paloalto TECHDOCS

CN-Series 防火牆部署模式

docs.paloaltonetworks.com

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

Last Revised

December 13, 2021

Table of Contents

快速入門 - CN-Series 防火牆部署

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 用於使用 Helm 進行 CN-Series 部署

透過以下步驟開始 CN-Series 部署：

1. 登入 CSP 帳戶然後啟動積分。
2. 建立部署設定檔。
3. 在 CN-Series 防火牆上安裝裝置憑證。
4. 安裝 Kubernetes 外掛程式並設定 CN-Series 的 Panorama。
5. 從 Palo Alto Networks GitHub 儲存庫，下載 CN-Series 部署檔案。從 Native-k8s 資料夾取得檔案，以用於原生 Kubernetes 內部部署或雲端部署
6. 使用或不使用 HELM 圖表儲存庫部署 CN-Series。

   建議使用 ***HELM*** 圖表部署 *CN-Series* 防火牆。

7. 設定 Panorama 保護 Kubernetes 部署

您可以選擇以下部署模式來部署 CN-Series 防火牆：

- 部署 CN-Series 防火牆作為 Kubernetes 服務（建議部署模式）- CN-Series 防火牆採用叢集部署模式。這種部署模式使用自動調整規模功能，透過基於原生 Kubernetes 的部署模型提高使用率、降低成本並增加規模。
- 將 CN-Series 防火牆部署為 DaemonSet- CN-Series 防火牆採用分散部署模式。當每個環境需要保護的節點數量較少時，建議採用此部署模式。
- 部署 CN-Series 防火牆作為 Kubernetes CNF- 此部署模式可以保護容器和非容器工作負載。您可以將其作為獨立的第三層進行部署。

CN-Series 防火牆的部署模式

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

在您檢閱 CN-Series 核心建置區塊以及使用 CN-Series 防火牆保護 Kubernetes 工作負載中的工作流程高階概觀之後，就可以開始部署 CN-Series 防火牆來保護相同叢集內容器之間的流量，以及容器與其他工作負載類型之間的流量（例如虛擬機器和裸機伺服器）。

如果您位於 OpenShift 環境上且需保護 5G 流量，請參閱 使用 CN-Series 防火牆保護 5G。

您需要 *kubectl* 或 *Helm* 這類標準 *Kubernetes* 工具來部署和管理 *Kubernetes* 叢集、應用程式和防火牆服務。*Panorama* 未設計成進行 *Kubernetes* 叢集部署和管理的協調器。進行叢集管理的範本是由「受管理 *Kubernetes* 提供者」所提供。*Palo Alto Networks* 提供社群支援的範本，以利用 Helm 和 Terraform 來部署 *CN-Series*。

- 部署 CN-Series 防火牆作為 Kubernetes 服務（建議部署模式）
- 將 CN-Series 防火牆部署為 DaemonSet
- 部署 CN-Series 防火牆作為 Kubernetes CNF
- 以獨立模式部署 Kubernetes CNF L3

從部署「*CN-Series* 作為 *DaemonSet*」移到「*CN-Series* 作為服務」之前（反之亦然），您必須刪除並重新套用 `plugin-serviceaccount.yaml`。

- 當您部署「*CN-Series* 作為 *DaemonSet*」時，`pan-plugin-cluster-mode-secret` 不得存在。
- 當您將 *CN-Series* 部署為 *Kubernetes* 服務時，必須要有 `pan-plugin-cluster-mode-secret`。

部署 CN-Series 防火牆作為 Kubernetes 服務（建議部署模式）

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 圖表的 CN-Series 部署

完成下列程序，以將 CN-Series 防火牆部署為 Kubernetes 服務。

開始之前，請確保 CN-Series YAML 檔案版本與 PAN-OS 版本相容。

- PAN-OS 10.1.2 或更新版本需要 YAML 2.0.2
- PAN-OS 10.1.0 和 10.1.1 需要 YAML 2.0.0 或 2.0.1

STEP 1 | 設定 Kubernetes 叢集。

1. 請驗證叢集具有足夠的版本。確保該叢集具有 CN-Series 先決條件資源以支援防火牆：

   ```
   kubectl get nodes
   ```

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

   檢視命令輸出之「容量」標題下的資訊，以查看所指定節點上可用的 CPU 和記憶體。

   CPU、記憶體和磁碟儲存體配置將取決於您的需求。請參閱 CN-Series 效能和調整規模。

   確保您具有下列資訊：

   - 收集「端點 IP 位址」，以在 Panorama 上設定 API 伺服器。Panorama 使用此 IP 位址來連線至 Kubernetes 叢集。
   - 從 Panorama 收集範本堆疊名稱、裝置群組名稱、Panorama IP 位址和日誌收集器群組名稱（選用）。
   - 收集 VM 驗證金鑰以及自動註冊 PIN ID 和值。
   - 將映像檔下載至其中的容器映像檔儲存庫位置。

STEP 2 | （選用）如果您已在 Panorama 的 Kubernetes 外掛程式中設定自訂憑證，則必須執行下列命令來建立憑證密碼。請不要從 ca.crt 變更檔案名稱。pan-cn-mgmt.yaml 和 pan-cn-ngfw.yaml 中的自定憑證數量是選用項目。

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

STEP 3 | 編輯 YAML 檔案，以提供部署 CN-Series 防火牆所需的詳細資料。

您需要取代 YAML 檔案中的映像檔路徑以包括私人登錄的路徑，以及提供必要參數。如需詳細資料，請參閱 CN-Series 部署 YAML 檔案中的可編輯參數。

STEP 4 | （僅限 AWS Outpost 上 EKS 的 CN-Series）更新儲存類別。若要支援在 AWS Outpost 上部署的 CN-Series，您必須使用儲存驅動程式 aws-ebs-csi-driver，確保 Outpost 在建立動態持續性磁碟區 (PV) 期間從 Outpost 拉出磁碟區。

1. 套用下列 yaml。

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

2. 驗證 ebs-sc 控制器是否正在執行。

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

3. 更新 pan-cn-storage-class.yaml 以符合下面的範例。

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

4. 將 **storageClassName: ebs-sc** 新增至下面所顯示位置中的 pan-cn-mgmt.yaml。

   ```
   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 # change this to 200Gi while using storageClassName for better disk iops - metadata: name: varlogpan spec: #storageClassName: pan-cn-storage-class //For better disk iops performance for dp logs accessModes: [ "ReadWriteOnce" ] storageClassName: ebs-sc resources: requests: storage:20Gi # change this to 200Gi while using storageClassName for better disk iops - 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 | 如果您要在 Kubernetes 環境中使用自動調整規模，請在繼續之前參閱水平 Pod 自動調整規模。

STEP 6 | 部署 CN-NGFW 服務。

1. 請驗證您已使用 pan-cni-serviceaccount.yaml 來建立服務帳戶。

   請參閱建立叢集驗證的服務帳戶。

2. 使用 Kubectl 來執行 pan-cni-configmap.yaml。

   kubectl apply -f pan-cni-configmap.yaml

3. 使用 kubectl 來執行 pan-cn-ngfw-svc.yaml。

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

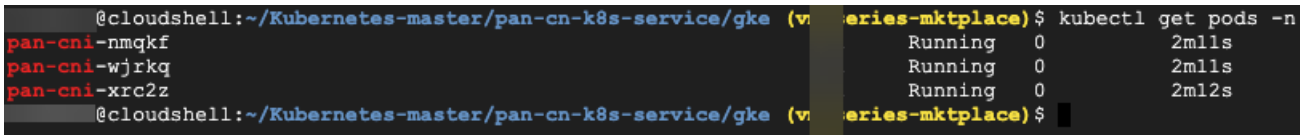
   必須在 *pan-cni.yaml* 之前部署此 *yaml*。

4. 使用 Kubectl 來執行 pan-cni.yaml。

   kubectl apply -f pan-cni.yaml

5. 請驗證您已修改 pan-cni-configmap 和 pan-cni YAML 檔案。
6. 執行下列命令，並確認您的輸出與下列範例相似。

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



STEP 7 | 部署 CN-MGMT StatefulSet。

管理平面預設會部署為可提供容錯的 StatefulSet。最多可以將 30 個防火牆 CN-NGFW Pod 連線至 CN-MGMT StatefulSet。

1. （僅為靜態佈建 PV 的必要項目）部署 CN-MGMT StatefulSet 的「永久性磁碟區 (PV)」。

   1. 建立目錄，以符合 pan-cn-pv-local.yaml 中所定義的本機磁碟區名稱。

      您需要至少 2 個背景工作節點上有六 (6) 個目錄。請登入將部署 CN-MGMT StatefulSet 以建立目錄的每個背景工作節點。例如，若要建立名為 /mnt/pan-local1 到 /mnt/pan-local6 的目錄，請使用命令：

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

   2. 修改 pan-cn-pv-local.yaml。

      符合 `nodeaffinity` 下方的主機名稱，並驗證您已修改上面您在 `spec.local.path` 中建立的目錄，然後部署檔案來建立新的 storageclass pan-local-storage 和本機 PV。

2. 驗證您已修改 pan-cn-mgmt-configmap 和 pan-cn-mgmt YAML 檔案。

   EKS 中的範例 pan-cn-mgmt-configmap。

   ```
    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>" # ctnr mode: "k8s-service", "k8s-ilbservice"
    PAN_CTNR_MODE_TYPE: "k8s-service" #Non-mandatory parameters #
    Recommended to have same name as the cluster name provided in
    Panorama Kubernetes plugin - helps with easier identification
    of pods if managing multiple clusters with same Panorama
    #CLUSTER_NAME: "<Cluster name>" #PAN_PANORAMA_IP2: "" #
    Comment out to use CERTs otherwise PSK for IPSec between
    pan-mgmt and pan-ngfw #IPSEC_CERT_BYPASS: "" # No values
    needed # Override auto-detect of jumbo-frame mode and
    force enable system-wide #PAN_JUMBO_FRAME_ENABLED: "true" #
    Start MGMT pod with GTP enabled.For complete functionality,
    need GTP # enable at Panorama as well. #PAN_GTP_ENABLED:
    "true" # Enable high feature capacities.These need high
    memory for MGMT pod and # higher/matching memory than
    specified below for NGFW pod. #PAN_NGFW_MEMORY="6Gi"
    #PAN_NGFW_MEMORY="40Gi" # For enabling faster datapath -
    AF_XDP, default is AF_PACKETV2.This requires kernel support.
    #PAN_DATA_MODE: "next-gen" #HPA params #PAN_CLOUD:"EKS"
   ```

```
#PAN_NAMESPACE_EKS:"EKSNamespace" #PUSH_INTERVAL:"15" #time
interval to publish metrics to AWS cloudwatch
```

範例 pan-cn-mgmt.yaml

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

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

3. 使用 Kubectl 來執行 yaml 檔案。

   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

   只有在您先前尚未完成建立叢集驗證的服務帳戶時，才必須執行 pan-mgmt-serviceaccount.yaml。

4. 驗證 CN-MGMT Pod 已啟動。

   這需要大約 5-6 分鐘。

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

STEP 8 | 部署 CN-NGFW Pod。

1. 驗證您已修改 PAN-CN-NGFW-CONFIGMAP 和 PAN-CN-NGFW 中詳述的 YAML 檔案。

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

2. 使用 Kubectl apply 來執行 pan-cn-ngfw-configmap.yaml。

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

3. 使用 Kubectl apply 來執行 pan-cn-ngfw.yaml。

   kubectl apply -f pan-cn-ngfw.yaml

4. 驗證 CN-NGFW Pod 正在執行。

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

STEP 9 | 在 CN-Series 上啟用水平 Pod 自動調整規模。

STEP 10 | 驗證您可以在 Kubernetes 叢集上看到 CN-MGMT、CN-NGFW 和 PAN-CNI。

kubectl -n kube-system get pods

STEP 11 | 標註應用程式 yaml 或命名空間，讓來自其新 Pod 的流量重新導向至防火牆。

您需要新增下列註釋，以將流量重新導向至 CN-NGFW 來進行檢查：

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

例如，對於「default」命名空間中的所有新 Pod：

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

在部分平台上，*pan-cni* 在 *CNI* 外掛程式鏈中未作用時，可以啟動應用程式 *Pod*。若要避免這類情況，您必須在應用程式 *Pod YAML* 中指定這裡顯示的磁碟區。

```
volumes: - name: pan-cni-ready hostPath: path: /var/log/
pan-appinfo/pan-cni-ready type:Directory
```

STEP 12 | （選用）特定流量可以根據 *PortInfo* 自訂資源來略過防火牆：

1. 套用 PortInfo 自訂資源定義 YAML

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

2. 使用 pan-cn-ngfw-port-cr.yaml 作為範例，以使用您想要略過的通訊協定協議和連接埠來建立 PortInfo 自訂資源。這從應用程式 Pod 觀點僅位於輸出方向，並支援 TCP 和 UDP，且最多 10 個個別連接埠（無連接埠範圍）。

   ```
   apiVersion: "paloaltonetworks.com/v1" kind:PortInfo metadata:
   name: "bypassfirewall" namespace: kube-system spec:
   portinfo:"TCP:8080,TCP:8081"
   ```

3. 套用您的 PortInfo 自訂資源 YAML。

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

4. 除了 pan-fw 註釋之外，還需要加上應用程式 Pod 的註釋。註釋應該出現在應用程式 Pod 啟動的那一刻。

   ```
   annotations: paloaltonetworks.com/firewall: pan-fw
   paloaltonetworks.com/bypassfirewall: kube-system/
  bypassfirewall
   ```

STEP 13 | 在叢集中部署應用程式。

在 CN-Series 上啟用水平 Pod 自動調整規模

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 圖表的 CN-Series 部署

水平 Pod 自動調整規模器 (HPA) 是所有雲端環境中可用的 Kubernetes 資源，可根據受監控度量，自動調整部署中的 CN-MGMT 和 CN-NGFW Pod 數目。HPA 會跨所有雲端環境使用兩個標準度量（CPU 和記憶體使用率）以及每個雲端環境特有的自訂度量。因此，每個雲端都需要特定的 yaml 檔案，才能在 AKS、EKS 和 GKE 中啟用 HPA。

HPA 使用雲端特定度量介面卡以從雲端環境的監控介面卡（例如 EKS 中的 CloudWatch）擷取度量資料，根據您定義的臨界值來決定何時擴充或縮減。您必須修改必要的 yaml 檔案，以設定最小和最大複本數目、每個度量的臨界值，以及自動調整防火牆時所使用的度量。

在 *PAN OS 10.1* 中，如果您使用 *CN-MGMT Pod HPA* 調整規模，則可以擴展許多 *CN-MGMT Pod* 而不連接到 *DP Pod*。建議盡量建立 *CN-MGMT Pod* 複本，以防止不必要的調整規模。

雲端環境	度量		平均值
AKS、EKS 和 GKE	CN-MGMT	panloggingrate	日誌計數
		pandataplaneslots	資料平面插槽計數
	CN-NFGW	dataplanecpuutilizationpct	CN-NGFW CPU 使用率的百分比
		dataplanepacketbufferutilization	CN-NGFW 封包緩衝區使用率的百分比
		pansessionactive	CN-NGFW 上的作用中工作階段數目
		pansessionutilization	工作階段使用率的百分比
		pansessionsslproxyutilization	工作階段 SSL Proxy 使用率的百分比
		panthroughput	輸送量 (kbps)

雲端環境	度量		平均值
		panpacketrate	封包速率（每秒封包數 (pps)）
		panconnectionspersecond	每秒連線數

在下面的範例中，是用於 EKS 的 pan-cn-hpa-dp.yaml 檔案。此範例使用資料平面 CPU 使用率百分比來自動調整 CN-NGFW Pod。在 25% 的情況下，將會擴充叢集。如果 CPU 使用率達到 50%，則叢集將會部署一個額外的 Pod。如果 CPU 使用率已達到 75%，則叢集將會部署兩個額外的 Pod。決定的方法是將度量總計除以度量臨界值，然後部署足夠的 Pod，將度量降低到叢集中所有 CN-NGFW Pod 的已設定臨界值。不過，叢集將不會部署超過 maxReplicas 的 CN-NGFW Pod。如果同時有多個度量超過臨界值，則叢集將會部署所需的 Pod 數目以處理較高的度量。

HPA 介面卡預設會每 15 秒輪詢度量介面卡一次。如果您指定的度量超過設定的臨界值 60 秒，則叢集將會部署額外的 CN-NGFW Pod。接著，叢集會先等待 300 秒（五分鐘），再決定是否需要額外的 CN-NGFW Pod。預設一次會部署一個 Pod。接著，叢集會在 300 秒後檢查度量（在此情況下為 CPU 使用率）。如果使用率下降到不再需要 Pod 的層級，則叢集將會刪除 Pod。然後，叢集將會再等待 60 秒，再決定是否可以移除另一個 Pod。

您可以修改下面顯示的所有值以及任何度量的值，使其適合您的部署。

```
kind:HorizontalPodAutoscaler apiVersion: autoscaling/v2beta2
 metadata: name: hpa-dp-eks namespace: kube-system spec:
 scaleTargetRef: apiVersion: apps/v1beta1 kind:Deployment name:
 pan-ngfw-dep minReplicas:1 maxReplicas:10 behavior: scaleDown:
 stabilizationWindowSeconds:300 policies: - type:Pods value:1
 periodSeconds:60 - type:Percent value:1 periodSeconds:60
 selectPolicy:Max scaleUp: stabilizationWindowSeconds:60 policies: -
 type:Pods value:1 periodSeconds:300 # assuming 5 mins for dp to be
 ready - type:Percent value:1 periodSeconds:300 # assuming 5 mins for
 dp to be ready selectPolicy:Max metrics: - type:External external:
 metric: name: dataplaneCpuUtilizationPct target: type:Value value:25
```

AKS

STEP 1 | 在叢集中部署 Azure Application Insights 執行個體。您必須提供必要的 Azure Application Insights 儀表金鑰和 Azure Application Insight APP ID API 金鑰作為 K8s 密碼。

STEP 2 | 從 Palo Alto Networks GitHub 儲存庫下載 AKS 特有 HPA yaml 檔案。

STEP 3 | 如果您的 CN-MGMT 部署在自訂命名空間中，則請使用自訂命名空間來更新 pan-cn-adapater.yaml。預設命名空間是 **kube-system**。

STEP 4 | 如果您尚未這麼做，則請更新 AKS 特有 **pan-cn-mgmt-configmap.yaml** 中的 HPA 參數。

```
 #PAN_CLOUD:"AKS" #HPA_NAME: "<name>" #unique name to identify hpa
resource per namespace or per tenant #PAN_INSTRUMENTATION_KEY:
"<>" #Azure APP Insight Instrumentation Key #PUSH_INTERVAL:"15"
#time interval to publish metrics to azure app insight
```

STEP 5 | 編輯 **pan-cn-hpa-secret.yaml**。

```
appinsights-appid: "<Azure App Insight Application ID obtained
 from API Access>" appinsights-key: "<Azure App Insight API Key
 created under API Access>" azure-client-id: "<Azure SP APP ID
 associated with corresponding resource group with monitoring
 reader access>" azure-client-secret: "<Azure SP Password
 associated with corresponding resource group with monitoring
 reader access>" azure-tenant-id: "<Azure SP tenant ID associated
 with corresponding resource group with monitoring reader access>"
```

STEP 6 | 將您上面建立的 HPA 名稱新增至 **pan-cn-custommetrics.yaml** 中的適當位置。

STEP 7 | 修改 **pan-cn-hpa-dp.yaml** 和 **pan-cn-hpa-mp.yaml**。

1. 輸入最小和最大複本數目。
2. （選用）變更縮減和擴充頻率值，以符合您的部署。如果您未變更這些值，則會使用預設值。
3. 針對您要用於調整規模的每個度量，複製下列區段。

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

4. 變更您要使用之度量的名稱，並將 **averageValue** 設定為上表所述的臨界值。如果您未變更這些值，則會使用預設值。
5. 儲存變更。

STEP 8 | 部署 HPA yaml 檔案。檔案必須依下面所述的順序進行部署。

1. 使用 Kubectl 來執行 pan-cn-hpa-secret.yaml

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

2. 使用 Kubectl 來執行 pan-cn-adapter.yaml

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

3. 使用 Kubectl 來執行 pan-cn-custommetrics.yaml

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

4. 使用 Kubectl 來執行 pan-cn-hpa-dp.yaml

   ```
   kubectl apply -f pan-cn-hpa-dp.yaml
   ```

5. 使用 Kubectl 來執行 pan-cn-hpa-mp.yaml

   ```
   kubectl apply -f pan-cn-hpa-mp.yaml
   ```

STEP 9 | 驗證您的部署。

- 使用 kubectl 來確認自訂度量命名空間中的自訂度量介面卡 Pod。

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

- 使用 kubectl 檢查 HPA 資源。

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

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

EKS

STEP 1 | 在「CN-Series 作為服務」叢集中，部署 Kubernetes 的 Amazon CloudWatch Metrics Adapter。您必須允許 CloudWatch 完整存取與 Kubernetes Pod 和叢集相關聯的兩個 IAM 角色。若要將自訂度量發佈至 CloudWatch，工作節點的角色必須要有 AWS 受管理政策 **CloudWatchAgentServerPolicy**，HPA 才能對其進行擷取。

STEP 2 | 從 Palo Alto Networks GitHub 儲存庫，下載 EKS 特有 HPA yaml 檔案。

STEP 3 | 如果您的 CN-MGMT 部署在自訂命名空間中，則請使用自訂命名空間來更新 pan-cn-adapater.yaml。預設命名空間是 **kube-system**。

STEP 4 | 修改 **pan-cn-hpa-dp.yaml** 和 **pan-cn-hpa-mp.yaml**。

1. 輸入最小和最大複本數目。
2. （選用）變更縮減和擴充頻率值，以符合您的部署。如果您未變更這些值，則會使用預設值。
3. 針對您要用於調整規模的每個度量，複製下列區段。

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

4. 變更您要使用之度量的名稱，並將 **averageValue** 設定為上表所述的臨界值。如果您未變更這些值，則會使用預設值。
5. 儲存變更。

STEP 5 | 部署 HPA yaml 檔案。檔案必須依下面所述的順序進行部署。

1. 使用 Kubectl 來執行 pan-cn-adapter.yaml

   kubectl apply -f pan-cn-adapter.yaml
2. 使用 Kubectl 來執行 pan-cn-externalmetrics.yaml

   kubectl apply -f pan-cn-externalmetrics.yaml
3. 使用 Kubectl 來執行 pan-cn-hpa-dp.yaml

   kubectl apply -f pan-cn-hpa-dp.yaml
4. 使用 Kubectl 來執行 pan-cn-hpa-mp.yaml

   kubectl apply -f pan-cn-hpa-mp.yaml

STEP 6 | 驗證您的部署。

- 使用 kubectl 來確認自訂度量命名空間中的自訂度量介面卡 Pod。

  kubectl get pods -n custom-metrics
- 使用 kubectl 檢查 HPA 資源。

  kubectl get hpa -n kube-system

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

將 CN-Series 防火牆部署為 DaemonSet

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 圖表的 CN-Series 部署

完成下列程序，以將 CN-Series 防火牆部署為 Daemonset。

開始之前，請確保 CN-Series YAML 檔案版本與 PAN-OS 版本相容。

- PAN-OS 10.1.2 或更新版本需要 YAML 2.0.2
- PAN-OS 10.1.0 和 10.1.1 需要 YAML 2.0.0 或 2.0.1

STEP 1 | 設定 Kubernetes 叢集。

1. 請驗證叢集具有足夠的版本。確保該叢集具有 CN-Series 先決條件資源以支援防火牆：

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

   檢視命令輸出之「容量」標題下的資訊，以查看所指定節點上可用的 CPU 和記憶體。

   CPU、記憶體和磁碟儲存體配置將取決於您的需求。請參閱 CN-Series 效能和調整規模。

   確保您具有下列資訊：

   - 收集「端點 IP 位址」，以在 Panorama 上設定 API 伺服器。Panorama 使用此 IP 位址來連線至 Kubernetes 叢集。
   - 從 Panorama 收集範本堆疊名稱、裝置群組名稱、Panorama IP 位址和日誌收集器群組名稱（選用）。
   - 收集授權碼以及自動註冊 PIN ID 和值。
   - 將映像檔下載至其中的容器映像檔儲存庫位置。

STEP 2 | （選用）如果您已在 Panorama 的 Kubernetes 外掛程式中設定自訂憑證，則必須執行下列命令來建立憑證密碼。請不要從 ca.crt 變更檔案名稱。pan-cn-mgmt.yaml 和 pan-cn-ngfw.yaml 中的自定憑證數量是選用項目。

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

STEP 3 | 編輯 YAML 檔案，以提供部署 CN-Series 防火牆所需的詳細資料。

您需要取代 YAML 檔案中的映像檔路徑以包括私人登錄的路徑，以及提供必要參數。如需詳細資料，請參閱 CN-Series 部署 YAML 檔案中的可編輯參數。

STEP 4 | 部署 CNI DaemonSet。

CNI 容器部署為 DaemonSet（一個節點一個 Pod），而且它會在節點上所部署之每個應用程式的 CN-NGFW Pod 上建立兩個執行個體。當您使用 kubectl 命令來執行 pan-cni YAML 檔案時，它會變成每個節點上服務鏈的一部分。

1. 請驗證您已使用 pan-cni-serviceaccount.yaml 來建立服務帳戶。

   請參閱建立叢集驗證的服務帳戶。

2. 使用 Kubectl 來執行 pan-cni-configmap.yaml。

   kubectl apply -f pan-cni-configmap.yaml

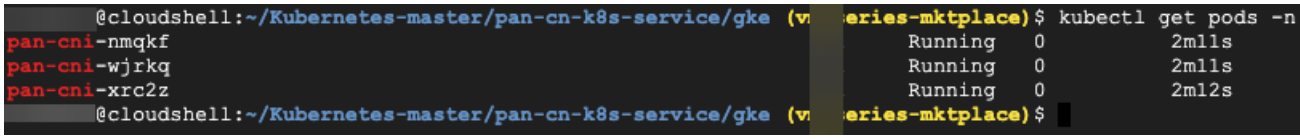
3. 使用 Kubectl 來執行 pan-cni.yaml。

   kubectl apply -f pan-cni.yaml

4. 請驗證您已修改 pan-cni-configmap 和 pan-cni YAML 檔案。

5. 執行下列命令，並確認您的輸出與下列範例相似。

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



STEP 5 | （僅限 AWS Outpost 上 EKS 的 CN-Series）更新儲存類別。若要支援在 AWS Outpost 上部署的 CN-Series，您必須使用儲存驅動程式 aws-ebs-csi-driver，確保 Outpost 在建立動態持續性磁碟區 (PV) 期間從 Outpost 拉出磁碟區。

1. 套用下列 yaml。

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

2. 驗證 ebs-sc 控制器是否正在執行。

   kubectl -n kube-system get pods

3. 更新 pan-cn-storage-class.yaml 以符合下面的範例。

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

4. 將 **storageClassName: ebs-sc** 新增至下面所顯示位置中的 pan-cn-mgmt.yaml。

```
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
# change this to 200Gi while using storageClassName
for better disk iops - metadata: name: varlogpan spec:
#storageClassName: pan-cn-storage-class //For better disk
iops performance for dp logs accessModes: [ "ReadWriteOnce" ]
storageClassName: ebs-sc resources: requests: storage:20Gi #
change this to 200Gi while using storageClassName for better
disk iops - 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 | 部署 CN-MGMT StatefulSet。

管理平面預設會部署為可提供容錯的 StatefulSet。最多可以將 30 個防火牆 CN-NGFW Pod 連線至 CN-MGMT StatefulSet。

1. （僅為靜態佈建 PV 的必要項目）部署 CN-MGMT StatefulSet 的「永久性磁碟區 (PV)」。

   1. 建立目錄，以符合 pan-cn-pv-local.yaml 中所定義的本機磁碟區名稱。

      您需要至少 2 個背景工作節點上有六 (6) 個目錄。請登入將部署 CN-MGMT StatefulSet 以建立目錄的每個背景工作節點。例如，若要建立名為 /mnt/pan-local1 到 /mnt/pan-local6 的目錄，請使用命令：

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

   2. 修改 pan-cn-pv-local.yaml。

      符合 `nodeaffinity` 下方的主機名稱，並驗證您已修改上面您在 `spec.local.path` 中建立的目錄，然後部署檔案來建立新的 storageclass pan-local-storage 和本機 PV。

2. 驗證您已修改 pan-cn-mgmt-configmap 和 pan-cn-mgmt YAML 檔案。

   EKS 中的範例 pan-cn-mgmt-configmap。

   ```
   Session Contents Restored 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: "x.y.z.a"
   PAN_DEVICE_GROUP: "dg-1" PAN_TEMPLATE_STACK: "temp-stack-1"
   PAN_CGNAME:"CG-EKS" # Intended License Bundle type - "CN-
  X-BASIC", "CN-X-BND1", "CN-X-BND2" # based on the authcode
   applied on the Panorama K8S plugin" PAN_BUNDLE_TYPE:"CN-X-
  BND2" #Non-mandatory parameters # Recommended to have same
   name as the cluster name provided in Panorama Kubernetes
   ```

```
 plugin - helps with easier identification of pods if managing
 multiple clusters with same Panorama #CLUSTER_NAME:"Cluster-
name" #PAN_PANORAMA_IP2: "passive-secondary-ip" # Comment
 out to use CERTs otherwise bypass encrypted connection to
 etcd in pan-mgmt. # Not using CERTs for etcd due to EKS bug
 ETCD_CERT_BYPASS: "" # No value needed # Comment out to use
 CERTs otherwise PSK for IPSec between pan-mgmt and pan-ngfw #
 IPSEC_CERT_BYPASS: "" # No values needed
```

範例 pan-cn-mgmt.yaml

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

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

3. 使用 Kubectl 來執行 yaml 檔案。

   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

   只有在您先前尚未完成建立叢集驗證的服務帳戶時，才必須執行 pan-mgmt-serviceaccount.yaml。

4. 驗證 CN-MGMT Pod 已啟動。

   這需要大約 5-6 分鐘。

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

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

STEP 7 | 部署 CN-NGFW Pod。

防火牆資料平面 CN-NGFW Pod 預設會部署為 DaemonSet。CN-NFGW Pod 執行個體可以保護節點上最多 30 個應用程式 Pod 的流量。

1. 驗證您已修改 PAN-CN-NGFW-CONFIGMAP 和 PAN-CN-NGFW 中詳述的 YAML 檔案。

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

2. 使用 Kubectl apply 來執行 pan-cn-ngfw-configmap.yaml。

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

3. 使用 Kubectl apply 來執行 pan-cn-ngfw.yaml。

   kubectl apply -f pan-cn-ngfw.yaml

4. 驗證所有 CN-NGFW Pod 都正在執行（叢集中一個節點會有一個 Pod）。

   這是 4 節點內部部署叢集的範例輸出。

   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-qsrm6 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 | 驗證您可以在 Kubernetes 叢集上看到 CN-MGMT、CN-NGFW 和 PAN-CNI。

kubectl -n kube-system get pods

```
pan-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-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 0 27hpan-ngfw-ds-qsrm6 1/1 Running 0 27hpan-ngfw-ds-vqk7z 1/1 Running 0 27hpan-ngfw-ds-zncqg 1/1 Running 0 27h
```

STEP 9 | 標註應用程式 yaml 或命名空間，讓來自其新 Pod 的流量重新導向至防火牆。

您需要新增下列註釋，以將流量重新導向至 CN-NGFW 來進行檢查：

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

例如，對於「default」命名空間中的所有新 Pod：

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

在部分平台上，*pan-cni* 在 *CNI* 外掛程式鏈中未作用時，可以啟動應用程式 *Pod*。若要避免這類情況，您必須在應用程式 *Pod YAML* 中指定這裡顯示的磁碟區。

```
volumes: - name: pan-cni-ready hostPath: path: /var/log/
pan-appinfo/pan-cni-ready type:Directory
```

STEP 10 | 在叢集中部署應用程式。

部署 CN-Series 防火牆作為 Kubernetes CNF

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 圖表的 CN-Series 部署

您現在可以在 Kubernetes 環境中將 CN-Series 部署為容器網路函數 (CNF)。

「CN-Series 作為精靈集」和「CN-Series 作為 kubernetes 服務」部署模式提供自動化安全性部署，並利用 Kubernetes 的自動調整規模功能。不過，這些部署模式的插入選項有限，並且不支援 I/O 加速。此外，其還會限制應用程式 Pod 的可實現輸送量，而 Pod 需要檢查和使用多個網路介面。

部署「CN-series 作為 Kubernetes-CNF」可解決透過雲端提供者的原生路由、vRouters 和機架頂部 (TOR) 交換器這類外部實體來使用服務函數鏈結 (SFC) 的流量的這些挑戰。「CN-series 作為 Kubernetes-CNF」部署模式不會影響應用程式 Pod。

完成下列程序，以部署「CN-series 作為 kubernetes-CNF」。

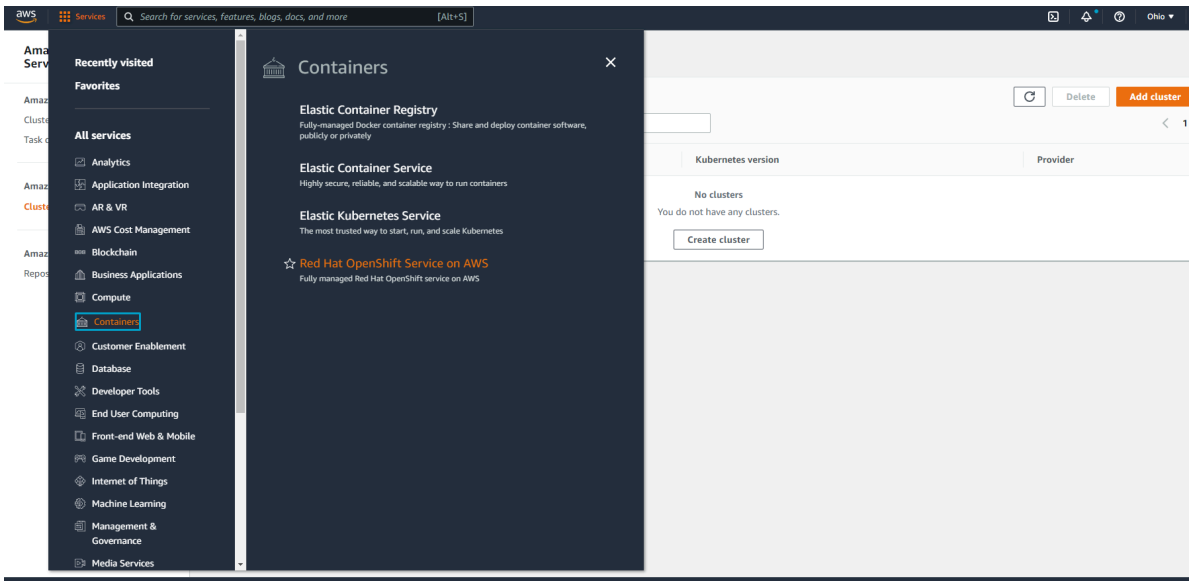
開始之前，請確保 CN-Series YAML 檔案版本與 PAN-OS 版本相容：

PAN-OS 10.2.0 或更新版本需要 YAML 3.0.0

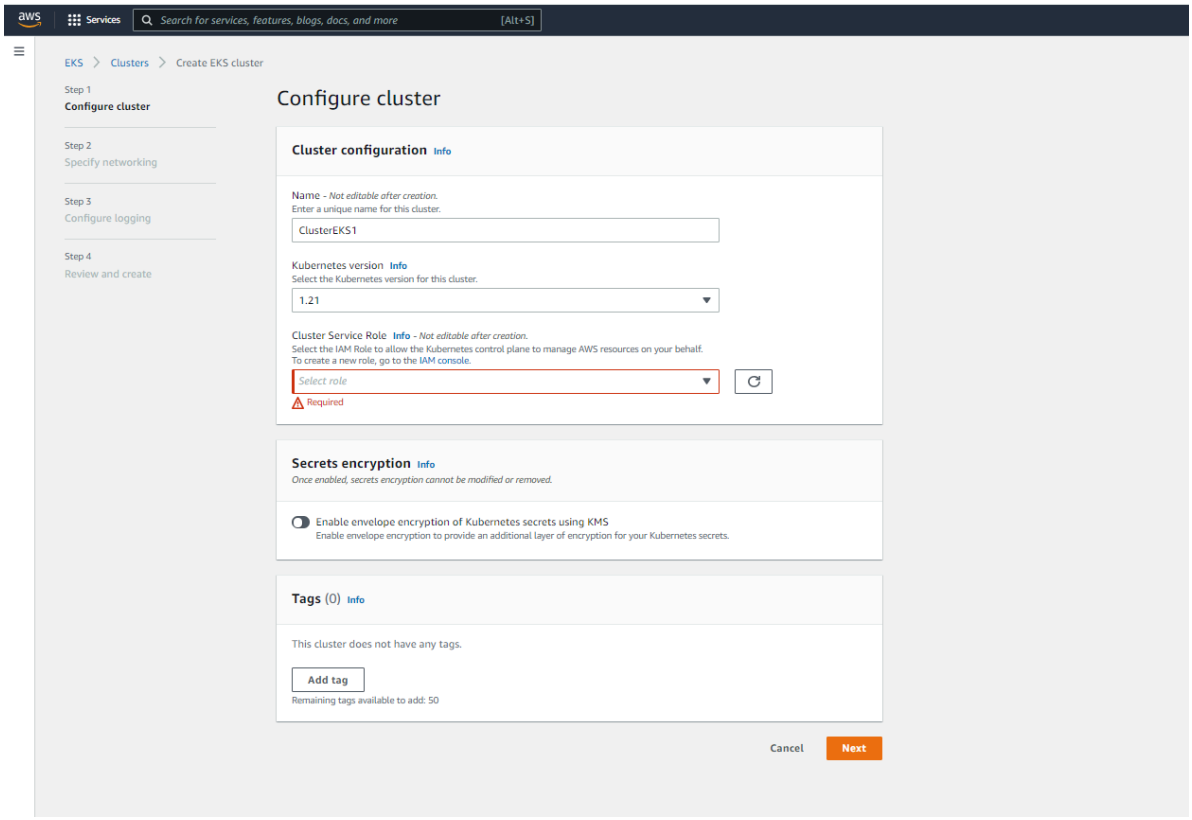
STEP 1 | 設定 Kubernetes 叢集。如需詳細資訊，請參閱建立 Amazon EKS 叢集以及 Pod 的多個網路介面。

若要在 AWS EKS 中建立叢集，請執行下列動作：

1. 按一下 **Services**（服務）導覽功能表，然後移至 **Containers**（容器）->**Elastic Kubernetes Service**（彈性 **Kubernetes** 服務）。



2. 按一下 **Create Cluster**（建立叢集）。
3. 填寫所需的詳細資訊，然後按一下 **Create**（建立）。



1. 請驗證叢集具有足夠的版本。確保該叢集具有 CN-Series 先決條件資源以支援防火牆：

   kubectl get nodes

   kubectl describe node <node-name>

   檢視命令輸出之「容量」標題下的資訊，以查看所指定節點上可用的 CPU 和記憶體。

   CPU、記憶體和磁碟儲存體配置將取決於您的需求。請參閱 CN-Series 效能和調整規模。

   確保您具有下列資訊：

   - 收集「端點 IP 位址」，以在 Panorama 上設定 API 伺服器。Panorama 使用此 IP 位址來連線至 Kubernetes 叢集。
   - 從 Panorama 收集範本堆疊名稱、裝置群組名稱、Panorama IP 位址和日誌收集器群組名稱（選用）。
   - 收集授權碼以及自動註冊 PIN ID 和值。
   - 將映像檔下載至其中的容器映像檔儲存庫位置。

STEP 2 | （選用）如果您已在 Panorama 的 Kubernetes 外掛程式中設定自訂憑證，則必須執行下列命令來建立憑證密碼。請不要從 ca.crt 變更檔案名稱。pan-cn-mgmt-0.yaml、pan-cn-mgmt-1.yaml、pan-cn-ngfw-0.yaml 和 pan-cn-ngfw.yaml-1 中的自訂憑證數量是選用項目。

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

STEP 3 | 編輯 YAML 檔案，以提供部署 CN-Series 防火牆所需的詳細資料。

您應該取代 YAML 檔案中的映像路徑以包括私人登錄的路徑，以及提供必要參數。如需詳細資料，請參閱 CN-Series 部署 YAML 檔案中的可編輯參數。

HA 中的 CN-Series-as-a-kubernetes-CNF 僅支援具有工作階段和設定同步的主動/被動 HA。

當您在 HA 中部署 CN-Series-as-a-kubernetes-CNF 時，將會有兩個 PAN-CN-MGMT-CONFIGMAP、PAN-CN-MGMT 和 PAN-CN-NGFW YAML 檔案各用於主動和被動節點，如下所示：

- pan-cn-mgmt-0.yaml
- pan-cn-mgmt-1.yaml
- pan-cn-mgmt-configmap-0.yaml
- pan-cn-mgmt-configmap-1.yaml
- pan-cn-ngfw-configmap-0.yaml
- pan-cn-ngfw-configmap-1.yaml

下列預設值定義於 pan-cn-mgmt-configmap-0.yaml 和 pan-cn-mgmt-configmap-1.yaml 檔案中。

pan-cn-mgmt-configmap-0.yaml:

```
metadata:
```

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

```
namespace: kube-system
```

```
data:
```

```
PAN_SERVICE_NAME: pan-mgmt-svc-0
```

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

pan-cn-mgmt-configmap-1.yaml:

```
metadata:
```

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

```
namespace: kube-system
```

```
data:
```

```
PAN_SERVICE_NAME: pan-mgmt-svc-1
```

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

您可以新增用於 CPU 釘選的 numa 選項。在 pan-cn-ngfw-configmap-0.yaml 和 pan-cn-ngfw-configmap-1.yaml 檔案中，新增 `PAN_NUMA_ENABLED` 參數的單一 numa 節點號碼。

若要在具有第 3 層支援的 HA 中成功部署 CN-Series-as-a-kubernetes-CNF，請執行下列動作：

- 在 HA 中，每個 Kubernetes 節點都至少應該有三個介面：管理（預設）、HA2 和資料介面。
- 針對 L3 模式的 CN-Series 防火牆，至少應該有兩個介面：管理（預設）和資料介面。
- 修改新的網路連接定義 YAML 檔案，並進行下列變更：
  - 在工作節點上，執行下列命令，以從超管理器介面擷取 **`pciBusID`** 值：

    ```
    lspci | grep -i ether
    ```

    例如：

    ```
    00:05.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
    ```

    ```
    00:06.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
    ```

    ```
    00:07.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
    ```

    ```
    00:08.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
    ```

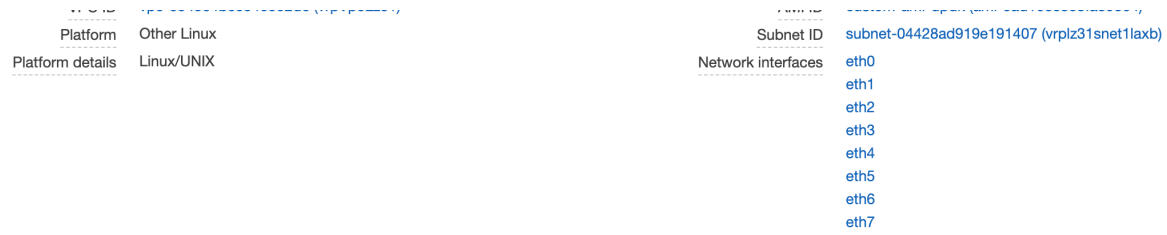
    ```
    00:09.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
    ```

    ```
    00:0a.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
    ```

    ```
    00:0b.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
    ```

    ```
    00:0c.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
    ```

    PCI 排序與 AWS EC2 UI 上所顯示的 eth 介面排序相同



將上面所擷取的 **pciBusID** 值新增至下列網路定義檔案：

```
net-attach-def-1.yaml
```

```
net-attach-def-2.yaml
```

```
net-attach-def-3.yaml
```

```
net-attach-def-ha2-0.yaml
```

```
net-attach-def-ha2-1.yaml
```

- 從 AWS 主控台的對應節點執行個體中擷取 HA2 介面的靜態 IP 位址，並將其新增至 `net-attach-def-ha2-0.yaml` 和 `net-attach-def-ha2-1.yaml` 檔案的 `address` 參數。

STEP 4 | 部署 CN-MGMT StatefulSet。

管理平面預設會部署為可提供容錯的 StatefulSet。只能將一個防火牆 CN-NGFW Pod 連線至 CN-MGMT StatefulSet。

1. （僅為靜態佈建 PV 的必要項目）部署 CN-MGMT StatefulSet 的「永久性磁碟區 (PV)」。
   1. 建立目錄，以符合 pan-cn-pv-local.yaml 中所定義的本機磁碟區名稱。

      您需要至少 2 個背景工作節點上有六 (6) 個目錄。請登入將部署 CN-MGMT StatefulSet 以建立目錄的每個背景工作節點。例如，若要建立名為 /mnt/pan-local1 到 /mnt/pan-local6 的目錄，請使用命令：

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

   2. 修改 pan-cn-pv-local.yaml。

      符合 `nodeaffinity` 下的主機名稱，並驗證您已在 `spec.local.path` 中修改上面所建立的目錄，然後部署檔案來建立新的 storageclass pan-local-storage 和本機 PV。
2. 驗證您已修改 pan-cn-mgmt-configmap 和 pan-cn-mgmt YAML 檔案。
3. 使用 Kubectl 來執行 yaml 檔案。

   kubectl apply -f pan-cn-mgmt-configmap-0.yaml

   kubectl apply -f pan-cn-mgmt-configmap-1.yaml

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

   kubectl apply -f pan-cn-mgmt-0.yaml

   kubectl apply -f pan-cn-mgmt-1.yaml

   只有在您先前尚未完成建立叢集驗證的服務帳戶時，才必須執行 pan-mgmt-serviceaccount.yaml。
4. 驗證 CN-MGMT Pod 已啟動。

   這需要大約 5-6 分鐘。

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

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

STEP 5 | 以 k8s-CNF 模式部署 CN-NGFW。

1. 驗證您是否已修改 YAML 檔案，如步驟 3 中所詳述。

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

   您應該確保已安裝 *multus daemonset*，並已建立網路連接定義檔案。*pan-cn-ngfw-configmap-0.yaml* 和 *pan-cn-ngfw-configmap-1.yaml* 檔案中的 *PAN_SERVICE_NAME* 參數值應該分別符合 *pan-cn-mgmt-0.yaml* 和 *pan-cn-mgmt-1.yaml* 檔案中的服務名稱參數值。

   針對 *HA* 支援，建議在不同的工作節點上部署 *DP Pod*。您可以從 ***yaml nodeSelector*** 欄位或開啟 *Pod* 反關聯性來確保這點。

   若要啟用 *HA* 支援，您應該確保下列 *YAML* 檔案中的 *PAN_HA_SUPPORT* 參數值為 ***true***：

   pan-cn-mgmt-configmap-0.yaml

   pan-cn-mgmt-configmap-1.yaml

   針對 *DP Pod* 的資料介面，應該視需要將 *CNI* 和介面資源新增至 *DP YAML* 檔案。例如：

   k8s.v1.cni.cncf.io/networks: net-attach-1,net-attach-2,net-attach-3

   若要啟用 *DPDK* 支援，您應該確保 *pan-cn-ngfw-configmap-0.yaml* 和 *pan-cn-ngfw-configmap-1.yaml* 檔案中的 *PAN_DATA_MODE* 參數值為 ***dpdk***。

   此外，*HUGEPAGE_MEMORY_REQUEST* 參數值應該符合 *pan-cn-ngfw-0.yaml* 和 *pan-cn-ngfw-1.yaml* 檔案中的巨型分頁記憶體要求。

   如需詳細資訊，請參閱 在 CN-Series 防火牆上設定 DPDK。

2. 使用 Kubectl apply 來執行 pan-cn-ngfw-configmap-0.yaml 和 pan-cn-ngfw-configmap-1.yaml。

   kubectl apply -f pan-cn-ngfw-configmap-0.yaml

```
kubectl apply -f pan-cn-ngfw-configmap-1.yaml
```

3. 使用 Kubectl apply 來執行 pan-cn-ngfw-0.yaml 和 pan-cn-ngfw-1.yaml。

   ```
   kubectl apply -f pan-cn-ngfw-0.yaml
   ```

   ```
   kubectl apply -f pan-cn-ngfw-1.yaml
   ```

4. 驗證 CN-NGFW Pod 正在執行。

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

STEP 6 | 部署 CN-NGFW Pod。請執行下列動作：

1. 驗證您是否已如 PAN-CN-NGFW-CONFIGMAP-0、PAN-CN-NGFW-CONFIGMAP-1、PAN-CN-NGFW-0 和 PAN-CN-NGFW-1 中所述修改 YAML 檔案。

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

2. 使用 Kubectl apply 來執行 pan-cn-ngfw-configmap.yaml。

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

3. 使用 Kubectl apply 來執行 pan-cn-ngfw.yaml。

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

4. 驗證 CN-NGFW Pod 正在執行。

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

STEP 7 | 驗證您是否可以在 Kubernetes 叢集上看到 CN-MGMT 和 CN-NGFW。執行下列命令：

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

以獨立模式部署 Kubernetes CNF L3

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 圖表的 CN-Series 部署

您可以在 Kubernetes 環境中以 L3 獨立模式部署 CN-Series 防火牆作為容器網路函數 (CNF)。

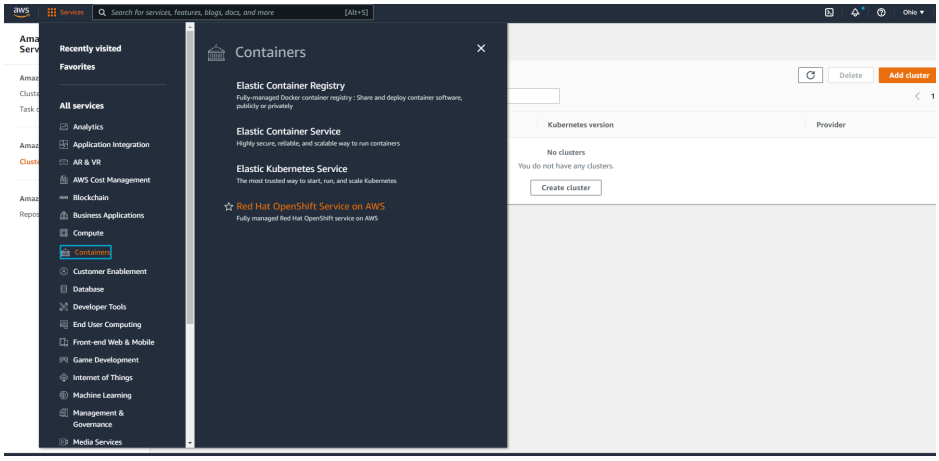
CN-Series 現在支援通過 vRouter 的流量，其中靜態路由設定為將流量重新導向至防火牆的資料平面介面。針對反向，透過 IPv4 IP 位址，使用 L3 政策型路由 (PBR) 將流量重新導定至相同防火牆。K8s 環境中介面的 IP 位址一般是透過 CNI 使用 DHCP 進行程式編碼。

若要以 L3 獨立模式部署 Kubernetes CNF，請執行下列動作：

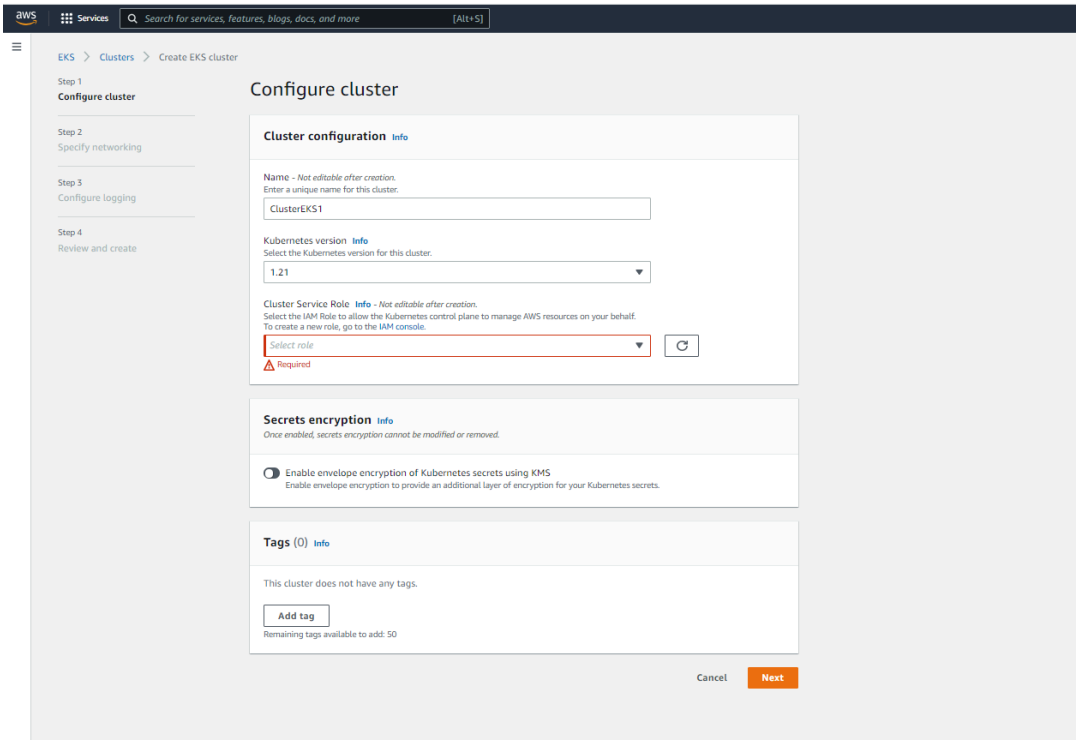
STEP 1 | 設定 Kubernetes 叢集。

若要在 AWS EKS 中建立叢集，請執行下列動作：

1. 按一下 **Services**（服務）導覽功能表，然後移至 **Containers**（容器）->**Elastic Kubernetes Service**（彈性 **Kubernetes** 服務）。



2. 按一下 **Create Cluster**（建立叢集）。
3. 填寫所需的詳細資訊，然後按一下 **Create**（建立）。



1. 請驗證叢集具有足夠的版本。確保該叢集具有 CN-Series 先決條件資源以支援防火牆：

   kubectl get nodes

   kubectl describe node <node-name>

   檢視命令輸出之「容量」標題下的資訊，以查看所指定節點上可用的 CPU 和記憶體。

CPU、記憶體和磁碟儲存體配置將取決於您的需求。請參閱 CN-Series 效能和調整規模

確保您具有下列資訊：

- 收集「端點 IP 位址」，以在 Panorama 上設定 API 伺服器。Panorama 使用此 IP 位址來連線至 Kubernetes 叢集。
- 從 Panorama 收集範本堆疊名稱、裝置群組名稱、Panorama IP 位址和日誌收集器群組名稱（選用）。
- 收集授權碼以及自動註冊 PIN ID 和值。
- 將映像檔下載至其中的容器映像檔儲存庫位置。

STEP 2 | 建立憑證密碼。（選用）如果您已在 Panorama 的 Kubernetes 外掛程式中設定自訂憑證，則必須執行下列命令來建立憑證密碼。請不要從 ca.crt 變更檔案名稱。pan-cn-mgmt-0.yaml 和 pan-cn-ngfw-0.yaml 中的自訂憑證數量是選用項目。

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

STEP 3 | 編輯 YAML 檔案，以提供部署 CN-Series 防火牆所需的詳細資料。

- pan-cn-mgmt-0.yaml
- pan-cn-mgmt-configmap-0.yaml
- pan-cn-ngfw-configmap-0.yaml

您應該取代 YAML 檔案中的映像路徑以包括私人登錄的路徑，以及提供必要參數。如需詳細資料，請參閱 CN-Series 部署 YAML 檔案中的可編輯參數。

下列預設值定義於 pan-cn-mgmt-configmap-0.yaml 檔案中。

pan-cn-mgmt-configmap-0.yaml:

```
metadata:
```

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

```
namespace: kube-systemdata
```

```
data:
```

```
PAN_SERVICE_NAME: pan-mgmt-svc-0
```

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

您可以新增用於 CPU 釘選的 numa 選項。在 pan-cn-ngfw-configmap-0.yaml 檔案中，新增 `PAN_NUMA_ENABLED` 參數的單一 numa 節點號碼。

若要在具有第 3 層支援的情況下成功部署 CN-Series-as-a-kubernetes-CNF，請執行下列動作：

- 每個 Kubernetes 節點都至少應該有三個介面：管理（預設）、HA2 連結和資料介面。
- 針對 L3 模式的 CN-Series 防火牆，至少應該有兩個介面：管理（預設）和資料介面。
- 修改新的網路連接定義 YAML 檔案，並進行下列變更：
  - 在工作節點上，執行下列命令，以從超管理器介面擷取 **`pciBusID`** 值：

    ```
    lspci | grep -i ether
    ```

例如：

```
00:05.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
```

```
00:06.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
```

```
00:07.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
```

```
00:08.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
```

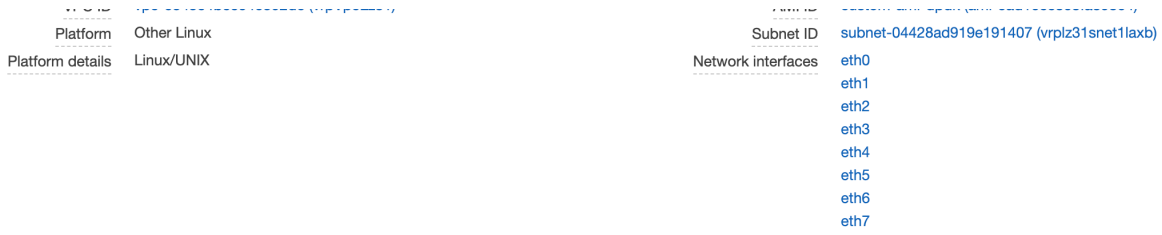
```
00:09.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
```

```
00:0a.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
```

```
00:0b.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
```

```
00:0c.0 乙太網路控制器：Amazon.com, Inc.彈性網路介面卡 (ENA)
```

PCI 排序與 AWS EC2 UI 上所顯示的 eth 介面排序相同



將上面所擷取的 **pciBusID** 值新增至下列網路定義檔案：

```
net-attach-def-1.yaml
```

```
net-attach-def-2.yaml
```

```
net-attach-def-3.yaml
```

STEP 4 | 部署 CN-MGMT StatefulSet。

管理平面預設會部署為可提供容錯的 StatefulSet。只能將一個防火牆 CN-NGFW Pod 連線至 CN-MGMT StatefulSet。

1. （僅為靜態佈建 PV 的必要項目）部署 CN-MGMT StatefulSet 的「永久性磁碟區 (PV)」。
   1. 建立目錄，以符合 pan-cn-pv-local.yaml 中所定義的本機磁碟區名稱。

      您需要至少 2 個背景工作節點上有六 (6) 個目錄。請登入將部署 CN-MGMT StatefulSet 以建立目錄的每個背景工作節點。例如，若要建立名為 /mnt/pan-local1 到 /mnt/pan-local6 的目錄，請使用命令：

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

   2. 修改 pan-cn-pv-local.yaml。

      符合 `nodeaffinity` 下方的主機名稱，並驗證您已修改上面您在 `spec.local.path` 中建立的目錄，然後部署檔案來建立新的 storageclass pan-local-storage 和本機 PV。
2. 驗證您已修改 pan-cn-mgmt-configmap 和 pan-cn-mgmt YAML 檔案。
3. 使用 Kubectl 來執行 yaml 檔案。

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

   kubectl apply -f pan-cn-mgmt-configmap-0.yaml

   kubectl apply -f $dir/pan-cn-mgmt-0.yaml

   kubectl apply -f $dir/net-attach-def-1.yaml

   kubectl apply -f $dir/net-attach-def-2.yaml

   kubectl apply -f $dir/pan-cn-mgmt-0.yaml

   kubectl apply -f $dir/pan-cn-ngfw-configmap-0.yaml

   kubectl apply -f $dir/pan-cn-ngfw-0.yaml

   只有在您先前尚未完成建立叢集驗證的服務帳戶時，才必須執行 pan-mgmt-serviceaccount.yaml。
4. 驗證 CN-MGMT Pod 已啟動。

   這需要大約 5-6 分鐘。

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

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

STEP 5 | 以 k8s-CNF 模式部署 CN-NGFW。

1. 驗證您是否已修改 YAML 檔案，如步驟 3 中所詳述。

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

   您應該確保已安裝 *multus daemonset*，並已建立網路連接定義檔案。*pan-cn-ngfw-configmap-0.yaml* 檔案中的 *PAN_SERVICE_NAME* 參數值應該符合 *pan-cn-mgmt-0.yaml* 檔案中的服務名稱參數值。

   針對 *CN-NFGW Pod* 的資料介面，應該視需要將 *CNI* 和介面資源新增至 *CN-NFGW YAML* 檔案。例如：

   k8s.v1.cni.cncf.io/networks: <interface-cni1>@eth1,<interface-cni2>@eth2

   若要啟用 *DPDK* 支援，您應該確保 *pan-cn-ngfw-configmap-0.yaml* 檔案中的 *PAN_DATA_MODE* 參數值為 ***dpdk***。

   此外，*HUGEPAGE_MEMORY_REQUEST* 參數值應該符合 *pan-cn-ngfw-0.yaml* 檔案中的巨型分頁記憶體要求。

   如需詳細資訊，請參閱 在 CN-Series 防火牆上設定 DPDK。

2. 使用 Kubectl apply 來執行 pan-cn-ngfw-configmap-0.yaml。

   kubectl apply -f pan-cn-ngfw-configmap-0.yaml

3. 使用 Kubectl apply 來執行 pan-cn-ngfw-0.yaml 和 pan-cn-ngfw-1.yaml。

   kubectl apply -f pan-cn-ngfw-0.yaml

4. 驗證 CN-NGFW Pod 正在執行。

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

STEP 6 | 部署 CN-NGFW Pod。請執行下列動作：

1. 驗證您已修改 PAN-CN-NGFW-CONFIGMAP-0 和 PAN-CN-NGFW-0 中詳述的 YAML 檔案。

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

2. 使用 Kubectl apply 來執行 pan-cn-ngfw-configmap.yaml。

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

3. 使用 Kubectl apply 來執行 pan-cn-ngfw.yaml。

   kubectl apply -f pan-cn-ngfw.yaml

4. 驗證 CN-NGFW Pod 正在執行。

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

STEP 7 | 驗證您是否可以在 Kubernetes 叢集上看到 CN-MGMT 和 CN-NGFW。執行下列命令：

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

```
root@master-1:~/CNv3-cnf/native# kubectl get pods -n kube-system
NAME                                      READY   STATUS    RESTARTS   AGE
calico-kube-controllers-694b4c9455-bxqbf      1/1     Running   4          246d
calico-node-fvr2c                             1/1     Running   23         246d
calico-node-js7v9                             1/1     Running   3          246d
calico-node-ssp9t                             1/1     Running   3          246d
coredns-dff8fc7d-87bnh                        1/1     Running   2          246d
coredns-dff8fc7d-l67mk                        1/1     Running   3          212d
dns-autoscaler-66498f5c5f-8kr4p               1/1     Running   2          246d
kube-apiserver-master-1                       1/1     Running   2          246d
kube-controller-manager-master-1              1/1     Running   2          246d
kube-multus-ds-5drrn                          1/1     Running   3          205d
kube-multus-ds-6vv4z                          1/1     Running   4          205d
kube-multus-ds-f6bhf                          1/1     Running   19         205d
kube-proxy-c4tth                              1/1     Running   2          246d
kube-proxy-fhtz9                              1/1     Running   2          246d
kube-proxy-gd5lj                              1/1     Running   21         246d
kube-scheduler-master-1                       1/1     Running   2          246d
kubernetes-dashboard-667c4c65f8-8wgtx         1/1     Running   4          246d
kubernetes-metrics-scraper-54fbb4d595-pp6qk   1/1     Running   2          246d
nginx-proxy-worker-1                          1/1     Running   27         246d
nginx-proxy-worker-2                          1/1     Running   2          246d
nodelocaldns-6nc4x                            1/1     Running   3          246d
nodelocaldns-d5s6g                            1/1     Running   4          246d
nodelocaldns-jcfzt                            1/1     Running   29         246d
pan-mgmt-sts-0-0                              1/1     Running   0          16m
pan-ngfw-dep-0-5ff468684f-2fnv6               1/1     Running   0          4m6s
root@master-1:~/CNv3-cnf/native# kubectl exec -it pan-mgmt-sts-0-0 -n kube-system -- bash
[root@pan-mgmt-sts-0-0 /]# ipsec status
Security Associations (1 up, 0 connecting):
       to-mp[2]: ESTABLISHED 3 minutes ago, 10.233.73.23[CN=pan-mgmt-svc-0.kube-system.svc]...10.233.73.24[CN=pan-fw.kube-system.svc]
       to-mp{1}:  INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: 20a5f62c_i abec4c31_o
       to-mp{1}:   0.0.0.0/0 === 169.254.202.2/32
[root@pan-mgmt-sts-0-0 /]# su admin

Warning: Your device is still configured with the default admin account credentials. Please change your password prior to deployment.
admin@pan-mgmt-sts-0-0> show jobs all

Enqueued              Dequeued            ID  PositionInQ                          Type                        Status Result Completed
------------------------------------------------------------------------------------------------------------------------------------
2022/02/25 10:41:22   10:41:30            5                                      Commit                        FIN     OK 10:42:16
2022/02/25 10:40:56   10:40:56            4                                    AutoCom                         FIN     OK 10:41:24
2022/02/25 10:32:47   10:32:47            3                                 CommitAll                          FIN     OK 10:33:24
2022/02/25 10:30:52   10:30:52            2                                    AutoCom                         FIN     OK 10:31:30

admin@pan-mgmt-sts-0-0> show panorama-status

Panorama Server 1 : 10.3.252.196
    Connected     : yes
    HA state      : Unknown
```

```
admin@pan-mgmt-sts-0-0> request plugins vm_series list-dp-pods

DP pods                          Licensed               License Type
-------                          ---------      ------------------------------------------------------------
pan-ngfw-dep-0-5ff468684f-2fnv6         yes             Threat Prevention, URL Filtering, Wildfire, DNS


admin@pan-mgmt-sts-0-0> debug show internal interface all

total configured hardware interfaces: 2

name                    id    speed/duplex/state            mac address
-------------------------------------------------------------------------------
ethernet1/1             16    10000/full/up                 00:0c:29:e7:ec:13
ethernet1/2             17    10000/full/up                 00:0c:29:e7:ec:3b

aggregation groups: 0


total configured logical interfaces: 2

name                id    vsys zone            forwarding              tag    address
------------------- ----- ---- --------------- ----------------------- ------ ------------------
ethernet1/1         16    1    trust           vr:vr1                  0      192.168.10.10/24
ethernet1/2         17    1    untrust         vr:vr1                  0      192.168.20.10/24
```

部署 CN-Series 防火牆

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

CN-Series 防火牆易於使用 Kubernetes 協調運作進行部署，以簡化網路安全性與持續整合/持續開發 (CI/CD) 流程的整合。CN-Series 防火牆的持續管理集中在 Panorama™ 網路安全性管理中，該管理主控台與所有 Palo Alto Networks 防火牆相同，為網路安全性團隊提供一個單一管理平台來管理組織的整體網路安全性動態。

本章包含以下部分：

CN-Series 部署檢查清單

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 圖表的 CN-Series 部署

若要部署 CN-Series 防火牆，您必須完成下列工作：

- ❑ 如果您尚未這樣做，請授權 CN-Series 防火牆—產生授權碼，並在準備部署 CN-Series 防火牆時將其放在隨手可即之處。
- ❑ 請檢閱 CN-Series 先決條件—開始部署之前，請確定您瞭解部署 CN-Series 防火牆所需的系統需求。
- ❑ 準備元件。
  - 在 Panorama 上產生 VM 驗證金鑰。
  - （選用）在 CN-Series 防火牆上安裝裝置憑證。
  - 建立叢集驗證的服務帳戶。
  - 部署 Panorama—您必須使用 Panorama 來設定、部署和管理 CN-Series 防火牆部署。如需部署和設定 Panorama 設備的詳細資訊，請參閱設定 Panorama。
  - 安裝適用於 CN-Series 的 Kubernetes 外掛程式。
  - 取得 CN-Series 部署的映像檔和檔案—存取 Palo Alto Network 儲存庫以下載 Docker 檔案，以及存取 GitHub 來取得在 Kubernetes 環境中部署 CN-Series 防火牆所需的 yaml 檔案。
- ❑ 部署 CN-Series 防火牆。
  - 編輯 HELM 圖表以適合您的部署—或者，您也可以編輯 yaml 檔案，然後先檢閱 CN-Series 部署 YAML 檔案中的可編輯參數，再部署 CN-Series 防火牆。必須修改 yaml 檔案中設定的許多參數，才能成功部署 CN-Series 防火牆。
  - 部署 CN-Series 防火牆作為 Kubernetes 服務（建議部署模式）.
  - 將 CN-Series 防火牆部署為 DaemonSet。
  - （選用）如果您要將 CN-Series 防火牆部署為 Kubernetes 服務，則可以在 CN-Series 上啟用水平 Pod 自動調整規模。水平 Pod 自動調整規模 (HPA) 允許您的 CN-Series 防火牆部署動態自動調整規模與您的 Kubernetes 環境。
  - 如果您要在 OpenShift 環境中部署 CN-Series，則請參閱在 OpenShift 部署 CN-Series 防火牆。
  - 如果您要使用 CN-Series 防火牆保護 5G 流量的安全，則請參閱 使用 CN-Series 防火牆保護 5G。

- 設定 Panorama 保護 Kubernetes 部署—在您已部署 CN-Series 防火牆後，請使用 Panorama 設定可啟用流量強制執行的安全性政策，並將這些政策推送至防火牆。

使用（建議）和不使用 Helm 圖表部署 CN-Series 防火牆

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

Helm 儲存庫包含圖表和範本，用於使用 Kubernetes 的 Helm Packet Manager 部署 Palo Alto Networks CN-series 容器化防火牆。

您可以從 GitHub 下載 CN-Series Helm 圖表。

- 準備使用 Helm 圖表和範本
- 使用 HELM 圖表（建議）部署 CN-Series 防火牆
- 透過 YAML 檔案部署 CN-Series 防火牆

準備使用 Helm 圖表和範本

安裝所需的軟體。這些指示列出最小版本，但除非指定上限，否則您可以在相同系列中安裝更新版本。

STEP 1 | 部署 CN-Series 防火牆 10.1.x、10.2.x、11.0.x 或 11.1.x 容器映像檔。

STEP 2 | 安裝 1.16 - 1.25 之間的 Kubernetes 版本，然後建立 Kubernetes 叢集。如需詳細瞭解您環境支援的 kubernetes 版本，請參閱 CN-Series 部署支援的環境。

STEP 3 | 將 Panorama 部署至可從 Kubernetes 叢集和您用來保護叢集之 CN-Series 防火牆存取的位置。

1. 確保 Panorama PAN-OS 版本是 10.x.x 或更新版本。
2. 安裝 Panorama 1.0.x 或 2.0.x 版的 Kubernetes 外掛程式。

STEP 4 | 安裝 Helm 用戶端 3.6.0 版或更新版本。

繼續使用 HELM 圖表（建議）部署 CN-Series 防火牆

或透過 YAML 檔案部署 CN-Series 防火牆。

使用 HELM 圖表（建議）部署 CN-Series 防火牆

使用此程序複製儲存庫，並從本機環境中部署。

STEP 1 | 在 Panorama 上產生 VM 驗證金鑰。

STEP 2 | 從 GitHub 複製儲存庫。

```
$ git clone https://github.com/PaloAltoNetworks/cn-series-helm.git
```

STEP 3 | 切換至所複製儲存庫的本機目錄。例如：

```
$ cd cn-series-helm
```

STEP 4 | 切換至部署的子目錄。

- 使用目錄 `helm_cnv1` 以將 CN-Series 部署為精靈集
- 使用目錄 `helm_cnv2` 以將 CN-Series 部署為服務。
- 使用目錄 `helm_cnv3` 以將 CN-Series 部署為 CNF。

STEP 5 | 下載 `plugin-serviceaccount.yaml` 的服務帳戶 YAML 並予以應用。服務帳戶會啟用 Panorama 向叢集進行驗證所需的權限來擷取 Kubernetes 標籤和資源資訊。此服務帳戶的名稱預設為 `pan-plugin-user`。執行以下命令以部署 `plugin-serviceaccount.yaml` 檔案：

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

```
kubectl -n kube-system get secrets | grep pan-plugin-user
```

檢視與此服務帳戶相關聯的祕密。

```
kubectl -n kube-system get secrets <secrets-from-above-command> -o
json >> cred.json
```

在此範例中，建立名稱為 `cred.json` 且包含祕密的認證檔案，並儲存此檔案。您需要將此檔案上傳至 Panorama，以設定用於監控為 CN-Series 防火牆安裝 Kubernetes 外掛程式中叢集的 Kubernetes 外掛程式。

在 ***Openshift*** 上，您必須為每個 *Openshift* 命名空間檔案手動部署 *`pan-cni-net-attach-def.yaml`* 才能部署 *Helm* 圖表。

STEP 6 | 編輯 `values.yaml` 檔案，以輸入您的設定資訊。以下值來自 `helm_cnv1` 子目錄。

```
# K8s 環境 # 有效的部署標籤為：[gke|eks|aks||native] # 有效的 multus 標
籤為：[enable|disable] 保持 multus 為 openshift 和原生部署啟用。叢
集：deployTo：eks multus：停用
```

```
# Panorama tags panorama: ip: "<Panorama-IP>" ip2: authKey:
 "<Panorama-auth-key>" deviceGroup: "<Panorama-device-group>"
 template: "<panorama-template-stack>" cgName: "<panorama-
collector-group>"
```

```
# MP container tags mp: initImage: gcr.io/pan-cn-series/
pan_cn_mgmt_init initVersion: latest image: gcr.io/pan-cn-series/
panos_cn_mgmt version:10.2.3 cpuLimit:4 # DP container tags
```

```
 dp: image: gcr.io/pan-cn-series/panos_cn_ngfw version:10.2.3
 cpuLimit:2 # CNI container tags cni: image: gcr.io/pan-cn-series/
pan_cni version: latest
```

STEP 7 | 檢視所呈現的 YAML 檔案。

```
helm install --debug --generate-name helm_cnv1/ --dry-run
```

STEP 8 | 對 helm 圖表執行 lint 檢查。

```
helm lint helm_cnv1/
```

STEP 9 | 部署 HELM 圖表。

```
helm install <deployment-name> helm_cnv1
```

解除安裝 *HELM* 圖表時不會刪除永久性磁碟區宣告。您必須事先清除這些宣告，以讓 *HELM* 正常安裝。

如需詳細瞭解 HELM，請參閱 HELM 經典：Kubernetes 套件管理員。

透過 YAML 檔案部署 CN-Series 防火牆

若要在不複製儲存庫的情況下進行部署，請將儲存庫新增至 Helm 用戶端。

STEP 1 | 在 Panorama 上產生 VM 驗證金鑰。

STEP 2 | 下載 `plugin-serviceaccount.yaml` 的服務帳戶 YAML 並予以應用。服務帳戶會啟用 Panorama 向叢集進行驗證所需的權限來擷取 Kubernetes 標籤和資源資訊。此服務帳戶的名稱預設為 `pan-plugin-user`。執行以下命令以部署 `plugin-serviceaccount.yaml` 檔案：

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

```
kubectl -n kube-system get secrets | grep pan-plugin-user
```

檢視與此服務帳戶相關聯的祕密。

```
kubectl -n kube-system get secrets <secrets-from-above-command> -o
json >> cred.json
```

在此範例中，建立名稱為 `cred.json` 且包含祕密的認證檔案，並儲存此檔案。您需要將此檔案上傳至 Panorama，以設定用於監控為 CN-Series 防火牆安裝 Kubernetes 外掛程式中叢集的 Kubernetes 外掛程式。

在 ***Openshift*** 上，您必須為每個 *Openshift* 命名空間檔案手動部署 *`pan-cni-net-attach-def.yaml`* 才能部署 *Helm* 圖表。

STEP 3 | 將 CN-Series 儲存庫新增至本機 Helm 用戶端。

將此命令輸入成一行：

```
$ helm repo add my-project https://paloaltonetworks.github.io/cn-series-helm
```

```
「cn-series」已新增至您的儲存庫
```

STEP 4 | 確認已將儲存庫新增至 Helm 用戶端。

```
$ helm search repo cn-series
```

STEP 5 | 選取 Kubernetes 叢集。

```
$ kubectl config set-cluster NAME
```

STEP 6 | 使用 Helm 圖表儲存庫部署。編輯下列命令以包括您的設定資訊。

```
$ helm install cn-series/cn-series --name="deployment name"
--set cluster.deployTo="gke|eks|aks|openshift"
--set panorama.ip="panorama hostname or ip"
--set panorama.ip2="panorama2 hostname or ip"
--set-string panorama.authKey="vm auth key"
--set panorama.deviceGroup="device group"
--set panorama.template="template stack"
--set panorama.cgName="collector group"
--set cni.image="container repo"
--set cni.version="container version"
--set mp.initImage="container repo"
--set mp.initVersion="container version"
--set mp.image="container repo"
--set mp.version="container version"
--set mp.cpuLimit="cpu max"
--set dp.image="container repo"
--set dp.version="container version"
--set dp.cpuLimit="cpu max"
```

解除安裝 *HELM* 圖表時不會刪除永久性磁碟區宣告。您必須事先清除這些宣告，以讓 *HELM* 正常安裝。

使用 Terraform 範本部署 CN-Series 防火牆

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Terraform 0.13.0 或更新版本

CN-Series 部署儲存庫包含部署 GKE、EKS 或 AKS 叢集的 Terraform 計劃。這些計劃可確保叢集節點大小調整和容器網路介面 (CNI) 支援叢集內的 CN-Series 防火牆部署。此儲存庫也提供 CN-Series 防火牆部署計劃以及可使用防火牆保護的範例 PHP Guestbook 應用程式。

此程序具有下列選用工作流程：

- 準備使用 Helm 圖表和範本
- 部署範例應用程式
- 使用 Terraform 部署 CN-Series 防火牆
- 設定 Panorama 的 Kubernetes 外掛程式

部署範例應用程式

Palo Alto Networks GitHub 儲存庫包括社群支援的範例應用程式以及名為 `guestbook.yml` 的 Kubernetes 資訊清單檔案。

此檔案會部署可使用 Redis 後端的簡單 PHP Guestbook Web 應用程式。

STEP 1 | 在 Palo Alto Networks GitHub 儲存庫 cn-series-deploy 目錄中，切換至 sample-application 目錄。

```
$ cd sample-application
```

STEP 2 | 部署 Guestbook 應用程式。

```
$ kubectl apply -f guestbook.yml
```

STEP 3 | 確認應用程式 Pod 已部署並達到 [Running（執行中）]，然後達到 [Ready（就緒）] 狀態。

```
$ kubectl get pods -n sample-app
```

```
NAME READY STATUS RESTARTS AGE frontend-69859f6796-96bs7
 1/1 Running 0 111m frontend-69859f6796-k2k4z 1/1 Running
 0 53m frontend-69859f6796-zwwbg 1/1 Running 0 111m redis-
master-596696dd4-5l5qv 1/1 Running 0 53m redis-slave-6bb9896d48-
dwhw2 1/1 Running 0 53m redis-slave-6bb9896d48-nhqzh 1/1 Running 0
 111m
```

STEP 4 | 列出用於判斷 Web 前端之公用 IP 位址的服務。

```
$ kubectl get services -n sample-app
```

您現在可以在 Panorama 上設定動態位址群組和安全性規則，以保護 Guestbook 應用程式的安全。

繼續使用 Terraform 部署 CN-Series 防火牆。

使用 Terraform 部署 CN-Series 防火牆

使用 Terraform 部署 CN-Series 防火牆。

STEP 1 | 使用本機 `cn-series\tfvars` 建立名為 `terraform.tfvars` 的檔案，並新增下列變數和其相關聯的值。

```
k8s_environment = ""          # Kubernetes environment
                               # (gke|eks|aks|openshift|
native) panorama_ip = ""              # Panorama IP address
 panorama_auth_key = ""        # Panorama auth key for VM-series
 registration panorama_device_group = ""    # Panorama device
 group panorama_template_stack = ""  # Panorama template stack
 panorama_collector_group = "" # Panorama log collector group
 k8s_dp_cpu = ""               # DP container CPU limit
```

STEP 2 | 驗證 Terraform 計劃。

```
$ terraform init
```

STEP 3 | 驗證 Terraform 計劃。

```
$ terraform plan
```

STEP 4 | 套用 Terraform 計劃。

```
$ terraform apply
```

STEP 5 | 驗證 Pod 已部署並準備就緒，且狀態為 [Running（執行中）]。

```
$ kubectl get pods -A
```

```
NAMESPACE NAME READY STATUS RESTARTS AGE ... kube-system pan-
cni-6kkxw 1/1 Running 0 26m kube-system pan-cni-tvx2b 1/1 Running
 0 26m kube-system pan-mgmt-sts-0 1/1 Running 0 26m kube-system
 pan-mgmt-sts-1 1/1 Running 0 26m kube-system pan-ngfw-ds-nrtrn 1/1
 Running 0 26m kube-system pan-ngfw-ds-rcmmj 1/1 Running 0 26m
```

您已準備好設定 Panorama 的 Kubernetes 外掛程式。

設定 Panorama 的 Kubernetes 外掛程式

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本

使用 Panorama 的 Kubernetes 外掛程式以將標籤傳播至 Panorama 裝置群組。

您可以使用 Kubernetes 外掛程式完成 Panorama 與 Kubernetes API 的整合。此外掛程式會學習新的標籤，並將它們傳播至 Panorama 裝置群組。這些標籤可以包括 Kubernetes 標籤、服務、命名空間和其他中繼資料，而使用這些中繼資料可以定義「動態位址群組」比對規則。

如果叢集認證檔案大於 *32KB*，在 *Panorama Kubernetes* 外掛程式上匯入認證檔案時，您會收到錯誤訊息。錯誤訊息會顯示錯誤原因為檔案大小。

如果叢集在 `ca.crt` 套件組合中有許多 *CA* 憑證，則 *Kubernetes* 外掛程式只需要最上層的 *CA* 憑證。務請僅保留最上層 *CA* 憑證，並從認證檔案中移除所有其他 *CA* 憑證和 `service.crt`。然後您便能使用此更新的認證檔案。

此程序假設您已安裝準備使用 Helm 圖表和範本中所列的支援軟體。

STEP 1 | 從 Kubernetes 主機擷取 pan-plugin-user 服務帳戶認證。

將每個命令都輸入成單行：

```
$ MY_TOKEN=`kubectl get serviceaccounts pan-plugin-user -n kube-system
  -o jsonpath='{.secrets[0].name}'`
$ kubectl get secret $MY_TOKEN -n kube-system -o json >
  ~/Downloads/pan-plugin-user.json
```

STEP 2 | 在 Panorama Kubernetes 外掛程式中建立叢集定義。

使用 Terraform 輸出中顯示的 Kubernetes 主要位址，以及位於 `~/Downloads/pan-plugin-user.json` 的 JSON 認證檔案。

定義您要從 Kubernetes API 匯入的標籤。

STEP 3 | 在 Panorama Kubernetes 外掛程式中建立通知群組定義。

此定義用來將從 Kubernetes API 學到的標籤傳播至 Panorama 裝置群組。

請執行以下步驟，在 Panorama Kubernetes 外掛程式中建立通知群組：

1. 選取 **Panorama** >**Plugins**（外掛程式）>**Kubernetes** >**Setup**（設定） >**Notify Groups**（通知群組）並 **Add**（新增）。

2. 輸入最多 31 個字元的通知群組 **Name**（名稱）。
3. 如果您除了針對叢集所建立的外部標記（預設值）之外，還想要共用內部標記，則請選取 **Enable sharing internal tags with Device Groups**（啟用與裝置群組共用內部標記）。
4. 選取您要向其註冊標記的裝置群組。

5. 按一下 **OK**（確定）。

STEP 4 | 在 Panorama 外掛程式中建立監控定義。

使用先前步驟中建立的叢集和通知群組定義。

STEP 5 | 提交至 Panorama。

STEP 6 | 若要確認 API 連線和 MP 容器註冊，請前往 [Monitoring Definition（監控定義）]，然後按一下 [Detailed Status（詳細狀態）] 和 [Cluster MPs（叢集 MP）]。

您現在已準備好部署應用程式，並使用 CN-Series 防火牆來保護應用程式。

使用 Rancher 協調流程部署 CN-Series 防火牆

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本

您現在可以使用 Rancher 協調流程和 PAN OS 10.1 將 CN-Series 防火牆部署為 Kubernetes 服務。Rancher 是您可用來部署 CN-Series 防火牆的開放原始碼容器協調流程平台。

對於支援 Rancher 叢集的 CN-Series 防火牆部署，您的 Panorama 執行個體必須要有 16 個 vCPU、32G 記憶體以及額外的 2 TB 磁碟。將會以可協助從 CN-Series 防火牆部署收集日誌的模式來部署 Panorama。

在內部部署 Rancher Kubernetes 叢集內部署 CN-Series 防火牆時，請執行下列動作：

- 請確認使用 CN-Series 防火牆保護 Kubernetes 叢集所需的元件可以使用。
- 確定 Kubernetes 叢集符合最低系統需求。如需更多詳細資訊，請參閱 CN-Series 系統需求。
- 執行 使用 Rancher 協調流程部署 CN-Series 防火牆。
-
- 修改 Rancher 叢集選項 YAML 檔案
- 安裝 CN-Series 防火牆的 Kubernetes 外掛程式。
- 授權 CN-Series 防火牆。
- 在 Rancher 上的 部署 CN-Series 防火牆作為 Kubernetes 服務（建議部署模式）。

Rancher 叢集部署

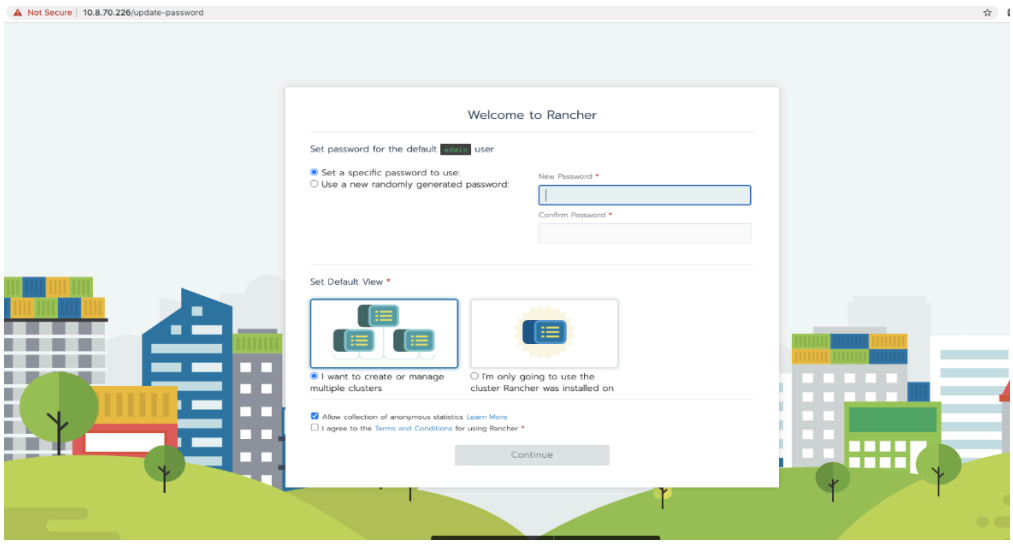
您可以使用下列兩個步驟來部署 Rancher：

1. 準備具有所支援 Linux 散佈和 4 GB 記憶體的 Linux 主機。在主機上安裝支援的 Docker 版本。
2. 啟動伺服器。

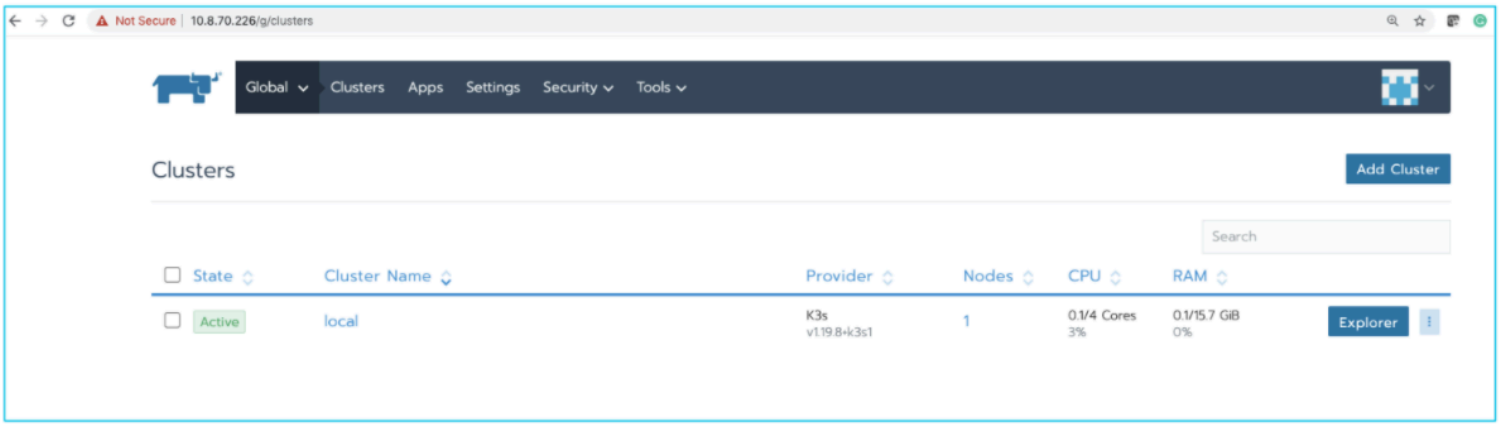
   若要安裝並執行 Rancher，請在主機上執行下列 Docker 命令：

```
$ sudo docker run --privileged -d --restart=unless-stopped -p 80:80
 -p 443:443 rancher/rancher
```

成功部署之後，您可以存取 Rancher 伺服器 UI，並為管理員使用者設定密碼。若要存取 Rancher 伺服器 UI，請開啟瀏覽器，然後前往已安裝容器的主機名稱或位址。系統會引導您完成設定第一個叢集。



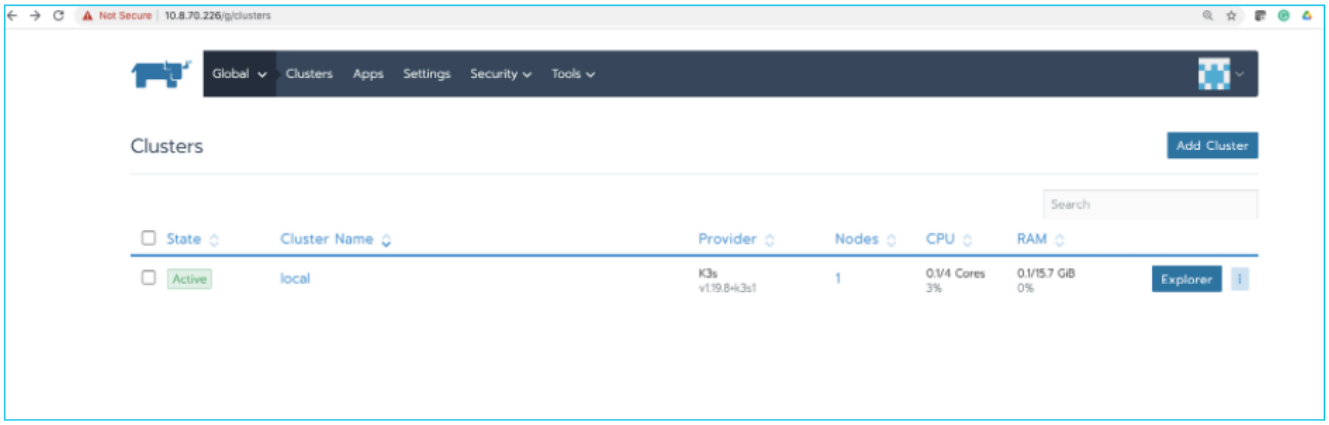
在建立管理使用者之後，將會建立本機叢集，如下所示：



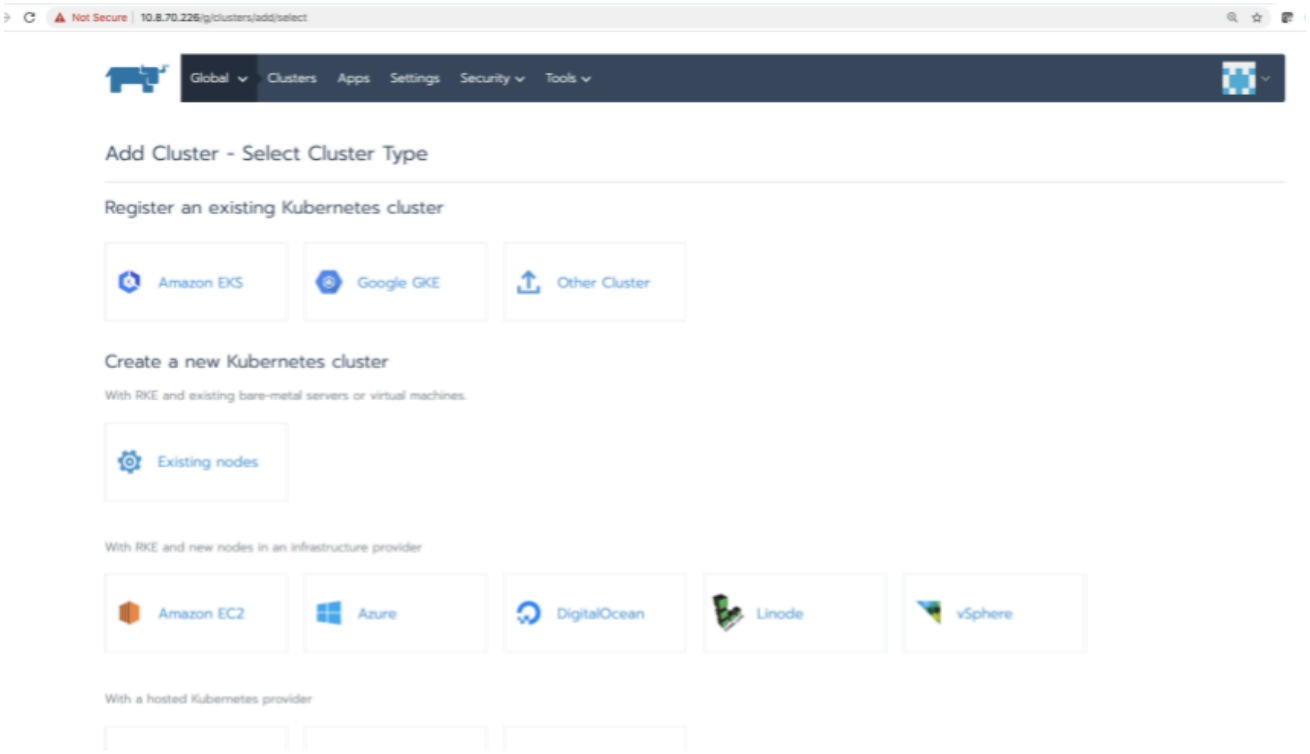
在 Rancher 叢集上設定主節點和工作節點

在 Rancher UI 上建立本機叢集之後，請設定主節點和工作節點，並執行下列動作：

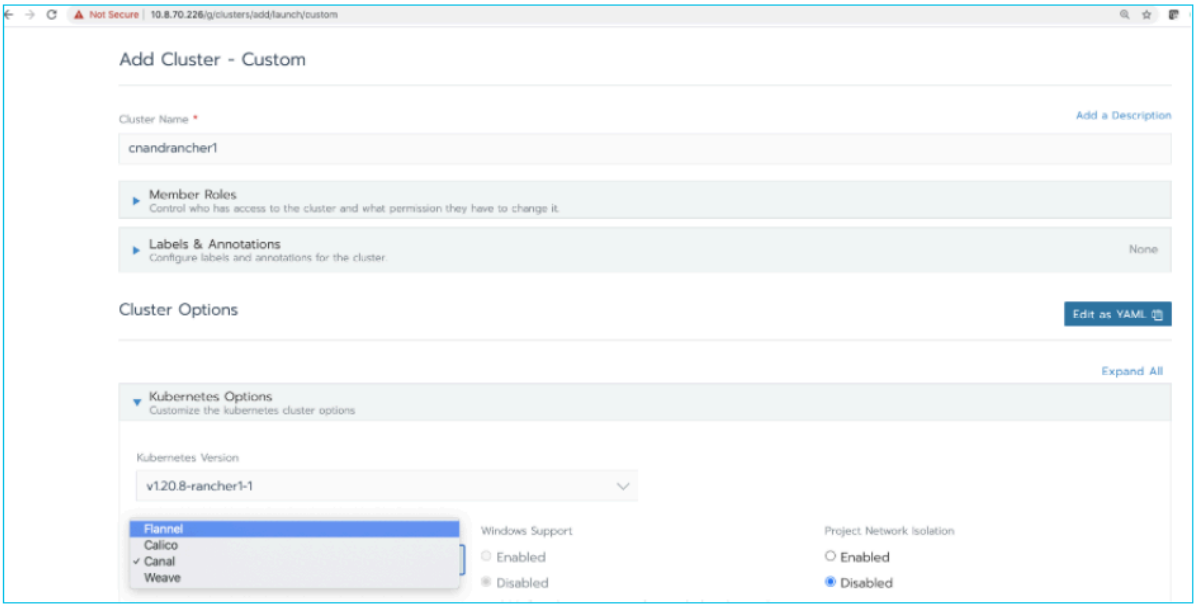
1. 前往 Rancher UI，然後按一下 **Add Cluster**（新增叢集）。



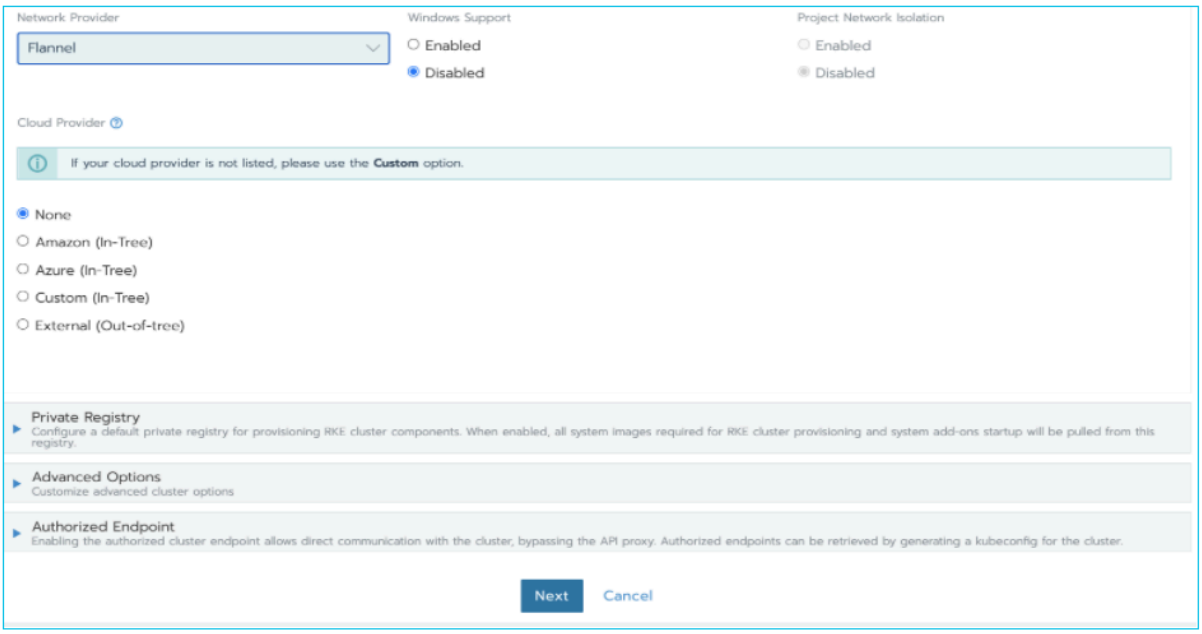
2. 按一下 **Existing nodes**（現有節點）。



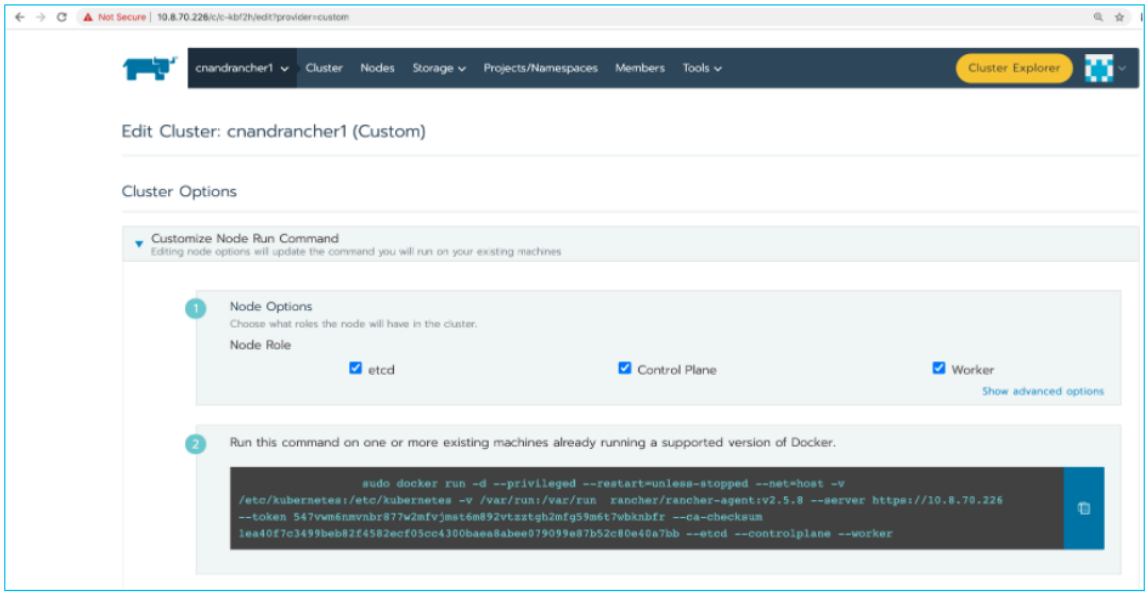
3. 輸入您的叢集名稱，然後從網路提供者下拉式清單中選取 Flannel。



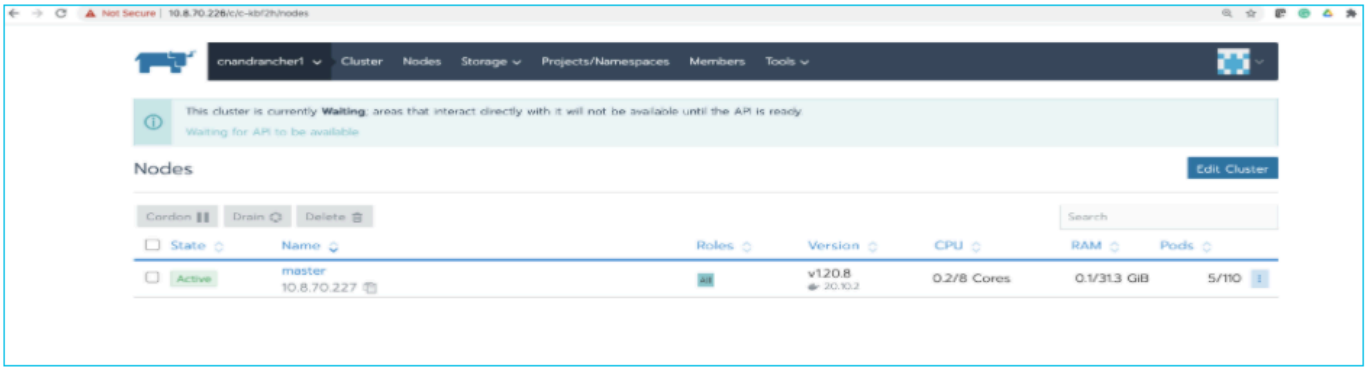
4. 保留所有其他欄位的預設值，然後按 **Next**（下一步）。



5. 在節點選項下，選取所有三個 **Node Role**（節點角色）選項，然後使用 SSH 將給定的命令執行到主節點。



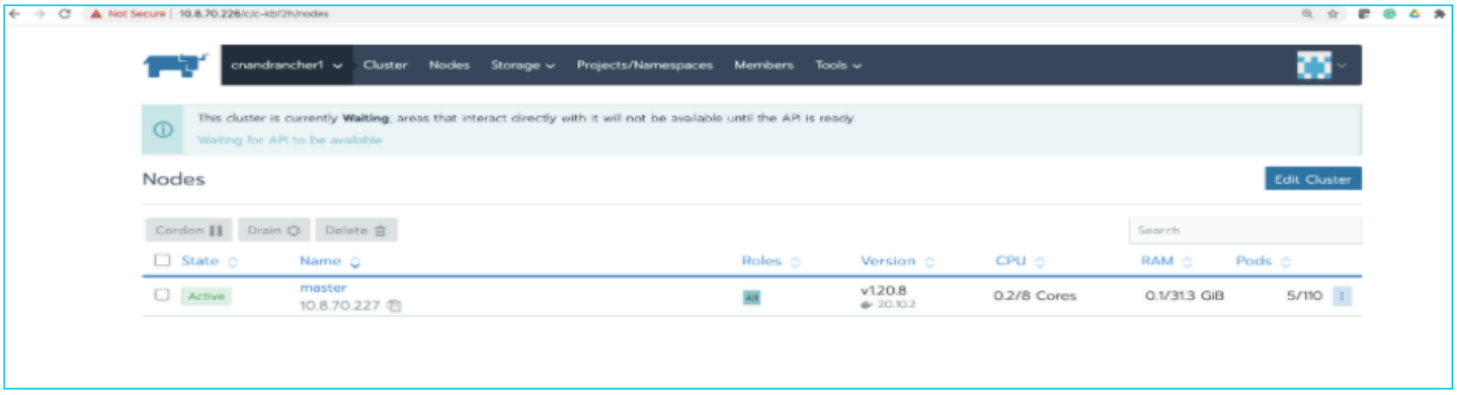
6. 確認已成功新增主節點。



7. 透過 SSH 進入每個工作節點並執行下列命令：

```
sudo docker run -d --privileged --restart=unless-stopped --
net=host -v /etc/kubernetes:/etc/kubernetes -v /var/run:/var/run
 rancher/rancher-agent:v2.5.8 --server https://10.8.70.226 --token
 547vwm6nmvnbr877w2mfvjmst6m892vtzztgh2mfg59m6t7wbknbfr --ca-
checksum
 1ea40f7c3499beb82f4582ecf05cc4300baea8abee079099e87b52c80e40a7bb
 --worker
```

在一個主節點和兩個工作節點上成功執行命令時，您會看到，Rancher 叢集已準備就緒，如下所示：



修改 Rancher 叢集選項 YAML 檔案

部署 CN-Series 防火牆之前，您必須修改叢集選項 YAML 檔案，如下所述。

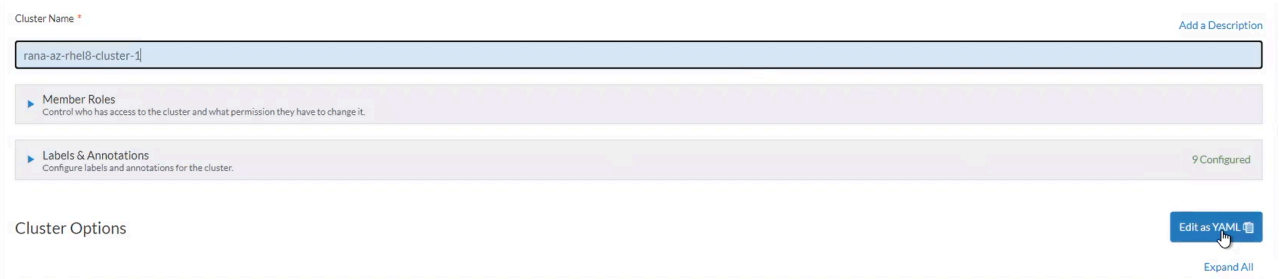
具有 *Rancher* 的 *CN-Series* 防火牆支援含 *k8s 1.20.5* 的 *Rancher 2.5* 或更新版本。

STEP 1 | 使用您先前建立的管理員認證登入 Rancher 入口網站。

STEP 2 | 按一下 **Navigation Menu**（導覽功能表），然後選取 **Cluster Management**（叢集管理）。

STEP 3 | 找到要修改的叢集，並按一下垂直省略符號功能表，然後選取 **Edit Config**（編輯設定）。

STEP 4 | 按一下 **Edit as YAML**（編輯為 **YAML**）。



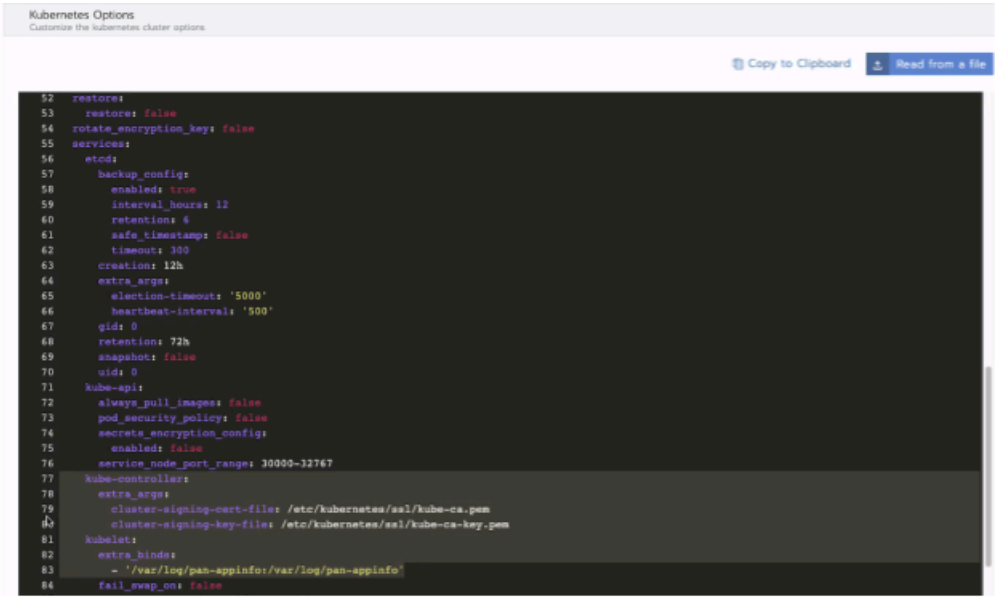
針對不同版本的 *Rancher*，請參閱 Rancher 文件。

STEP 5 | 在現有 YAML 檔案的 **Services** 區段下新增下列幾行。

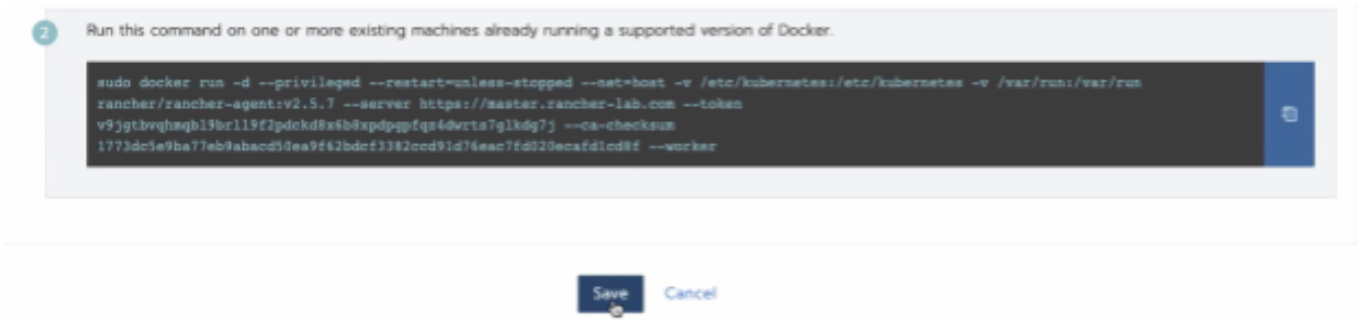
```
kube-controller: extra_args: cluster-signing-cert-file: "/etc/
kubernetes/ssl/kube-ca.pem" cluster-signing-key-file: "/etc/
kubernetes/ssl/kube-ca-key.pem"
```

```
kubelet: extra_binds: - ’ /mnt:/mnt:rshared’  - '/var/log/pan-
appinfo:/var/log/pan-appinfo’
```

如果您使用「*/mnt*」以外的儲存路徑，則應該確保修改 *extra_binds* 下的儲存路徑。



STEP 6 | 按一下 **Save**（儲存），並等到叢集升級變成使用中，再部署 CN-Series 防火牆。



CN-Series 部署 YAML 檔案中的可編輯參數

YAML 檔案包括數個可編輯參數，而下列各表列出您必須修改才能成功部署 CN-Series 防火牆的參數。

- PAN-CN-MGMT-CONFIGMAP
- PAN-CN-MGMT-SECRET
- PAN-CN-MGMT
- PAN-CN-NGFW-CONFIGMAP
- PAN-CN-NGFW
- PAN-CNI-CONFIGMAP
- PAN-CNI
- PAN-CNI-MULTUS

PAN-CN-MGMT-CONFIGMAP

PAN-CN-MGMT-CONFIGMAP	
進階路由 （Kubernetes 3.0.0 部署的必要項目） `PAN_ADVANCED_ROUTING:” true”`	如果您要搭配使用進階路由與 Kubernetes 3.0.0 外掛程式，則必須設定先在 PAN-OS 中予以啟用，然後在範本堆疊上手動設定。啟用之後，請提交並推送設定。如需詳細資訊，請參閱進階路由。
Panorama IP 位址 `PAN_PANORAMA_IP:`	包括 CN-MGMT Pod 將連線的 Panorama IP 位址。如果您已在高可用性 (HA) 設定中設定 Panorama 管理伺服器，則請提供主要主動 Panorama 的 IP 位址。 您可以在 **Dashboard**（儀表板） > **General Information**（一般資訊）上找到 Panorama IP 位址。
裝置群組名稱 `PAN_DEVICE_GROUP:`	指定您要將 CN-NGFW Pod 指派至其中的裝置群組名稱。從 Panorama，您將相同的原則推送至由 CN-MGMT Pod 配對所管理 (或屬於 PAN-SERVICE-NAME) 的所有 CN-NGFW Pod。 您可以在 **Panorama** > **Device Groups**（裝置群組）上找到裝置群組名稱。
範本堆疊名稱	允許您設定可讓防火牆 (CN-NGFW Pod) 在網路上作業的設定。

PAN-CN-MGMT-CONFIGMAP	
PAN_TEMPLATE_STACK:	您可以在 **Panorama** > **Templates**（範本）上找到範本堆疊名稱。
日誌收集器群組名稱 PAN_PANORAMA_CGNAME:	啟用 CN-NGFW 防火牆上所產生日誌的日誌儲存體。沒有「收集器群組」，就不會儲存防火牆日誌。 您可以在 **Panorama** > **Collector Groups**（收集器群組）上找到收集器群組名稱。
（選用） #CLUSTER_NAME:	指定叢集名稱。CN-MGMT Pod 的主機名稱會結合 PAN-CN-MGMT.yaml 中所定義的 StatefulSet 名稱與這個選用 CLUSTER_NAME。如果您在相同的 Panorama 設備上管理多個叢集，則此主機名稱可讓您識別與不同叢集相關聯的 Pod。最佳作法是在這裡以及 Panorama 的 Kubernetes 外掛程式上使用相同的名稱。
（選用）Panorama HA 對等節點 IP 位址 #PAN_PANORAMA_IP2:	高可用性設定中所設定 Panorama 對等節點（被動次要）的 IP 位址。請驗證 PAN_PANORAMA_IP 是主要主動 Panorama 的項目。 您可以在 **Panorama** > **High Availability**（高可用性） > **Setup**（設定）上找到 Panorama HA 對等 IP 位址。
（GTP 的必要項目）GTP 安全性 #PAN_GTP_ENABLED: "true"	在 CN-Series 防火牆上，針對 GTP 安全性，啟用此參數。在您啟用 GTP 之後，可以使用 Panorama 來設定 GTP 安全性以及監視防火牆上的 GTP 流量。
（Jumbo Frame 支援的必要項目，如果主要 CNI 未使用 Jumbo Frame）Jumbo Frame 模式 #PAN_JUMBO_FRAME_ENABLED: "true"	啟動期間的 CN-MGMT Pod 會使用 eth0 MTU 來自動偵測是否啟用 Jumbo Frame 模式。因此，如果您的次要 CNI 使用 Jumbo Frame，則主要 CNI 未使用時，您必須定義 PAN_JUMBO_FRAME_ENABLED:"True"，以在 CN-Series 防火牆上啟用 Jumbo Frame 模式。 您必須先進行這項變更，再部署 CN-MGMT StatefulSet。

PAN-CN-MGMT-CONFIGMAP	

（彈性系統資源配置的必要項目）
• CN-Series 作為 DaemonSet
`#PAN_NGFW_MEMORY:"42Gi"`
• CN-Series 作為 K8s 服務
`#PAN_NGFW_MEMORY:"6.5Gi"`
`#PAN_NGFW_MEMORY:"42Gi"`
針對「*5G-Native* 安全性」，建議使用 *48Gi*
	如果您需要較高的容量，而且想要設定更多記憶體來處理部署需求，則請使用此參數來定義記憶體值。
	• CN-Series 作為 DaemonSet
	小型容量等於或小於 42Gi，而大型容量大於 42Gi。
	• CN-Series 作為 K8s 服務
	小型容量小於 6.5Gi、中型容量介於 6.5Gi 與 42Gi 之間，而大型容量大於 42Gi。
	這項變更也需要 *pan-cn-ngfw.yaml* 上具有相同或更高的記憶體配置。

（選用）AF-XDP `#PAN_DATA_MODE: “next-gen”`	需要此參數，才能啟用位址系列 eXpress 資料路徑 (AF-XDP)。 AF-XDP 是一種基於 eBPF 的通訊端，針對適用於雲端原生服務的高效能封包處理進行最佳化，以提高有效輸送量。這需要核心 5.4 版或更新版本。此外，不支援巨型模式：EKS 無法使用此參數，因為預設會啟用巨型模式。 此外，PAN-CN-NGFW中需要特權模式。
（啟用 HPA 的必要項目） （AKS 和 GKE） `#HPA_NAME` （僅限 EKS）`#PAN_NAMESPACE_EKS` （僅限 AKS）`#PAN_INSTRUMENTATION_KEY`	需要幾個參數，才能在作為服務的 CN-Series 防火牆上啟用水平 Pod 自動調整規模 (HPA)。 • 對於每個環境，您必須提供唯一名稱，以識別每個命名空間或每個租戶的 HPA 資源。 • 對於 AKS 部署，您必須提供 Azure Application Insight 儀表金鑰。

下列預設值定義於 *pan-cn-mgmt-configmap.yaml* 檔案中。

```
metadata: name: pan-mgmt-config namespace: kube-system
 data:PAN_SERVICE_NAME: pan-mgmt-svc PAN_MGMT_SECRET: pan-mgmt-
secret
```

這些預設值可讓您使用這些檔案來取得快速概念證明。如果您想要修改這些項目（例如，部署最多管理 30 個 PAN-NGFW Pod 的多個 PAN-MGMT Pod 容錯配對），則必須修改 `pan-mgmt-`

PAN-CN-MGMT-CONFIGMAP
`svc`，以使用另一個服務名稱。當您修改這些值時，必須更新其他 YAML 檔案中的對應參照，以符合您在此檔案中所定義的值。

PAN-CN-MGMT-SECRET

PAN-CN-MGMT-SECRET	
VM 驗證金鑰 `PAN_PANORAMA_AUTH_KEY:`	可讓 Panorama 驗證防火牆，以便將每個防火牆新增為受管理裝置。部署留存期需要 VM 驗證金鑰。如果連線要求缺少有效金鑰，則 CN-Series 防火牆將無法向 Panorama 註冊。 請參閱安裝 CN-Series 防火牆的 Kubernetes 外掛程式。

CN-Series 的裝置憑證
`CN-SERIES-AUTO-REGISTRATION-PIN-ID`
`CN-SERIES-AUTO-REGISTRATION-PIN-VALUE`
	防火牆需要裝置憑證以取得任何網站授權權利，以及安全地存取 Palo Alto 雲端提供的服務。在 Palo Alto Networks CSP 上產生 PIN ID 和 PIN 值，並在 PIN 到期前使用該 PIN。例如：
	`CN-SERIES-AUTO-REGISTRATION-PIN-ID:`
	`"01cc5-0431-4d72-bb84-something”`
	`CN-SERIES-AUTO-REGISTRATION-PIN-VALUE:`
	`"12·····.13e"`
	CN-SERIES-AUTO-REGISTRATION-API-CSP 的下列其他欄位已設為註解，而且不是必要項目：*`"certificate.paloaltonetworks.com"`*
	請參閱在 CN-Series 防火牆上安裝裝置憑證。

PAN-CN-MGMT

PAN-CN-MGMT	
CN-MGMT 防火牆之 Init 容器映像檔的映像檔路徑 `initContainers: - name: pan-mgmt-init image: <your-private-registry-image-path>`	init 容器會產生憑證，以用於保護 CN-MGMT Pod 執行個體之間以及 CN-MGMT Pod 與 CN-NGFW Pod 之間的通訊。 編輯映像檔路徑，指向已將 CN-MGMT 容器的 Docker 映像檔上傳至其中的位置。
CN-MGMT 映像檔容器的映像檔路徑： `initContainers: - name: pan-mgmt image: <your-private-registry-image-path>`	編輯映像檔路徑，指向已將 CN-MGMT 容器的 Docker 映像檔上傳至其中的位置。
CN-MGMT 防火牆的主機名稱 `kind:StatefulSet metadata: name: pan-mgmt-sts`	CN-MGMT 防火牆的主機名稱衍生方式是結合 StatefulSet 名稱與您可能已在 `pan-cn-mgmt-configmap.yaml` 中定義的選用叢集名稱。 CN-MGMT Pod 的預設主機名稱是 pan-mgmt-sts-0 和 pan-mgmt-sts-1，因為 StatefulSet 名稱是 pan-mgmt-sts，而且未定義叢集名稱。 如果主機名稱超過 *30* 個字元，則會將名稱截斷於 *30* 個字元。
（如果您已定義彈性系統資源配置的記憶體，則為必要項目）	如果您已針對 `#PAN_NGFW_MEMORY:"40Gi"` 配置超過或等於 40Gi 的記憶體值（在 `pan-cn-mgmt-configmap.yaml` 中），則請確定 `request` 和 `limit` 中具有相同的 CPU 和記憶體值，以達到下者下方的較高容量使用率： `containers: resources: requests: # configurable based on desired logging, capacities cpu:"4" memory:"16.0Gi" limits: cpu:"4" memory:"16.0Gi"` 針對「5G-Native 安全性」，建議值為 cpu=4、memory=16Gi
（僅適用於內部部署或自我管理原生 Kubernetes 部署）	針對自我管理部署，預設設定具有「storageClassName: local」。

PAN-CN-MGMT	
`storageClassName: local`	如果您的叢集具有動態佈建的「永久性磁碟區 (PV)」，則必須修改「storageClassName: local」以符合該 storageClass，或在使用 DefaultStorageClass 時移除這些行。 如果您的叢集沒有動態佈建 PV，則叢集管理員可以建立具有所提供 `pan_cn_pv_local.yaml` 的靜態 PV，而此檔案具有 2 組 PV，而每一組各適用於每個 PAN-CN-MGMT statefulSet Pod。您可以修改 `pan_cn_pv_local.yaml` 以符合設定中的磁碟區，以及在部署 `PAN-CN-MGMT.yaml` 之前進行部署。

PAN-CN-NGFW-CONFIGMAP

除非您需要變更下列項目，否則不需要修改任何 PAN 值：

- `PAN_SERVICE_NAME: pan-mgmt-svc`

  服務名稱應該符合您在 PAN-CN-MGMT-CONFIGMAP 上定義的服務名稱。

- `FAILOVER_MODE: failopen`

  您可以將此項目變更為 failclose。這只有在 CN-NGFW 無法取得授權時才會作用。

  - 在 fail-open 模式中，防火牆將會接收並送出封包，而不需要進行檢查。轉換為 fail-open 模式時會導致內部重新啟動和流量短暫中斷。
  - 在 fail-close 模式中，防火牆將會捨棄所有收到的封包。fail-close 模式也會關閉 CN-NFGW，並釋出已配置讓其他授權 CN-NFGW 使用該插槽的插槽。

- CPU 釘選—在 `pan-cn-ngfw-configmap.yaml` 中，會停用 CPU 釘選和超執行緒處理。請不要切換此設定來啟用專用實體核心的 CPU 釘選，而非具有超執行緒處理的邏輯核心，除非由「Palo Alto Networks 支援」指導。

  ```
  PAN_CPU_PINNING_ENABLED:"True"/"False"
  PAN_HYPERTHREADING_ENABLE:"True"/"False"
  ```

PAN-CN-NGFW

PAN-CN-NGFW	
Image path for the CN-NGFW container image image `containers: - name: pan-ngfw-container image:`	編輯映像檔路徑，指向已將 CN-NGFW 容器的 Docker 映像檔上傳至其中的位置。

PAN-CN-NGFW	
`<your-private-registry-image-path>`	
（如果您已定義彈性系統資源配置的記憶體，則為必要項目）	如果您已針對 `#PAN_NGFW_MEMORY:"40Gi"` 配置超過或等於 40Gi 的記憶體值（在 `pan-cn-mgmt-configmap.yaml` 中），則請確定 `request` 和 `limit` 中具有相同的 CPU 和記憶體值，以達到下者下方的保證 QoS： `containers: resources: requests: #configurable based on desired throughput, number of running pods cpu:"1" memory:"40.0Gi" limits: cpu:"1" memory:"40.0Gi"` 針對「5G-Native 安全性」，建議值為 cpu=12、memory=48Gi。

註：
• 下列註釋識別 PAN-NGFW daemonset：
`paloaltonetworks.com/app: pan-ngfw-ds`
請不要修改此值。
• 下列註釋識別防火牆名稱（「pan-fw」）：
`paloaltonetworks.com/firewall: pan-fw`
在 `pan-cni-configmap.yaml` 中，`cni_network_config: “firewall”` 中的這個防火牆名稱必須完全符合
而且，在用來部署每個應用程式 Pod 的應用程式 yaml 中，此註釋應該完全符合。
	每個節點上的 CN-NGFW Pod 都會保護具有註釋的應用程式 Pod 和命名空間：
	`paloaltonetworks.com/firewall: pan-fw`
	請將此註釋保持現狀。

（選用）AF-XDP `imagePullPolicy:Always securityContext: capabilities: #add: ["NET_ADMIN","NET_RAW","NET_BROADCAST","NET_BIND_SERVICE"]`	您必須將 `privileged: true` 新增至左側顯示的區段。需要此參數，才能啟用位址系列 eXpress 資料路徑 (AF-XDP)。 您也必須在PAN-CN-MGMT-CONFIGMAP中啟用 AF-XDP。

PAN-CN-NGFW	
`add: ["ALL"] privileged: true resources:`	

PAN-CNI-CONFIGMAP

這些是選用參數。

PAN-CNI-CONFIGMAP	
應用程式 Pod 可能屬於的防火牆名稱清單： `"firewall": [ "pan-fw" ]`	不需要修改時，如果您變更 `pan-cn-ngfw.yaml` 中的註釋 `paloaltonetworks.com/firewall: pan-fw`，則必須取代 `"firewall": [ "pan-fw" ]` 中的值，使其相符。
`"exclude_namespaces": []`	不需要修改時，如果您想要排除特定命名空間，則請將它新增至 **`"exclude_namespaces"`**，以忽略該命名空間中的應用程式 Pod 註釋，而且不會將流量重新導向至 CN-NGFW Pod 以進行檢查。
`"security_namespaces": [ "kube-system" ]`	新增已在 security_namespaces 中部署 CN-NGFW daemonset 的命名空間。預設命名空間是 kube-system。

`“interfaces”`	在應用程式 Pod 中新增介面，而您要從此應用程式 Pod 中將流量重新導向至 CN-NGFW Pod 以進行檢查。預設只會檢查 eth0 流量，而且您可以使用逗號區隔字串清單形式來新增其他介面，例如 [“eth0”，“net1”，“net 2”]。
	`cni_network_config:`
	`{ "cniVersion":"0.3.0", "name": "pan-cni", "type": "pan-cni", "log_level": "debug", "appinfo_dir": "/var/log/pan-appinfo", "mode": "daemonset", "firewall": [ "pan-fw" ], "interfaces": ["eth0", "net1", "net2", "net3"],`
	`}`

PAN-CNI-CONFIGMAP	
	除了此項目之外，您也必須將 *pan-cni* 附加至應用程式 *Pod* 中的 *k8s.v1.cni.cncf.io/networks* 註釋。 例如： `metadata: name: testpod annotations: paloaltonetworks.com/firewall: pan-fw k8s.v1.cni.cncf.io/networks: sriov-net1, sriov-net2, macvlan-conf, pan-cni` *CN-Series* 目前不支援 *DPDK*，而且不允許應用程式 *Pod* 使用 *DPDK*。如果應用程式未自動調整為非 *DPDK* 模式，則您可能需要修改應用程式 *Pod*。
（僅限 CN-Series 作為 Kubernetes 服務） “dpservicename” “dpservicenamespace”	將 CN-Series 部署為服務時，需要 dpservicename 和 dpservicenamespace。依預設，dpservicename 是「pan-ngfw-svc」而 dpservicenamespace 是「kube-system」。

PAN-CNI

PAN-CNI	
具有 CNI 二進位檔以及每個節點上 CNI 網路設定檔案之 PAN-CNI 容器映像檔的映像檔路徑。 `containers: name: install-pan-cni image: <your-private-registry-image-path>`	編輯映像檔路徑，指向已將 PAN-CNI 容器的 Docker 映像檔上傳至其中的位置。

PAN-CNI-MULTUS

如果您要在 Kubernetes 的自我管理或原生實作（例如使用 VMware TKG+）上使用 Multus CNI，則請使用 `pan-cni-multus.yaml`，而非 `pan-cni.yaml`。

使用 CN-Series 防火牆保護 5G

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

若要查看和控制私人企業的 5G 流量以及 Kubernetes 上 Mobile Operator Networks 中的 5G Mobile Packet Core 部署，請檢閱下列各節來瞭解支援的環境以及如何修改 YAML 檔案來解除鎖定 CN-Series 防火牆上的 GTP 安全性和 5G-Native 安全性。除了在部署 CN-Series 防火牆時啟用這些功能之外，您還必須啟用 Panorama 來取得「GTP 安全性」和/或 SCTP 安全性。

容器執行時期	Docker CRI-O Containerd
Kubernetes 版本	1.17 到 1.27
雲端提供者受管理 Kubernetes	• AWS EKS（CN-Series 的 1.17 到 1.27 作為精靈集和服務部署模式。） • AWS EKS（CN-Series 的 1.17 到 1.22 作為 CNF 部署模式。） • AWS EKS（CN-Series 的 1.22 到 1.27 作為 CN 叢集部署。） • AWS Outpost 上的 EKS（1.17 到 1.25） *AWS Outpost 上 EKS 的 CN-Series 不支援 SR-IOV 或 Multus。* • Azure AKS（1.17 到 1.27） 在 *Azure AKS* 中，*PAN-OS 11.0.2* 是支援 *kubernetes 1.25* 及更新版本的所需最低版本。

	• GCP GKE（1.17 到 1.27） 包括 *GKE* 資料平面 *V2*。 • OCI OKE (1.23)
客戶受管理 Kubernetes	在公用雲端或內部部署資料中心上。 請確定此表格列出 Kubernetes 版本、「CNI 類型」和「主機 VM OS」版本。 VMware TKG + 1.1.2 版 • 基礎架構平台—vSphere 7.0 • Kubernetes 主機 VM OS—Photon OS
Kubernetes 主機 VM	作業系統： • Ubuntu 16.04 • Ubuntu 18.04 • Ubuntu-22.04 • RHEL/Centos 7.3 和更新版本 • CoreOS 21XX、22XX • 容器最佳化 OS
	Linux 核心版本： • 4.18 或更新版本（僅限 K8s 服務模式） • 啟用 AF_XDP 模式所需的 5.4 或更新版本。如需詳細資訊，請參閱 CN-Series 部署 YAML 檔案中的可編輯參數。
	Linux 核心 Netfilter：Iptables
CNI 外掛程式	CNI Spec 0.3 和更新版本： • AWS-VPC • Azure • Calico • Flannel • Weave • 針對 Openshift、OpenshiftSDN

	• 在作為 DaemonSet 的 CN-Series 防火牆上，支援下列項目。
	  • Multus
	  • 橋接器
	  • SR-IOV
	  • Macvlan

OpenShift	• 4.2、4.4、4.5、4.6、4.7、4.8、4.9、4.10、4.11、4.12 和 4.13 版。 *OpenShift 4.7* 僅適用於「*CN-Series* 作為 *DaemonSet*」。 *PAN-OS 11.0.2* 是支援 *4.12* 及更新版本的所需最低版本。 • AWS 上的 OpenShift

容器執行時期	版本
CN-Series 防火牆 Kubernetes 外掛程式 Panorama	PAN-OS 10.0.3 或更新版本 1.0.1 或更新版本 10.0.0 或更新版本
下列是 YAML 檔案中用來部署 CN-Series 防火牆的所有可編輯參數清單：如需更多詳細資訊，請參閱 CN-Series 部署 yaml 檔案中的可編輯參數和 CN-Series 核心建置區塊。	
啟用 GTP	在 pan-cn-mgmt-configmap.yaml 中，先設定：`PAN_GTP_ENABLED :"True"`，再部署 CN-MGMT StatefulSet。
啟用 Jumbo Frame 模式	在 pan-cn-mgmt-configmap.yaml 中，先設定：`PAN_JUMBO_FRAME_ENABLED:"True"`，再部署 CN-MGMT StatefulSet。 啟動期間的 CN-MGMT Pod 會使用「eth0」MTU 來自動偵測是否啟用 Jumbo Frame 模式。因此，如果您的次要 CNI 使用 Jumbo Frame，則主要 CNI 未使用時，您必須定義 `PAN_JUMBO_FRAME_ENABLED:"True"`，以在 CN-Series 防火牆上啟用 Jumbo Frame 模式。

容器執行時期	版本
	CN-Series 目前不支援 *DPDK*，而且不允許應用程式 *Pod* 使用 *DPDK*。如果應用程式未自動調整為非 *DPDK* 模式，則您可能需要修改應用程式 *Pod*。
啟用系統資源彈性	如果您需要較高的輸送量，而且想要設定更多記憶體來處理 pan-cn-mgmt-configmap.yaml set 上的部署需求：PAN_NGFW_MEMORY="48Gi" 針對範本處理 *(Helm)*，則需要採用針對 *CN-NGFW Pod* 所配置的相同變數。啟用較大記憶體使用量時，*CN-MGMT StatefulSet* 只支援一個 *CN-NGFW Pod*。
設定 5G 的 vCPU、記憶體	CN-MGMT Pod（在 pan-cn-mgmt.yaml 中）和 NGFW Pod（在 pan-cn-ngfw.yaml 中）的建議設定是 "request" 和 "limit" 中具有相同的 cpus 和 memory 值，以達到保證的 QoS。 針對 CN-MGMT Pod，建議值是 cpu=4、memory=16Gi。若要控制 CN-MGMT Pod 的放置（例如，在與 CN-NGFW Pod 部署所在位置相同或不同的節點上），請在 k8s 中使用節點選取器功能。
	針對 CN-NGFW Pod，建議值是 cpu=12、memory=48Gi。若要控制 CN-NGFW Pod 的放置（例如，在與 CN-NGFW Pod 部署所在位置相同或不同的節點上），請在 k8s 中使用節點選取器功能。
選取 CNI yaml 檔案	Multus CNI 是作為呼叫其他 CNI 外掛程式的中繼外掛程式。在 OpenShift 環境上，預設會啟用 Multus，讓您可以使用 `pan-cni.yaml`。在支援 Multus 但為選用的其他環境（例如具有自我管理（原生）環境）上，使用 `pan-cni-multus.yaml`，而非 `pan-cni.yaml`。

在繼續部署 CN-Series 防火牆之前，請同時查看 CN-Series 防火牆的系統需求。

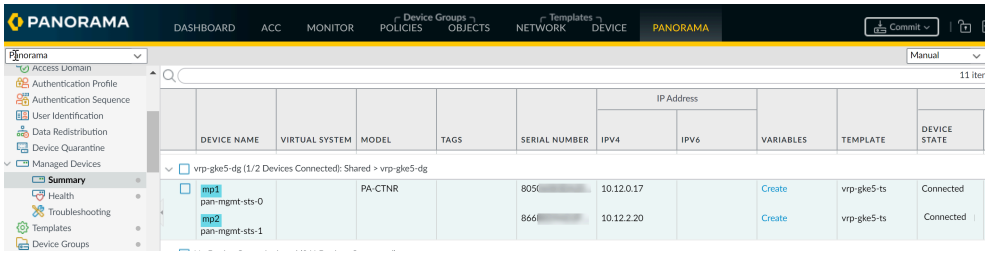
設定 Panorama 保護 Kubernetes 部署

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 圖表的 CN-Series 部署

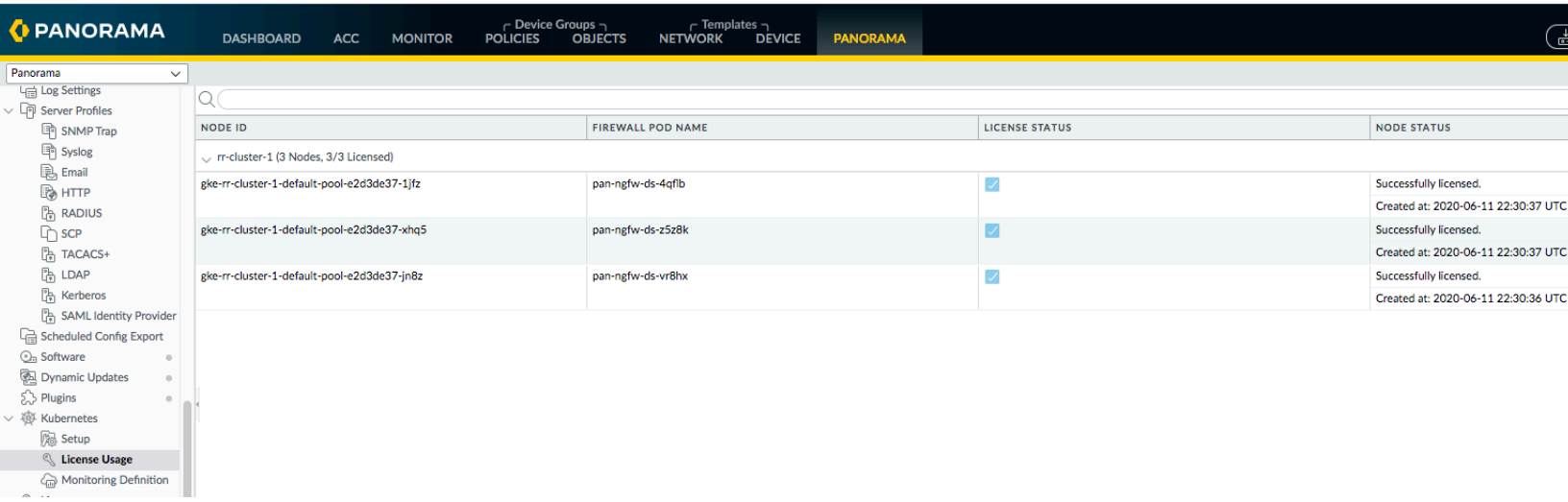
在您安裝 CN-Series 的 Kubernetes 外掛程式以及部署 CN-Series 防火牆來監控 Kubernetes 叢集以及設定可啟用流量強制執行的安全性政策之後，需要完成下列工作

STEP 1 | 驗證已在 Panorama 上註冊 CN-MGMT Pod，並且已授權 CN-NGFW Pod。

1. 選取 **Panorama** > **Managed Devices**（受管理的裝置） > **Summary**（摘要）。



2. 選取 **Panorama** > **Plugins**（外掛程式） > **Kubernetes** > **License Usage**（授權使用），驗證叢集內的每個節點都已獲配置一個授權權杖。



STEP 2 | 建立「日誌轉送」設定檔，以將日誌轉送給 Panorama。

設定檔會為將在防火牆上產生的不同日誌定義目的地。

1. 從 **Device Group**（裝置群組）下拉式清單中，選取您針對 k8s 部署所部署的裝置群組。
2. 選取 **Objects**（物件）> **Log Forwarding**（日誌轉送），然後按一下 **Add**（新增）。
3. 輸入用來識別設定檔的 **Name**（名稱）。如果您要將設定檔自動指派給新的安全性規則和區域，則請輸入 **`default`**。如果您不想要預設設定檔，或想要覆寫現有預設設定檔，則請在將設定檔指派給安全性規則時輸入可協助您識別設定檔的 **Name**（名稱）。
4. **Add**（新增）要轉送的日誌類型。
5. 按一下 **OK**（確定）。

STEP 3 | 設定 Kubernetes 外掛程式，以將標記推送至指定的裝置群組。

您必須新增包含 Kubernetes 叢集名稱的監視定義，而 Panorama 會從此叢集中擷取預先定義標籤和通知群組（選用）。

如果 *CN-Series* 部署在 *kube* 系統以外的命名空間中，則需要通知群組。

通知群組是可接收標記更新的裝置群組清單。針對 Kubernetes 外掛程式，通知群組應該包括叢集外部的防火牆 (表示它們不屬於與 Kubernetes 叢集相同的裝置群組，而且會從此叢集中收集屬性)。

因為您在 YAML 檔案中指定用來部署 CN-Series 防火牆的裝置群組名稱，則 Kubernetes 外掛程式會自動學習叢集內部的所有裝置群組，而且預設會自動將所有預先定義的標記推送至這些裝置群組。

Kubernetes 外掛程式使用「Kubernetes 祕密」來動態學習每個叢集內的裝置群組。每次部署 CN-MGMT StatefulSet 時，都會將「祕密」發佈至 Kubernetes API 伺服器，而且 Panorama 會在下個監視間隔學習到它。

1. 設定用於監視叢集的 Kubernetes 外掛程式。
2. 新增通知群組。新增通知群組，並選取可接收與 Kubernetes 叢集相關之標記的裝置群組。
   1. 選取 **Panorama** > **Plugins**（外掛程式） > **Kubernetes** > **Setup**（設定） > **Notify Groups**（通知群組）並 **Add**（新增）。
   2. 輸入最多 31 個字元的通知群組 **Name**（名稱）。
   3. 如果您除了針對叢集所建立的外部標記（預設值）之外，還想要共用內部標記，則請選取 **Enable sharing internal tags with Device Groups**（啟用與裝置群組共用內部標記）。
   4. 選取您要向其註冊標記的裝置群組。

      針對您選取的「通知群組」，Panorama 只會推送外部標記。

      外部標記是任何可從叢集外部到達的標記，例如針對叢集 IP 位址的外部服務 IP 位址和連接埠、所有節點和節點連接埠的外部 IP 位址以及外部負載平衡器 IP 位址和連接埠或節點連接埠所產生的標記。

      內部標記包括內部叢集 IP 位址、Pod IP 位址、節點和節點連接埠的詳細資料。

      Panorama 預設會將所有找到的標記（根據您選取的「標籤篩選」）推送至與叢集相關聯的「裝置群組」，而這定義於用來部署 CN-MGMT Pod 的 YAML 檔案中。
3. 新增每個叢集的監視定義。
   1. 選取 **PanoramaPlugins** > **Kubernetes** > **Monitoring Definition**（監視定義）和 **Add**（新增）。
   2. 輸入監視定義的 **Name**（名稱）。
   3. 選取您要監視的 **Cluster**（叢集）。

4. （選用）選取您要傳送 IP 位址與標記對應資訊的 **Notify Group**（通知群組）。

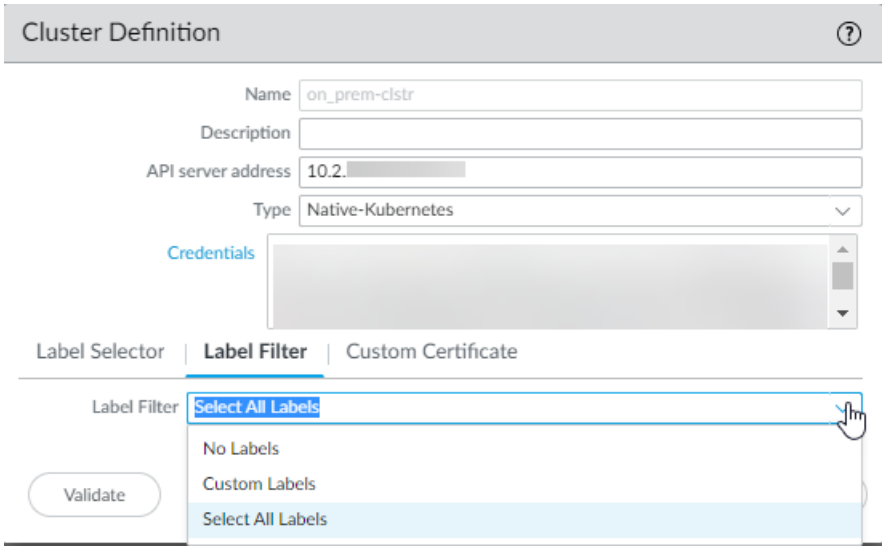
   預設會與叢集內的所有 CN-NFGW Pod 共用標記。

5. 按一下 **OK**（確定）儲存您的變更。

4. **Commit**（提交）至 Panorama。

STEP 4 | （選用）設定 Kubernetes 外掛程式，以從應用程式 YAML 檔案中擷取使用者定義的標籤。

1. 選取 **PanoramaPlugins** > **Kubernetes** > **Setup**（設定） > **Cluster**（叢集），然後從清單中選取叢集定義。
2. 從下列選項中，選取標籤篩選：



1. **No Labels**（無標籤）—不建立 Kubernetes 標籤的任何標記。
2. **Custom Labels**（自訂標籤）—只建立您所關心之標籤的標記。

   若要使用自訂標籤，您必須先將 Kubernetes 部署中的 YAML 檔案設為註釋，然後使用下列任何組合來產生對應 IP 位址的自訂標記：

   指定命名空間、索引鍵和值。使用 * 表示全部。這三個輸入都有效時，外掛程式會建立標記。

   指定命名空間和索引鍵，以建立該命名空間內所有相符索引鍵的標記。

   指定命名空間，只建立該命名空間內每個標籤的標記。
3. **Select All Labels**（選取所有標籤）—建立所有 Kubernetes 標籤的標記，包括任何自訂標籤。

3. 新增標籤選取器運算式。

   標籤選取器符合 Kubernetes 叢集內指定的標籤，並將與標籤相關聯的 IP 位址對應至單一標記。如需所支援前置詞的清單，請參閱 Kubernetes 屬性的 IP 位址與標記對應。

   針對每個標籤選取器，Panorama 會產生可作為動態位址群組中比對準則的標記，並可讓您強制執行安全性原則：

   1. **Tag Prefix**（標記前置詞）—每個標記結尾可協助您輕鬆地識別標記的詞組。例如，標籤選取器 k8s.cl_<clustername>.<selector-name> 符合所有符合這些選取器的叢集 IP、所

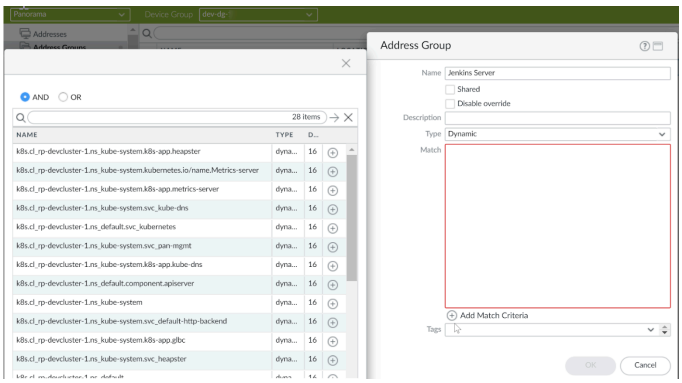
有符合選取器的 Pod IP。根據您設定的內容，這些可能位在所有命名空間或特定命名空間中。

2. **Namespace**（命名空間）—* 表示所有命名空間，或輸入命名空間的值。
3. **Label Selector Filter**（標籤選取器篩選）—Kubernetes 外掛程式支援標籤索引鍵和標籤值的集合型和相等型選取器。系統支援以下相等型選取器—key = value；key ==value；key != value，例如 app = redis。您也可以在運算式中使用逗號區隔清單形式來指定多個選取器，例如 app == web、tier != backend。系統支援以下集合型選取器—key in (value1,value2)、key notin (value1, value2)、key、!key，例如 tier notin (frontend, backend)。
4. **Apply On**（套用於）—要對其套用此項目的資源類型是 [Service（服務）]、Pod、[All（全部）]。

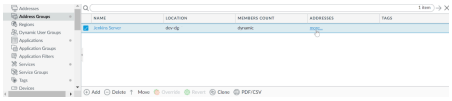
STEP 5 | 設定「動態位址群組」。

1. 選取「裝置群組」來管理 CN-NGFW Pod。
2. 選取 **Object**（物件） > **Address Groups**（位址群組）。
3. 按一下 **Add**（新增），再輸入位址群組的 **Name**（名稱）和 **Description**（說明）。
4. 在 **Type**（類型）中選取 **Dynamic**（動態）。

STEP 6 | 按一下 **Add Match Criteria**（新增比對準則），並選取 **AND** 或 **OR** 運算子，然後選取您在篩選或比對時要對照的屬性。



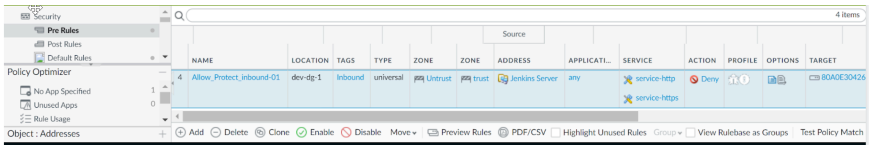
STEP 7 | 按一下 **OK**（確定），並 **Commit on Panorama**（在 **Panorama** 上提交）。



使用 **[more...**（更多...）**]** 連結來檢視與物件（在此範例中，是叢集中的 Jenkins 伺服器）相關聯的 IP 位址。

STEP 8 | 建立「安全性原則」規則來強制執行流量。

1. 選取 **Policies**（政策） > **Security**（安全性）。
2. 按一下 **Add**（新增），然後輸入原則的 **Name**（名稱）和 **Description**（說明）。
3. 新增 **Source Zone**（來源區域）以指定流量來源於哪個區域。
4. 新增流量將終止於哪個 **Destination Zone**（目的地區域）。
5. 對於 **Destination Address**（目的地位址），請選取您剛才建立的動態位址群組。
6. 針對流量指定動作—**Deny**（拒絕），並選擇性地將預設安全性設定檔附加至規則。
7. 選取 **Actions**（動作）頁籤，然後選取所建立的 **Log Forwarding**（日誌轉送）設定檔。
8. 按一下 **Commit**（交付）。



您也可以在命名空間內套用東西向流量的「安全性」原則。例如，如果您在稱為臨時叢集的叢集內有兩個命名空間 stage-ns 和 db-ns，而在此叢集中，投票應用程式的前端 Pod 部署在 stage-NS 中，而 Redis 後端 Pod 是在 DB-NS 命名空間中執行。當您將此叢集新增至 Panorama 上的 Kubernetes 外掛程式以進行監視時，會擷取標籤中繼資料來建立標記。您可以使用這些標記來強制執行「安全性」原則規則。若要這麼做，您需要：

- 確定您用來部署前端和後端應用程式的「命名空間」或 YAML 檔案是以 paloaltonetworks.com/firewall: pan-fw 標註。
- 建立前端和後端 Pod 的動態位址群組。

  您必須在與叢集相關聯的裝置群組中設定動態位址群組，然後先選取前端伺服器的標籤。然後，重複建立後端伺服器之另一個動態位址群組的程序。
- 新增安全性政策規則以允許 Redis 應用程式從前端 Pod 到後端 Pod 的流量。

  來源是前端伺服器的動態位址群組、目的地是後端伺服器的動態位址群組，而動作是「允許」。

Kubernetes 屬性的 IP 位址與標記對應

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

Panorama 上的 Kubernetes 外掛程式會建立 Kubernetes 叢集中預先定義標記的標記、Pod 和服務的使用者定義標籤，以及服務物件。

外掛程式會建立下列 Kubernetes 物件的標記：

- Pod 類別：ReplicaSets、DaemonSets、StatefulSets
- 服務類型：ClusterIP、NodePort、LoadBalancer
- 服務物件：port、targetPort、nodePort 和 Pod 介面

Panorama 上的 Kubernetes 外掛程式預設會從您要在 Panorama 上監視的每個 Kubernetes 叢集中擷取下列預先定義的標記，並以下面列出的格式來建立標記。您接著可以使用這些標記作為「動態位址群組」中的比對準則，並強制執行與每個標記相關聯之基礎 IP 位址的「安全性」原則。

每個標籤的最大長度為 *127* 個字元。如果標籤超過最大字元計數，則予以截斷。如果兩個截斷的標籤相同，則會在標籤中新增唯一雜湊，以區分它們彼此。

您可以使用 Kubernetes 外掛程式，將 Kubernetes 叢集內部署的 Pod、節點、命名空間和服務的 IP 位址與標記對應散佈至實體或 VM-Series 防火牆，即使您尚未在該叢集中部署 CN-Series 防火牆也是一樣。

預先定義的標記	**Panorama** 上的標記格式	收集到的 **IP** 位址
DaemonSet	k8s.cl_<cluster-name>.ns_<namespace>.ds_<pod-name>	Pod IP 位址
ReplicaSet	k8s.cl_<cluster-name>.ns_<namespace>.rs_<pod-name>	Pod IP 位址
StatefulSet	k8s.cl_<cluster-name>.ns_<namespace>.ss_<pod-name>	Pod IP 位址
服務	k8s.cl_<cluster-name>.ns_<namespace>.svc_<svc-name>	叢集 IP 位址 Pod IP 位址
外部服務	k8s.cl_<cluster-name>.ns_<namespace>.exsvc_<svc-name>	外部服務 IP 位址 LoadBalancer IP 位址
節點	k8s.cl_<cluster-name>.nodes	所有節點的私人 IP 位址
外部節點	k8s.cl_<cluster-name>.ex_nodes	所有節點的公用 IP 位址
命名空間	k8s.cl_<cluster-name>.ns_<namespace>	命名空間中的所有叢集 IP 位址 命名空間中的所有 Pod IP 位址

預先定義的標記	**Panorama** 上的標記格式	收集到的 **IP** 位址
介面	• k8s.cl_<cluster-name>.ns_<namespace>.ds_<daemonset-name>.if_<interface> • k8s.cl_<cluster-name>.ns_<namespace>.rs_<replicaset-name>.if_<interface> • k8s.cl_<cluster-name>.ns_<namespace>.ss_<statefulset-name>.if_<interface>	部署中每個 Pod 上所有介面的所有 IP 位址。

如果您使用標籤來組織 Kuberbetes 叢集內的 Pod 和服務，則 Panorama 上的 Kubernetes 外掛程式可以查詢這些標籤，並為您建立標記。下列是支援的使用者定義標籤：

使用者定義的標記	**Panorama** 上的標記格式	收集到的 **IP** 位址
標籤	k8s.cl_<cluster-name>.ns_<namespace>.<label-key>.<label-value>	該命名空間中符合所指定標籤的所有叢集 IP 位址。 該命名空間中符合所指定標籤的所有 Pod IP 位址。
標籤選取器	k8s.cl_<cluster-name>.<selector-name>	符合所指定選取器的所有叢集 IP 位址。 符合所指定選取器的所有 Pod IP 位址。

標籤選取器符合 Kubernetes 叢集內 Pod 和服務的指定標籤，並將與標籤相關聯的 IP 位址對應至單一標記。Kubernetes 外掛程式支援標籤索引鍵和標籤值的集合型和相等型選取器。

下列是支援的相等型選取器：

- `key = value; key ==`
- `value; key != value, for example, app = redis`

您也可以在運算式中使用逗號區隔清單形式來指定多個選取器。例如：

```
app == web, tier != backend
```

下列是支援的集合型選取器：

- key in (value1, value2)
- key notin (value1, value2)，例如，tier notin (frontend, backend)
- key

- !key

針對受監視的「服務物件」，外掛程式會使用下列命名配置來產生 port、targetPort 和 nodePort 服務物件的連接埠：

```
<namespace>-<svc_name>-<type>-<port_value>-<hash>
```

雜湊確保即使您有跨 k8s 叢集的重疊命名空間和服務名稱，服務物件還是唯一的。

啟用檢查已標記的 VLAN 流量

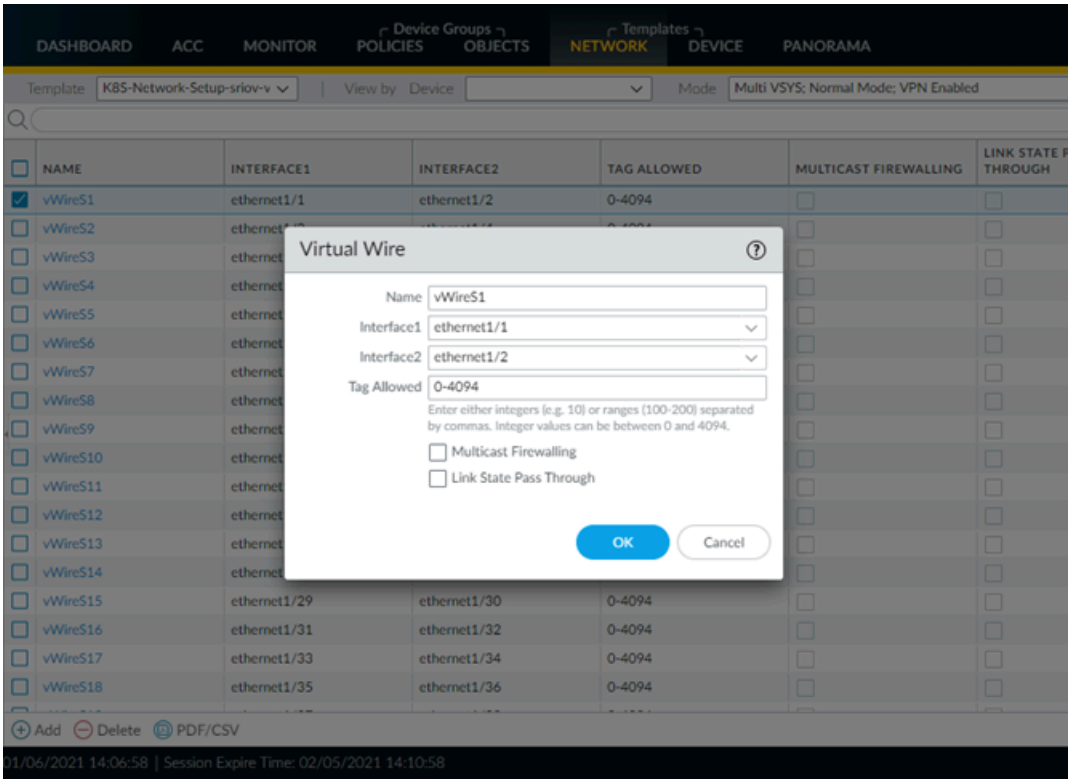
我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

完成下列程序，讓 CN-Series 防火牆檢查已標記的 VLAN 流量。若要檢查 VLAN 標記的流量，您必須更新 Panorama 上所有虛擬連線的設定，以允許所有 VLAN 標記。然後，您必須標註應用程式 Pod YAML 檔案，以將 VLAN 標記指派給應用程式 Pod 介面。此註釋會將套用至透過防火牆傳送之封包的標記告知 CN-NGFW。

不支援雙 *VLAN* 標記。

STEP 1 | 在 CN-NGFW 的所有介面上啟用所有 VLAN。

1. 登入 Panorama。
2. 選取 **Network**（網路）> **Virtual Wires**（虛擬連線）。
3. 從 **Template**（範本）下拉式清單中，選取 **K8S-Network-Setup** 範本。
4. 選取第一個虛擬連線。
5. 將 **Tag Allowed**（允許的標記）設為 0-4094。
6. 針對每個虛擬連線，重複此程序。
7. **Commit**（提交）您的變更。



STEP 2 | 附加具有下列註釋的應用程式 Pod YAML 檔案，以根據介面套用靜態 VLAN ID。

一個介面只支援一個 *VLAN* 標記。

paloaltonetworks.com/interfaces: '[{"name" : "eth0"}, {"name" : "net1", "vlan" : <VLAN-ID> } {"name" : "net2", "vlan" : <VLAN-ID> }]’

For example:

```
 annotations: k8s.v1.cni.cncf.io/networks: bridge-conf-1,bridge-
conf-2,bridge-conf-0,pan-cni paloaltonetworks.com/firewall: pan-fw
 paloaltonetworks.com/interfaces: '[ {"name" : "eth0"}, {"name" :
 "net1", "vlan" :101 }, {"name" : "net2", "vlan" :102 }, {"name" :
 "net3", "vlan" :103 } ]'
```

啟用 IPVLAN

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • Panorama 執行 PAN-OS 10.1.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

IPVLAN 是虛擬網路裝置的驅動程式，可在容器化環境中用來存取主機網路。在 L2 模式中，不論主機網路內建立多少 IPVLAN 裝置，IPVLAN 都會將單一 MAC 位址公開到外部網路。所有邏輯 IP 介面都會使用相同的 MAC 位址。這可讓您避免在父系 NIC 上使用混合式模式，並防止 NIC 或切換器上的可能 MAC 限制。

您現在可以在具有下列限制的 CN-Series 防火牆上使用 IPVLAN。

- 需要 PAN-OS 10.1.2 和更新版本
- 僅限 IPv4
- 僅限 L2 模式
- 一個介面一個 IP 位址
- 如果您要使用 Multus，則請部署 **pan-cni-multus.yaml**，而不是 **pan-cni.yaml**。此外，您必須在部署 Multus 應用程式 Pod 的每個命名空間中部署 pan-cni-net-attach-def.yaml。

相同主機中的 *IPVLAN* 子介面通訊（共用相同的父系介面）無法運作。

您必須為應用程式 Pod yaml 檔案加上註釋，才能啟用 IPVLAN；啟用 IPVLAN 時不需要變更任何 CN-Series yaml 檔案。以下是 IPVLAN 的網路連接定義範例。請注意，模式設定為 “**l2**”。CN-Series 防火牆僅支援 L2 模式。

```
cat ipvlan-nw-10.yaml apiVersion: "k8s.cni.cncf.io/v1"
 kind:NetworkAttachmentDefinition metadata: name: ipvlan-conf-10
 spec: config: '{ "cniVersion":"0.3.0", "name": "ipvlan-conf-10",
 "type": "ipvlan", "master": "eth1", "mode": "l2", "ipam": { "type":
 "static", "addresses": [ { "address":"10.154.102.89/24" } ] } }'
```

在 Panorama 上解除安裝 Kubernetes 外掛程式

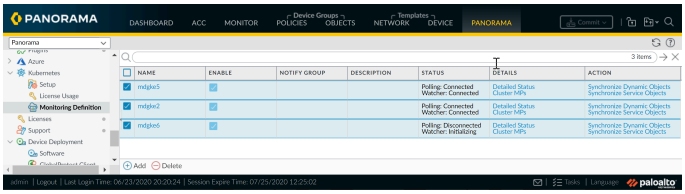
使用下列工作流程以解除安裝 Panorama 上的 Kubernetes 外掛程式，讓您可以成功將所有權杖都傳回給 Palo Alto Networks 授權伺服器，然後清除驗證碼。此工作流程可讓您確定權杖可用於另一個 Panorama。如果您已使用高可用性設定來部署 Panorama 管理伺服器，則必須先完成主動主要 Panorama 上的步驟，再移至被動主要 Panorama 對等節點。

STEP 1 | 如果已部署於 HA 設定中，則登入主動主要 Panorama 對等節點。

1. 從外掛程式中移除所有叢集設定。

   1. 刪除監視定義。

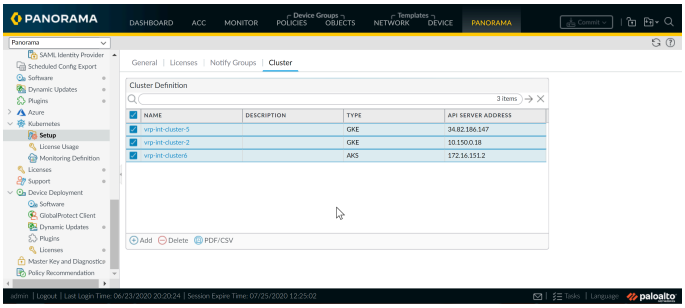
      選取 **Plugins**（外掛程式） > **Kubernetes** > **Monitoring Definition**（監控定義），並選取監控定義，然後選取 **Delete**（刪除）

      。

   2. 刪除 Kubernetes 叢集定義。

      選取 **Plugins**（外掛程式） > **Kubernetes** > **Set up**（設定） > **Cluster**（叢集），並選取叢集定義，然後選取 **Delete**（刪除）。

2. 提交您在 Panorama 上的變更。

   Commit（提交） > **Commit to Panorama**（提交至 **Panorama**）。

3. 驗證使用的權杖計數為零。

   確認所有權杖都已傳回給授權伺服器。

4. 執行清除驗證碼，並確定授權欄驗證碼是 [None（無）]。

5. 移除設定，並提交變更。

   1. 選取 **Plugins**（外掛程式），並尋找您已安裝的 Kubernetes 外掛程式版本，然後 **Remove Config**（移除設定）。

   2. **Commit**（提交） > **Commit to Panorama**（提交至 **Panorama**）。

6. 解除安裝 Kubernetes 外掛程式。

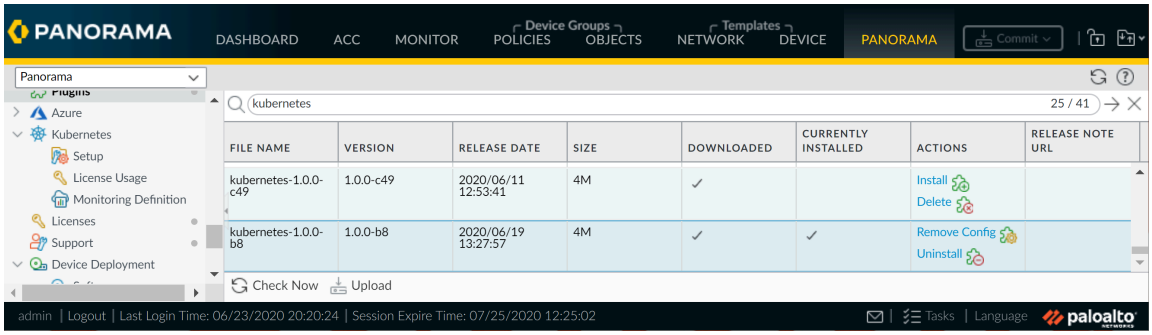
7. 暫停主動 Panorama 端點。

   選取 **Panorama** > **High Availability**（高可用性），然後在 [Operational Commands（操作命令）] 區段中按一下 **Suspend local Panorama**（暫停本機 **Panorama**）連結。

STEP 2 | 登入其他 Panorama 對等節點。

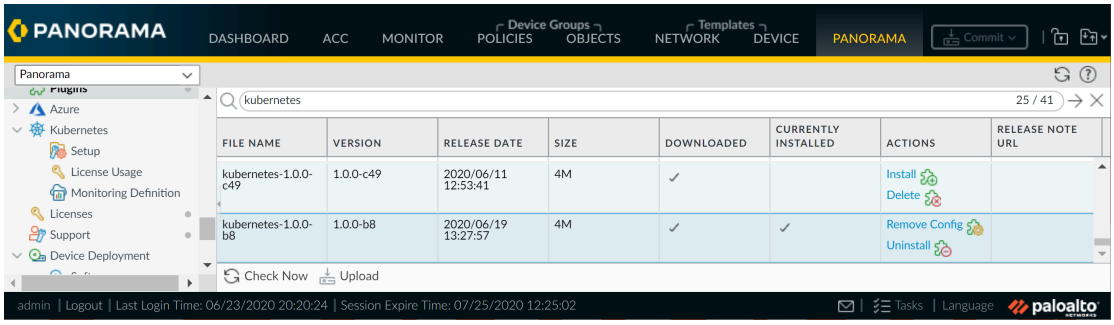
現在，此對等節點是主動次要對等節點。

1. 選取 **Plugins**（外掛程式），並尋找您已安裝的 Kubernetes 外掛程式版本，然後 **Remove Config**（移除設定）。



2. 解除安裝外掛程式。

   1. 選取 **Plugins**（外掛程式），並尋找您已安裝的 Kubernetes 外掛程式版本，然後 **Uninstall**（解除安裝）。



   2. 驗證解除安裝成功。

清除 Panorama 上 CN-Series 防火牆的驗證碼

只有在您移除外掛程式設定並在清除驗證碼之前認可變更時，才會使用下面所列的因應措施。此因應措施可讓您將權杖釋回授權伺服器，讓您可以將它用於另一個 Panorama 設備。

STEP 1 | 1。新增新的外掛程式使用者，並提交變更。

1. 選取 **Panorama** > **Administrators**（管理員）。
2. **Add**（新增）稱為 **__kubernetes** 的新使用者。
3. **Commit**（提交） > **Commit to Panorama**（提交至 **Panorama**）。

STEP 2 | 清除 Panorama 上的驗證碼。

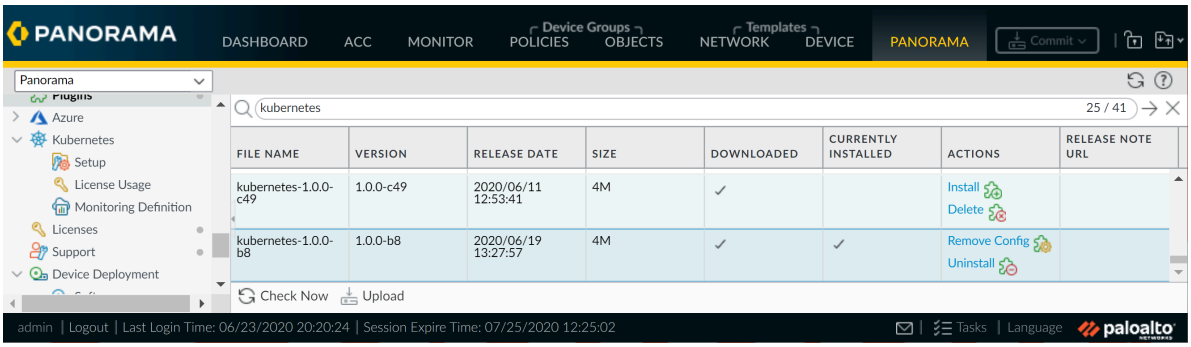
1. 選取 **Panorama** > **Plugins**（外掛程式） > **Kubernetes** > **Setup**（設定） > **Licenses**（授權）。
2. 選取 **Activate/update using authorization code**（使用授權碼啟用/更新）和 **Clear Auth Code**（清除驗證碼）。
3. 驗證授權欄顯示驗證碼 **None**（無）。

STEP 3 | 刪除您已在步驟 1 中建立的外掛程式使用者 **__kubernetes**。

STEP 4 | Commit（提交）您的變更。

STEP 5 | 解除安裝外掛程式。

1. 選取 **Plugins**（外掛程式），並尋找您已安裝的 Kubernetes 外掛程式版本，然後 **Uninstall**（解除安裝）。



2. 驗證解除安裝成功。

CN-Series 上不支援的功能

除非下面另有說明，否則 CN-Series 無法使用 PAN-OS 上支援的下列功能：

功能	DaemonSet	K8s 服務	CNF 模式	HSF 模式
驗證	否。	否。	否。	否。
日誌至 Cortex Data Lake	否。	否。	否。	否。
企業 DLP	否。	否。	否。	否。
非 vWire 介面	否。	否。	是	是
IoT Security	否。	否。	否。	否。
IPv6	是	否。	是	否。
NAT	否。	否。	是	否。
基於原則的轉送	否。	否。	是	否。
QoS	否。	否。	否。	否。
SD-WAN	否。	否。	否。	否。
使用者-ID	否。	否。	是	否。
WildFire 內嵌 ML	否。	否。	否。	否。
SaaS 內嵌	否。	否。	否。	否。
IPSec	否。	否。	否。	否。
通道內容檢查	否。	否。	否。	否。

CN-Series 防火牆的高可用性和 DPDK 支援

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

高可用性 (HA) 是一種單一群組中配置兩個防火牆的組態，且這兩個防火牆的組態會同步處理，防止單點在網路上失效。兩個防火牆對等間的活動訊號連線可確保當其中一個對等損壞時能夠無縫容錯移轉。在兩個裝置叢集中設定防火牆可提供備援能力，並能讓您確保業務連續性。

本章包含以下部分：

CN-Series 防火牆作為 Kubernetes CNF 的高可用性支援

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

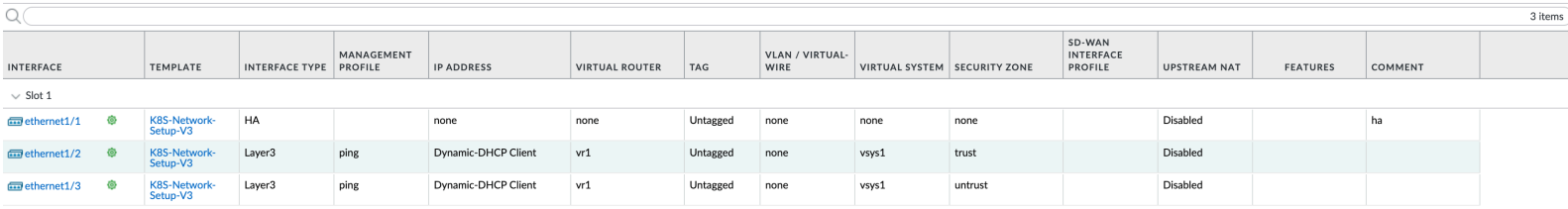
高可用性 (HA) 是一種單一群組中配置兩個防火牆的組態，且這兩個防火牆的組態會同步處理，防止單點在網路上失效。兩個防火牆對等間的活動訊號連線可確保當其中一個對等損壞時能夠無縫容錯移轉。在兩個裝置叢集中設定防火牆可提供備援能力，並能讓您確保業務連續性。

您現在可以在 HA 中部署 CN-series-as-a-kubernetes-CNF。此部署模式僅支援具有工作階段和設定同步的主動/被動 HA。

當您在 HA 中部署 CN-Series-as-a-Kubernetes CNF 時，將會有兩個 PAN-CN-MGMT-CONFIGMAP、PAN-CN-MGMT 和 PAN-CN-NGFW YAML 檔案各用於主動和被動節點。

若要在具有第 3 層支援的 HA 中成功部署 CN-Series 防火牆作為 Kubernetes CNF，請執行下列動作：

- 在 HA 中，每個 Kubernetes 節點都至少應該有三個介面：管理（預設）、HA2 介面和資料介面。
- 針對 L3 模式的 CN-Series 防火牆，至少應該有兩個介面：管理（預設）和資料介面。



3 items

INTERFACE	TEMPLATE	INTERFACE TYPE	MANAGEMENT PROFILE	IP ADDRESS	VIRTUAL ROUTER	TAG	VLAN / VIRTUAL-WIRE	VIRTUAL SYSTEM	SECURITY ZONE	SD-WAN INTERFACE PROFILE	UPSTREAM NAT	FEATURES	COMMENT
Slot 1													
ethernet1/1	K8S-Network-Setup-V3	HA		none	none	Untagged	none	none	none		Disabled		ha
ethernet1/2	K8S-Network-Setup-V3	Layer3	ping	Dynamic-DHCP Client	vr1	Untagged	none	vsys1	trust		Disabled		
ethernet1/3	K8S-Network-Setup-V3	Layer3	ping	Dynamic-DHCP Client	vr1	Untagged	none	vsys1	untrust		Disabled		

- 修改新的網路連接定義 YAML 檔案，並進行下列變更：
  - 確定下列 YAML 檔案中的 PAN_HA_SUPPORT 參數值為 **true**：

    ```
    pan-cn-mgmt-configmap-0.yaml
    ```

    ```
    pan-cn-mgmt-configmap-1.yaml
    ```

  - 執行下列命令，以從超管理器介面擷取 **pciBusID** 值：

    ```
    ethtool -i interface name
    ```

    將上面所擷取的 **pciBusID** 值新增至下列網路定義檔案：

    ```
    net-attach-def-1.yaml
    ```

    ```
    net-attach-def-2.yaml
    ```

    ```
    net-attach-def-3.yaml
    ```

    ```
    net-attach-def-ha2-0.yaml
    ```

    ```
    net-attach-def-ha2-1.yaml
    ```

  - 從 AWS 主控台的對應節點執行個體中擷取 HA2 介面的靜態 IP 位址，並將其新增至 net-attach-def-ha2-0.yaml 和 net-attach-def-ha2-1.yaml 檔案的 address 參數。

  如果您要使用 **Advanced Routing**（進階路由），則請考慮只有 EKS 和內部環境中才支援以 CNF 模式所部署的 CN-Series 防火牆。如果您要搭配使用 **Advanced Routing**（進階路由）與 Kubernetes 3.0.0 外掛程式，則必須在範本堆疊上手動將其設定；在檔案 pan-cn-mgmt-console.yaml 中，設定旗標 PAN_ADVANCED_ROUTING:”true”。

AWS EKS 上 CN-Series 防火牆的高可用性

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

您現在可以在 HA 中部署 CN-Series-as-a-Kubernetes-CNF。此部署模式僅支援具有工作階段和設定同步的主動/被動 HA。

AWS 環境不支援在具 *IPV6* 的 *HA* 中進行 *CN-Series-as-a-Kubernetes CNF* 部署。

為確保備援，您可以在 AWS 上部署 CN-Series 防火牆，採用主動/被動高可用性 (HA) 設定。主動端點連續不斷地將其組態及工作階段資訊與設定相同的被動端點保持同步。如果主動設備關閉，兩部設備間的活動訊號連線可確保故障復原。您可以透過次要 IP 移動，在 HA 的 AWS EKS 上部署 CN-Series 防火牆。

若要確保您網際網路型應用程式的所有流量都會通過防火牆，您可以設定 AWS 進入路由。AWS 進入路由功能可讓您建立路由表與 AWS 網際網路閘道的關聯，並新增透過 CN-Series 防火牆將應用程式流量重新導向的路由規則。這樣的重新導向可確保所有網際網路流量都會通過防火牆，且無需重新設定應用程式端點。

次要移動

當主動對等故障時，被動對等會偵測到此失敗狀況，並變成主動。此外，被動對等還會對 AWS 基礎結構觸發 API 呼叫，將已設定的次要 IP 位址從失敗對等的資料平面介面移給自己。此外，AWS 會更新路由表，以確保流量導向至主動防火牆實例。這兩個操作可確保容錯移轉之後還原輸入和輸出流量工作階段。此選項可讓您利用 DPDK 來改善 CN-Series 防火牆執行個體的效能。

HA 的 IAM 角色

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

AWS 要求所有 API 請求必須使用其簽發的憑證進行加密簽署。為了啟用 CN-Series 防火牆（將部署為 HA 配對）的 API 權限，您必須建立政策，並在AWS 身分識別和存取管理 (IAM) 服務中將該政策附加至角色。角色必須在啟動時附加至 CN-Series 防火牆。此政策將權限給予 IAM 角色來起始必要的 API 動作，以便觸發容錯移轉時，將介面或次要 IP 位址從主動對等移至被動對等。

如需建立政策的詳細指示，請參閱有關建立客戶受管理政策的 AWS 文件。如需建立 IAM 角色，定義哪些帳戶或 AWS 服務可承擔該角色，以及承擔角色後應用程式可使用哪些 API 動作及資源的詳細指示，請參閱 Amazon EC2 的 IAM 角色。

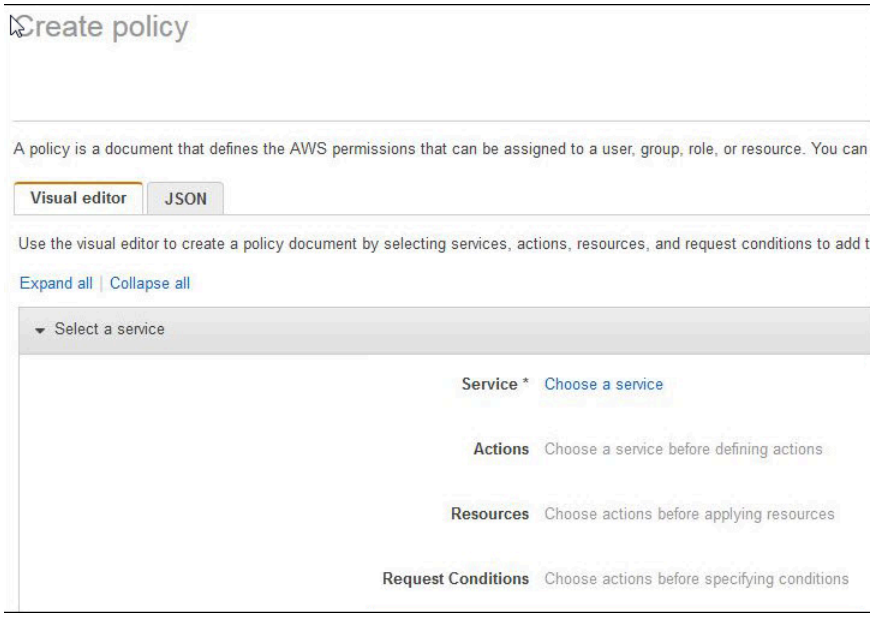
在 AWS 主控台設定的 IAM 原則必須（至少）擁有下列動作與資源：

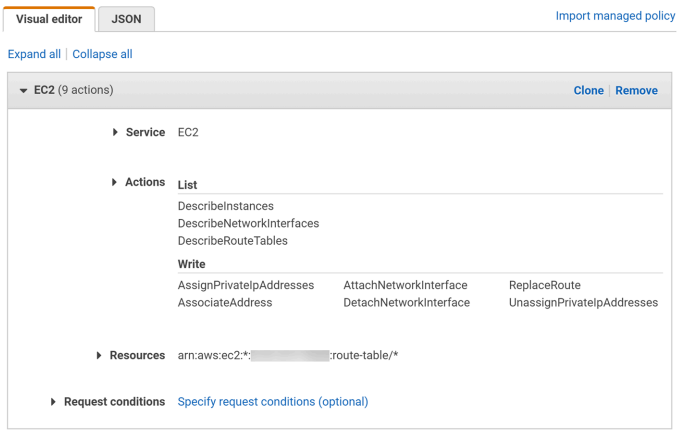
啟用 HA 需要下列 IAM 動作、權限和資源。

IAM 動作、權限或資源	說明	次要 IP 移動
AttachNetworkInterface	允許將 ENI 連接至實例。	✓
DescribeNetworkInterfaces	擷取 ENI 參數以便將介面連接至實例。	✓
DetachNetworkInterface	允許將 ENI 從 EC2 實例分離。	✓
DescribeInstances	允許取得 VPC 中 EC2 實例的相關資訊。	✓
AssociateAddress	允許將主要 IP 位址相關聯的公共 IP 位址，從被動介面移至主動介面。	✓
AssignPrivateIpAddresses	允許將次要 IP 位址和相關聯的公共 IP 位址，指派給被動對等的介面。	✓
DescribeRouteTables	允許擷取與 CN-Series 防火牆執行個體相關聯的所有路由表。	✓
ReplaceRoute	允許更新 AWS 路由表項目。	✓
GetPolicyVersion	允許擷取 AWS 政策版本資訊。	✓
GetPolicy	允許擷取 AWS 政策資訊。	✓
ListAttachedRolePolicies	允許擷取連接至特定 IAM 角色的所有受管理政策的清單。	✓
ListRolePolicies	允許擷取內嵌於特定 IAM 角色中的內嵌政策名稱清單。	✓
GetRolePolicy	允許擷取內嵌於特定 IAM 角色中的特定內嵌政策。	✓

IAM 動作、權限或資源	說明	次要 IP 移動
policy	允許存取 IAM 政策 Amazon Resource Name (ARN)。	✓
role	允許存取 IAM 角色 ARN。	✓
route-table	允許存取路由表 Amazon Resource Name (ARN) 以便於容錯移轉時更新。	✓
萬用字元 (*)	在 ARN 欄位中使用 * 作為萬用字元。	✓

下列螢幕擷取畫面顯示以上針對次要 IP HA 所述 IAM 角色的存取管理設定：





次要 IP 移動 HA 需要的最低權限為：{"Version":"2012-10-17","Statement":
[{"Sid":"VisualEditor0","Effect":"Allow","Action":
["ec2:AttachNetworkInterface","ec2:DetachNetworkInterface","ec2:DescribeInstances","ec2:DescribeNetworkInterfaces

“ec2:AssignPrivateIpAddresses”,“ec2:AssociateAddress”,“ec2:DescribeRouteTables”],"Resource": "*"}{"Sid":"VisualEditor1","Effect":"Allow","Action": "ec2:ReplaceRoute", "Resource": "arn:aws:ec2:*:*:route-table/*"}]}

HA 連結

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

HA 配對中的裝置會使用 HA 連結來同步資料以及維護狀態資訊。在 AWS 上，CN-Series 防火牆使用下列連接埠：

- 控制連結—HA1 連結用來交換 Hello、活動訊號和 HA 狀態資訊，以及路由的管理平面同步。此連結也用於隨端點同步主動或被動設備上的組態變更。

  用於 HA1 的管理連接埠。使用於明碼通訊的 TCP 連接埠 28769 和 28260，或使用於加密通訊的連接埠 28（TCP 上的 SSH）。
- 資料連結—HA2 連結可用於在 HA 配對中同步設備之間的執行階段、轉送表格、IPSec 安全性關聯和 ARP 表格。HA2 連結中的資料流永遠為單一方向性（HA2 保持運作除外）；其流向會從主動設備流往被動設備。

  Ethernet1/1 必須指派為 HA2 連結；如此才能在 HA 中將 CN-Series 防火牆部署在 AWS。HA 資料連結可設定為使用 IP（通訊協定編號 99）或 UDP（埠號 29281）作為傳輸用途。

AWS 上的 CN-Series 防火牆不支援用於 HA1 或 HA2 的備份連結。

活動訊號輪詢與您好訊息

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

防火牆使用您好訊息和活動訊號來驗證端點設備可回應及可操作。您好訊息會以設定的您好間隔在端點間傳送，以確認裝置的狀態。活動訊號是在控制連結上對 HA 端點的 ICMP ping，而該端點會回應 ping 以建立設備間的連線與回應。如需可觸發容錯移轉的 HA 計時器的詳細資訊，請參閱 HA 計時器。（CN-Series 防火牆的 HA 計時器與 PA-5200 系列防火牆的 HA 計時器相同）。

裝置優先順序及先佔

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

可對 HA 配對中的裝置指定裝置優先順序值，以表示喜好的裝置在容錯移轉後可擔任主動角色及管理流量。若要在 HA 配對中使用特定設備來主動保護流量，您必須在兩個防火牆上啟用先佔行為並為每個設備指定設備優先順序。數值較小的設備就等於有較高的優先順序，表示將其指定為主動設備並管理所有網路上的流量。另一部設備則進入被動狀態，並與主動設備的設定和狀態資訊同步，以便隨時在發生故障時轉換為主動狀態。

較低的數值會在首次部署期間變成使用中狀態。如果先部署較高的數值並停用先佔，則較高的數值將會變成使用中狀態。

針對 *AWS* 上 *CN-Series* 防火牆中的 *HA*，不建議使用先佔。

預設會停用防火牆上的先佔。啟用後，先佔行為允許優先順序較高（數值較小）的防火牆在故障復原後繼續擔任主動設備。出現先佔行為時，該事件會記錄在系統日誌中。

若要新增優先順序，您應該確保 `pan-cn-mgmt-configmap-0.yaml` 和 `pan-cn-mgmt-configmap-1.yaml` 檔案中的 `PAN_HA_PRIORITY` 參數值設定為數值。

例如：

```
PAN_HA_PRIORITY: “10”
```

HA 計時器

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client 對於使用 Helm 的 CN-Series 部署

高可用性 (HA) 計時器是用來偵測防火牆失敗及觸發故障復原。若要減少設定 HA 計時器的複雜度，您可以從三個設定檔選取：**Recommended**（建議）、**Aggressive**（積極）和 **Advanced**（進階）。這些設定檔會自動填入最佳的 HA 計時器值，供特定的防火牆平台啟用更快速的 HA 部署。

為一般的故障復原計時器設定使用 **Recommended**（建議）的設定檔，並為較快速的故障復原計時器設定使用 **Aggressive**（積極）設定檔。**Advanced**（進階）設定檔可讓您自訂計時器值以符合您的網路需求。

AWS 上 **CN-Series** 防火牆上的 **HA** 計時器	建議的/積極設定檔的預設值
提升保留時間	2000/500 毫秒
Hello 間隔	8000/8000 毫秒
活動訊號間隔	2000/1000 毫秒
最大擺動旗標數	3/3
先佔保留時間	1/1 分鐘
監控失敗維持時間	0/0 毫秒
其他主機維持時間	500/500 毫秒

使用次要 IP 在 AWS EKS 上設定主動/被動 HA

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本

完成下列程序，以將新的 CN-Series 防火牆部署為具有次要 IP 位址的 HA 配對。

STEP 1 | 在針對 HA 配對部署 CN-Series 防火牆之前，請確保下列事項：

- HA 對等雙方都部署在相同的 AWS 可用性區域。請參閱 HA 的 IAM 角色。
- 部署執行個體時，建立 IAM 角色，並將角色指派給執行 CN-Series 防火牆的工作節點。
- 主動和被動防火牆至少必須各有三個介面：管理介面、HA2 介面和資料介面。

  管理介面預設將會用作 HA1 介面。
- 在與叢集位於相同可用性區域的 AWS 上建立網路介面。在 `eni` 上新增標記，使其不受 AWS 管理，並且可供 multus 使用：

  `node.k8s.amazonaws.com/no_manage:True`
- 確認網路與安全性元件已適當定義。
  - 啟用網際網路通訊。預設 VPC 包括網際網路閘道，如果您在預設子網路中安裝 CN-Series 防火牆，則該防火牆可以存取網際網路。
  - 建立子網路。子網路是指派給 VPC 的 IP 位址範圍區段，您可以在該 VPC 中啟動 EC2 實例。CN-Series 防火牆必須屬於公用子網路，才能設定該防火牆存取網際網路。
  - 建立包含防火牆資料介面的資料安全性群組。此外，設定安全性來允許所有流量，以便防火牆強制執行安全性。必須如此，才能在容錯移轉期間維持現有工作階段。
  - 為私人子網路的路由表新增路由，以確定流量可在 VPC 中整個子網路與安全性群組之間路由（若適當的話）。

在 *EKS* 上部署 *CN-Series* 防火牆時，如果 *`http-put-response-hop-limit`* 值設定為預設值 *1*，則 *IMDSv2* 權杖擷取會失敗。您必須確保在啟用 *IMDSv2* 時將躍點限制值設定為等於或大於 *3*。

例如：

執行下列命令：

```
aws ec2 modify-instance-metadata-options --instance-id
<your-instance-id> --http-tokens required --http-endpoint
enabled --http-put-response-hop-limit 3
```

STEP 2 | 在 EKS 上部署 CN-Series 防火牆。

1. 在每個 HA 對等上，設定乙太網路 1/1 作為 HA2 介面。
   1. 開啟 Amazon EC2 主控台。
   2. 選取 [Network Interface（網路介面）]，然後選取您的網路介面。
   3. 選取 **Actions**（動作） > **Manage IP Addresses**（管理 **IP** 位址）。
   4. 將此欄位空白，以允許 AWS 動態指派 IP 位址，或輸入 CN-Series 防火牆子網路範圍內的 IP 位址。這將指派 HA2 介面的次要 IP。
   5. 按一下 **Yes**（是）和 **Update**（更新）。
   6. 選取 **Actions**（動作） > **Change Source/Dest**（變更來源/**Dest**）。勾取 ， 然後選取 **Disable**（停用）。
   7. 在第二個（成為被動）HA 對等上重複此程序。
2. 將次要 IP 位址新增至第一個（成為主動）HA 對等上的資料平面介面。
   1. 選取 **Network Interface**（網路介面），然後選取您的網路介面。
   2. 選取 **Actions**（動作） > **Manage IP Addresses**（管理 **IP** 位址） > **IPv4 Addresses**（**IPv4** 位址） > **Assign new IP**（指派新 **IP**）。
   3. 將此欄位空白，以允許 AWS 動態指派 IP 位址，或輸入 CN-Series 防火牆子網路範圍內的 IP 位址。
   4. 按一下 **Yes**（是）和 **Update**（更新）。
3. 將次要彈性（公共）IP 位址與主動對等的不受信任介面建立關聯。
   1. 選取 **Elastic IPs**（彈性 **IP**），然後選取要建立關聯的彈性 IP 位址。
   2. 選取 **Actions**（動作） > **Associate Elastic IP**（與彈性 **IP** 位址建立關聯）。
   3. 在 **Resource Type**（資源類型）下方，選取 **Network Interface**（網路介面）。
   4. 選取要與彈性 IP 位址建立關聯的網路介面。
   5. 按一下 **Associate**（關聯）。
4. 若要檢查輸出流量，請在子網路路由表中新增項目，將下一個躍點設定為防火牆信任介面。
   1. 選取 **VPC** > **Route Tables**（路由表）。
   2. 選擇子網路路由表。
   3. 選取 **Actions**（動作） > **Edit routes**（編輯路由） > **Add route**（新增路由）。
   4. 輸入 **Destination**（目的地） CIDR 區塊或 IP 位址。
   5. 在 **Target**（目標）中，輸入防火牆信任介面的網路介面。
   6. 按一下 **Save routes**（儲存路由）。
5. 若要使用 AWS 進入路由，請建立路由表，並將網際網路閘道與路由表建立關聯。然後，新增項目，將下一個躍點設定為主動防火牆不受信任介面。
   1. 選取 **Route Tables**（路由表） > **Create Route Table**（建立路由表）。

2. （選用）輸入路由表的描述性 **Name tag**（名稱標籤）。
3. 按一下 **Create**（建立）。
4. 按一下您的路由表，選取 **Actions**（動作） > **Edit edge associations**（編輯邊緣關聯）。
5. 選取 **Internet gateways**（網際網路閘道），然後選擇您的 VPC 網際網路閘道。
6. 按一下 **Save**（儲存）。
7. 按一下您的路由表，選取 **Actions**（動作） > **Edit routes**（編輯路由）。
8. 在 **Target**（目標）中，選取 **Network Interface**（網路介面），然後選擇主動防火牆的不受信任介面。
9. 按一下 **Save routes**（儲存路由）。

STEP 3 | 啟用 HA。

若要啟用 HA 支援，您應該確保下列 YAML 檔案中的 PAN_HA_SUPPORT 參數值為 true：

- `pan-cn-mgmt-configmap-0.yaml`
- `pan-cn-mgmt-configmap-1.yaml`

系統會自動設定對等 HA1 IP 位址。

STEP 4 | 從 AWS 主控台的對應節點執行個體中擷取 HA2 介面的靜態 IP 位址，並將其新增至 `net-attach def-ha2-0.yaml` 和 `net-attach-def-ha2-1.yaml` 檔案的 address 參數。

（選用）修改 **HA2 Keep-alive**（**HA2** 保持活動）封包的 **Threshold**（閾值）。依預設，可啟用 **HA2 Keep-alive**（**HA2** 保持運作）對端點之間 HA2 資料連結進行監控。如果發生故障且超過此臨界值（預設值為 10000 毫秒），則會發生定義的動作。發生 HA2 keep-alive（HA2 保持運作）故障時，會產生重要系統日誌訊息。

您可以在兩個設備都設定 ***HA2 keep-alive***（***HA2*** 保持運作）選項，或僅設定 *HA* 配對中的一個設備。如果您在一部設備上啟用此選項，僅該設備會傳送保持運作訊息。

STEP 5 | 驗證防火牆在主動/被動 HA 中是否配對。

1. 存取兩個防火牆上的 **Dashboard**（儀表板），然後檢視高可用性 Widget。
2. 在主動 HA 對等上，按一下 **Sync to peer**（同步處理至對等）。
3. 確認防火牆已配對並同步。
   - 在被動防火牆上：本機防火牆狀態應顯示為 **Passive**（被動），而 **Running Config**（執行中設定）應顯示為 Synchronized（已同步）。
   - 在主動防火牆上：本機防火牆狀態應顯示為 **Active**（主動），而執行中設定應顯示為 **synchronized**（已同步）。
4. 從防火牆 command line interface（命令列介面 - CLI），執行下列命令：
   - 確認容錯移轉備妥情況：

     ```
     show plugins vmw_series aws ha state
     ```
   - 顯示次要 IP 對應：

     ```
     show plugins vm_series aws ha ips
     ```

在 CN-Series 防火牆上設定 DPDK

我可以在哪裡使用這個？	我需要哪些內容？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • Panorama 執行 PAN-OS 10.2.x 或更高版本 • Helm 3.6 or above version client

資料平面開發套件 (DPDK) 提供資料平面應用程式中快速封包處理的簡單架構。

只有 *CN-Series* 防火牆作為 *Kubernetes* 容器網路函數 *(CNF)* 才支援 *DPDK* 模式。

DPDK 模式不支援 *DHCP IPAM*。

系統需求

若要執行 DPDK 應用程式，您必須在目標機器上進行下列自訂。

- 核心設定—在主機 OS 核心中啟用 HUGETLBFS 選項。
- **KNI** 和 **UIO/VFIO**—在主機 OS 核心中插入 KNI 和 UIO/VFIO。

- 巨型分頁

1. 保留巨型分頁

   - 在 Pod 啟動之前，在執行階段期間保留巨型分頁。將所需的巨型分頁數目新增至 `/sys/kernel/` 目錄中對應至特定分頁大小（以 KB 為單位）的 `nr_hugepages` 檔案。例如，如果需要 1024 個 2M 分頁，則請針對單一節點系統使用下列命令。

     ```
     echo 1024 > /sys/kernel/mm/hugepages/hugepages-2048kB/
     nr_hugepages
     ```

   - 在啟動期間保留巨型分頁。例如，若要將 4G 記憶體的巨型分頁保留為 4 個 1G 分頁，應該將下列選項傳遞給核心。

     ```
     default_hugepagesz=1G hugepagesz=1G hugepages=4
     ```

2. 搭配使用巨型分頁與 **DPDK**—為巨型分頁建立裝載點，因為 PanOS 10.2 使用 DPDK 次要程序。

   下列範例命令會建立大小為 1 GB 的巨型分頁以供 DPDK 使用。

   ```
   mkdir /mnt/huge mount -t hugetlbfs pagesize=1GB /mnt/huge
   ```

3. 使用下列命令啟用巨型分頁之後，請重新啟動主機上的 kubelet 服務。

   ```
   sudo systemctl restart kubelet
   ```

4. 檢查 `/sys/fs/cgroup/hugetlb/kubepods.slice/hugetlb.2MB.limit_in_bytes` 以確保大小與巨型分頁大小相符。如果大小與巨型分頁大小不相符，則請使用下列命令來更新大小。

   ```
   echo 2147483648 > /sys/fs/cgroup/hugetlb/kubepods.slice/
   hugetlb.2MB.limit_in_bytes
   ```

   在 *Pod* 中，應用程式可以配置以及使用預先配置的多種大小的巨型分頁。應用程式會使用資源名稱 *hugepages-<size>*，以透過容器層級資源需求來使用巨型分頁。例如，*hugepages-2Mi* 或 *hugepages-1Gi*。

   與 *CPU* 或記憶體不同，巨型分頁不支援過度提交。

啟用特權模式，以存取主機裝置空間。若要列出網路裝置，並將其繫結至容器，請將 `/sys` 裝載至容器，讓 *DPDK* 可以存取目錄內的檔案。

下列是在 *DPDK* 上啟用巨型分頁的程式碼片段。

```
requests: cpu:"1" memory:"4Gi" hugepages-2Mi:4Gi limits:
 cpu:"1" memory:"4Gi" hugepages-2Mi:4Gi volumeMounts:
 - mountPath: /sys name: sys - mountPath: /dev name:
 dev - mountPath: /dev/shm name: dshm - mountPath: /
run/tmp name: hosttmp - mountPath: /etc/pan-fw-sw
 name: sw-secret envFrom: - configMapRef: name: pan-
ngfw-config-0 env: - name:CPU_REQUEST valueFrom:
 resourceFieldRef: containerName: pan-ngfw-container
 resource: requests.cpu - name:CPU_LIMIT valueFrom:
 resourceFieldRef: containerName: pan-ngfw-container
 resource: limits.cpu - name:MEMORY_REQUEST valueFrom:
 resourceFieldRef: containerName: pan-ngfw-container
 resource: requests.memory - name:MEMORY_LIMIT
 valueFrom: resourceFieldRef: containerName: pan-ngfw-
container resource: limits.memory - name:MY_POD_UUID
 valueFrom: fieldRef: fieldPath: metadata.uid -
 name:MY_NODE_NAME valueFrom: fieldRef: fieldPath:
 spec.nodeName - name:MY_POD_NAME valueFrom: fieldRef:
 fieldPath: metadata.name - name:MY_POD_NAMESPACE
 valueFrom: fieldRef: fieldPath: metadata.namespace
 - name:MY_POD_SERVICE_ACCOUNT valueFrom: fieldRef:
 fieldPath: spec.serviceAccountName - name:MY_POD_IP
 valueFrom: fieldRef: fieldPath: status.podIP volumes:
 - name: sys hostPath: path: /sys - name: dev hostPath:
 path: /dev - name: hosttmp hostPath: path: /tmp/pan -
 name: dshm emptyDir: medium:Memory - name: sw-secret
 secret: secretName: pan-fw-sw
```

- **NUMA** 和 **CPU** 釘選—多個 DPDK 程序不能在相同核心上執行，因為其會導致記憶體集區快取損壞和其他問題。次要程序會釘選至不同的核心。使用 *configmap* 中的 CPU 釘選選項來控制次要程序。
- 設定和 **Pod** 變更
  - 在 `pan-cn-ngfw-configmap-0.yaml` 和 `pan-cn-ngfw-configmap-1.yaml` 中，啟用 `PAN_DATA_MODE: "dpdk"`。

    DPDK 不是 *CN-Series-as-a-kubernetes-CNF* 的預設模式。

  - 符合 `#HUGEPAGE_MEMORY_REQUEST` 參數與 `pan-cn-ngfw-configmap-0.yaml` 和 `pan-cn-ngfw-configmap-1.yaml` 中的巨型分頁記憶體要求。

    如果巨型分頁記憶體無法使用，則預設為 *MMAP*。

如需詳細資訊，請參閱 DPDK 系統需求。

您可以在內部部署工作節點和 AWS EKS 叢集上設定 DPDK

- 在內部部署工作節點上設定 DPDK
- 在 AWS EKS 上設定 DPDK

在內部部署工作節點上設定 DPDK

STEP 1 | 安裝以下相依性：

在要設定 DPDK 的工作節點上執行所有命令。

- 如果是 CentOS：

```
yum groupinstall 'Development Tools' -y yum install net-tools
 pciutils -y yum install git gcc make -y yum install numactl-
devel -y yum install which -y yum install -y sudo libhugetlbfs-
utils libpcap-devel kernel kernel-devel kernel-headers yum
 update -y yum install epel-release -y yum install python36 -y
```

- 如果是 Ubuntu OS：

```
sudo apt install build-essential sudo apt-get install libnuma-
dev
```

STEP 2 | 安裝相依性之後：

- 從 https://fast.dpdk.org/rel/ 下載 DPDK tar 檔案。編譯步驟請參考 DPDK 文件。

```
wget https://fast.dpdk.org/rel/dpdk-19.11.9.tar.xz
```

- 將檔案解壓縮。

```
tar -xvf dpdk-19.11.9.tar.xz cd dpdk-stable-19.11.9
```

- 編譯檔案。編譯後的檔案將位於 `x86_64-native-linuxapp-gcc` 子資料夾

```
make install T=x86_64-native-linuxapp-gcc
```

STEP 3 | 在執行階段以統計方式或動態方式插入已編譯的核心模組 (modprobe/insmod)。如需更多詳細資訊，請參閱核心模組。

```
cd x86_64-native-linuxapp-gcc/kmod insmod igb_uio.ko insmod
 rte_kni.ko
```

如果您在 *Ubuntu* 上看到錯誤—*`insmod`*：錯誤：無法插入模組 *`igb_uio.ko`*，先插入 *`uio`* 模組。

```
modprobe uio
```

STEP 4 | 使用特定散布方式在開機時插入模組。或者，您可以建立一個在每次系統啟動時執行 modprobe/insmod 命令的服務。

```
cp <service-file> to /etc/systemd/system sudo systemctl daemon-
reload
```

STEP 5 | 啟動並掛載 2048K 的 2M 巨型分頁。

您也可以使用步驟 4 的服務指令碼啟動巨型分頁。

```
echo 2048 > /sys/devices/system/node/node0/hugepages/
hugepages-2048/nr_hugepages echo 4292967296 > /sys/fs/cgroup/
hugetlb/kubepods.slice/hugetlb.2MB.limit_in_bytes mkdir /mnt/huge
 mount -t hugetlbfs nodev /mnt/huge
```

STEP 6 | 建立 VM 的快照以供日後使用。

在 AWS EKS 上設定 DPDK

在 AWS EKS 上，每個 Pod 都會有一個由 Amazon VPC CNI 外掛程式所指派的網路介面。使用 Multus，您可以建立具有多個介面的 Pod。

STEP 1 | 如果您還沒有 AWS 帳戶，則請建立 AWS 帳戶。

STEP 2 | 使用自訂 AMI 建立 EKS 叢集。如需詳細資訊，請參閱建立 Amazon EKS 叢集。

STEP 3 | 修改 VPC 和節點設定。如需詳細資訊，請參閱 AWS EKS 文件。

STEP 4 | (Multus) 將多個 ENI 新增至 EKS 節點，並載入 KNI 和 UIO 驅動程式。

- 使用下列標記，以將多個 ENI 新增至 EKS 節點。

  ```
  'Key': 'node.k8s.amazonaws.com/no_manage', 'Value': 'true'
  ```

  偵測到該標記時，Multus CNI 接著可以使用此介面。如需詳細資訊，請參閱 AWS 文件。

- 在 AWS CLI 中執行下列命令。

  ```
  aws ec2 create-network-interface --subnet-id <>
   --description "test" --groups <> --region=us-
  west-1 --tag-specifications 'ResourceType=network-
  interface,Tags=[{Key='node.k8s.amazonaws.com/
  no_manage',Value='true'}]' aws ec2 attach-network-interface --
  network-interface-id <> --instance-id <> --device-index 2
  ```

- （如果您未使用自訂 AMI）在工作節點上啟用巨型分頁。

  ```
  echo 1024 > /sys/devices/system/node/node0/hugepages/
  hugepages-2048kB/nr_hugepages mkdir -p /mnt/huge mount -t
   hugetlbfs nodev /mnt/huge service kubelet restart
  ```