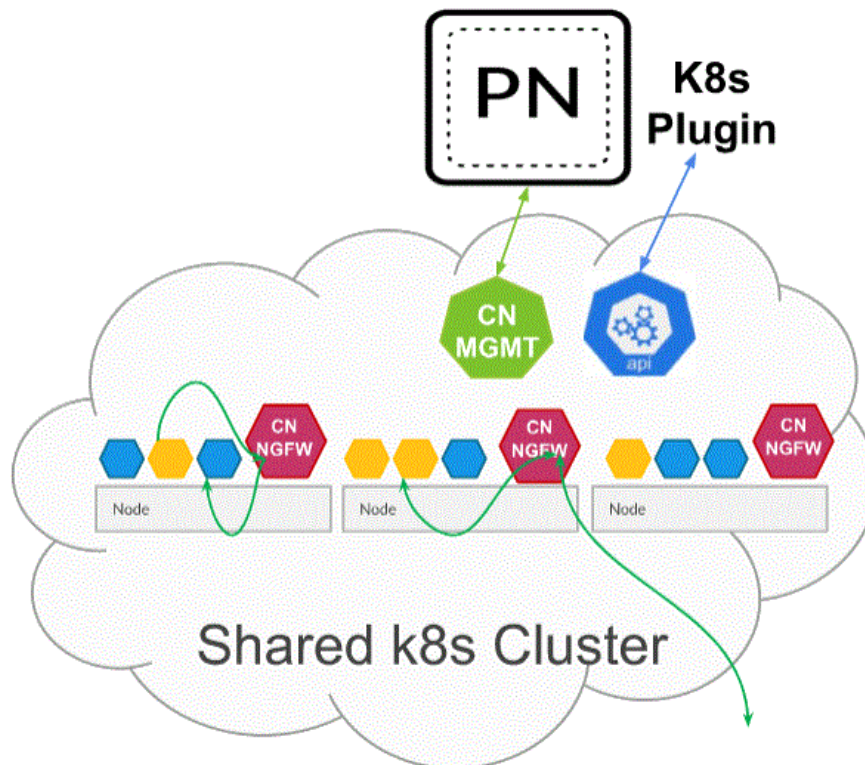


PAN_PANORAMA_CG_NAME のパラメータ値が、作成したログコレクター名と同じであることを確認する必要があります。



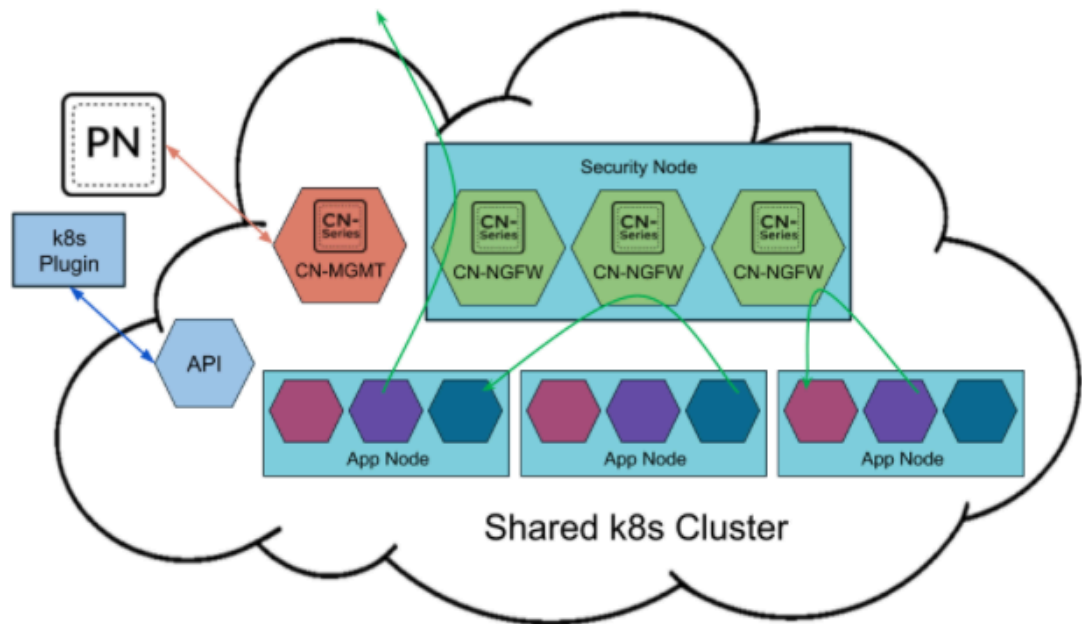
詳細については、[Editable parameters in CN-Series deployment yaml files \(CN-Series デプロイメント yaml ファイルの編集可能なパラメーター\)](#)を参照してください。

STEP 4 | CN-NGFW サービスをデプロイします。次の手順を実行します。



Kubernetes サービスとしてデプロイすると、CN-NGFW のインスタンスをセキュリティノードにデプロイでき、アプリケーションポッドトラフィックは、検査と適用のために利用可能な CN-NGFW インスタンスにリダイレクトされます。

- 📋 **CN-Series** ファイアウォールを *Kubernetes* サービスとして *OKE* にデプロイする場合、[pan-cn-k8s-service](#) ネイティブ フォルダの *yaml* ファイルを使用できます。



1. pan-cni-serviceaccount.yaml を使用してサービス アカウントが作成されたことを確認します。

クラスタ認証用のサービス アカウントの作成を参照してください。

2. Kubectl を使用して pan-cni-configmap.yaml を実行します。

kubectl apply -f pan-cni-configmap.yaml

3. kubectl を使用して pan-cn-ngfw-svc.yaml を実行します。

kubectl apply -f pan-cn-ngfw-svc.yaml



この yaml は pan-cni.yaml の前にデプロイする必要があります。

4. Kubectl を使用して pan-cni.yaml を実行します。

kubectl apply -f pan-cni.yaml

5. pan-cni-configmap YAML ファイルと pan-cni YAML ファイルが変更されたことを確認します。

6. 以下のコマンドを実行して、出力が以下の例のようになっていることを確認します。

```
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $ kubectl get pods -n kube-system | grep pan-cni
pan-cni-nmqkf          Running    0          2m11s
pan-cni-wjrkq          Running    0          2m11s
pan-cni-xrc2z          Running    0          2m12s
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $
```


STEP 5 | CN-MGMT StatefulSet をデプロイします。

デフォルトで、管理プレーンは耐障害性を提供する StatefulSet としてデプロイされます。1 つの CN-MGMT StatefulSet に最大 30 個のファイアウォール CN-NGFW ポッドを接続できます。

1. pan-cn-mgmt-configmap YAML ファイルと pan-cn-mgmt YAML ファイルが変更されたことを確認します。

OKE から pan-cn-mgmt-configmap をサンプリングします。

```
apiVersion: v1 kind:ConfigMap metadata: name: pan-mgmt-config
namespace: kube-system data: PAN_SERVICE_NAME: pan-mgmt-svc
PAN_MGMT_SECRET: pan-mgmt-secret # パノラマ設定 PAN_PANORAMA_IP:
"<panorama-ip>" PAN_DEVICE_GROUP: "<panorama-device-group>"
PAN_TEMPLATE_STACK: "<panorama-template-stack>" PAN_CGNAME:
"<panorama-collector-group>" PAN_CTNR_MODE_TYPE: "k8s-
service" #必須ではないパラメーター # Panorama Kubernetes プラ
グインで提供されるクラスター名と同じ名前にすることをお勧めします - 同
じ Panorama で複数のクラスターを管理する場合、ポッドを簡単に識別で
きます #CLUSTER_NAME: "<Cluster name>" #PAN_PANORAMA_IP2:
"" # CERT を使用する場合はコメントアウトします。それ以外の場
合は、pan-mgmt と pan-ngfw の間で IPsec の PSK を使用します
#IPSEC_CERT_BYPASS: "" # 値は必要ありません # ジャンボ フレーム モー
ドの自動検出をオーバーライドし、システム全体で強制的に有効にします #
PAN_JUMBO_FRAME_ENABLED: "true" # GTP を有効にして MGMT ポッドを
開始します。完全な機能を実現するには、Panorama でも GTP # を有効にする必
要があります。#PAN_GTP_ENABLED: 「true」 # 高い機能容量を有効にします。
これらには、MGMT ポッド用に大量のメモリが必要であり、NGFW ポッド用に以下
で指定されている # 以上のメモリが必要です。 # システム要件のドキュメント
を参照して、各メモリ プロファイルでサポートされている # サポートされている
NGFW CPU の最大サイズを確認してください。 #PAN_NGFW_MEMORY: "6.5Gi"
#PAN_NGFW_MEMORY: "48Gi" #PAN_NGFW_MEMORY: "56Gi"
```

pan-cn-mgmt-dynamic-pv.yaml のサンプル

```
initContainers: - name: pan-mgmt-init image: <your-private-
registry-image-path> command: ["/usr/bin/pan_start.sh"]
imagePullPolicy: Always
```

```
containers: - name: pan-mgmt image: <your-private-registry-
image-path> terminationMessagePolicy: FallbackToLogsOnError
```

2. Kubectl を使用して yaml ファイルを実行します。

```
kubectl apply -f pan-cn-mgmt-configmap.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-crd.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-cr.yaml
```

```
kubectl apply -f pan-cn-mgmt-secret.yaml
```

```
kubectl apply -f pan-cn-mgmt-dynamic-pv.yaml
```

pan-mgmt-serviceaccount.yaml は、[クラスター認証用のサービスアカウントの作成](#)を以前に完了していない場合にのみ実行する必要があります。

3. 次のコマンドを実行して、CN-MGMT ポッドが起動していることを確認します。

```
kubectl get pods -l app = pan-mgmt -n kube-system
```

これには、5～6分かかります。

STEP 6 | CN-NGFW ポッドをデプロイします。

1. PAN-CN-NGFW-CONFIGMAP と PAN-CN-NGFW に詳述されているように YAML ファイルが変更されたことを確認します。

```
containers: - name: pan-ngfw-container image: <your-private-registry-image-path>
```

2. Kubectl apply を使用して pan-cn-ngfw-configmap.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-configmap.yaml
```

3. Kubectl apply を使用して pan-cn-ngfw.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw.yaml
```

4. CN-NGFW ポッドがデプロイされたことを確認します。

```
kubectl get pods -n kube-system -l app=pan-ngfw -o wide
```

STEP 7 | Kubernetes クラスター上の CN-MGMT、CN-NGFW、および PAN-CNI が表示されていることを確認します。

```
kubectl -n kube-system get pods
```

STEP 8 | 新しいポッドからのトラフィックがファイアウォールにリダイレクトされるようにアプリケーション yml または名前空間に注釈を付けます。

検査のためにトラフィックを CN-NGFW にリダイレクトするには、以下のアノテーションを追加する必要があります：

```
annotations: paloaltonetworks.com/firewall: pan-fw
```

たとえば、「default」名前空間のすべての新しいポッドの場合：

```
kubectl annotate namespace default paloaltonetworks.com/  
firewall=pan-fw
```



一部のプラットフォームでは、CNI プラグイン チェーン内で *pan-cni* がアクティブになっていない状態でアプリケーション ポッドが開始する可能性があります。このようなシナリオを回避するには、アプリケーションポッド YAML にここに示すようにボリュームを指定する必要があります。

```
volumes: - name: pan-cni-ready hostPath: path: /var/log/  
pan-appinfo/pan-cni-ready type: ディレクトリ
```

STEP 9 | クラスタでアプリケーションをデプロイします。

CN-Series ファイアウォールを OKE に DaemonSet としてデプロイする

どこで使用できますか?	何が必要ですか?
<ul style="list-style-type: none">CN-Seriesデプロイメント	<ul style="list-style-type: none">CN-Series 10.2.x or above Container ImagesPanoramaPAN-OS 10.2.x以降のバージョンを実行しているHelm 3.6 or above version clientHelmを使用したCNシリーズのデプロイメント用

CN-Series ファイアウォールを OKE プラットフォーム に Daemonset としてデプロイするには、次の手順を実行します。



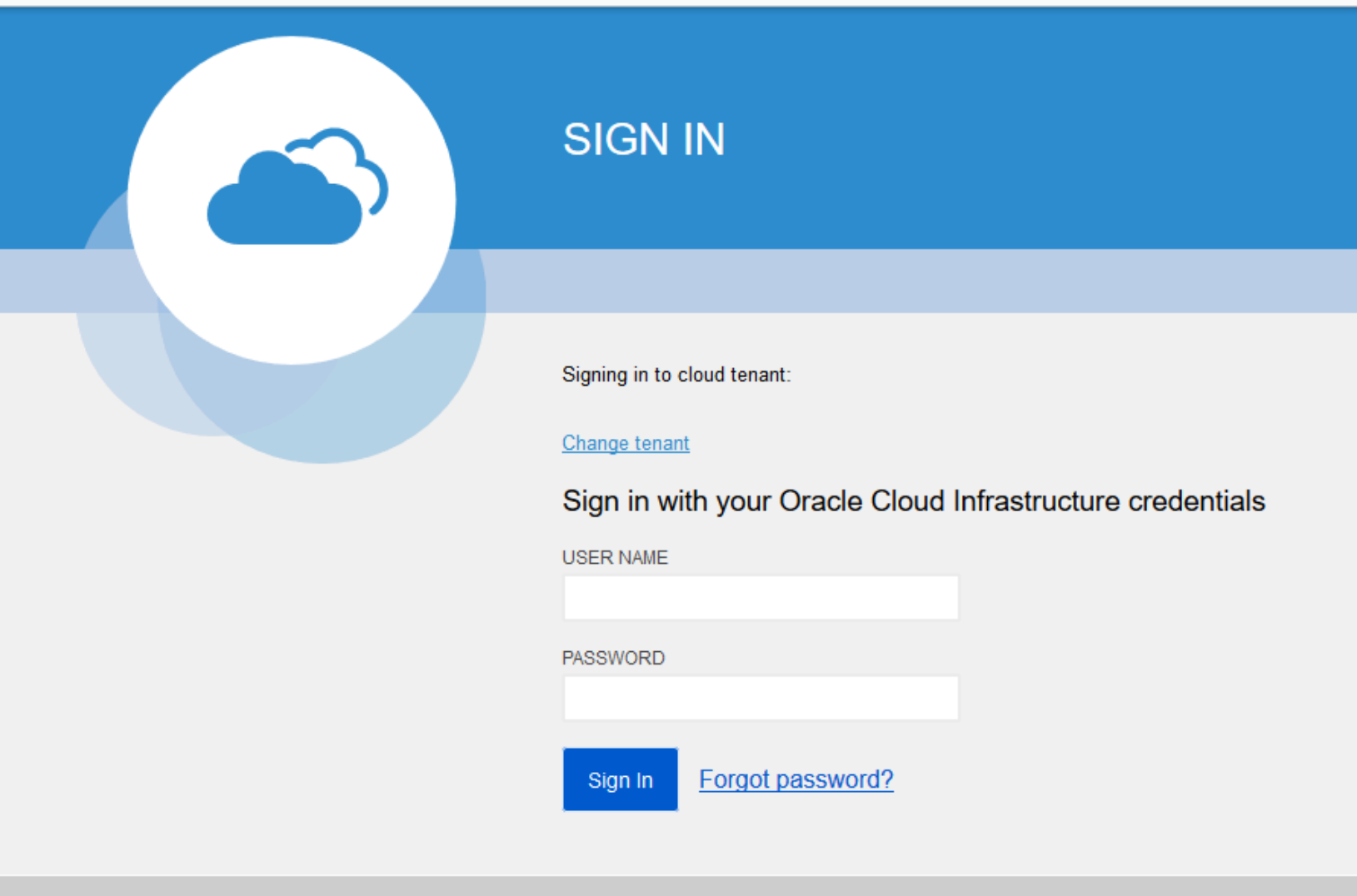
Oracle Linux 8.5 OSは、OKE に CN-Series ファイアウォールを導入するための唯一の認定環境です。

STEP 1 | Kubernetes クラスタをセットアップします。

OKE でクラスタを作成するには、次のように実施します。

1. Oracle Cloud Infrastructure にログインします。

ORACLE Cloud Infrastructure



The image shows the Oracle Cloud Infrastructure (OCI) Sign In page. It features a blue header with the OCI logo and the text 'SIGN IN'. Below the header is a large white circle containing a blue cloud icon. To the right of the icon, the text 'SIGN IN' is displayed. Below this, the text 'Signing in to cloud tenant:' is shown, followed by a link 'Change tenant'. The main section is titled 'Sign in with your Oracle Cloud Infrastructure credentials'. It contains two input fields: 'USER NAME' and 'PASSWORD'. Below the 'PASSWORD' field is a blue 'Sign In' button and a link 'Forgot password?'.

Signing in to cloud tenant:

[Change tenant](#)

Sign in with your Oracle Cloud Infrastructure credentials

USER NAME

PASSWORD

[Sign In](#) [Forgot password?](#)

2. ナビゲーションメニューをクリックし、ソリューションとプラットフォームに移動して、開発者サービスをクリックします。
3. **Kubernetes Clusters**をクリックします。
4. コンパートメントを選択し、クラスタの作成をクリックします。



Clusters *in* Tutorial2 Compartment



Clusters Requirements: [Preparing for Container Engine for Kubernetes](#)

[Show more information](#)

Create Cluster

Name

Status

Node Pools

VCN

Version

Create

No clusters exist. Create one to get started.

5. [クラスタを作成] ダイアログ ボックスにて、カスタム作成をクリックし、ワークフローを起動をクリックします。
6. **Create Cluster** ページで、クラスタ名 とその他の詳細を入力します。
7. 次へをクリックして、新しいクラスタについて入力した詳細を確認します。
8. レビューページで、**Create Cluster** をクリックします。

ORACLE Cloud

US West (Phoenix)

Cluster Creation

Create Cluster

Preview

Resources to be created

Basic Information

Cluster Name:

cluster1

Compartment:

Tutorial2

Version:

v1.18.10

Network

Compartment:

Tutorial2

VCN Name:

oke-vcn-quick-

cluster1-4baf5729a

Network Security Groups:

Not Enabled

Kubernetes API Private Endpoint:

Auto

Assigned

Kubernetes API Public Endpoint:

Auto

Assigned

Kubernetes CIDR Block:

10.96.0.0/16

Create Cluster

[Cancel](#)

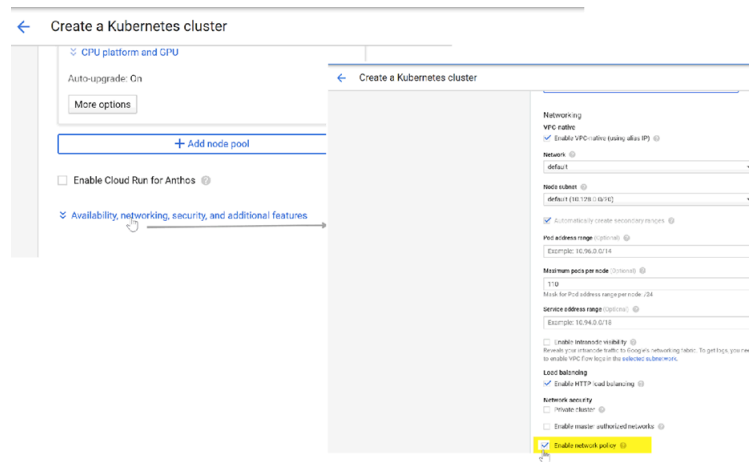
and Privacy

Cookie Preferences

Copyright © 2019, Oracle and/or its



クラスタが *GKE* 上に存在する場合は、クラスタ管理者が相互通信を許可するポッドを指定できるように、*Kubernetes* ネットワーク ポリシー *API* を有効にしてください。この *API* は、*CN-NGFW* ポッドと *CN-MGMT* ポッドが通信するために必要です。



クラスタのリソースが適切であることを確認します。ファイアウォールをサポートするための [CN シリーズの前提条件](#) リソースがクラスタにあることを確認します。

```
kubectl get nodes
```

```
kubectl describe node <node-name>
```

コマンド出力の[容量]見出しの下にある情報を表示して、指定したノードで使用可能な CPU とメモリーを確認します。

CPU、メモリー、およびディスク ストレージの割り当てはニーズによって異なります。「[CN シリーズのパフォーマンスとスケーラビリティ](#)」を参照してください。

以下の情報があることを確認してください。

- Panorama 上で API サーバーをセットアップするためのエンドポイント IP アドレスを収集します。

Cluster Definition

Name

on_prem-clstr

Description

API server address

10.2.

Type

Native-Kubernetes

Credentials

Label Selector

Label Filter

Custom Certificate

0 items

→

×

TAG PREFIX	NAMESPACE	LABEL SELECTOR FILTER	APPLY ON
------------	-----------	-----------------------	----------

+

 Add

-

 Delete

Validate

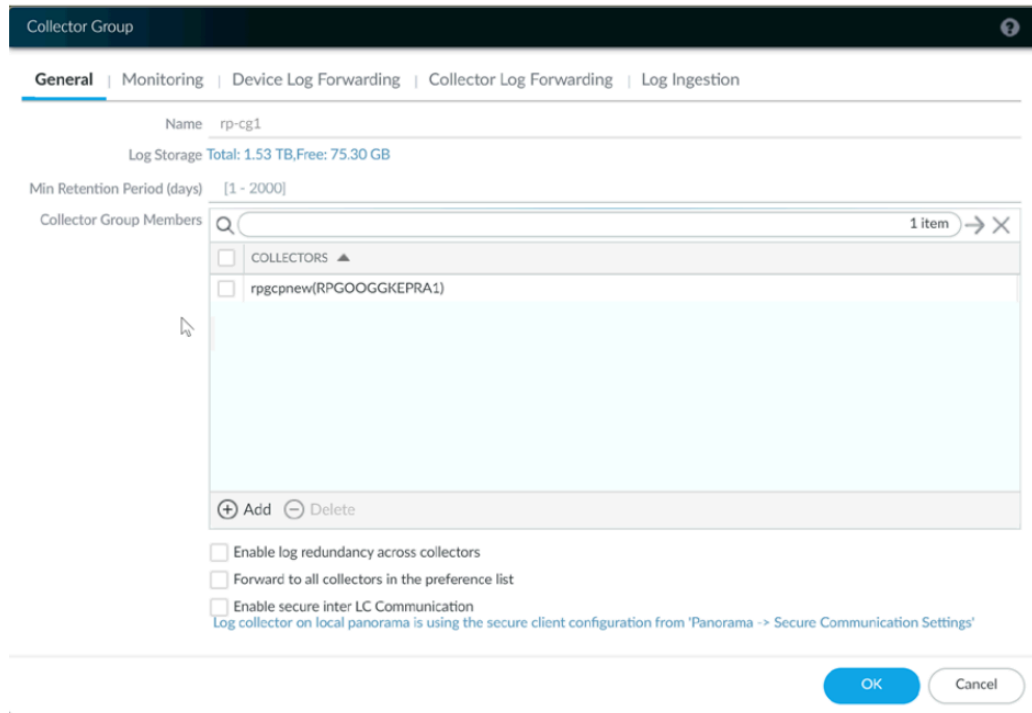
OK

Cancel

Panorama は、この IP アドレスを使用して、Kubernetes クラスタに接続します。

詳細については、[クラスタをモニタリングするための Kubernetes プラグインの設定](#)をご覧ください。

- テンプレート スタック名、デバイス グループ名、Panorama IP アドレス、およびオプションで Panorama からログ コレクタ グループ名を収集します。



詳細については、[親 デバイス グループとテンプレート スタックの作成](#)を参照してください。

- [認証コードと自動登録の PIN ID と値](#)を収集します。
- イメージをダウンロードしたコンテナ イメージ リポジトリの場所。

STEP 2 | (任意) Panorama の Kubernetes プラグインでカスタム証明書を設定した場合は、次のコマンドを実行して証明書シークレットを作成する必要があります。ファイル名を `ca.crt` から変更しないでください。pan-cn-mgmt-dynamic-pv.yaml および pan-cn-ngfw.yaml のカスタム証明書のボリュームはオプションです。

```
kubectl -n kube-system create secret generic custom-ca --from-file = ca.crt
```

STEP 3 | YAML ファイルを編集して、CN-Series ファイアウォールをデプロイするために必要な詳細を記入します。

非公開の Google Container レジストリへのパスを含め、必要なパラメータを提供するには、YAML ファイル内のイメージパスを置き換える必要があります。詳細は、「[CN-Series 配置の yaml ファイルで編集可能なパラメータ](#)」を参照してください。

STEP 4 | CNI DaemonSet をデプロイします。

CNI コンテナは、DaemonSet (ノードあたり 1 つのポッド) としてデプロイされ、ノード上にデプロイされたアプリケーションごとに 2 つずつのインターフェースを CN-NGFW ポッド上

に作成します。kubectl コマンドを使用して pan-cni YAML ファイルを実行すると、それが各ノード上のサービス チェーンに組み込まれます。



CN-Series ファイアウォールを *Kubernetes* サービスとして *OKE* にデプロイする場合、[pan-cn-k8s-daemonset](#) ネイティブフォルダーの *yaml* ファイルを使用できます。

1. CN-Series ファイアウォールには、Kubernetes クラスター リソースと通信することを許可する最小権限を備えた 3 つのサービス アカウントが必要です。[Create Service Accounts for Cluster Authentication with CN-Series](#)(CN-Seriesを使用したクラスター認証用サービスアカウントの作成)を作成し、pan-cni-serviceaccount.yamlを使用してサービスアカウントを作成したことを確認します。
2. Kubectl を使用して pan-cni-configmap.yaml を実行します。
kubectl apply -f pan-cni-configmap.yaml
3. Kubectl を使用して pan-cni.yaml を実行します。
kubectl apply -f pan-cni.yaml
4. pan-cni-configmap YAML ファイルと pan-cni YAML ファイルが変更されたことを確認します。
5. 以下のコマンドを実行して、出力が以下の例のようになっていることを確認します。

```
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $ kubectl get pods -n kube-system | grep pan-cni
pan-cni-nmqkf          Running    0          2m11s
pan-cni-wjrkq          Running    0          2m11s
pan-cni-xrc2z          Running    0          2m12s
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $
```

STEP 5 | CN-MGMT StatefulSet をデプロイします。

デフォルトで、管理プレーンは耐障害性を提供する StatefulSet としてデプロイされます。1 つの CN-MGMT StatefulSet に最大 30 個のファイアウォール CN-NGFW ポッドを接続できます。

1. pan-cn-mgmt-configmap YAML ファイルと pan-cn-mgmt YAML ファイルが変更されたことを確認します。

pan-cn-mgmt-configmap のサンプル

```
apiVersion: v1 kind: ConfigMap metadata: name: pan-mgmt-config
namespace: kube-system data: PAN_SERVICE_NAME: pan-mgmt-svc
PAN_MGMT_SECRET: pan-mgmt-secret # パノラマ設定 PAN_PANORAMA_IP:
"<panorama-ip>" PAN_DEVICE_GROUP: "<panorama-device-group>"
PAN_TEMPLATE_STACK: "<panorama-template-stack>" PAN_CGNAME:
"<panorama-collector-group>" # 必須ではないパラメーター # Panorama
Kubernetes プラグインで提供されるクラスター名と同じ名前にすることをお勧めします - 同じ Panorama #CLUSTER_NAME で複数のクラスターを管理する場合、ポッドを簡単に識別できます: "<Cluster name>"
#PAN_PANORAMA_IP2: "" # CERT を使用する場合はコメントアウトします。それ以外の場合は、pan-mgmt と pan-ngfw の間の IPSec の PSK
```

```
#IPSEC_CERT_BYPASS: "" # 値は必要ありません # ジャンボ フレーム モードの自動検出をオーバーライドし、システム全体で強制的に有効にします #
PAN_JUMBO_FRAME_ENABLED: "true" # GTP を有効にして MGMT ポッドを開始します。完全な機能を実現するには、Panorama でも GTP # を有効にする必要があります。#PAN_GTP_ENABLED: 「true」 # 高い機能容量を有効にします。これらには、MGMT ポッド用に大量のメモリが必要であり、NGFW ポッド用に以下で指定されている # 以上のメモリが必要です。 # システム要件のドキュメントを参照して、各メモリ プロファイルでサポートされている # サポートされている NGFW CPU の最大サイズを確認してください。 #PAN_NGFW_MEMORY:"6.5Gi" #PAN_NGFW_MEMORY:"48Gi" #PAN_NGFW_MEMORY:"56Gi"
```

pan-cn-mgmt-dynamic-pv.yaml のサンプル

```
initContainers: - name: pan-mgmt-init image: <your-private-registry-image-path>
```

```
containers: - name: pan-mgmt image: <your-private-registry-image-path> terminationMessagePolicy:FallbackToLogsOnError
```

2. Kubectl を使用して yaml ファイルを実行します。

```
kubectl apply -f pan-cn-mgmt-configmap.yaml
```

```
kubectl apply -f pan-cn-mgmt-secret.yaml
```

```
kubectl apply -f pan-cn-mgmt-dynamic-pv.yaml
```

CN-Seriesを使用したクラスター認証用のサービス アカウントの作成を以前に完了していない場合にのみ、pan-mgmt-serviceaccount.yaml を実行する必要があります。

3. CN-MGMT ポッドが起動していることを確認します。

これには、5 ～ 6 分かかります。

```
kubectl get pods -l app=pan-mgmt -n kube-system を使用します。
```

```
NAME READY STATUS RESTARTS AGEpan-mgmt-sts-0 1/1
```

```
Running 0 27hpan-mgmt-sts-1 1/1 Running 0 27h
```

STEP 6 | CN-NGFW ポッドをデプロイします。

デフォルトで、ファイアウォール データプレーン CN-NGFW ポッドは DaemonSet としてデプロイされます。CN-NGFW ポッドのインスタンスは、1 つのノード上で最大 30 個のアプリケーション ポッドのトラフィックを保護することができます。

1. PAN-CN-NGFW-CONFIGMAP と PAN-CN-NGFW に詳述されているように YAML ファイルが変更されたことを確認します。

```
containers: - name: pan-ngfw-container image: <your-private-registry-image-path>
```

2. Kubectl apply を使用して pan-cn-ngfw-configmap.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-configmap.yaml
```

3. Kubectl apply を使用して pan-cn-ngfw.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw.yaml
```

4. すべての CN-NGFW ポッド (クラスタ内のノードあたり 1 つずつ) が実行していることを確認します。

これは、4 ノード オンプレミス クラスタからの出力例です。

```
kubectl get pods -n kube-system -l app=pan-ngfw -o wide
```

```
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
```

```
pan-ngfw-ds-8g5xb 1/1 Running 0 27h 10.233.71.113 rk-k8-node-1 <none> <none>
```

```
pan-ngfw-ds-qsr6 1/1 Running 0 27h 10.233.115.189 rk-k8-vm-worker-1 <none> <none>
```

```
pan-ngfw-ds-vqk7z 1/1 Running 0 27h 10.233.118.208 rk-k8-vm-worker-3 <none> <none>
```

```
pan-ngfw-ds-zncqg 1/1 Running 0 27h 10.233.91.210 rk-k8-vm-worker-2 <none> <none>
```

STEP 7 | Kubernetes クラスタ上の CN-MGMT、CN-NGFW、および PAN-CNI が表示されていることを確認します。

```
kubectl -n kube-system get pods
```

```
0 27hpan-cni-5fhbg 1/1 Running
0 27hpan-cni-9j4rs 1/1 Running
0 27hpan-cni-ddwb4 1/1 Running
0 27hpan-cni-fwfrk 1/1 Running
0 27hpan-cni-h57lm 1/1 Running
0 27hpan-cni-h57lm 1/1 Running
0 27hpan-cni-j62rk 1/1 Running
0 27hpan-cni-lmxdz 1/1 Running
0 27hpan-mgmt-sts-0 1/1 Running
0 27hpan-mgmt-sts-1 1/1 Running
0 27hpan-ngfw-ds-8g5xb 1/1 Running
27hpan-ngfw-ds-qsr6 1/1 Running
0 27hpan-ngfw-ds-vqk7z 1/1 Running
0 27hpan-ngfw-ds-zncqg 1/1 Running
```

STEP 8 | 新しいポッドからのトラフィックがファイアウォールにリダイレクトされるようにアプリケーション yaml または名前空間に注釈を付けます。

検査のためにトラフィックを CN-NGFW にリダイレクトするには、以下のアノテーションを追加する必要があります：

```
annotations: paloaltonetworks.com/firewall: pan-fw
```

たとえば、「default」名前空間のすべての新しいポッドの場合：

```
kubectl annotate namespace default paloaltonetworks.com/
firewall=pan-fw
```



一部のプラットフォームでは、CNI プラグイン チェーン内で *pan-cni* がアクティブになっていない状態でアプリケーション ポッドが開始する可能性があります。このようなシナリオを回避するには、アプリケーションポッド YAML にここに示すようにボリュームを指定する必要があります。

```
volumes: - name: pan-cni-ready hostPath: path: /var/log/
pan-appinfo/pan-cni-ready type: ディレクトリ
```

STEP 9 | クラスタでアプリケーションをデプロイします。

CN-Series ファイアウォールを EKS にデプロイする

どこで使用できますか？	何が必要ですか？
<ul style="list-style-type: none"> CN-Seriesデプロイメント 	<ul style="list-style-type: none"> CN-Series 10.1.x or above Container Images PanoramaPAN-OS 10.1.x以降のバージョンを実行している Helm 3.6 or above version clientHelmを使用したCNシリーズのデプロイメント用

CNシリーズ ビルディング ブロックとCNシリーズ ファイアウォールによるKubernetesの安全な環境内のワークフローの概要を確認したら、同じクラスタ内のコンテナ間やコンテナと他のワークロード タイプ(仮想マシンやベアメタル サーバーなど)間のトラフィックを保護するためのAWS EKS プラットフォームにおけるCN-Seriesファイアウォールのデプロイから始めることができます。



Kubernetes クラスタ、アプリケーション、およびファイアウォール サービスをデプロイして管理するためには、*kubectl* や *Helm* などの標準の *Kubernetes* ツールが必要です。

詳細については、「*Helm* チャートとテンプレートを使用した [CN-Seriesファイアウォールのデプロイ](#)」を参照してください。*Panorama* は、*Kubernetes* クラスタのデプロイメントと管理用のオーケストレーターになるようには設計されていません。クラスタ管理用のテンプレートがマネージド *Kubernetes* プロバイダから提供されています。*Palo Alto Networks* は、[Helm](#) および [Terraform](#) で *CN-Series* をデプロイするためのコミュニティサポートのテンプレートを提供しています。

- [CN-Series ファイアウォールを AWS EKS に Kubernetes サービスとしてデプロイする](#)
- [CN-Series ファイアウォールを AWS EKS に DaemonSet としてデプロイする](#)
- [AWS Marketplace から CN-Seriesをデプロイする](#)



CN-Series を *DaemonSet* としての *CN-Series* からサービスとしての *CN-Series*、またはその逆に移行する前に、*plugin-serviceaccount.yaml* を削除して再適用する必要があります。詳細については、「[クラスター認証用サービスアカウントの作成](#)」を参照してください。

- *CN-Series* を EKS 上の *DaemonSet* として デプロイする場合、*pan-plugin-cluster-mode-secret* が存在してはいけません。
- *CN-Series* を EKS 上の *Kubernetes* サービスとしてデプロイする場合は、*pan-plugin-cluster-mode-secret* が存在する必要があります。

CN-Series ファイアウォールを AWS EKS に Kubernetes サービスとしてデプロイする

どこで使用できますか？	何が必要ですか？
<ul style="list-style-type: none">• CN-Seriesデプロイメント	<ul style="list-style-type: none">• CN-Series 10.1.x or above Container Images• PanoramaPAN-OS 10.1.x以降のバージョンを実行している• Helm 3.6 or above version clientHelmを使用したCNシリーズのデプロイメント用

以下の手順を実行して、CN-Series ファイアウォールを Kubernetes サービスとしてデプロイします。

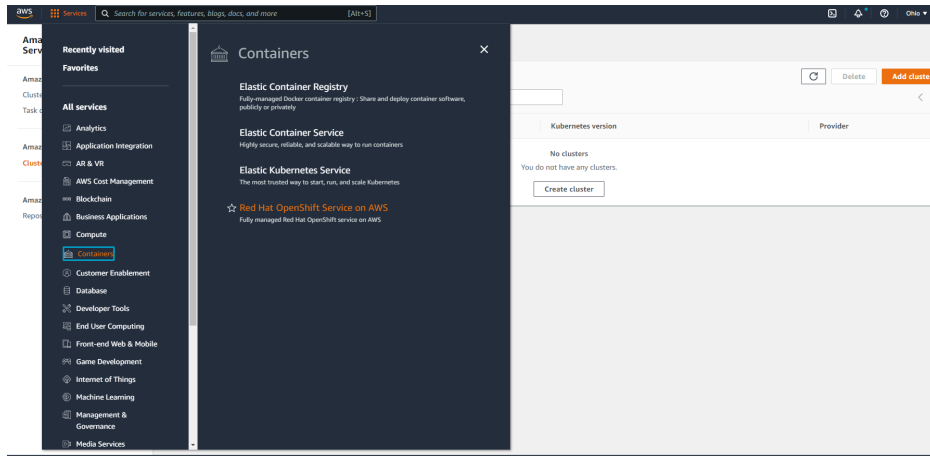
開始する前に、CN-Series YAML ファイルのバージョンが PAN-OS バージョンと互換性があることを確認します。

- PAN-OS10.1.2 以降には YAML 2.0.2 が必要です
- PAN-OS10.1.0 および 10.1.1 には YAML 2.0.0 または 2.0.1 が必要です

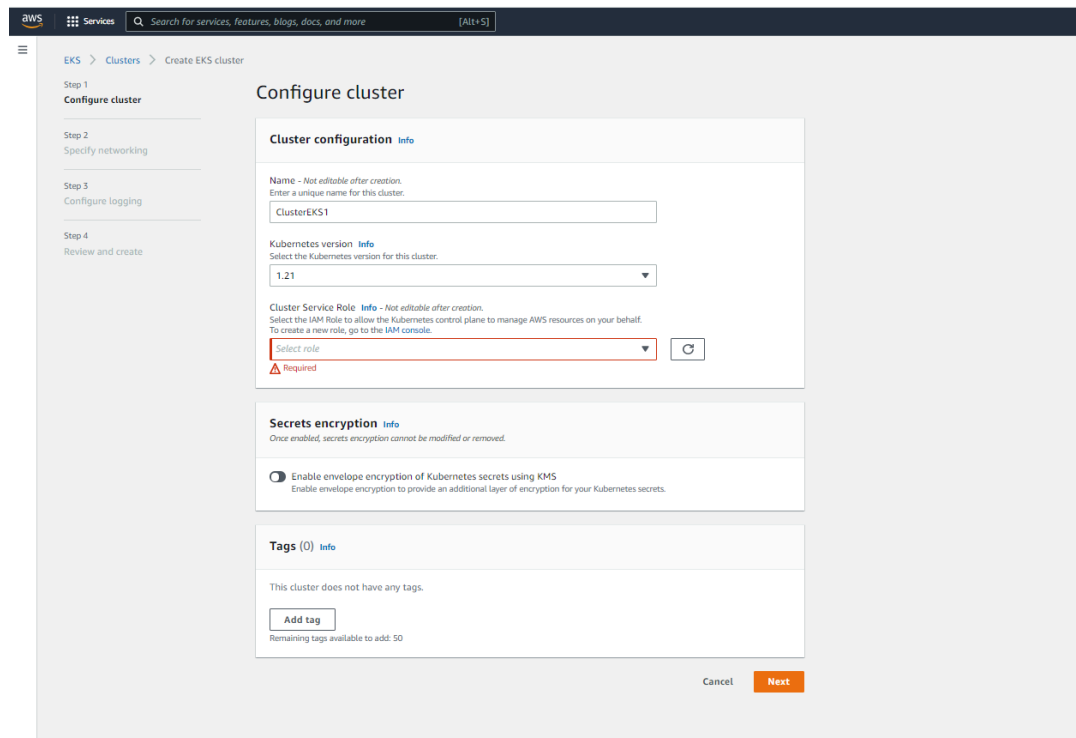
STEP 1 | Kubernetes クラスターをセットアップします。

AWS EKS でクラスターを作成するには、次の手順を実行します。

1. サービスナビゲーションメニューをクリックし、コンテナ->**Elastic Kubernetes Service**に移動します。



2. [クラスターの作成]をクリックします。
3. 必要な詳細を入力し、[作成] をクリックします。



1. クラスターのリソースが適切であることを確認します。クラスターにファイアウォールをサポートするための**CNシリーズの前提条件**のリソースが含まれていることを確認します。

kubectl get nodes

kubectl describe node <node-name>

コマンド出力の[容量]見出しの下にある情報を表示して、指定したノードで使用可能な CPU とメモリーを確認します。

CPU、メモリー、およびディスク ストレージの割り当てはニーズによって異なります。[CNシリーズのパフォーマンスとスケーリング](#)を参照してください。

以下の情報があることを確認してください。

- Panorama 上で API サーバーをセットアップするためのエンドポイント IP アドレスを収集します。Panorama は、この IP アドレスを使用して、Kubernetes クラスタに接続します。
- テンプレート スタック名、デバイス グループ名、Panorama IP アドレス、およびオプションで Panorama からログ コレクタ グループ名を収集します。
- [認証コード](#)と[自動登録の PIN ID と値](#)を収集します。
- イメージをダウンロードしたコンテナ イメージ リポジトリの場所。

STEP 2 | (任意) Panorama の Kubernetes プラグインでカスタム証明書を設定した場合は、次のコマンドを実行して証明書シークレットを作成する必要があります。ファイル名を `ca.crt` から変更しないでください。pan-cn-mgmt.yaml および pan-cn-ngfw.yaml のカスタム証明書のボリュームはオプションです。

```
kubectl -n kube-system create secret generic custom-ca --from-file = ca.crt
```

STEP 3 | YAML ファイルを編集して、CN-Series ファイアウォールをデプロイするために必要な詳細を記入します。

プライベート レジストリへのパスを含み、必要なパラメータを提供するように YAML ファイル内のイメージパスを置き換える必要があります。詳細は、「[CN-Series 配置の yaml ファイルで編集可能なパラメータ](#)」を参照してください。

STEP 4 | ストレージクラスを更新します。AWS Outpost にデプロイされた CN-Series をサポートするには、ストレージドライバー `aws-ebs-csi-driver` を使用する必要があります。これにより、動的永続ボリューム (PV) の作成中に Outpost が Outpost からボリュームをプルします。

1. 以下の yml を適用します。

```
kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/
deploy/kubernetes/overlays/stable/?ref=release-0.10"
```

2. `ebs-sc` コントローラーが実行されていることを確認します。

```
kubectl -n kube-system get pods
```

3. 以下の例に一致するように `pan-cn-storage-class.yml` を更新します。

```
apiVersion: v1 kind:StorageClass apiVersion: storage.k8s.io/
v1 metadata: name: ebs-sc provisioner: ebs.csi.aws.com
volumeBindingMode: WaitForFirstConsumer parameters: type: gp2
```

4. 以下に示す場所で、**storageClassName : ebs-sc** を `pan-cn-mgmt.yml` に追加します。

```
volumeClaimTemplates: - metadata: name: panlogs spec:
#storageClassName: pan-cn-storage-class //For better disk
iops performance for logging accessModes: [ "ReadWriteOnce" ]
storageClassName: ebs-sc // resources: requests: storage:20Gi
# ディスク IOPS を向上させるためにstorageClassName を使用
しているときにこれを 200Gi に変更します - metadata: name:
varlogpan spec: #storageClassName: pan-cn-storage-
class // dp ログのディスク IOPS パフォーマンスを向上させるため
accessModes: [ "ReadWriteOnce" ] storageClassName: ebs-
sc resources: requests: storage:20Gi # ディスク IOPS 向
上のために storageClassName を使用している間、これを 200Gi に
変更します - metadata: name: varcores spec: accessModes:
[ "ReadWriteOnce" ] storageClassName: ebs-sc resources:
requests: storage:2Gi - metadata: name: panplugincfg spec:
accessModes: [ "ReadWriteOnce" ] storageClassName: ebs-sc
resources: requests: storage:1Gi - metadata: name: panconfig
spec: accessModes: [ "ReadWriteOnce" ] storageClassName:
ebs-sc resources: requests: storage:8Gi - metadata:
name: panplugins spec: accessModes: [ "ReadWriteOnce" ]
storageClassName: ebs-sc resources: requests: storage:200Mi
```

STEP 5 | Kubernetes 環境でオートスケーリングを使用している場合は、次の手順を実行します。

1. CN-Series で、[Kubernetes 用 Amazon CloudWatch メトリクス アダプター](#)をサービスクラスターとしてデプロイします。CloudWatch に、Kubernetes ポッドとクラスターに関連付けられた両方の IAM ロールへの完全なアクセスを許可する必要があります。カスタムメトリクスを CloudWatch に公開するには、HPA がそれらを取得できるように、ワーカーノードのロールに AWS 管理ポリシー **CloudWatchAgentServerPolicy** が必要です。
2. [Palo Alto Networks GitHub リポジトリ](#)から EKS 固有の HPA yml ファイルをダウンロードします。

3. CN-MGMT がカスタム名前空間にデプロイされている場合は、カスタム名前空間を使用して `pan-cn-adapater.yaml` を更新します。デフォルトの名前空間は **kube-system** です。
4. **pan-cn-hpa-dp.yaml** と **pan-cn-hpa-mp.yaml** を変更します。
 1. レプリカの最小数と最大数を入力します。
 2. (任意) スケールダウンを変更し、デプロイメントに合わせて頻度値をスケールアップします。これらの値を変更しない場合は、デフォルト値が使用されます。
 3. スケーリングに使用する各メトリックについて、以下のセクションをコピーします。

```
- type:Pods pods: metric: name: pansessionactive target:
  type:AverageValue averageValue:30
```

4. 使用するメトリックの名前を変更し、**averageValue** を上記の表で説明したしきい値に設定します。これらの値を変更しない場合は、デフォルト値が使用されます。
5. 変更を保存します。

詳細については、Horizontal Pod Autoscalingを参照してください。
5. HPA yaml ファイルをデプロイします。ファイルは、以下に説明する順序でデプロイする必要があります。
 1. Kubectl を使用して `pan-cn-adapter.yaml` を実行します
`kubectl apply -f pan-cn-adapter.yaml`
 2. Kubectl を使用して `pan-cn-externalmetrics.yaml` を実行します
`kubectl apply -f pan-cn-externalmetrics.yaml`
 3. Kubectl を使用して `pan-cn-hpa-dp.yaml` を実行します
`kubectl apply -f pan-cn-hpa-dp.yaml`
 4. Kubectl を使用して `pan-cn-hpa-mp.yaml` を実行します
`kubectl apply -f pan-cn-hpa-mp.yaml`
6. デプロイメントを確認します。

kubectl を使用して、カスタム メトリックス名前空間内のカスタム メトリック アダプターポッドを確認します。

```
kubectl get pods -n custom-metrics
```

kubectl を使用して、HPA リソースを確認します。

```
kubectl get hpa -n kube-system
```

```
kubectl describe hpa <hpa-name> -n kube-system
```

STEP 6 | CN-NGFW サービスをデプロイします。

1. pan-cni-serviceaccount.yaml を使用してサービス アカウントが作成されたことを確認します。

クラスタ認証用のサービス アカウントの作成を参照してください。

2. Kubectl を使用して pan-cni-configmap.yaml を実行します。

```
kubectl apply -f pan-cni-configmap.yaml
```

3. kubectl を使用して pan-cn-ngfw-svc.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-svc.yaml
```



この yaml は pan-cni.yaml の前にデプロイする必要があります。

4. Kubectl を使用して pan-cni.yaml を実行します。

```
kubectl apply -f pan-cni.yaml
```

5. pan-cni-configmap YAML ファイルと pan-cni YAML ファイルが変更されたことを確認します。

6. 以下のコマンドを実行して、出力が以下の例のようになっていることを確認します。

```
kubectl get pods -n kube-system | grep pan-cni
```

```
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v-series-mktplace) $ kubectl get pods -n kube-system | grep pan-cni
pan-cni-nmqkf          Running    0          2m11s
pan-cni-wjrkq          Running    0          2m11s
pan-cni-xrc2z          Running    0          2m12s
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v-series-mktplace) $
```

STEP 7 | CN-MGMT StatefulSet をデプロイします。

デフォルトで、管理プレーンは耐障害性を提供する StatefulSet としてデプロイされます。1 つの CN-MGMT StatefulSet に最大 30 個のファイアウォール CN-NGFW ポッドを接続できます。

1. (静的にプロビジョニングされた PV のみに必要) CN-MGMT StatefulSet 用の永続ボリューム (PV) をデプロイします。

1. pan-cn-pv-local.yaml で定義されたローカル ボリューム名と一致するディレクトリを作成します。

少なくとも 2 つのワーカー ノード上に 6 つのディレクトリが必要です。CN-MGMT StatefulSet をデプロイする各ワーカー ノードにログインして、ディレクトリを作成

します。たとえば、/mnt/pan-local1 から /mnt/pan-local6 という名前のディレクトリを作成するには、次のコマンドを使用します。

```
mkdir -p /mnt/pan-local1 /mnt/pan-local2 /mnt/pan-local3 /
mnt/pan-local4 /mnt/pan-local5 /mnt/pan-local6
```

2. pan-cn-pv-local.yaml を変更します。

nodeaffinity の下でホスト名を一致させ、上記で spec.local.path に作成したディレクトリが変更されたことを確認してから、そのファイルをデプロイして、新しいストレージクラス pan-local-storage とローカル PV を作成します。

2. pan-cn-mgmt-configmap YAML ファイルと pan-cn-mgmt YAML ファイルが変更されたことを確認します。

EKS から pan-cn-mgmt-configmap をサンプリングします。

```
apiVersion: v1 kind:ConfigMap metadata: name: pan-mgmt-
config namespace: kube-system data: PAN_SERVICE_NAME: pan-
mgmt-svc PAN_MGMT_SECRET: pan-mgmt-secret # Panorama 設
定 PAN_PANORAMA_IP: "<panorama-IP>" PAN_DEVICE_GROUP:
"<panorama-device-group>" PAN_TEMPLATE_STACK: "<panorama-
template-stack>" PAN_CGNAME: "<panorama-collector-
group>" # ctnr mode: "k8s-service", "k8s-ilb-service"
PAN_CTNR_MODE_TYPE: "k8s-service" #必須でないパラメータ #
Panorama Kubernetes プラグインで提供されるクラスタ名と同じ名前
を持つことを推奨 - 同じ Panorama で複数のクラスタを管理する場合、
ポッドの識別が容易になります #CLUSTER_NAME: "<Cluster name>"
#PAN_PANORAMA_IP2: "" # CERT を使用する場合はコメントアウト
します。それ以外の場合は、pan-mgmt と pan-ngfw 間の IPsec 用に
PSK を使用します #IPSEC_CERT_BYPASS: "" # 値は不要 # jumbo-
frame モードの自動検出をオーバーライドし、システム全体を強制的に有
効にします #PAN_JUMBO_FRAME_ENABLED: "true" # GTP を有効にし
て MGMT を起動します。完全な機能を実現するには、Panorama でも GTP
# を有効にする必要があります。#PAN_GTP_ENABLED: 「true」# 高い
機能容量を有効にします。これらは MGMT ポッドには高いメモリを必要と
し、NGFW ポッドには以下の指定以上のメモリが必要です。#PAN_NGFW_MEMORY =
"6Gi" #PAN_NGFW_MEMORY = "40Gi" # より高速なデータパス-AF_XDP を有
効にするには、デフォルトは AF_PACKETV2 です。これにはカーネルのサポート
が必要です。#PAN_DATA_MODE: "次世代" #HPA params #PAN_CLOUD: "EKS"
#PAN_NAMESPACE_EKS: "EKSNamespace" #PUSH_INTERVAL: "15" #AWS
cloudwatch にメトリクスを発行する間隔
```

pan-cn-mgmt.yaml のサンプル

```
initContainers: - name: pan-mgmt-init image: <your-private-
registry-image-path>
```

```
containers: - name: pan-mgmt image: <your-private-registry-
image-path> terminationMessagePolicy:FallbackToLogsOnError
```

3. Kubectl を使用して yaml ファイルを実行します。

```
kubectl apply -f pan-cn-mgmt-configmap.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-crd.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-cr.yaml
```

```
kubectl apply -f pan-cn-mgmt-secret.yaml
```

```
kubectl apply -f pan-cn-mgmt.yaml
```

pan-mgmt-serviceaccount.yaml は、[クラスター認証用のサービスアカウントの作成](#)を以前に完了していない場合にのみ実行する必要があります。

4. CN-MGMT ポッドが起動していることを確認します。

これには、5 ～ 6 分かかります。

```
kubectl get pods -l app=pan-mgmt -n kube-system
```

 を使用します。

STEP 8 | CN-NGFW ポッドをデプロイします。

1. PAN-CN-NGFW-CONFIGMAP と PAN-CN-NGFW に詳述されているように YAML ファイルが変更されたことを確認します。

```
containers: - name: pan-ngfw-container image: <your-private-registry-image-path>
```

2. Kubectl apply を使用して pan-cn-ngfw-configmap.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-configmap.yaml
```

3. Kubectl apply を使用して pan-cn-ngfw.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw.yaml
```

4. CN-NGFW ポッドがデプロイされたことを確認します。

```
kubectl get pods -n kube-system -l app=pan-ngfw -o wide
```

STEP 9 | CN-Series で水平ポッド自動スケーリングを有効にします。

STEP 10 | Kubernetes クラスタ上の CN-MGMT、CN-NGFW、および PAN-CNI が表示されていることを確認します。

```
kubectl -n kube-system get pods
```

STEP 11 | 新しいポッドからのトラフィックがファイアウォールにリダイレクトされるようにアプリケーション yml または名前空間に注釈を付けます。

検査のためにトラフィックを CN-NGFW にリダイレクトするには、以下のアノテーションを追加する必要があります：

```
annotations: paloaltonetworks.com/firewall: pan-fw
```

たとえば、「default」名前空間のすべての新しいポッドの場合：

```
kubectl annotate namespace default paloaltonetworks.com/  
firewall=pan-fw
```



一部のプラットフォームでは、CNI プラグイン チェーン内で *pan-cni* がアクティブになっていない状態でアプリケーション ポッドが開始する可能性があります。このようなシナリオを回避するには、アプリケーションポッド YAML にここに示すようにボリュームを指定する必要があります。

```
volumes: - name: pan-cni-ready hostPath: path: /var/log/  
pan-appinfo/pan-cni-ready type: ディレクトリ
```

STEP 12 | クラスタでアプリケーションをデプロイします。

CN-Series ファイアウォールを AWS EKS に DaemonSet としてデプロイする

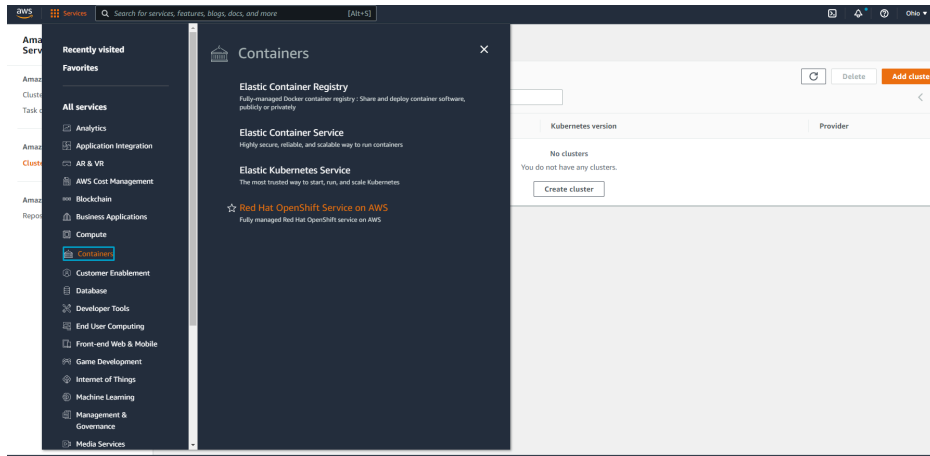
どこで使用できますか？	何が必要か？
<ul style="list-style-type: none">• CN-Seriesデプロイメント	<ul style="list-style-type: none">• CN-Series 10.1.x or above Container Images• PanoramaHelmを使用してCN-SeriesをデプロイするためのPAN-OS 10.1.xまたはそれ以上のバージョンを実行している場合• Helm 3.6 or above version client 以下の手順を完了し、

CN-SeriesファイアウォールをAWS EKS上のdameonsetとしてデプロイします。

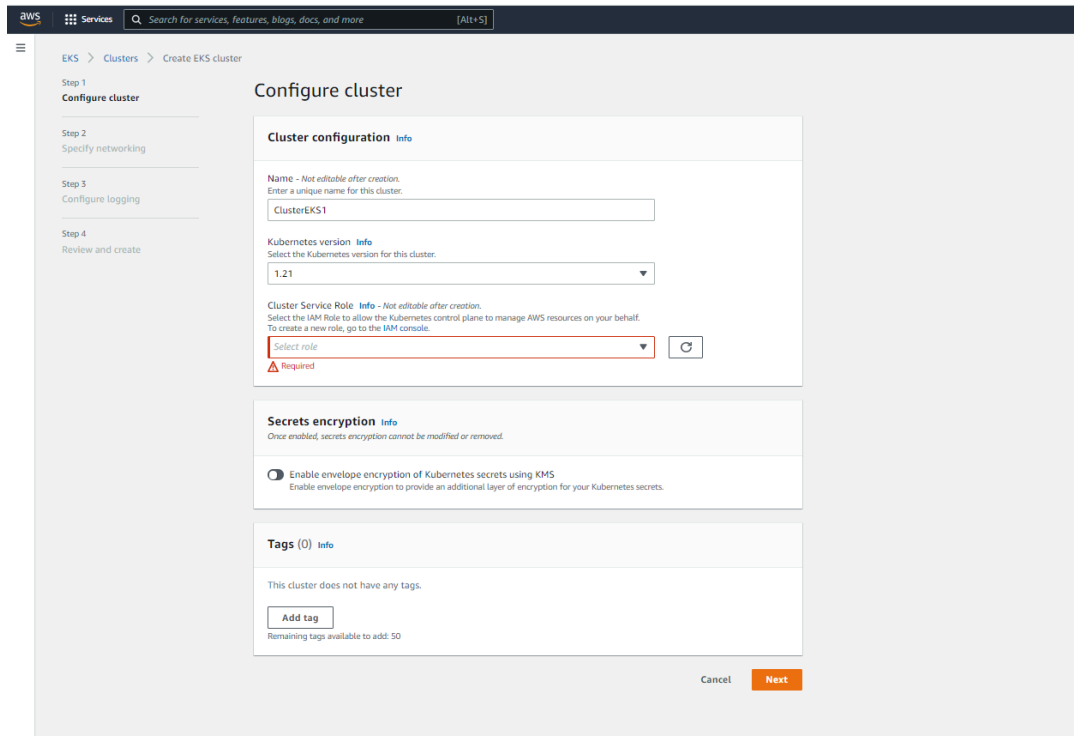
STEP 1 | Kubernetes クラスターをセットアップします。

AWS EKS でクラスターを作成するには、次の手順を実行します。

1. サービスナビゲーションメニューをクリックし、コンテナ->**Elastic Kubernetes Service**に移動します。



2. [クラスターの作成]をクリックします。
3. 必要な詳細を入力し、[作成] をクリックします。



クラスタのリソースが適切であることを確認します。クラスタにファイアウォールをサポートするための**CNシリーズの前提条件**のリソースが含まれていることを確認します。

```
kubectl get nodes
```

```
kubectl describe node <node-name>
```

コマンド出力の[容量]見出しの下にある情報を表示して、指定したノードで使用可能な CPU とメモリーを確認します。

CPU、メモリー、およびディスク ストレージの割り当てはニーズによって異なります。[CN シリーズのパフォーマンスとスケーリング](#)を参照してください。

以下の情報があることを確認してください。

- Panorama 上で API サーバーをセットアップするためのエンドポイント IP アドレスを収集します。

Cluster Definition

Name: on_prem-clstr

Description:

API server address: 10.2...

Type: Native-Kubernetes

Credentials

Label Selector | Label Filter | Custom Certificate

0 items

TAG PREFIX	NAMESPACE	LABEL SELECTOR FILTER	APPLY ON
------------	-----------	-----------------------	----------

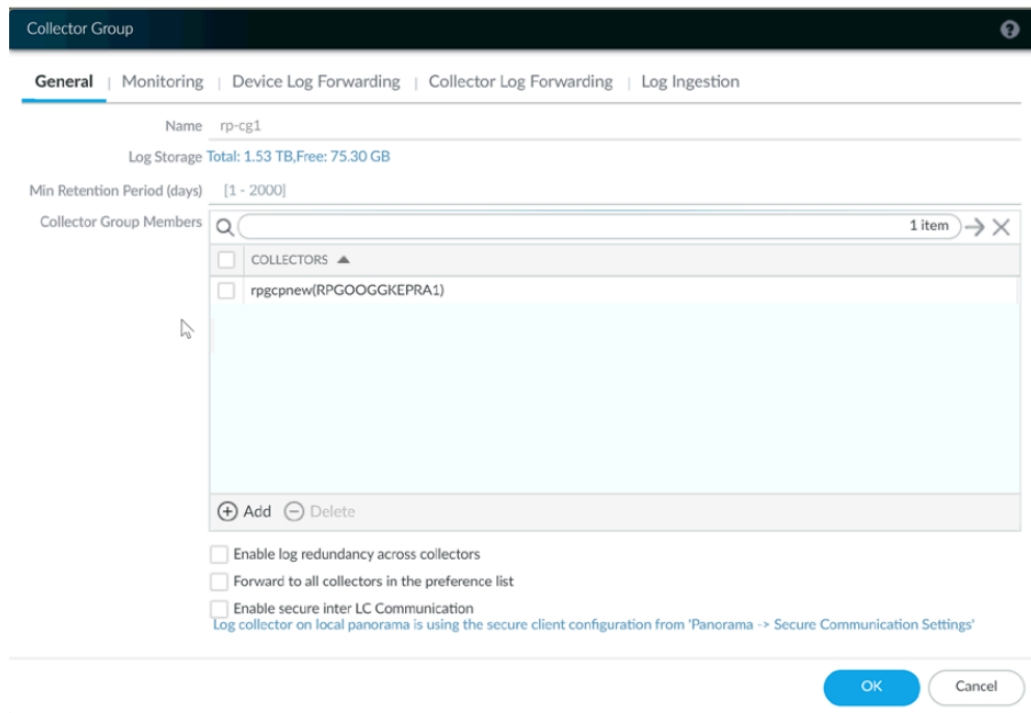
+ Add - Delete

Validate OK Cancel

Panorama は、この IP アドレスを使用して、Kubernetes クラスタに接続します。

詳細については、[クラスタをモニタリングするための Kubernetes プラグインの設定](#)をご覧ください。

- テンプレート スタック名、デバイス グループ名、Panorama IP アドレス、およびオプションで Panorama からログ コレクタ グループ名を収集します。



詳細については、[親 デバイス グループとテンプレート スタックの作成](#)を参照してください。

- [認証コードと自動登録の PIN ID と値](#)を収集します。
- イメージをダウンロードしたコンテナ イメージ リポジトリの場所。

STEP 2 | (任意) Panorama の Kubernetes プラグインでカスタム証明書を設定した場合は、次のコマンドを実行して証明書シークレットを作成する必要があります。ファイル名を `ca.crt` から変更しないでください。 `pan-cn-mgmt.yaml` および `pan-cn-ngfw.yaml` のカスタム証明書のボリュームはオプションです。

```
kubectl -n kube-system create secret generic custom-ca --from-file = ca.crt
```

STEP 3 | YAML ファイルを編集して、CN-Series ファイアウォールをデプロイするために必要な詳細を記入します。

非公開の Google Container レジストリへのパスを含め、必要なパラメータを提供するには、YAML ファイル内のイメージパスを置き換える必要があります。詳細は、「[CN-Series 配置の yaml ファイルで編集可能なパラメータ](#)」を参照してください。

STEP 4 | CNI DaemonSet をデプロイします。

CNI コンテナは、DaemonSet (ノードあたり 1 つのポッド) としてデプロイされ、ノード上にデプロイされたアプリケーションごとに 2 つずつのインターフェースを CN-NGFW ポッド上

に作成します。kubectl コマンドを使用して pan-cni YAML ファイルを実行すると、それが各ノード上のサービス チェーンに組み込まれます。

1. CN-Series ファイアウォールには、Kubernetes クラスター リソースと通信することを許可する最小権限を備えた 3 つのサービス アカウントが必要です。 [Create Service Accounts for Cluster Authentication \(クラスター認証用サービスアカウントの作成\)](#) を作成し、pan-cni-serviceaccount.yaml を使用してサービスアカウントを作成したことを確認します。
2. Kubectl を使用して pan-cni-configmap.yaml を実行します。

```
kubectl apply -f pan-cni-configmap.yaml
```

3. Kubectl を使用して pan-cni.yaml を実行します。

```
kubectl apply -f pan-cni.yaml
```

4. pan-cni-configmap YAML ファイルと pan-cni YAML ファイルが変更されたことを確認します。
5. 以下のコマンドを実行して、出力が以下の例のようになっていることを確認します。

```
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v-series-mktplace) $ kubectl get pods -n kube-system | grep pan-cni
pan-cni-nmqkf          Running    0          2m11s
pan-cni-wjtkq          Running    0          2m11s
pan-cni-xrc2z          Running    0          2m12s
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v-series-mktplace) $
```

STEP 5 | ストレージクラスを更新します。AWS Outpost にデプロイされた CN-Series をサポートするには、ストレージドライバ aws-ebs-csi-driver を使用する必要があります。これにより、動的永続ボリューム (PV) の作成中に Outpost が Outpost からボリュームをプルします。

1. 以下の yaml を適用します。

```
kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-0.10"
```

2. ebs-sc コントローラーが実行されていることを確認します。

```
kubectl -n kube-system get pods
```

3. 以下の例に一致するように pan-cn-storage-class.yaml を更新します。

```
apiVersion: v1 kind:StorageClass apiVersion: storage.k8s.io/v1
metadata: name: ebs-sc provisioner: ebs.csi.aws.com
volumeBindingMode: WaitForFirstConsumer parameters: type: gp2
```

4. 以下に示す場所で、**storageClassName : ebs-sc** を pan-cn-mgmt.yaml に追加します。

```
volumeClaimTemplates: - metadata: name: panlogs spec:
  #storageClassName: pan-cn-storage-class //For better disk
  iops performance for logging accessModes: [ "ReadWriteOnce" ]
  storageClassName: ebs-sc // resources: requests: storage:20Gi
  # ディスク IOPS を向上させるためにstorageClassName を使用
  しているときにこれを 200Gi に変更します - metadata: name:
  varlogpan spec: #storageClassName: pan-cn-storage-
  class // dp ログのディスク IOPS パフォーマンスを向上させるため
```



```
accessModes: ["ReadWriteOnce"] storageClassName: ebs-sc resources: requests: storage:20Gi # ディスク IOPS 向上のために storageClassName を使用している間、これを 200Gi に変更します - metadata: name: varcores spec: accessModes: ["ReadWriteOnce"] storageClassName: ebs-sc resources: requests: storage:2Gi - metadata: name: panplugincfg spec: accessModes: ["ReadWriteOnce"] storageClassName: ebs-sc resources: requests: storage:1Gi - metadata: name: panconfig spec: accessModes: ["ReadWriteOnce"] storageClassName: ebs-sc resources: requests: storage:8Gi - metadata: name: panplugins spec: accessModes: ["ReadWriteOnce"] storageClassName: ebs-sc resources: requests: storage:200Mi
```

STEP 6 | CN-MGMT StatefulSet をデプロイします。

デフォルトで、管理プレーンは耐障害性を提供する StatefulSet としてデプロイされます。1 つの CN-MGMT StatefulSet に最大 30 個のファイアウォール CN-NGFW ポッドを接続できます。

1. (静的にプロビジョニングされた PV のみに必要) CN-MGMT StatefulSet 用の永続ボリューム (PV) をデプロイします。

1. pan-cn-pv-local.yaml で定義されたローカル ボリューム名と一致するディレクトリを作成します。

少なくとも 2 つのワーカー ノード上に 6 つのディレクトリが必要です。CN-MGMT StatefulSet をデプロイする各ワーカー ノードにログインして、ディレクトリを作成します。たとえば、/mnt/pan-local1 から /mnt/pan-local6 という名前のディレクトリを作成するには、次のコマンドを使用します。

```
mkdir -p /mnt/pan-local1 /mnt/pan-local2 /mnt/pan-local3 /mnt/pan-local4 /mnt/pan-local5 /mnt/pan-local6
```

2. pan-cn-pv-local.yaml を変更します。

nodeaffinity の下でホスト名を一致させ、上記で spec.local.path に作成したディレクトリが変更されたことを確認してから、そのファイルをデプロイして、新しいストレージクラス pan-local-storage とローカル PV を作成します。

2. pan-cn-mgmt-configmap YAML ファイルと pan-cn-mgmt YAML ファイルが変更されたことを確認します。

EKS から pan-cn-mgmt-configmap をサンプリングします。

```
復元されたセッション コンテンツ apiVersion: v1 kind: ConfigMap
metadata: name: pan-mgmt-config namespace: kube-system
data: PAN_SERVICE_NAME: pan-mgmt-svc PAN_MGMT_SECRET: pan-mgmt-secret # Panorama 設定 PAN_PANORAMA_IP: "x.y.z.a"
PAN_DEVICE_GROUP: "dg-1" PAN_TEMPLATE_STACK: "temp-stack-1"
PAN_CGNAME: "CG-EKS" # 意図されたライセンス バンドル タイプ - "CN-X-BASIC", "CN-X-BND1", "CN-X-BND2" # Panorama K8S プラグインに適用された認証コードに基づく " PAN_BUNDLE_TYPE: "CN-X-BND2" # 必須ではな
```

いパラメーター # Panorama Kubernetesプラグインで提供されるクラスター名と同じ名前にすることをお勧めします-同じ Panorama #CLUSTER_NAME で複数のクラスターを管理する場合、ポッドを簡単に識別できます。"Cluster-name"
 #PAN_PANORAMA_IP2: "passive-secondary-ip" # CERTを使用する場合はコメントアウトします。それ以外の場合は、pan-mgmt の etcd への暗号化された接続をバイパスします。# EKS バグのために etcd に CERT を使用しない
 ETCD_CERT_BYPASS: "" # 値は必要ありません # CERT を使用するようにコメントアウトしてください。それ以外の場合は、pan-mgmtとpan-ngfw 間のIPSec に PSK を使用します。 # IPSEC_CERT_BYPASS: "" # 値は必要ありません

pan-cn-mgmt.yaml のサンプル

```
initContainers: - name: pan-mgmt-init image: <your-private-registry-image-path>
```

```
containers: - name: pan-mgmt image: <your-private-registry-image-path> terminationMessagePolicy:FallbackToLogsOnError
```

3. Kubectl を使用して yaml ファイルを実行します。

```
kubectl apply -f pan-cn-mgmt-configmap.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-crd.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-cr.yaml
```

```
kubectl apply -f pan-cn-mgmt-secret.yaml
```

```
kubectl apply -f pan-cn-mgmt.yaml
```

pan-mgmt-serviceaccount.yamlは、[CN-Seriesファイアウォールを使用したクラスター認証用のサービスアカウントの作成](#)を以前に完了していない場合にのみ実行する必要があります。

4. CN-MGMT ポッドが起動していることを確認します。

これには、5～6分かかります。

kubectl get pods -l app=pan-mgmt -n kube-system を使用します。

```
NAME READY STATUS RESTARTS AGEpan-mgmt-sts-0 1/1 Running 0
27hpan-mgmt-sts-1 1/1 Running 0 27h
```

STEP 7 | CN-NGFW ポッドをデプロイします。

デフォルトで、ファイアウォール データプレーン CN-NGFW ポッドは DaemonSet としてデプロイされます。CN-NGFW ポッドのインスタンスは、1 つのノード上で最大 30 個のアプリケーション ポッドのトラフィックを保護することができます。

1. PAN-CN-NGFW-CONFIGMAP と PAN-CN-NGFW に詳述されているように YAML ファイルが変更されたことを確認します。

```
containers: - name: pan-ngfw-container image: <your-private-registry-image-path>
```

2. Kubectl apply を使用して pan-cn-ngfw-configmap.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-configmap.yaml
```

3. Kubectl apply を使用して pan-cn-ngfw.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw.yaml
```

4. すべての CN-NGFW ポッド (クラスタ内のノードあたり 1 つずつ) が実行していることを確認します。

これは、4 ノード オンプレミス クラスタからの出力例です。

```
kubectl get pods -n kube-system -l app=pan-ngfw -o wide
```

```
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
```

```
pan-ngfw-ds-8g5xb 1/1 Running 0 27h 10.233.71.113 rk-k8-node-1 <none> <none>
```

```
pan-ngfw-ds-qsr6 1/1 Running 0 27h 10.233.115.189 rk-k8-vm-worker-1 <none> <none>
```

```
pan-ngfw-ds-vqk7z 1/1 Running 0 27h 10.233.118.208 rk-k8-vm-worker-3 <none> <none>
```

```
pan-ngfw-ds-zncqg 1/1 Running 0 27h 10.233.91.210 rk-k8-vm-worker-2 <none> <none>
```

STEP 8 | Kubernetes クラスタ上の CN-MGMT、CN-NGFW、および PAN-CNI が表示されていることを確認します。

```
kubectl -n kube-system get pods
```

```
0 27hpan-cni-5fhbg 1/1 Running
0 27hpan-cni-9j4rs 1/1 Running
0 27hpan-cni-ddwb4 1/1 Running
0 27hpan-cni-fwfrk 1/1 Running
0 27hpan-cni-h57lm 1/1 Running
0 27hpan-cni-h57lm 1/1 Running
0 27hpan-cni-j62rk 1/1 Running
0 27hpan-cni-lmxdz 1/1 Running
0 27hpan-mgmt-sts-0 1/1 Running
0 27hpan-mgmt-sts-1 1/1 Running
0 27hpan-ngfw-ds-8g5xb 1/1 Running
27hpan-ngfw-ds-qsr6 1/1 Running
0 27hpan-ngfw-ds-vqk7z 1/1 Running
0 27hpan-ngfw-ds-zncqg 1/1 Running
```

STEP 9 | 新しいポッドからのトラフィックがファイアウォールにリダイレクトされるようにアプリケーション yaml または名前空間に注釈を付けます。

検査のためにトラフィックを CN-NGFW にリダイレクトするには、以下のアノテーションを追加する必要があります：

```
annotations: paloaltonetworks.com/firewall: pan-fw
```

たとえば、「default」名前空間のすべての新しいポッドの場合：

```
kubectl annotate namespace default paloaltonetworks.com/
firewall=pan-fw
```



一部のプラットフォームでは、CNI プラグイン チェーン内で *pan-cni* がアクティブになっていない状態でアプリケーション ポッドが開始する可能性があります。このようなシナリオを回避するには、アプリケーションポッド YAML にここに示すようにボリュームを指定する必要があります。

```
volumes: - name: pan-cni-ready hostPath: path: /var/log/
pan-appinfo/pan-cni-ready type: ディレクトリ
```

STEP 10 | クラスタでアプリケーションをデプロイします。


AWS Marketplace から CN-Seriesをデプロイする

どこで使用できますか?	何が必要ですか?
<ul style="list-style-type: none">• CN-Seriesデプロイメント	<ul style="list-style-type: none">• CN-Series 10.1.x or above Container Images• PanoramaPAN-OS 10.1.x以降のバージョンを実行している• Helm 3.6 or above version clientHelmを使用したCNシリーズのデプロイメント用

CN-Series ファイアウォールは、[AWS Marketplace](#)を通じて AWS EKS にデプロイされた Kubernetes サービスとしてライセンスを取得できます。CNシリーズは、1か月、1年、2年、または3年のライセンスを取得し、EKS 1.19以降または Redhat Openshift 4.7 以降にデプロイできます。


 この製品はプレビュー版です。

このライセンスを使用するには、Kubernetes ワーカーノードに添付されている IAM ポリシーを更新する必要があります。

 CN-Series のデプロイにAWS Marketplaceから購入した PAYG ライセンスを使用している場合は、KubernetesのPanorama プラグインに認証コードを追加しないでください。


STEP 1 | 次の前提条件を完了します。

1. EKS または Redhat OpenShift クラスターを作成します。
2. Panorama をデプロイし、Kubernetes プラグインをインストールします。

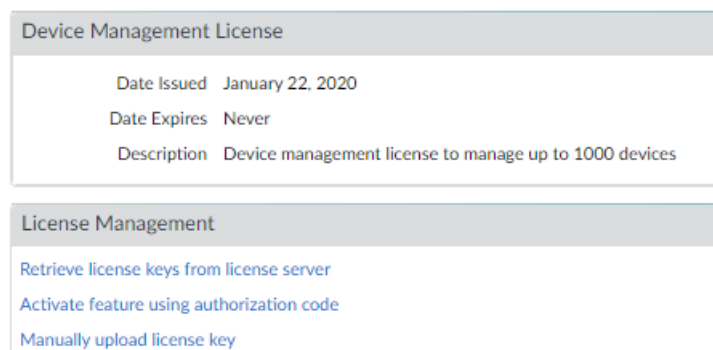
 ライセンスされた Panorama インスタンスがすでに AWS にデプロイされている場合は、この手順をスキップしてください。

1. Amazon EC2インスタンスに[Panoramaをインストール](#)します。
2. [CNシリーズ用Kubernetesプラグインをインストール](#)します。
3. Panorama がインストールされたら、CN-Series チーム（cn-series-aws-marketplace@paloaltonetworks.com）に電子メールを送信して、Panoramaのライセンスをリクエストしてください。フルネーム、会社の電子メール、会社名、発注書番号、AWS アカウント名、および AWS アカウント ID を含めてください。

STEP 2 | シリアル番号とライセンスを Panorama に適用します。

1. Panorama Web インターフェイスにログインします。
2. **[Panorama] > [Setup (セットアップ)] > [Management (管理)]**を選択し、編集  アイコンをクリックします。
3. Panorama の **Serial Number** (シリアル番号) (受注完了電子メールに記載) を入力し、**OK** をクリックします。
4. **Panorama > ライセンス**を選択します。
5. **Activate feature using authorization code** (認証コードを使用した機能のアクティベーション) をクリックします。
6. ファイアウォール管理ライセンス認証コードを入力し、**OK**をクリックしてライセンスをアクティブ化します。
7. ファイアウォール管理ライセンスがアクティブ化されていることを確認します。

デバイス管理ライセンスセクションが表示され、ライセンスの発行日、ライセンスの有効期限、およびファイアウォール管理ライセンスの説明が表示されます。

**STEP 3 |** IAM ポリシーを更新し、そのポリシーを Kubernetes ワーカーノードに添付します。

1. AWS 管理コンソールにログインして IAM コンソールを開きます。
1. **Policies** (ポリシー)を選択します。
2. ポリシーリストから、**AWSLicenseManagerConsumptionPolicy**および**AWSMarketplaceMeteringRegisterUsage**を選択します。
3. アクションを選択してから、添付を選択します。
4. ポリシーを添付するワーカーノード ID を選択します。ID を選択したら、ポリシーの添付をクリックします。

STEP 4 | `plugin-serviceaccount.yaml`をダウンロードし、Helm チャートをデプロイする前にyaml を適用します。

```
kubectl apply -f plugin-serviceaccount.yaml
```

STEP 5 | [AWS Marketplace](#) にアクセスし、**AWS Marketplace**の[リスト](#)からCN-Series を見つけます。

STEP 6 | Continue to Subscribe（続行して登録）をクリックします。

STEP 7 | 購入するライセンスの数を入力します。各ライセンス資格は、CN-Series のデプロイメントで使用される1つの vCPU と同等です。

デプロイメントのニーズを満たすために必要な vCPU の数については、[CN-Series のシステム要件](#)および[CN-Series のパフォーマンスとスケーリング](#)を参照してください。

STEP 8 | 設定に進むをクリックします。これにより、AWS アカウントにライセンスが追加されます。

1. フルフィルメントオプションとして**Helm**チャートを選択します。
2. ソフトウェアバージョンの最新バージョンを選択します。

[< Product Detail](#) [Subscribe](#) [Configure](#)

Configure this software

Choose a fulfillment option and software version to launch this software.

Fulfillment option	Supported services
<div>Helm Chart ▼</div>	<ul style="list-style-type: none">• Amazon EKS• Amazon EKS Anywhere• Self-managed Kubernetes
Software version	Fulfillment option description
<div>Version1.2.2 (Nov 22, 2021) ▼</div>	Deploy CN-Series on EKS and RedHat Openshift using Helm Chart

STEP 9 | [Continue to Launch (続行して起動)] をクリックします。

1. 起動ターゲット (Amazon が管理する **Kubernetes** または自己が管理する **Kubernetes**) を選択します。自己管理モードは、Redhat OpenShift にデプロイされます。
2. AWS Marketplace のリストに表示される起動手順に従ってください。この手順は、起動ターゲットによって異なります。

- **Amazon が管理する Kubernetes**

1. 起動手順のステップ 1 からコマンドをコピーします。
2. コピーしたコマンドを更新して、クラスター名を追加します。

--cluster<ENTER_YOUR_CLUSTER_NAME_HERE>

3. コピーしたコマンドをEKSクラスターで実行します。

Step 1: Create an AWS IAM role and Kubernetes service account

Use the following command to create an AWS IAM role and Kubernetes service account.

```
kubectl create namespace kube-system  
  
eksctl create iamserviceaccount \  
  --name my-service-account \  
  --namespace kube-system \  
  --cluster <ENTER_YOUR_CLUSTER_NAME_HERE>
```

Copy

4. 起動手順のステップ 2 から Helm チャートコマンドをコピーします。
5. Helm のインストール情報を更新して、Panorama IP、Panorama 認証キー、デバイス グループ名、テンプレート スタック名、および収集グループ名を含めます。**cluster.deployTo**を**eks**に設定します。

```
helm install cn-series-helm \ --namespace kube-system ./awsmp-chart/* \ --set serviceAccount.create=false  
  \ --set serviceAccount.name=my-service-account \ --set cluster.deployTo=eks \ --set  
  panorama.ip=Panorama-IP \ --set panorama.ip2=Panorama-IP2 \ --set panorama.authKey=000xxxxxxx  
  \ --set panorama.deviceGroup=Panorama-DG  
  \ --set panorama.template=Panorama-TS \
```

```
--set panorama.cgName=Panorama-CG \ --set
imagePullSecrets=awsmp-image-pull-secret
```

Step 2: Launch the software

Use the following commands to launch this software by installing a Helm chart on your Amazon EKS cluster.

```
export HELM_EXPERIMENTAL_OCI=1

aws ecr get-login-password \
  --region us-east-1 | helm registry login \
  --username AWS \
```

[Copy](#)

6. 上記の値を更新した後、EKS クラスターにて helm install コマンドを実行します。
- セルフマネージド型 **Kubernetes**
 1. 起動手順のステップ 1 を完了して、ライセンストークンと IAM ロールを作成します。

Step 1: Create a license token and IAM role

Choose **Create token** to generate a license token and AWS IAM role. These will be used to access the AWS License Manager APIs for billing and metering. You can use an existing token if you have one.

[Create token](#)

2. 起動手順のステップ 2 からコマンドをコピーします。
3. コピーしたコマンドを更新して、トークン値を追加します。

AWSMP_TOKEN =<CREATE_TOKEN_ABOVE>

4. コピーしたコマンドを OpenShift クラスターで実行します。

Step 2: Save the token and IAM role as a Kubernetes secret

Use the following commands to save the license token and IAM role as a secret in the cluster. The secret will be used in a following step when launching the software.

```
kubectl create namespace kube-system
kubectl create serviceaccount my-service-account --namespace kube-system

AWSMP_TOKEN=<CREATE_TOKEN_ABOVE>
AWSMP_ROLE_ARN=arn:aws:iam::018147215560:role/service-role/AWSMarketplaceLicenseT
```

[Copy](#)

5. 起動手順のステップ 3 から Helm チャートコマンドをコピーします。
6. Helm のインストール情報を更新して、Panorama IP、Panorama 認証キー、デバイス グループ名、テンプレート スタック名、および収集グループ名を含めます。**cluster.deployTo**を**openshift**に設定します。

```
helm install cn-series-helm \ --namespace kube-system ./
awsmp-chart/* \ --set serviceAccount.create=false
```

```
\ --set serviceAccount.name=my-service-account
\ --set cluster.deployTo=eks|openshift \ --set
panorama.ip=Panorama-IP \ --set panorama.ip2=Panorama-
IP2 \ --set panorama.authKey=000xxxxxxx
\ --set panorama.deviceGroup=Panorama-DG
\ --set panorama.template=Panorama-TS \
--set panorama.cgName=Panorama-CG \ --set
imagePullSecrets=awsmp-image-pull-secret
```

Step 3: Launch the software

Use the following commands to launch the software by installing a Helm chart from Amazon Elastic Container Registry (ECR).

```
export HELM_EXPERIMENTAL_OCI=1

aws ecr get-login-password \
  --region us-east-1 | helm registry login \
  --username AWS \
```

[Copy](#)

7. 上記の値を更新した後、OpenShift クラスターにてhelm installコマンドを実行します。

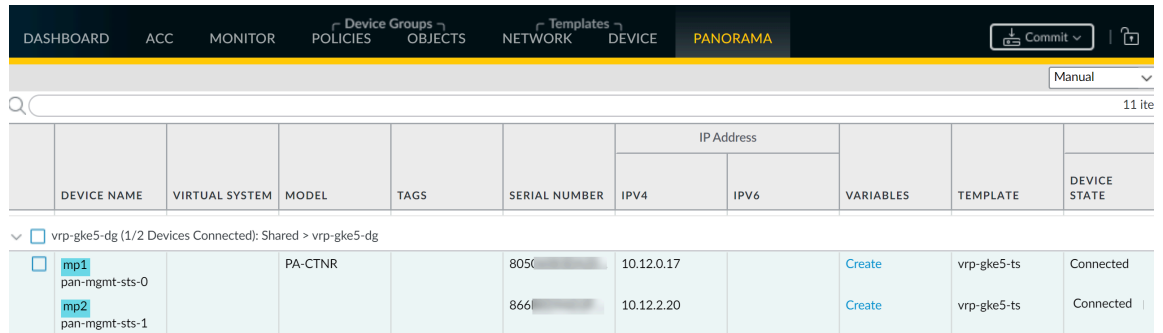
STEP 10 | ライセンスがアカウントに正常に追加されたことを確認します。

1. AWS License Manager に移動します。
2. **Granted Licenses** (付与されたライセンス)を選択し、CN-Series for AWS Marketplace のリストを見つけます。
3. **Entitlements** (資格)にて、ライセンスの総数と消費されたライセンスの数を確認できます。

Entitlements							
An entitlement is a right to use, access, or consume an application or resource.							
<input type="text" value="Search"/> < 1 > ⌵							
Name	Value	Max count	Usage	Units	Overages	Allow check in	
vCPU	-	1000	5	Count	Not Allowed	Allowed	
AWS::Marketplace::Usage	Enabled	-	-	None	-	Not Allowed	

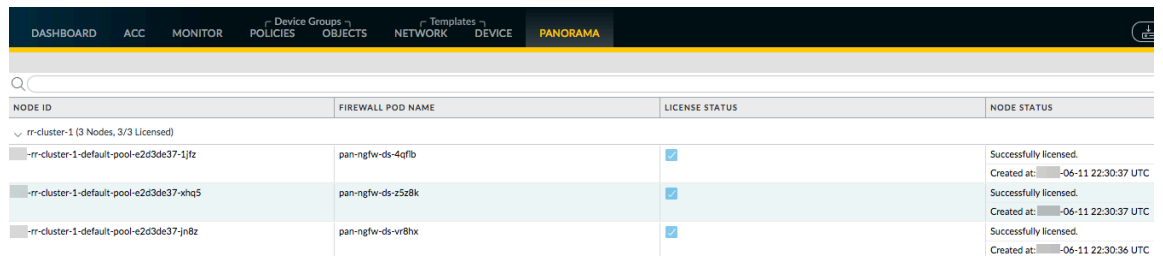
STEP 11 | CN-Series ファイアウォールが Panorama に表示されることを確認します。

1. Panorama にログインします。
2. CN-MGMT ポッドを表示するには、**Panorama** > 管理対象デバイス > サマリーを選択します。



PANORAMA										
Manual										
11 items										
	DEVICE NAME	VIRTUAL SYSTEM	MODEL	TAGS	SERIAL NUMBER	IP Address		VARIABLES	TEMPLATE	DEVICE STATE
						IPV4	IPV6			
vrp-gke5-dg (1/2 Devices Connected): Shared > vrp-gke5-dg										
<input type="checkbox"/>	mp1 pan-mgmt-sts-0		PA-CTNR		805C	10.12.0.17		Create	vrp-gke5-ts	Connected
<input type="checkbox"/>	mp2 pan-mgmt-sts-1				866	10.12.2.20		Create	vrp-gke5-ts	Connected

3. CN-NGFW ポッドにライセンスが付与されていることを確認するには、**Panorama** > プラグイン > **Kubernetes** > ライセンス使用状況を選択し、各ポッドにライセンストークンが割り当てられていることを確認します。



PANORAMA			
NODE ID	FIREWALL POD NAME	LICENSE STATUS	NODE STATUS
rr-cluster-1 (3 Nodes, 3/3 Licensed)			
rr-cluster-1-default-pool-e2d3de37-1jtz	pan-ngfw-ds-4qfb	<input checked="" type="checkbox"/>	Successfully licensed. Created at: 06-11 22:30:37 UTC
rr-cluster-1-default-pool-e2d3de37-9hq5	pan-ngfw-ds-z5z8k	<input checked="" type="checkbox"/>	Successfully licensed. Created at: 06-11 22:30:37 UTC
rr-cluster-1-default-pool-e2d3de37-jn8z	pan-ngfw-ds-vr8hx	<input checked="" type="checkbox"/>	Successfully licensed. Created at: 06-11 22:30:36 UTC

CN-Series ファイアウォールを AliCloud (ACK) に Kubernetes サー ビスとしてデプロイする

どこで使用できますか?	何が必要ですか?
<ul style="list-style-type: none"> CN-Seriesデプロイメント 	<ul style="list-style-type: none"> CN-Series 10.1.x or above Container Images PanoramaPAN-OS 10.1.xまたはPAN-OS 10.2.xバージョンを実行している

CNシリーズ コアビルディング ブロックとCNシリーズ ファイアウォールによるKubernetesの安全なワークロード内のワークフローの概要を確認したら、同じクラスタ内のコンテナ間やコンテナと他のワークロード タイプ(仮想マシンやベアメタル サーバーなど)間のトラフィックを保護するためのCNシリーズ ファイアウォールのデプロイから始めることができます。

plugin-serviceaccount.yamlファイルを確実に適用する必要があります。詳細については、「[クラスタ認証用サービスアカウントの作成](#)」を参照してください。



- CN-Series ファイアウォールを Kubernetes サービスとして ACK にデプロイする場合、`pan-plugin-cluster-mode-secret` が存在する必要があります。

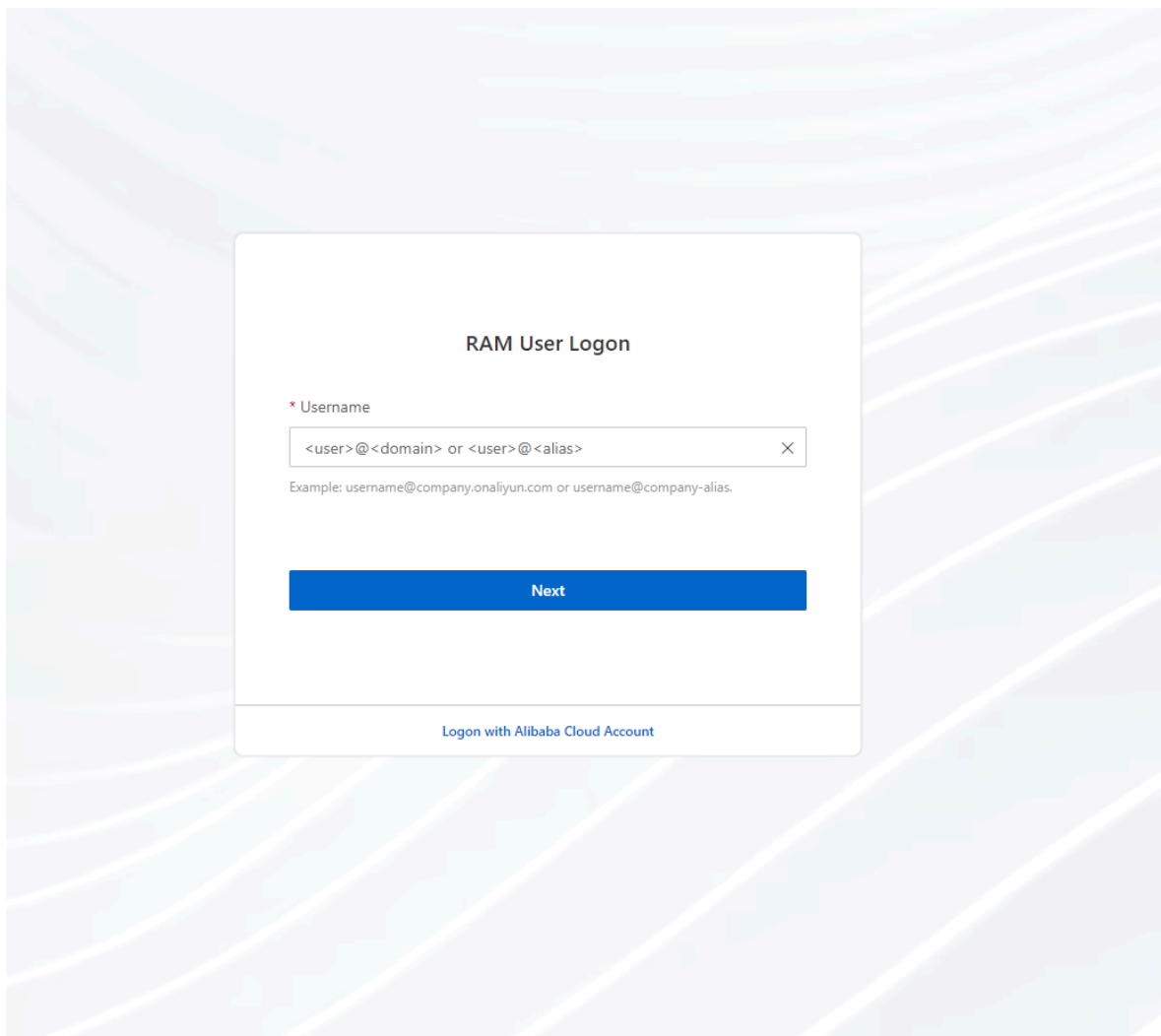
開始する前に、CN-Series YAML ファイルのバージョンが PAN-OS バージョンと互換性があることを確認します。詳細については、[CN-SeriesYAML](#)を参照してください。

以下の手順を実行して、CN-Series ファイアウォールを Kubernetes サービスとして ACK プラットフォームにデプロイします。

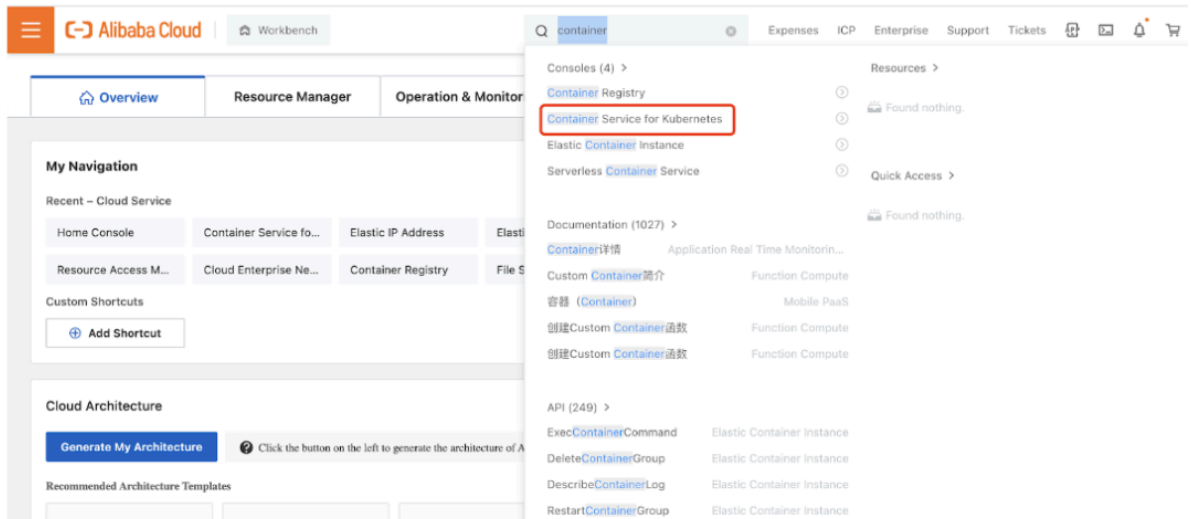
STEP 1 | Kubernetes クラスタをセットアップします。

ACKでクラスタを作成するには、次のようにします。

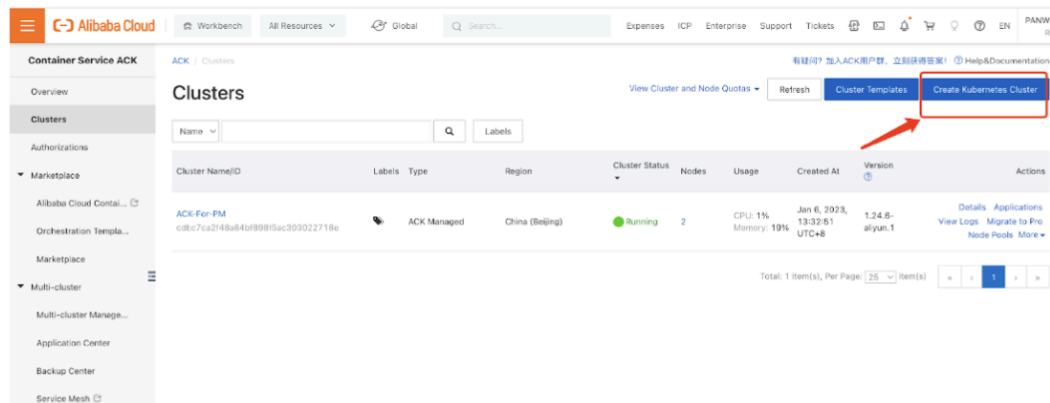
1. RAMログイン資格情報を使用して[RAM User Logon \(RAMユーザーログオン\)](#)にログオンします。



2. 上部のナビゲーションバーで、クラスタを作成するリージョンを選択し、ビジネス要件に基づいてリソースグループを選択します。
 - クラスタのリージョンは、クラスタの作成後に変更できません。
 - デフォルトでは、アカウント内のすべてのリソースグループが表示されます。
3. 検索バーメニューで**Container Service for Kubernetes (Kubernetes向けコンテナサービス)**を検索します。



4. **[Create Kubernetes Cluster (Kubernetesクラスタの作成)]**をクリックします。



5. クラスタを作成するには、ウィザードの指示に従ってソフトウェアパラメータ、ハードウェアパラメータ、および基本パラメータを設定する必要があります。これらの必須パラメータの設定の詳細については、「[Create a Cluster on ACK \(ACKでクラスタを作成する\)](#)」

を参照してください。次の手順は、ACKプラットフォームでのクラスタ作成のサンプルを示しています。



CN-Series on Alibaba cloud ACKは、Terway Networkプラグインのみをサポートしています

- [VPC]、[Network Plugin (ネットワークプラグイン)]、[vSwitch]を選択します。

The screenshot shows the 'Network' configuration page in the ACK console. The 'VPC' dropdown is set to 'vpc-xiaofang'. The 'Network Plug-in' is set to 'Terway'. The 'vSwitch' section shows a table of available vSwitches, with 'cn-pod2' selected.

	name	vsw-id	zone	cidr-block	ip-count
<input type="checkbox"/>	inside	vsw-2zej8ngtuyp6r6qy1eol	Beijing Zone C	10.101.2.0/24	252
<input type="checkbox"/>	outside	vsw-2zerc7sn6emhk9mq4lzy7	Beijing Zone C	10.101.1.0/24	252
<input type="checkbox"/>	mgmt	vsw-2zepoq1k3a7zx1pk2laf	Beijing Zone C	10.101.0.0/24	252
<input checked="" type="checkbox"/>	cn-pod2	vsw-2ze5v4zny1j58rzzdd19t	Beijing Zone A	10.101.102.0/24	243

- [POD v Switch] を選択します。

Pod vSwitch

AllZoneA (2 / 1)

	inside	vsw-2zej8ngtuy6r6qy1eoil	Beijing Zone C	10.101.2.0/24	252
	outside	vsw-2zerc7sn6emhk9mq4lzy7	Beijing Zone C	10.101.1.0/24	252
	mgmt	vsw-2zepoq1k3a7zx1pk2iafs	Beijing Zone C	10.101.0.0/24	252
<input checked="" type="checkbox"/>	cn-pod2	vsw-2ze5v4zny1j58rzzdd19t	Beijing Zone A	10.101.102.0/24	252
<input checked="" type="checkbox"/>	cn-pod1	vsw-2zex1z33lu6ffu72ko5ry	Beijing Zone A	10.101.101.0/24	252
<input type="checkbox"/>	cn-node-ip	vsw-2ze5nzjrkzio4sbf5d2n9	Beijing Zone A	10.101.10.0/24	252

Create vSwitch

The prefix length of the VSwitch address is recommended to be no greater than 19 bits.

Service CIDR

192.168.0.0/16

Recommended Value:192.168.0.0/16

Valid values: 10.0.0.0/16-24, 172.16-31.0.0/16-24, and 192.168.0.0/16-24.

- **[Configure SNAT (SNATを設定)]**、**[Access to API Server (APIサーバーへアクセス)]**、**[Security Groups (セキュリティグループ)]**、**[Resource Group (リソースグループ)]**を選択します。

Configure SNAT ☒ Configure SNAT for VPC
Nodes and applications in the cluster have Internet access. If the VPC that you select has a NAT gateway, ACK uses this NAT gateway to enable Internet access. If the VPC does not have a NAT gateway, ACK automatically creates a NAT gateway and configures SNAT rules. For more information, see [NAT Gateway bill of materials](#).

Access to API Server [SLB Instance Specifications](#)
By default, an internal-facing SLB instance is created for the API server. You can modify the specification of the SLB instance. If you delete the SLB instance, the API server cannot access the API server.

☒ Expose API Server with EIP
If you select this check box, the internal-facing SLB instance is associated with an EIP. This allows you to access the API server of the cluster over the Internet.

RDS Whitelist [Select RDS Instance](#)
We recommend that you go to the RDS console to add the CIDR blocks of the specified nodes and specified pods to a whitelist of the RDS instance. (If the RDS instance is not in the running state, the node pool cannot be scaled out.)

Security Group [Create Basic Security Group](#) [Create Advanced Security Group](#)
To use a basic security group, the total number of pods in the cluster cannot exceed 2,000 if you select the Terway network plug-in. Otherwise, you must use an advanced security group. [Security group overview](#)

Deletion Protection ☐ Enable
Cluster Cannot Be Deleted in Console or by Calling API

Resource Group [Create Resource Group](#)
To create a resource group, click [here](#).

- **[Quantity (数量)]**、**[Operating System (オペレーティングシステム)]**を選択し、次に**[Node Pool Configurations (ノードプール構成)]**の**[Logon Type (ログオンタイプ)]**を選択します。

Instance type is used. The actual instance types used to create nodes are subject to inventory availability.

ecs.sn2nec.xlarge (4 vCPU 16 GiB, General purpose type family with enhanced network performance sn2nec) Move Up Move Down

Quantity: 2 unit(s)

Nodes will be evenly assigned to your selected vswitches.
A standard managed cluster can contain up to 100 nodes. To use a larger cluster, create a professional managed cluster.

System Disk: SSD Disk 120 GiB

Mount Data Disk: You have selected 0 disks and can select 10 more.
[Disk Parameters and Performance](#) [Add Data Disk](#) [Recommended](#)

Operating System: Alibaba Cloud Linux 3.2104

Security: Disable Reinforcement based on classified protection CIS Reinforcement ⓘ

Reinforcement

Logon Type: Key Pair Password Later

Key Pair: key-par-Alibaba [Create a key pair](#)

[ACK Billing](#) SLB Price: ¥ 0.100 /Hours EIP Price: ¥ 0.800 /GB ECS Price: ¥ 4.91 /Hours [Prev: Cluster Configurations](#) [Next: Compute Resources](#)

- **[Public Network (パブリックネットワーク)]**タブに移動し、**[Service Discovery (サービス検出)]**、**[Volume Plug-in (ボリュームプラグイン)]**、**[Monitoring Agents (監視エージェント)]**のチェックボックスをオフにします。

The screenshot shows the 'Component Configurations' step in the AliCloud ACK console. The 'Ingress' section has 'Nginx Ingress' selected. The 'SLB Network Type' is 'Public Network' and 'SLB Specifications' is 'slb.s1.small'. The 'Service Discovery' section has 'Install NodeLocal DNSCache' unchecked. The 'Volume Plug-in' section has 'CSI' selected. The 'Monitoring Agents' section has 'Install CloudMonitor Agent on ECS Instance' and 'Enable Prometheus Monitoring' both unchecked. A red box highlights the 'Service Discovery' and 'Monitoring Agents' sections, with the text 'unselection all' written next to it.

ACK Billing SLB Price: ¥ 0.220 /hours
¥ 0.800 /GB EIP Price: ¥ 0.800 /GB ECS Price: ¥ 4.91 /Hours Prev: Node Pool Configu

6. 利用規約のチェックボックスを選択します。

RAM Role Authorization Check	Passed
Dependent Service Activation Status	Passed
Auto Scaling Status Check	Passed
Service Quota Check	Passed
System Disk Size Check	Passed
Data Disk Size Check	Passed
Account Balance Check	Passed

Terms of Service

During the cluster creation process, the following operations may be performed depending on cluster configurations:

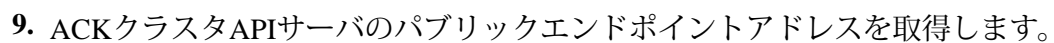
- Create ECS instances, configure a public key to enable SSH login from master nodes to other nodes, and configure the Kubernetes cluster through Cloudinit.
- Create a security group that allows access to the VPC network over ICMP.
- Create VPC routing rules.
- Create a NAT gateway and Elastic IP addresses.
- Create a RAM role and grant it the following permissions: query, create, and delete ECS instances, create and delete cloud disks, and all permissions on SLB instances, CloudMonitor, VPC, Log Service, and NAS. The Kubernetes cluster dynamically creates SLB instances, cloud disks, and VPC routing rules based on your settings.
- Create an internal SLB instance and open port 6443.
- When you use a dedicated or managed Kubernetes cluster, the system collects log and monitoring information about control components on master nodes to help ensure cluster stability.

☒ I have read and understand the preceding statement. I also have read and accept the [Terms of Service and Disclaimer](#).

ACK Billing SLB Price: ¥ 0.220 /Hours
EIP Price: ¥ 0.800 /GB
ECS Price: ¥ 4.91 /Hours

Prev: Component Configurations

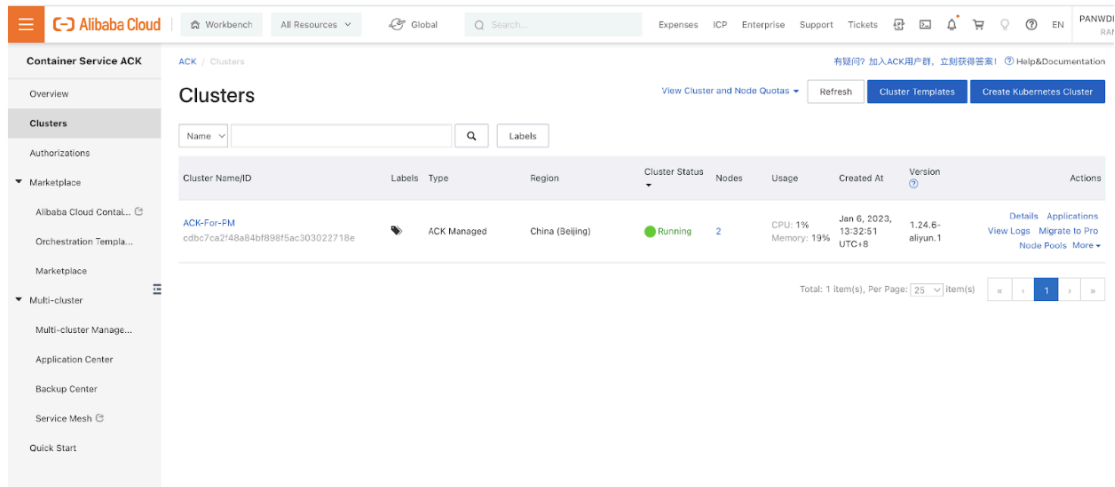
7. [クラスタの作成]をクリックします。
8. API server KeyをチェックしてACKクラスタにログインし、以下のコンテンツをローカルコンピュータの\$HOME / . kube/configにコピーします。



The screenshot shows the Alibaba Cloud ACK console interface. The left sidebar contains navigation menus for 'Nodes', 'Workloads', 'Network', and 'Configurations'. The main content area displays the details for the 'ACK-For-PM' cluster. The 'Basic Information' tab is active, showing the cluster ID, status (Running), region (China (Beijing)), and time zone (Asia/Shanghai). The 'Cluster Information' section lists various endpoints and configurations, with the 'API Server Public Endpoint' highlighted as `https://47.93.191.191:6443`.

Basic Information	
Cluster ID: cdbc7ca2f48a84bf998f5ac303022718e	Running
Region: China (Beijing)	Time Zone: Asia/Shanghai

Cluster Information	
API Server Public Endpoint	https://47.93.191.191:6443 Change EIP Unbind EIP
API Server Internal Endpoint	https://10.101.10.169:6443 Set access control Troubleshoot connection issues
Service CIDR	192.168.0.0/16
RRSA OIDC	Enable RRSA Configure RAM permissions for service accounts to isolate permissions among pods
Kube-proxy Mode	ipvs
Network Plug-in	terway-eniip
Custom Certificate SANs	Update
Testing Domain	*cdbc7ca2f48a84bf998f5ac303022718e.cn-beijing.alicontainer.com Rebind Domain Name



クラスタのリソースが適切であることを確認します。デフォルトのGKE ノードプール仕様は、CN-Series ファイアウォールには適していません。クラスタにファイアウォールをサポートするためのCN-Series前提条件リソースがあることを確認する必要があります。

```
kubectl get nodes
```

```
kubectl describe node <node-name>
```

コマンド出力の[容量]見出しの下にある情報を表示して、指定したノードで使用可能な CPU とメモリーを確認します。

CPU、メモリ、およびディスク ストレージの割り当てはニーズによって異なります。CN-Seriesのパフォーマンスとスケーラビリティを参照してください。

次の情報があることを確認する必要があります。

- PanoramaにAPI サーバーを設定するためのエンドポイント IP アドレスを収集します。

The screenshot shows the 'Cluster Definition' window. The 'Name' field contains 'on_prem-clstr'. The 'API server address' field contains '10.2...'. The 'Type' dropdown is set to 'Native-Kubernetes'. Below this is a 'Label Selector' section with a search bar and a table. The table has four columns: 'TAG PREFIX', 'NAMESPACE', 'LABEL SELECTOR FILTER', and 'APPLY ON'. The table is currently empty. At the bottom of the window are three buttons: 'Validate', 'OK', and 'Cancel'.

Panorama は、この IP アドレスを使用して、Kubernetes クラスタに接続します。

- テンプレートスタック名、デバイス グループ名、Panorama IP アドレス、およびオプションで ログ コレクタグループ名をPanorama から収集します。

詳細については、[親デバイス グループとテンプレート スタックの作成](#)を参照してください。

- [VM 認証キーと自動登録の PIN ID と値](#)を収集します。
- イメージをダウンロードしたコンテナ イメージ リポジトリの場所。

STEP 2 | (任意) Panorama の Kubernetes プラグインでカスタム証明書を設定した場合は、次のコマンドを実行して証明書シークレットを作成する必要があります。ファイル名を ca.crt から変更しないでください。pan-cn-mgmt.yaml および pan-cn-ngfw.yaml のカスタム証明書のボリュームはオプションです。

```
kubectrl -n kube-system create secret generic custom-ca --from-file = ca.crt
```

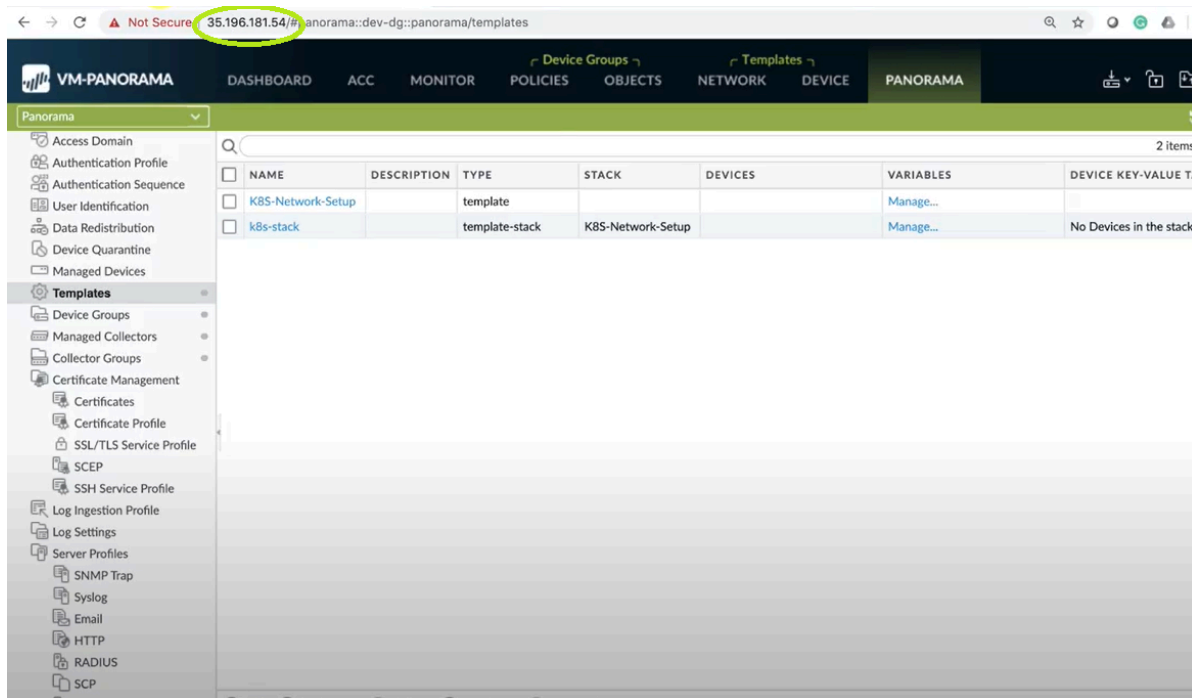
STEP 3 | YAML ファイルを編集して、CN-Series ファイアウォールをデプロイするために必要な詳細を記入します。


```
apiVersion: v1 kind:ConfigMap metadata: name: pan-mgmt-config
namespace: kube-system data: PAN_SERVICE_NAME: pan-mgmt-
svc PAN_MGMT_SECRET: pan-mgmt-secret # Panorama settings
PAN_PANORAMA_IP: "<panorama-IP>" PAN_DEVICE_GROUP: "<panorama-
device-group>" PAN_TEMPLATE_STACK: "<panorama-template-stack>"
PAN_CGNAME: "<panorama-collector-group>" PAN_CTNR_MODE_TYPE: "k8s-
service"
```

```
apiVersion: v1 kind:Secret metadata: name: pan-mgmt-secret
namespace: kube-system type:Opaque stringData: # Panorama Auth
Key PAN_PANORAMA_AUTH_KEY: "<panorama-auth-key>" # Thermite
```

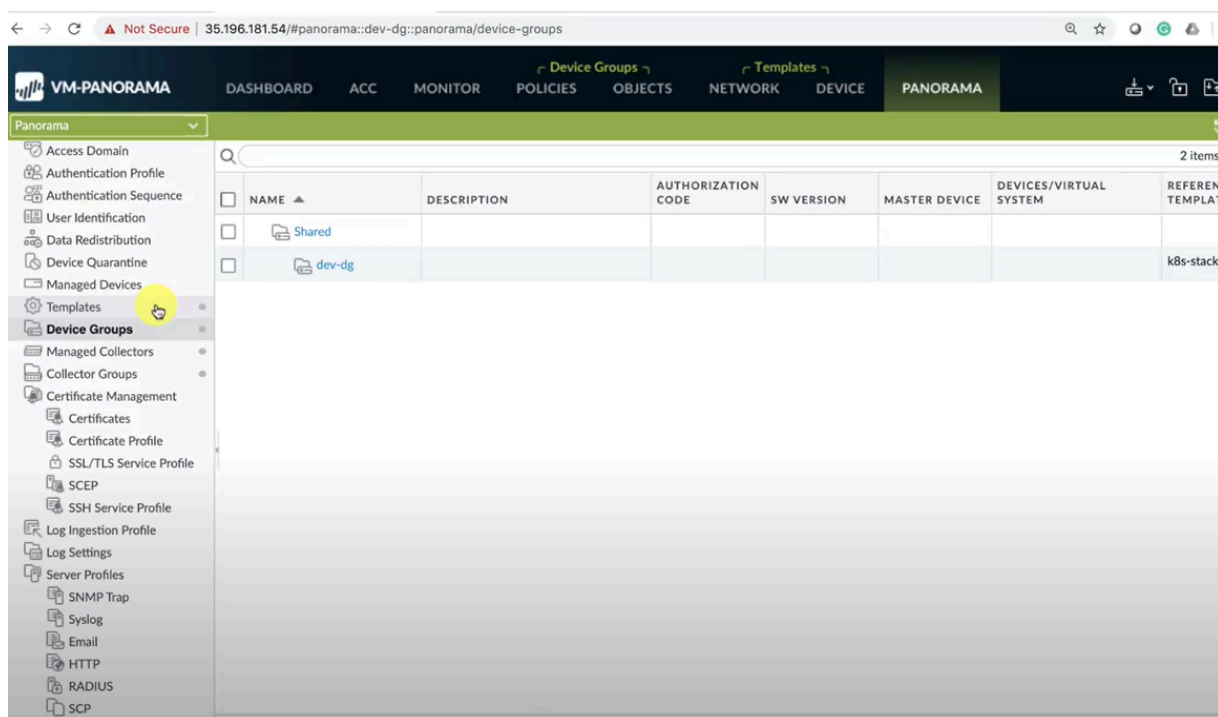
```
Certificate retrieval CN-SERIES-AUTO-REGISTRATION-PIN-ID: "<PIN
Id>" CN-SERIES-AUTO-REGISTRATION-PIN-VALUE: "<PIN-Value>"
```

以下の図に示すように、YAML ファイルの PAN_PANORAMA_IP パラメータの値が実際の Panorama IP アドレスと一致していることを確認する必要があります。

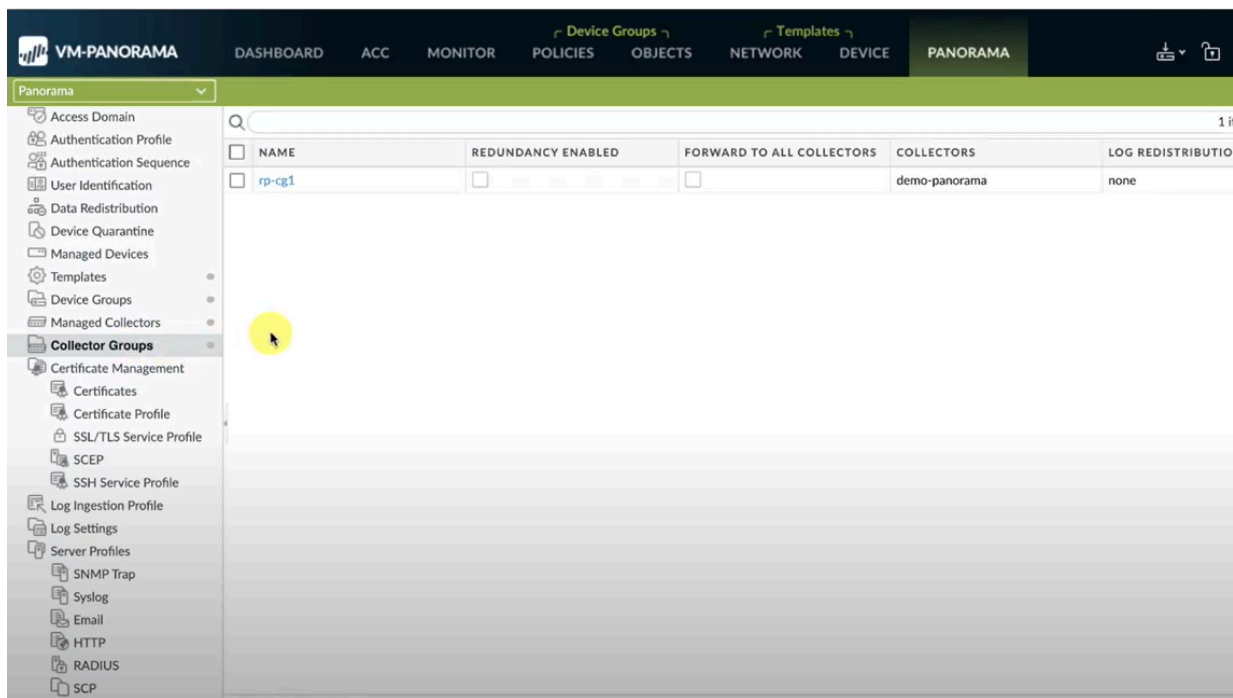


 YAML ファイルの最新バージョンは、[Palo Alto Networks Kubernetes Security のリポジトリ - CN Series](#) で入手できます。「Switch branches/tags (分岐/タグの切り替え)」ドロップダウンメニューから、最新の分岐またはタグを選択できます。

以下の図に示すように、YAML ファイルの PAN_DEVICE_GROUP と PAN_TEMPLATE のパラメータ値が、Panorama で作成したデバイスグループとテンプレートスタックの名前と一致していることを確認する必要があります。



PAN_PANORAMA_CG_NAME のパラメータ値が、作成したログコレクター名と同じであることを確認する必要があります。



詳細については、CN-Seriesの[yamlファイルの編集可能なパラメータ](#)を参照してください。

STEP 4 | CN-NGFW サービスをデプロイします。次の手順を実行します。

Kubernetesサービスとしてデプロイすると、CN-NGFW のインスタンスをセキュリティノードにデプロイでき、アプリケーションポッドトラフィックは、検査と執行のために利用可能な CN-NGFW インスタンスにリダイレクトされます。

1. pan-cni-serviceaccount.yaml を使用してサービス アカウントが作成されたことを確認します。

[クラスタ認証用のサービス アカウントの作成](#)を参照してください。

2. Kubectl を使用して pan-cni-configmap.yaml を実行します。

```
kubectl apply -f pan-cni-configmap.yaml
```

3. kubectl を使用して pan-cn-ngfw-svc.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-svc.yaml
```



この yaml は pan-cni.yaml の前にデプロイする必要があります。

4. Kubectl を使用して pan-cni.yaml を実行します。

```
kubectl apply -f pan-cni.yaml
```

5. pan-cni-configmap YAML ファイルと pan-cni YAML ファイルが変更されたことを確認します。

6. 以下のコマンドを実行して、出力が以下の例のようになっていることを確認します。

```
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $ kubectl get pods -n kube-system | grep pan-cni
pan-cni-nmqkf          Running    0          2m11s
pan-cni-wjtkq          Running    0          2m11s
pan-cni-xrc2z          Running    0          2m12s
@cloudshell:~/Kubernetes-master/pan-cn-k8s-service/gke (v series-mktplace) $
```



Alicloud ACKは標準メトリックベースのオートスケーリングのみをサポートしています。

STEP 5 | CN-MGMT StatefulSet をデプロイします。

デフォルトで、管理プレーンは耐障害性を提供する StatefulSet としてデプロイされます。1つの CN-MGMT StatefulSet に最大 30 個のファイアウォール CN-NGFW ポッドを接続できます。

1. (静的にプロビジョニングされた PV のみに必要) CN-MGMT StatefulSet 用の永続ボリューム (PV) をデプロイします。

1. pan-cn-pv-local.yaml で定義されたローカル ボリューム名と一致するディレクトリを作成します。

少なくとも 2 つのワーカー ノード上に 6 つのディレクトリが必要です。CN-MGMT StatefulSet をデプロイする各ワーカー ノードにログインして、ディレクトリを作成

します。たとえば、/mnt/pan-local1 ~ /mnt/pan-local6 のディレクトリを作成するには、次のコマンドを実行します。

```
mkdir -p /mnt/pan-local1 /mnt/pan-local2 /mnt/pan-local3 /mnt/pan-local4 /mnt/pan-local5 /mnt/pan-local6
```

2. pan-cn-pv-local.yaml を変更します。

nodeaffinity の下でホスト名を一致させ、上記で spec.local.path に作成したディレクトリが変更されたことを確認してから、そのファイルをデプロイして、新しいストレージ クラス pan-local-storage とローカル PV を作成します。



pan-cn-mgmt.yaml ファイルでは、*volumeClaimTemplates* の作成時にストレージクラス名を *alicloud-disk-available* として追加する必要があります。

以下に例を示します。

```
storageClassName: alicloud-disk-available
```

収納サイズは、すべてのPVで20G以上が必要です。

2. pan-cn-mgmt-configmap YAML ファイルと pan-cn-mgmt YAML ファイルが変更されたことを確認します。

pan-cn-mgmt.yaml のサンプル

```
initContainers: - name: pan-mgmt-init image: <your-private-registry-image-path>
```

```
containers: - name: pan-mgmt image: <your-private-registry-image-path> terminationMessagePolicy: FallbackToLogsOnError
```

3. Kubectl を使用して yaml ファイルを実行します。

```
kubectl apply -f pan-cn-mgmt-configmap.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-crd.yaml
```

```
kubectl apply -f pan-cn-mgmt-slot-cr.yaml
```

```
kubectl apply -f pan-cn-mgmt-secret.yaml
```

```
kubectl apply -f pan-cn-mgmt.yaml
```

pan-mgmt-serviceaccount.yaml は、[クラスター認証用のサービスアカウントの作成](#)を以前に完了していない場合にのみ実行する必要があります。

4. 次のコマンドを実行して、CN-MGMT ポッドが起動していることを確認します。


```
kubectl get pods -l app = pan-mgmt -n kube-system
```

これには、5～6分かかります。

STEP 6 | CN-NGFW ポッドをデプロイします。

1. PAN-CN-NGFW-CONFIGMAP と PAN-CN-NGFW に詳述されているように YAML ファイルが変更されたことを確認します。

```
containers: - name: pan-ngfw-container image: <your-private-registry-image-path>
```

2. Kubectl apply を使用して pan-cn-ngfw-configmap.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw-configmap.yaml
```

3. Kubectl apply を使用して pan-cn-ngfw.yaml を実行します。

```
kubectl apply -f pan-cn-ngfw.yaml
```

4. CN-NGFW ポッドがデプロイされたことを確認します。

```
kubectl get pods -n kube-system -l app=pan-ngfw -o wide
```

STEP 7 | Kubernetes クラスタ上の CN-MGMT、CN-NGFW、および PAN-CNI が表示されていることを確認します。

```
kubectl -n kube-system get pods
```

STEP 8 | 新しいポッドからのトラフィックがファイアウォールにリダイレクトされるようにアプリケーション yaml または名前空間に注釈を付けます。

検査のためにトラフィックを CN-NGFW にリダイレクトするには、以下のアノテーションを追加する必要があります：

```
annotations : paloaltonetworks.com/firewall : pan-fw
```

たとえば、「default」名前空間のすべての新しいポッドの場合：

```
kubectl annotate namespace default paloaltonetworks.com/firewall=pan-fw
```


STEP 9 | クラスタでアプリケーションをデプロイします。

CN-Series を OpenShift にデプロイする

どこで使用できますか？	何が必要ですか？
<ul style="list-style-type: none"> CN-Series OpenShift環境へのデプロイメント 	<ul style="list-style-type: none"> CN-Series 10.1.x or above Container Images PanoramaPAN-OS 10.1.x以降のバージョンを実行している

pan-cni は、アプリケーション ポッドのデフォルトの "eth0" インターフェース上のトラフィックを保護します。マルチホーム ポッドを使用している場合は、ブリッジベースの接続を使用して他のポッドやホストと通信するように設定された追加のインターフェースを保護するように CN-NGFW ポッドを設定できます。アプリケーション YAML 内の注釈に応じて、各ポッドに関連付けられたすべてのインターフェースまたは選択された数のインターフェースからのトラフィックを検査するように CN-Series ファイアウォールを設定できます。

pan-cni は、ネットワークを作成しないため、他の CNI プラグインのように IP アドレスを必要としません。


-  *OpenShift* で *CN-Series* を *Kubernetes* サービスとしてデプロイするには、*PAN-OS 10.1.3* 以降が必要です。さらに、*OpenShift* 上の *Kubernetes* サービスとしての *CN-Series* は、インターフェース **eth0** のみを保護します。

STEP 1 | クラスタをデプロイします。

クラウド プラットフォーム ベンダーのドキュメントを参照し、OpenShift のバージョンと CNI が CN-Series に対してサポートされていることを確認してください。[CN シリーズ ファイアウォールのイメージ ファイル](#)を取得し、[CN-Series yaml ファイル](#)で編集可能なパラメータを確認します。

STEP 2 | CN-Series を使用した Secure Kubernetes ワークロードに含まれるワークフローを使用します。

サービス認証情報を作成し、ファイアウォール YAML をデプロイします。

-  注:サービス認証情報ファイルが 10KB を超えている場合は、ファイルを gzip で圧縮し、圧縮したファイルの base64 エンコーディングを実施してから、ファイルの内容を *Panorama CLI* または *API* にアップロードまたは貼り付ける必要があります。

STEP 3 | Multus CNI プラグインと連動するように PAN-CNI プラグインを設定します。

OpenShift 上の Multus CNI は、他の CNI プラグインを呼び出す「メタプラグイン」として機能します。アプリケーションごとに、以下の作業を行う必要があります。

1. すべてのポッド名前空間で PAN-CNI NetworkAttachmentDefinition をデプロイします。

```
kubectl apply -f pan-cni-net-attach-def.yaml -n <target-namespace>
```

2. アプリケーション YAML を変更します。

pan-cni-net-attach-def.yaml をデプロイしたら、アプリケーション ポッド yaml に注釈を追加します。

```
paloaltonetworks.com/firewall: pan-fw
```

```
k8s.v1.cni.cncf.io/networks: pan-cni
```

上記注釈内に他のネットワークが含まれている場合は、検査する必要のあるネットワークの後ろに **pan-cni** を追加します。**pan-cni** の後ろのネットワークは、リダイレクトされず、検査されません。



ポッドに複数のネットワーク インターフェースが含まれている場合は、*pan-cni-configmap.yaml* の "interfaces" の下で、*CN-NGFW* ポッドがトラフィックを検査する対象のインターフェース名を指定する必要があります。

以下に例を示します。

```
template: metadata: annotations: paloaltonetworks.com/
firewall: pan-fw k8s.v1.cni.cncf.io/networks: bridge-conf,
macvlan-conf, sriov-conf, pan-cni
```



CNシリーズはRedHat OpenShiftバージョン4.13以上で、*Kubernetes Service*デプロイモードと*DaemonSet*モードで*OVN-Kubernetes Container Network Interface (CNI)*プラグインをサポートするようになりました。

CN-Series を OpenShift にデプロイする

どこで使用できますか？	何が必要ですか？
<ul style="list-style-type: none"> CN-Seriesデプロイメント 	<ul style="list-style-type: none"> CN-Series 10.1.x or above Container Images PanoramaPAN-OS 10.2.xバージョン以降を実行している

CN-Seriesコンテナファイアウォールが、[RedHat Openshiftプラットフォーム](#)のOperator Hubで利用できるようになりました。RedHat Operator Hubから直接CN-Seriesのコンテナファイアウォールを導入、設定、運用できます。

Openshift Operator HubでのCN-Seriesの前提条件。

OpenshiftオペレータハブにCN-Seriesファイアウォールを導入するための前提条件は次のとおりです。

- CN-Series ファイアウォールのライセンス取得CN-Series ファイアウォール ライセンスは、Kubernetes plugin on Panorama によって管理されます。CN-Series ファイアウォールをデプロイする準備ができたなら、認証コードを生成し、手元に置いておきます。詳細については、「[License the CN-Series Firewall \(CN-Series ファイアウォールのライセンス\)](#)」を参照してください。
- [Panorama](#) で VM 認証キーを生成します。
- [VM-Series](#)ファイアウォールへのデバイス証明書のインストール
- クラスタ認証用にサービス アカウントを作成する
- Panorama のデプロイ-CN-Series ファイアウォールのデプロイメントを設定、デプロイ、および管理するには、Panorama を使用する必要があります。Panorama アプライアンスのデプロイとセットアップの詳細については、[Panorama の設定](#)を参照してください。
- [CN-Series](#)ファイアウォール用のKubernetesプラグインをインストールします。
- OpenShiftクラスタは、[CN-Series前提条件](#)に準拠している必要があります。
- [Palo Alto Networks Customer Service Portal \(CSP\)](#) へのアクセスがあり、[Flexクレジット](#)を取得していることを確認します。
- OpenShiftライセンスとOpenShiftでリソースを作成する権限を持つアカウントを持つRedHatの顧客であることを確認します。
- OpenShiftクラスタが[CN-Series前提条件](#)に準拠していることを確認します。

詳しくは、[RedHat Openshift Operator Hub](#)でCN-Seriesを簡単に導入する方法をご覧ください。

OpenShift OperatorハブにCN-Seriesをデプロイする。

pan-cni は、アプリケーション ポッドのデフォルトの **eth0** インターフェース上のトラフィックを保護します。マルチホーム ポッドを使用している場合は、ブリッジベースの接続を使用して他のポッドやホストと通信するように設定された追加のインターフェースを保護するように CN-NGFW ポッドを設定できます。アプリケーション YAML 内の注釈に応じて、各ポッドに関連付けられたすべてのインターフェースまたは選択された数のインターフェースからのトラフィックを検査するように CN-Series ファイアウォールを設定できます。

pan-cni はネットワークを作成しないため、他の CNI プラグインのように IP アドレスを必要としません。

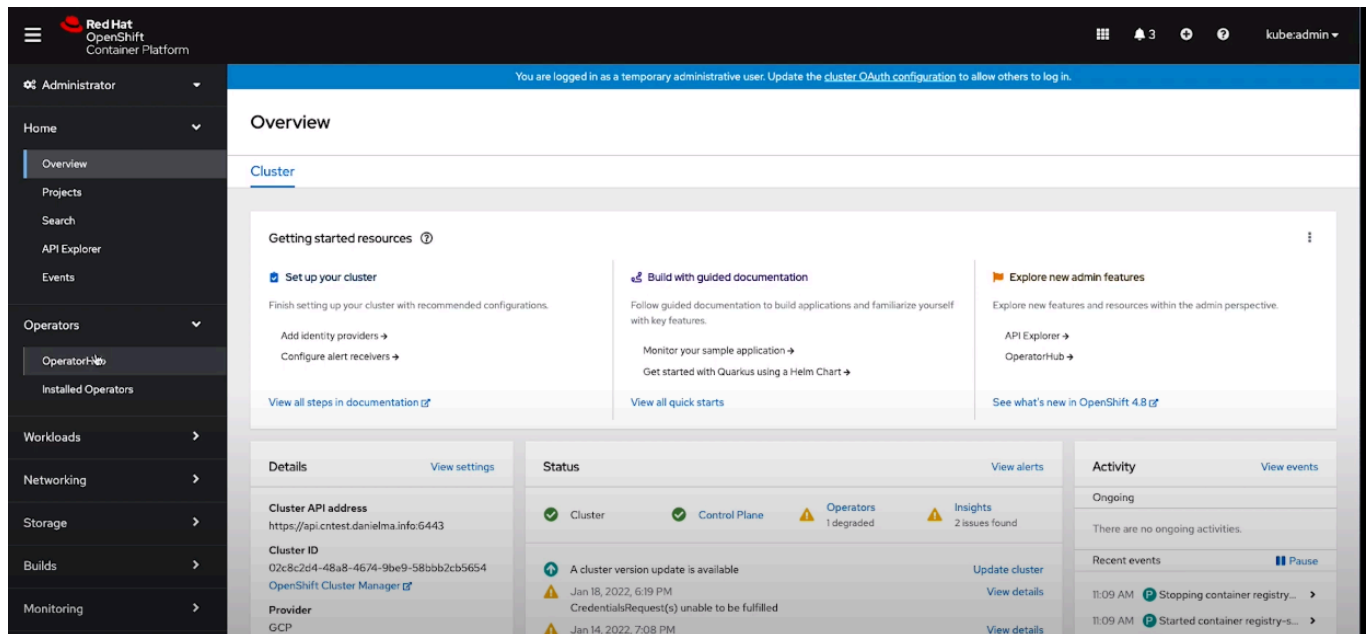


OpenShift OperatorハブにCN-Seriesを導入するには、PAN-OS 10.2以降が必要です。

Redhat OpenShiftオペレーターハブにCN-Seriesファイアウォールを導入する手順は、次のとおりです。

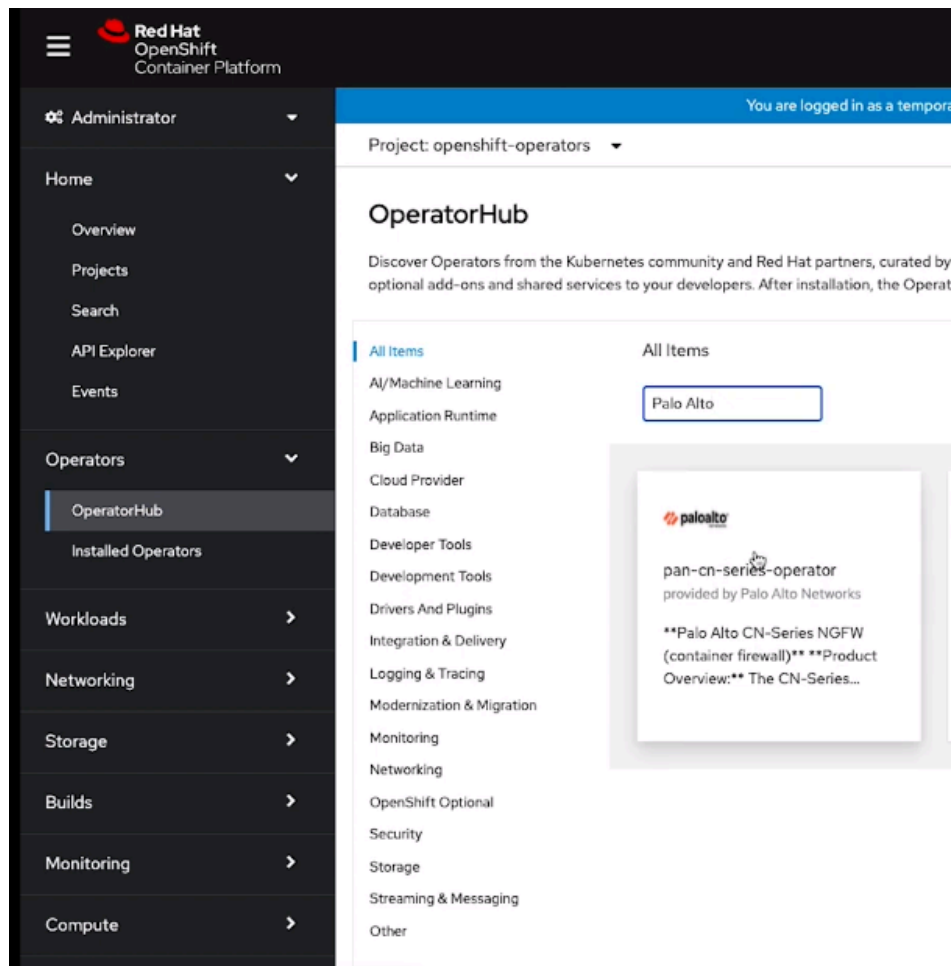
STEP 1 | Redhat OpenShiftコンテナコンソールにログインします。

STEP 2 | [Operators (オペレータ)]に移動し、[OperatorHub]をクリックします。



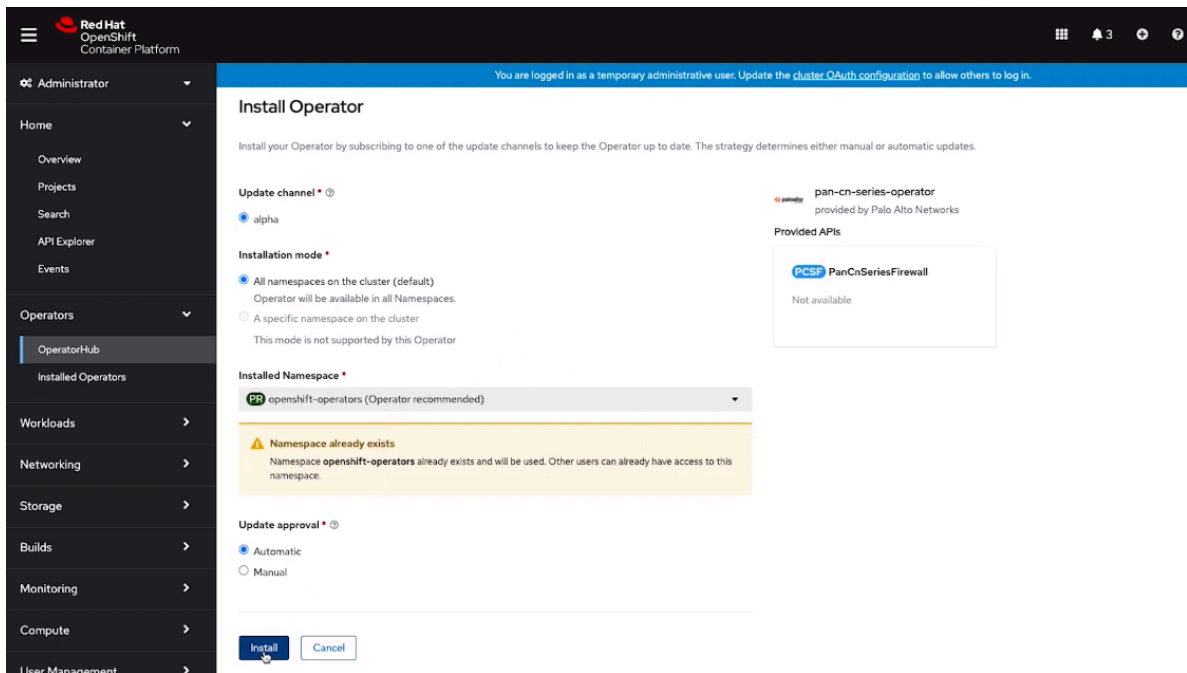
STEP 3 | [演算子]検索ボックスに「Palo Alto」と入力します。

STEP 4 | **pan-cn-series-operator** をクリックします。



pan-cn-series-operator タイルをクリックすると、インストールウィンドウが開きます。

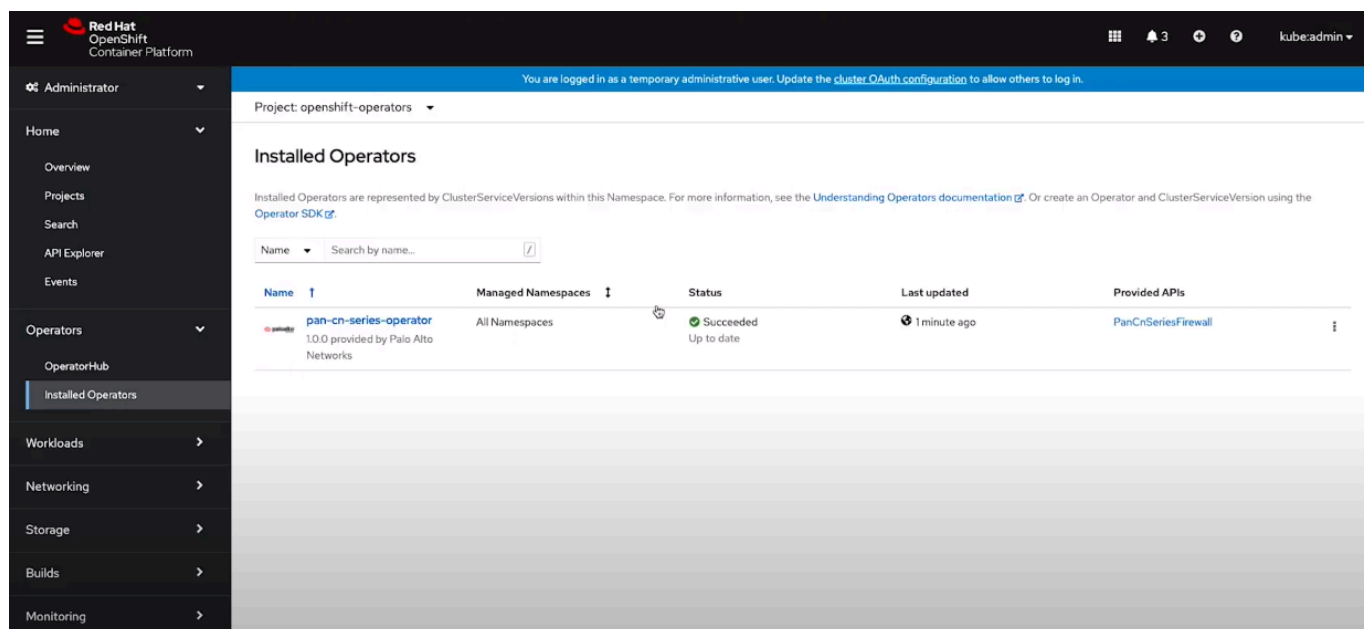
STEP 5 | 「**Install (インストール)**」をクリックして、OpenShift クラスタに pan-cn-series オペレータをインストールします。



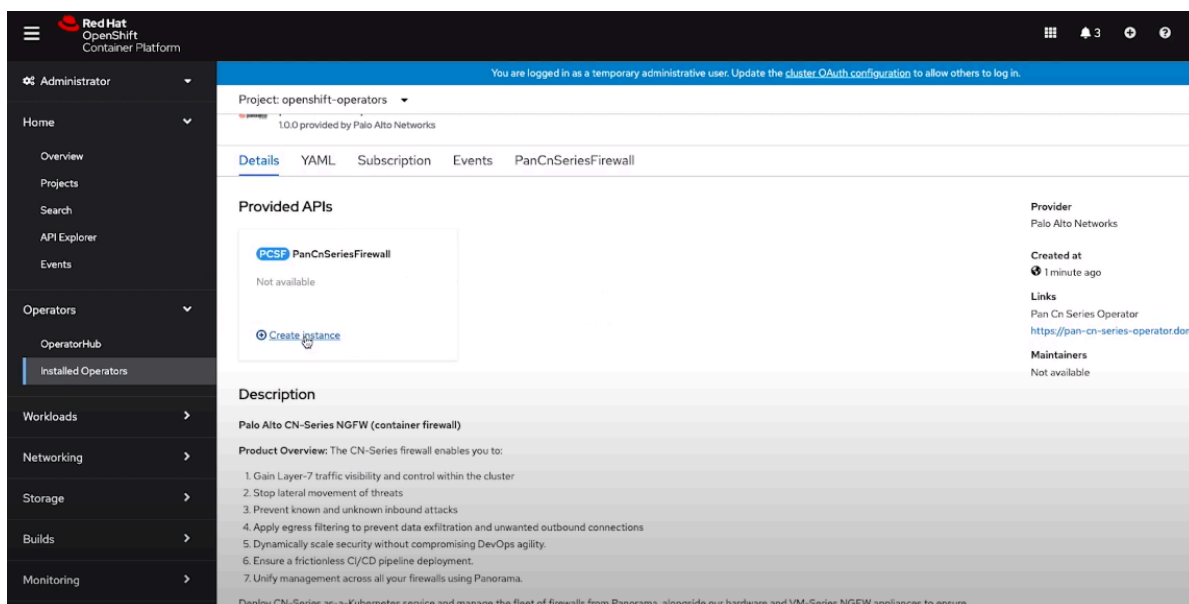
📋 ここで示す次の展開手順の前に、**インストール前手順**を完了します。

📋 サービス認証情報ファイルが **10KB** を超えている場合は、ファイルを **gzip** で圧縮し、圧縮したファイルの **base64** エンコーディングを実施してから、ファイルの内容を **Panorama CLI** または **API** にアップロードまたは貼り付ける必要があります。

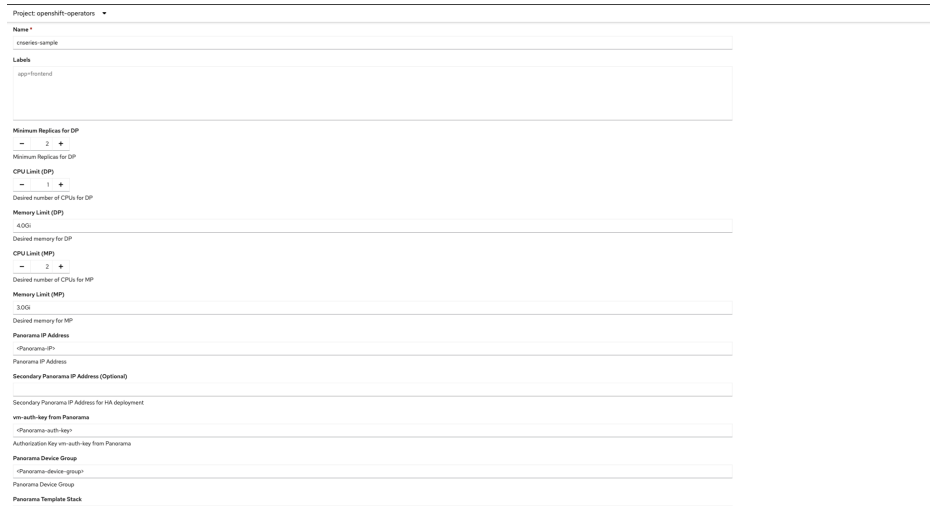
STEP 6 | 操作メニューで、[**Installed Operators** (インストール済オペレーター)]に移動し、インストールした**pan-cn-series-operator**をクリックします。



STEP 7 | **Create Instance** (インスタンスを作成) をクリックします。



STEP 8 | 一意のオペランド名を入力します。



STEP 9 | [Minimum Replicas for DP (DPの最小レプリカ数)]、[Memory Unit (メモリーユニット)]を入力し、DPとMPポッド用の[vCPU Limit]を入力します。vCPU の制限については、「[CN-Series Key Performance Metrics \(CNシリーズ主要パフォーマンス メトリクス\)](#)」を参照してください。

STEP 10 | Panorama IPアドレスを入力します。



STEP 11 | (任意)HA展開のセカンダリPanoramaIPアドレスを入力します。

STEP 12 | CN-Series PanoramaAuth Key (認証キー)を入力します。

STEP 13 | Panorama デバイスグループを入力します。

STEP 14 | Panorama テンプレートスタックを入力します。

STEP 15 | Panorama ログコレクタグループ名を入力します。

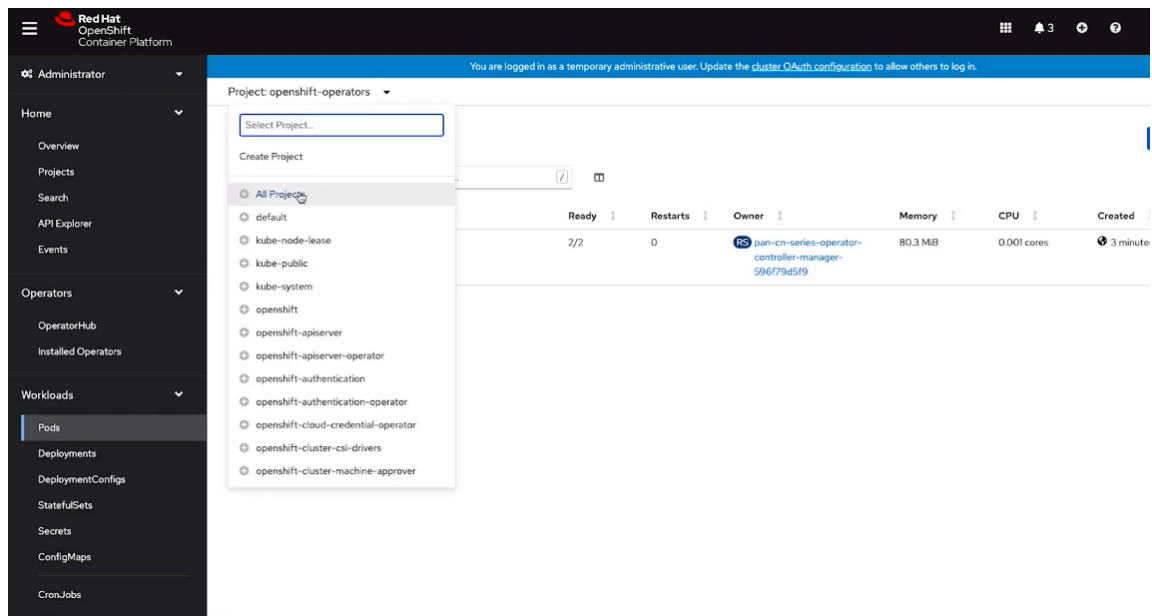
STEP 16 | (オプション) CSP(カスタマーサポートポータル)の**PIN ID**、**PIN**値、代替**URL**を入力します。

STEP 17 | PAN-OS のバージョンに基づいて、**CN シリーズ コンテナ レジストリ** コンソールで DP、MP、および CNI の適切なイメージにリンクします。

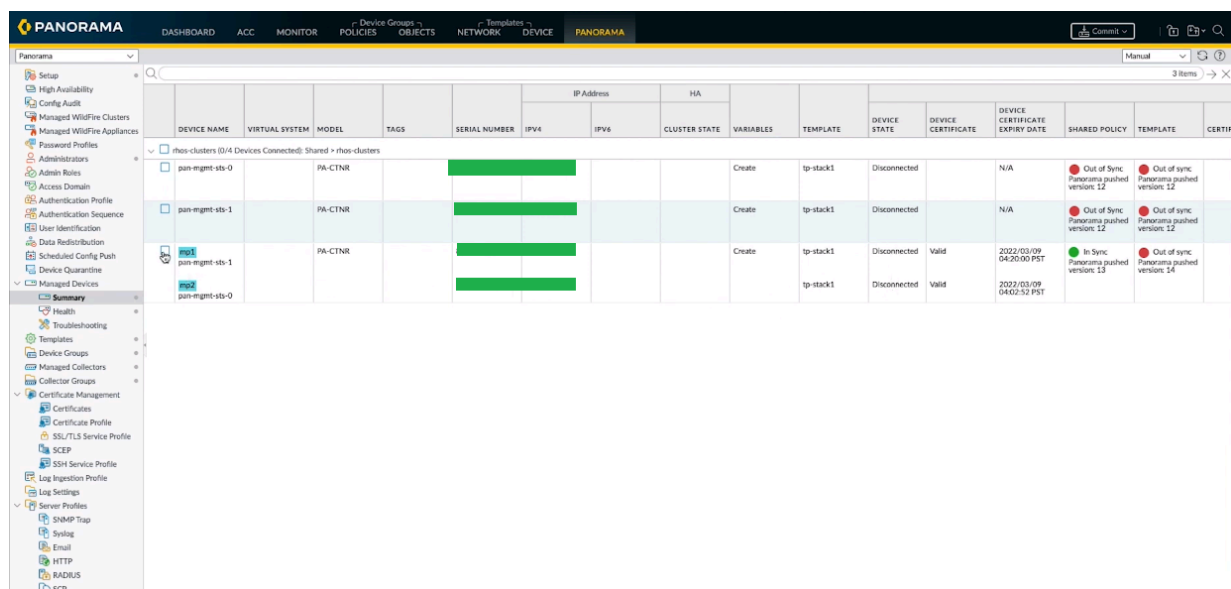
STEP 18 | 作成をクリックします。

STEP 19 | ナビゲーションメニューで、ポッドに移動します。

STEP 20 | プロジェクト[OpenShift-operators]を選択してから [kube-system] に移動し、オペランドの一部としてデプロイした CNI、管理、およびデータプレーンポッドの名前とステータスを表示します。



Panorama上でファイアウォールの導入状況を確認できます。導入後5分以内にデバイス状態が「接続済み」に変わります。



STEP 21 | Multus CNI プラグインと連動するように PALO ALTO NETWORKS-CNI プラグインを設定します。

OpenShift 上の Multus CNI は、他の CNI プラグインを呼び出す「メタプラグイン」として機能します。アプリケーションごとに、以下の作業を行う必要があります。

1. 次のコマンドを実行して、すべてのポッド名前空間に `pan-cni-net-attach-def.yaml` をデプロイします。

```
kubectl apply -f pan-cni-net-attach-def.yaml -n <target-namespace>
```

2. アプリケーション YAML を変更します。

`pan-cni-net-attach-def.yaml` をデプロイしたら、アプリケーション ポッド `yaml` に以下の注釈を追加します。

```
paloaltonetworks.com/firewall: pan-fw
```

```
k8s.v1.cni.cncf.io/networks: pan-cni
```

上記注釈内に他のネットワークが含まれている場合は、検査する必要があるネットワークの後ろに **pan-cni** を追加します。**pan-cni** の後ろのネットワークは、リダイレクトされず、検査されません。



ポッドに複数のネットワーク インターフェースが含まれている場合は、`pan-cni-configmap.yaml` の `interfaces` セクションの下で、`CN-NGFW` ポッドがトラフィックを検査する対象のインターフェース名を指定する必要があります。

以下に例を示します。

```
テンプレート: metadata: annotations: paloaltonetworks.com/
firewall: pan-fw k8s.v1.cni.cncf.io/networks: pan-cni
```

