paloalto TECHDOCS

CN 系列防火墙部署模式

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

快速入门 - CN 系列防火墙部署

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

开始使用以下步骤部署 CN-Series：

1. 登录到 CSP 帐户，然后激活积分。
2. 创建部署配置文件。
3. 在 CN 系列防火墙上安装设备证书。
4. 为 CN 系列安装 Kubernetes 插件并设置 Panorama。
5. 从 Palo Alto Networks GitHub 存储库下载 CN 系列部署文件。从 Native-k8s 文件夹获取文件，以用于原生 Kubernetes 本地或云部署
6. 使用或不使用 HELM 图表存储库部署 CN-Series。

   建议使用 ***HELM*** 图表部署 *CN* 系列防火墙。

7. 配置 Panorama 以保护 Kubernetes 部署

您可以选择以下部署模式来部署 CN 系列防火墙：

- 将 CN 系列防火墙部署为 Kubernetes 服务（推荐的部署模式）- 采用集群部署模式部署 CN 系列防火墙。这种部署模式使用自动扩展功能，提高了利用率，降低了成本，并通过基于原生 kubernetes 的部署模式提高了扩展性。
- 将 CN 系列防火墙部署为守护进程- 采用分布式部署模式部署 CN 系列防火墙。当每个环境中需要保护的节点数量较少时，这种部署模式更为合适。
- 将 CN 系列防火墙部署为 Kubernetes CNF- 这种部署模式可以同时保护容器和非容器工作负载。您可以将其部署为独立的第 3 层部署。

CN 系列防火墙的部署模式

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

在使用 CN 系列防火墙保护 Kubernetes 工作负载的安全中查看 CN 系列核心构建块和该工作流程的高级概述后，您可以开始部署 CN 系列防火墙，以保护同一集群中容器之间，以及容器和其他工作负载类型（例如虚拟机和裸机服务器）之间的流量。

如果在 OpenShift 环境中，请参阅；如需保护 5G 流量，请参阅使用 CN 系列防火墙保护 5G。

您需要标准 *Kubernetes* 工具（例如 *kubectl* 或 *Helm*）部署和管理 *Kubernetes* 集群、应用程序和防火墙服务。*Panorama* 并非旨在成为 *Kubernetes* 集群部署和管理的 *Orchestrator*。托管 *Kubernetes* 提供商已提供用于集群管理的模板。您还可以使用社区支持的模板部署具有 Helm 和 Terraform 的 *CN* 系列。

- 将 CN 系列防火墙部署为 Kubernetes 服务（推荐的部署模式）
- 将 CN 系列防火墙部署为守护进程
- 将 CN 系列防火墙部署为 Kubernetes CNF
- 以独立模式部署 Kubernetes CNF L3

从*CN* 系列即 *DaemonSet* 部署迁移到 *CN* 系列即服务之前（反之亦然），您必须删除并重新应用 `plugin-serviceaccount.yaml`。

- 部署 *CN* 系列即 *DaemonSet* 时，不能存在 `pan-plugin-cluster-mode-secret`。
- 部署 *CN* 系列即 *Kubernetes* 服务时，必须存在 `pan-plugin-cluster-mode-secret`。

将 CN 系列防火墙部署为 Kubernetes 服务（推荐的部署模式）

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 图表部署 CN 系列的 Helm 3.6 or above version client

完成以下过程以将 CN 系列防火墙部署为 Kubernetes 服务。

在开始之前，请确保 CN 系列 YAML 文件版本与 PAN OS 版本兼容。

- PAN-OS 10.1.2 或更高版本需要 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 | 设置 Kubernetes 集群。

1. 验证集群是否有足够的资源。确保该群集具有 CN 系列先决条件资源，以便支持防火墙。

   ```
   kubectl get nodes
   ```

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

   查看命令输出中容量标题下的信息，以了解指定节点上可用的 CPU 和内存。

   CPU、内存和磁盘存储分配将取决于您的需求。请参阅 CN 系列的性能和扩展性。

   确保有以下信息：

   - 收集端点 IP 地址，用于在 Panorama 上设置 API 服务器。Panorama 使用此 IP 地址连接到 Kubernetes 集群。
   - 从 Panorama 收集模板堆栈名称、设备组名称、Panorama IP 地址和可选的日志收集器组名称。
   - 收集 VM 身份验证密钥以及自动注册 PIN ID 和值。
   - 将映像下载到的容器映像存储库的位置。

STEP 2 | （可选）如果您在 Kubernetes 插件中为 Panorama 配置了自定义证书，则必须通过执行以下命令来创建证书密钥。不要更改 ca.crt 中的文件名。pan-cn-mgmt.yaml 和 pan-cn-ngfw.yaml 中自定义证书的数量是可选的。

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

STEP 3 | 编辑 YAML 文件以提供部署 CN 系列防火墙所需的详细信息。

您需要替换 YAML 文件中的映像路径，以包括私有注册表的路径并提供所需的参数。有关详细信息，请参阅CN 系列部署 YAML 文件中的可编辑参数。

STEP 4 | （仅限 AWS Outpost 上 EKS 的 CN 系列）更新存储类。要支持部署在 AWS Outpost 上的 CN 系列，您必须使用存储驱动程序 aws-ebs-csi-driver，以确保 Outpost 在动态持久性卷 (PV) 创建期间从 Outpost 拉取卷。

1. 应用以下 yaml。

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

2. 验证 ebs-sc 控制器是否正在运行。

   kubectl -n kube-system get pods

3. 更新 pan-cn-storage-class.yaml 以匹配以下示例。

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

4. 将 **storageClassName: ebs-sc** 添加到 pan-cn-mgmt.yaml 的如下所示位置。

```
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 | 如果您在 Kubernetes 环境中使用自动缩放，请在继续之前参阅水平 Pod 自动缩放。

STEP 6 | 部署 CN-NGFW 服务。

1. 使用 pan-cni-serviceaccount.yaml 文件验证您是否已创建服务帐户。

   请参阅创建用于集群身份验证的服务帐户。

2. 使用 Kubectl 运行 pan-cni-configmap.yaml 文件。

   kubectl apply -f pan-cni-configmap.yaml

3. 使用 kubectl 运行 pan-cn-ngfw-svc.yaml。

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

   该 *yaml* 必须在 *pan-cni.yaml* 之前部署。

4. 使用 Kubectl 运行 pan-cni.yaml 文件。

   kubectl apply -f pan-cni.yaml

5. 验证是否已修改 pan-cni-configmap 和 pan-cni YAML 文件。
6. 运行以下命令并验证输出是否与以下示例相似。

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

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

STEP 7 | 部署 CN-MGMT StatefulSet。

默认情况下，部署管理平面作为提供容错功能的 StatefulSet。最多可以将 30 个防火墙 CN-NGFW Pod 连接到 CN-MGMT StatefulSet。

1. （仅对于静态配置的 PV 为必需）为 CN-MGMT StatefulSet 部署持久卷 (PV)。

   1. 创建目录以匹配 pan-cn-pv-local.yaml 文件中定义的本地卷名称。

      在至少 2 个工作节点上需要六 (6) 个目录。登录将在其中部署 CN-MGMT StatefulSet 的每个工作节点，以创建目录。例如，要创建名为 /mnt/pan-local1 到 /mnt/pan-local6 的目录，请使用以下命令：

      ```
      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。

      在 `nodeaffinity` 下匹配主机名，并验证是否已修改在 `spec.local.path` 中创建的上述目录，然后部署文件以创建新的存储类 pan-local-storage 和本地 PV。

2. 验证是否已修改 pan-cn-mgmt-configmap 和 pan-cn-mgmt YAML 文件。

   EKS 中的 pan-cn-mgmt-configmap 示例。


```
 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
```

pan-cn-mgmt.yaml 文件示例

```
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 运行 yaml 文件。

   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

   如果您之前未完成为集群身份验证创建服务帐户，则必须运行 pan-mgmt-serviceaccount.yaml。

4. 验证 CN-MGMT Pod 是否启动。

   大约需要 5 至 6 分钟的时间。

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

STEP 8 | 部署 CN-NGFW Pod。

1. 验证是否已按 PAN-CN-NGFW-CONFIGMAP 和 PAN-CN-NGFW 中的详细说明修改 YAML 文件。

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

2. 使用 Kubectl apply 运行 pan-cn-ngfw-configmap.yaml 文件。

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

3. 使用 Kubectl apply 运行 pan-cn-ngfw.yaml 文件。

   kubectl apply -f pan-cn-ngfw.yaml

4. 确认已部署 CN-NGFW Pod。

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

STEP 9 | 在 CN 系列上启用水平 Pod 自动缩放.

STEP 10 | 验证您是否可以在 Kubernetes 集群上看到 CN-MGMT、CN-NGFW 和 PAN-CNI。

kubectl -n kube-system get pods

STEP 11 | 为应用程序 yaml 文件或命名空间添加注释，以便将来自其新 Pod 的流量重定向到防火墙。

您需要添加以下注解，以将流量重定向到 CN-NGFW 来进行检查：

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

例如，对于“default”命名空间中的所有新 Pod：

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

在某些平台上，当在 *CNI* 插件链中未激活 *pan-cni* 时，可以启动应用程序 *Pod*。为避免此类情况，您必须按如下在应用程序 *Pod YAML* 中指定卷。

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

STEP 12 | （可选）某些流量可以绕过基于 *PortInfo* 自定义资源的防火墙：

1. 应用 PortInfo 自定义资源定义 YAML

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

2. 以 pan-cn-ngfw-port-cr.yaml 为例，创建一个 PortInfo 自定义资源，其中包含要绕过的协议和端口。它仅在应用程序 Pod 的出站方向上，支持 TCP 和 UDP，最多包含 10 个单独的端口（而不是端口范围）。


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

3. 应用 PortInfo 自定义资源 YAML。

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

4. 除了 pan-fw 注释，还要注释应用 Pod。注释应在应用 Pod 启动时显示。

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

STEP 13 | 在集群中部署应用程序。

在 CN 系列上启用水平 Pod 自动缩放

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 图表部署 CN 系列的 Helm 3.6 or above version client

水平 Pod 自动所放弃 (HPA) 是在所有云环境中可用的 Kubernetes 资源，它可以根据监控的指标自动扩展部署中 CN-MGMT 和 CN-NGFW Pod 的数量。HPA 在所有云环境中使用两个标准指标（CPU 和内存利用率）以及特定于每个云环境的自定义指标。因此，每个云都需要特定的 yaml 文件才能在 AKS、EKS 和 GKE 中启用 HPA。

HPA 使用特定于云的指标适配器从云环境中的监控适配器（例如 EKS 中的 CloudWatch）检索指标数据，从而根据您定义的阈值确定何时向上或向下扩展。您必须修改必要的 yaml 文件来设置最小和最大副本数、每个指标的阈值以及在自动扩展防火墙时使用的指标。

在 *PAN OS 10.1* 中，如果使用 *CN-MGMT Pod HPA* 扩展功能，则可以扩展许多没有连接 *DP Pod* 的 *CN-MGMT Pod*。建议创建 *CN-MGMT Pod* 的最大副本数，以防止不必要的扩展。

云环境	指标		平均值
AKS、EKS 和 GKE	CN-MGMT	panloggingrate	日志计数
		pandataplaneslots	数据平面插槽计数

云环境	指标		平均值
	CN-NFGW	dataplanecpuutilizationpct	CN-NGFW CPU 利用率百分比
		dataplanepacketbufferutilization	CN-NGFW 数据包缓冲区利用率百分比
		pansessionactive	CN-NGFW 上的活动会话数
		pansessionutilization	会话利用率百分比
		pansessionsslproxyutilization	会话 SSL 代理利用率百分比
		panthroughput	以 kbps 为单位的吞吐量
		panpacketrate	以每秒数据包数为单位的数据包速率 (pps)
		panconnectionspersecond	每秒连接数

下面的示例中是 EKS 的 pan-cn-hpa-dp.yaml 文件。此示例使用数据平面 CPU 利用率百分比自动缩放 CN-NGFW Pod。集群将向上扩展 (25#)。如果 CPU 利用率达到 50%，集群将再部署一个 Pod。如果 CPU 利用率达到 75#，集群将再部署两个 Pod。这取决于将总指标除以指标阈值，然后部署足够多的 Pod，才能在集群中的所有 CN-NGFW Pod 中将指标降低到配置阈值。但是，集群部署的 CN-NGFW Pod 不会超过 maxReplicas。如果多个指标同时超过阈值，集群将部署必要数量的 Pod 来满足更高指标的要求。

默认情况下，HPA 适配器每 15 秒轮询一次指标适配器。如果您指定的指标超过配置的阈值 60 秒，集群将部署额外的 CN-NGFW Pod。然后，集群会等待 300 秒（五分钟），然后再决定是否需要额外的 CN-NGFW Pod。默认情况下，一次部署一个 Pod。接着，集群会在 300 秒后检查指标（本例中为 CPU 利用率）。如果利用率下降到不再需要 Pod 的水平，集群将删除一个 Pod。然后，集群将再等待 60 秒，然后再决定是否可以移除另一个 Pod。

您可以修改下面显示的所有值以及任何指标的值，使其适合您的部署。

```
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 | 在集群中部署 Azure 应用程序见解实例。您必须提供所需的 Azure 应用程序见解工具密钥和 Azure 应用程序洞察 App-ID API 密钥作为 K8s 密钥。

STEP 2 | 从 Palo Alto Networks GitHub 存储库下载 AKS 特定的 HPA yaml 文件。

STEP 3 | 如果 CN-MGMT 部署在自定义命名空间中，请使用自定义命名空间更新 pan-cn-adapater.yaml。默认命名空间为 **kube-system**。

STEP 4 | 如果未执行此操作，请在 AKS 特定的 **pan-cn-mgmt-configmap.yaml** 中更新 HPA 参数。

```
 #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 | 编辑 **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 | 将在上面创建的 HPA 名称添加到 **pan-cn-custommetrics.yaml** 中的相应位置。

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

1. 输入副本的最小和最大数量。
2. （可选）更改缩小和放大频率值以使其适合您的部署。如果不更改这些值，则会使用默认值。
3. 为要用于缩放的每个指标复制以下部分。

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

4. 更改要使用的指标的名称，然后将 **AveragValue** 设置为上表中所述的阈值。如果不更改这些值，则会使用默认值。
5. 保存更改。

STEP 8 | 部署 HPA yaml 文件。必须按照下面所述的顺序部署这些文件。

1. 使用 Kubectl 运行 pan-cn-hpa-secret.yaml

   kubectl apply -f pan-cn-hpa-secret.yaml
2. 使用 Kubectl 运行 pan-cn-adapter.yaml

   kubectl apply -f pan-cn-adapter.yaml
3. 使用 Kubectl 运行 pan-cn-custommetrics.yaml

   kubectl apply -f pan-cn-custommetrics.yaml
4. 使用 Kubectl 运行 pan-cn-hpa-dp.yaml

   kubectl apply -f pan-cn-hpa-dp.yaml
5. 使用 Kubectl 运行 pan-cn-hpa-mp.yaml

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

STEP 9 | 验证部署。

- 使用 kubectl 验证自定义指标命名空间中的自定义指标适配器 Pod 是否存在。

  kubectl get pods -n custom-metrics
- 使用 kubectl 检查 HPA 资源。

  kubectl get hpa -n kube-system

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

EKS

STEP 1 | 在 CN 系列即服务集群中部署适用于 Kubernetes 的 Amazon CloudWatch 指标适配器。您必须允许 CloudWatch 完全访问与 Kubernetes Pod 和集群关联的两个 IAM 角色。要将自定义指标发布到 CloudWatch，Worker 节点的角色必须具有 AWS 托管策略 **CloudWatch AgentServerPolicy**，HPA 才能检索这些指标。

STEP 2 | 从 Palo Alto Networks GitHub 存储库下载 EKS 特定的 HPA yaml 文件。

STEP 3 | 如果 CN-MGMT 部署在自定义命名空间中，请使用自定义命名空间更新 pan-cn-adapater.yaml。默认命名空间为 **kube-system**。

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

1. 输入副本的最小和最大数量。
2. （可选）更改缩小和放大频率值以使其适合您的部署。如果不更改这些值，则会使用默认值。
3. 为要用于缩放的每个指标复制以下部分。

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

4. 更改要使用的指标的名称，然后将 **AveragValue** 设置为上表中所述的阈值。如果不更改这些值，则会使用默认值。
5. 保存更改。

STEP 5 | 部署 HPA yaml 文件。必须按照下面所述的顺序部署这些文件。

1. 使用 Kubectl 运行 pan-cn-adapter.yaml

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

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

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

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

STEP 6 | 验证部署。

- 使用 kubectl 验证自定义指标命名空间中的自定义指标适配器 Pod 是否存在。

  kubectl get pods -n custom-metrics
- 使用 kubectl 检查 HPA 资源。

  kubectl get hpa -n kube-system

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

将 CN 系列防火墙部署为守护进程

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 图表部署 CN 系列的 Helm 3.6 or above version client

完成以下过程以将 CN 系列防火墙部署为 Daemonset。

在开始之前，请确保 CN 系列 YAML 文件版本与 PAN OS 版本兼容。

- PAN-OS 10.1.2 或更高版本需要 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 | 设置 Kubernetes 集群。

1. 验证集群是否有足够的资源。确保该群集具有 CN 系列先决条件资源，以便支持防火墙。

   ```
   kubectl get nodes
   ```

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

   查看命令输出中容量标题下的信息，以了解指定节点上可用的 CPU 和内存。

   CPU、内存和磁盘存储分配将取决于您的需求。请参阅 CN 系列的性能和扩展性。

   确保有以下信息：

   - 收集端点 IP 地址，用于在 Panorama 上设置 API 服务器。Panorama 使用此 IP 地址连接到 Kubernetes 集群。
   - 从 Panorama 收集模板堆栈名称、设备组名称、Panorama IP 地址和可选的日志收集器组名称。
   - 收集授权代码以及自动注册 PIN ID 和值。
   - 将映像下载到的容器映像存储库的位置。

STEP 2 | （可选）如果您在 Kubernetes 插件中为 Panorama 配置了自定义证书，则必须通过执行以下命令来创建证书密钥。不要更改 ca.crt 中的文件名。pan-cn-mgmt.yaml 和 pan-cn-ngfw.yaml 中自定义证书的数量是可选的。

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

STEP 3 | 编辑 YAML 文件以提供部署 CN 系列防火墙所需的详细信息。

您需要替换 YAML 文件中的映像路径，以包括私有注册表的路径并提供所需的参数。有关详细信息，请参阅CN 系列部署 YAML 文件中的可编辑参数。

STEP 4 | 部署 CNI DaemonSet。

CNI 容器作为 DaemonSet 部署（每个节点一个 Pod），并且在 CN-NGFW Pod 上为节点上部署的每个应用程序创建两个接口。使用 kubectl 命令运行 pan-cni YAML 文件时，该容器将成为每个节点上服务链的一部分。

1. 使用 pan-cni-serviceaccount.yaml 文件验证您是否已创建服务帐户。

   请参阅创建用于集群身份验证的服务帐户。

2. 使用 Kubectl 运行 pan-cni-configmap.yaml 文件。

   kubectl apply -f pan-cni-configmap.yaml

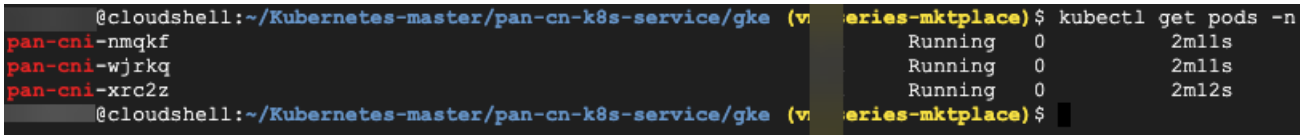
3. 使用 Kubectl 运行 pan-cni.yaml 文件。

   kubectl apply -f pan-cni.yaml

4. 验证是否已修改 pan-cni-configmap 和 pan-cni YAML 文件。

5. 运行以下命令并验证输出是否与以下示例相似。

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



STEP 5 | （仅限 AWS Outpost 上 EKS 的 CN 系列）更新存储类。要支持部署在 AWS Outpost 上的 CN 系列，您必须使用存储驱动程序 aws-ebs-csi-driver，以确保 Outpost 在动态持久性卷 (PV) 创建期间从 Outpost 拉取卷。

1. 应用以下 yaml。

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

2. 验证 ebs-sc 控制器是否正在运行。

   kubectl -n kube-system get pods

3. 更新 pan-cn-storage-class.yaml 以匹配以下示例。

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

4. 将 **storageClassName: ebs-sc** 添加到 pan-cn-mgmt.yaml 的如下所示位置。

```
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。

默认情况下，部署管理平面作为提供容错功能的 StatefulSet。最多可以将 30 个防火墙 CN-NGFW Pod 连接到 CN-MGMT StatefulSet。

1. （仅对于静态配置的 PV 为必需）为 CN-MGMT StatefulSet 部署持久卷 (PV)。

   1. 创建目录以匹配 pan-cn-pv-local.yaml 文件中定义的本地卷名称。

      在至少 2 个工作节点上需要六 (6) 个目录。登录将在其中部署 CN-MGMT StatefulSet 的每个工作节点，以创建目录。例如，要创建名为 /mnt/pan-local1 到 /mnt/pan-local6 的目录，请使用以下命令：

      ```
      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。

      在 `nodeaffinity` 下匹配主机名，并验证是否已修改在 `spec.local.path` 中创建的上述目录，然后部署文件以创建新的存储类 pan-local-storage 和本地 PV。

2. 验证是否已修改 pan-cn-mgmt-configmap 和 pan-cn-mgmt YAML 文件。

   EKS 中的 pan-cn-mgmt-configmap 示例。

```
 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
```

pan-cn-mgmt.yaml 文件示例

```
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 运行 yaml 文件。

   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

   如果您之前未完成为集群身份验证创建服务帐户，则必须运行 pan-mgmt-serviceaccount.yaml。

4. 验证 CN-MGMT Pod 是否启动。

   大约需要 5 至 6 分钟的时间。

   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。

默认情况下，将防火墙数据平面 CN-NGFW Pod 作为 DaemonSet 部署。CN-NFGW Pod 的实例可保护节点上最多 30 个应用程序 Pod 的流量。

1. 验证是否已按 PAN-CN-NGFW-CONFIGMAP 和 PAN-CN-NGFW 中的详细说明修改 YAML 文件。

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

2. 使用 Kubectl apply 运行 pan-cn-ngfw-configmap.yaml 文件。

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

3. 使用 Kubectl apply 运行 pan-cn-ngfw.yaml 文件。

   kubectl apply -f pan-cn-ngfw.yaml

4. 验证所有 CN-NGFW Pod 是否正在运行（集群中每个节点一个）。

   以下是 4 个节点本地集群的输出示例。

   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 | 验证您是否可以在 Kubernetes 集群上看到 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 | 为应用程序 yaml 文件或命名空间添加注释，以便将来自其新 Pod 的流量重定向到防火墙。

您需要添加以下注解，以将流量重定向到 CN-NGFW 来进行检查：

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

例如，对于“default”命名空间中的所有新 Pod：

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

在某些平台上，当在 *CNI* 插件链中未激活 *pan-cni* 时，可以启动应用程序 *Pod*。为避免此类情况，您必须按如下在应用程序 *Pod YAML* 中指定卷。

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

STEP 10 | 在集群中部署应用程序。

将 CN 系列防火墙部署为 Kubernetes CNF

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 图表部署 CN 系列的 Helm 3.6 or above version client

现在，您可以在 Kubernetes 环境中将 CN 系列部署为 Container Network Function (CNF)。

CN 系列即 DaemonSet 和 CN 系列即 Kubernetes 服务部署模式提供了自动化安全部署，并利用了 Kubernetes 的自动扩展功能。但是，这些部署模式的插入选项有限，不支持 I/O 加速。此外，它们还限制需要检查和使用多个网络接口的应用程序 Pod 的可实现吞吐量。

部署 CN 系列即 kubernetes-CNF 解决了通过外部实体（例如云提供商的本机路由、vRouters 和架顶式 (TOR) 交换机等）使用服务功能链 (SFC) 的流量所面临的这些挑战。CN 系列即 kubernetes-CNF 部署模式不会影响应用程序 Pod。

完成以下过程以部署 CN 系列即 kubernetes-CNF。

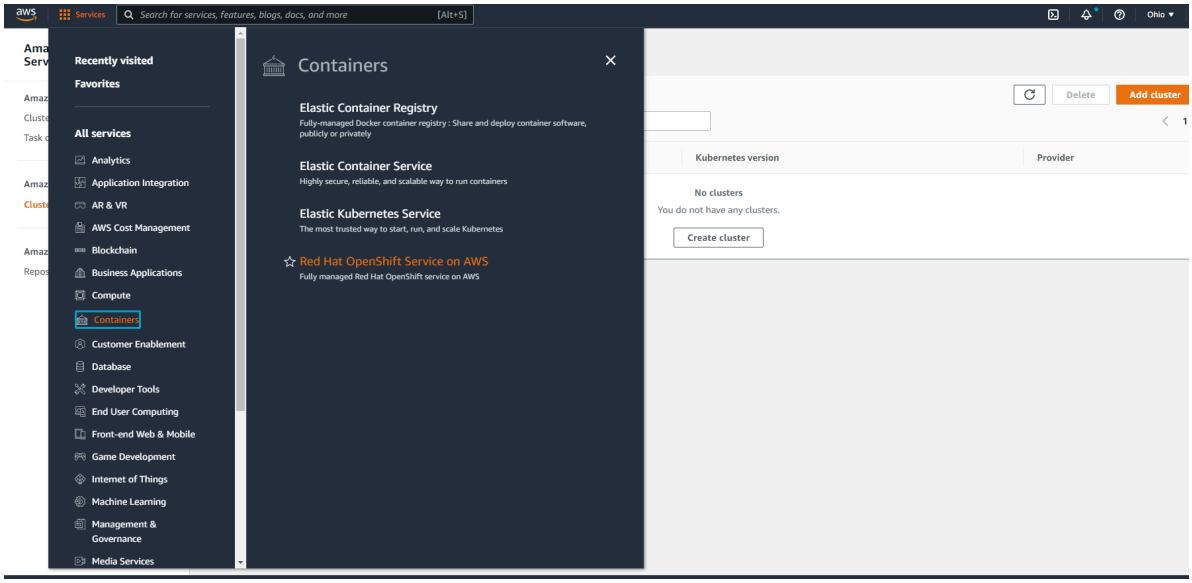
开始之前，请确保 CN 系列 YAML 文件版本与 PAN-OS 版本兼容：

PAN-OS 10.2.0 或更高版本需要 YAML 3.0.0

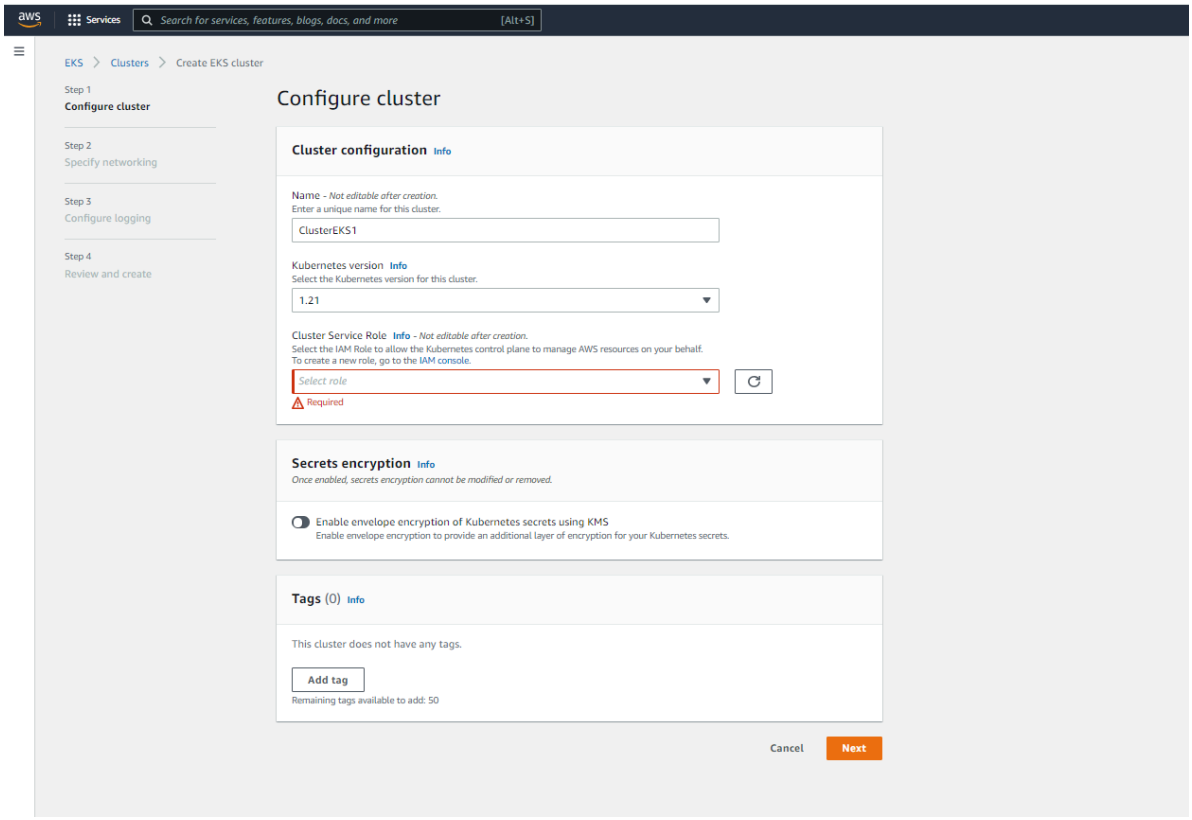
STEP 1 | 设置 Kubernetes 集群。有关更多信息，请参阅创建 Amazon EKS 集群和 Pod 的多个网络接口。

要在 AWS EKS 中创建集群，请执行以下操作：

1. 单击 **Services**（服务）导航菜单，转到 **Containers**（容器）->**Elastic Kubernetes Service**（**Elastic Kubernetes** 服务）。



2. 单击 **Create Cluster**（创建集群）。
3. 填写所需的详细信息，然后单击 **Create**（创建）。



1. 验证集群是否有足够的资源。确保该群集具有 CN 系列先决条件资源，以便支持防火墙。

   ```
   kubectl get nodes
   ```

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

   查看命令输出中容量标题下的信息，以了解指定节点上可用的 CPU 和内存。

   CPU、内存和磁盘存储分配将取决于您的需求。请参阅 CN 系列的性能和扩展性。

   确保有以下信息：

   - 收集端点 IP 地址，用于在 Panorama 上设置 API 服务器。Panorama 使用此 IP 地址连接到 Kubernetes 集群。
   - 从 Panorama 收集模板堆栈名称、设备组名称、Panorama IP 地址和可选的日志收集器组名称。
   - 收集授权代码以及自动注册 PIN ID 和值。
   - 将映像下载到的容器映像存储库的位置。

STEP 2 | （可选）如果您在 Kubernetes 插件中为 Panorama 配置了自定义证书，则必须通过执行以下命令来创建证书密钥。不要更改 ca.crt 中的文件名。pan-cn-mgmt-0.yaml、pan-cn-mgmt-1.yaml、pan-cn-ngfw-0.yaml 和 pan-cn-ngfw.yaml-1 中的自定义证书数量是可选的。

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

STEP 3 | 编辑 YAML 文件以提供部署 CN 系列防火墙所需的详细信息。

您要替换 YAML 文件中的映像路径，以包括私有注册表的路径并提供所需的参数。有关详细信息，请参阅CN 系列部署 YAML 文件中的可编辑参数。

HA 中的 CN-Series-as-a-kubernetes-CNF 仅支持具有会话和配置同步功能的主动/被动 HA。

在 HA 中部署 CN-seriesas-a-Kubernetes-CNF 时，主动节点和被动节点将有两个 PAN-CN-MGMT-CONFIGMAP、PAN-CN-MGMT 和 PAN-CN-NGFW YAML 文件，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

以下默认值在 pan-cn-mgmt-configmap-0.yaml 和 pan-cn-mgmt-configmap-1.yaml 文件中定义。

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
```


您可以为 CPU 固定添加 numa 选项。在 pan-cn-ngfw-configmap-0.yaml 和 pan-cn-ngfw-configmap-1.yaml 文件中为 `PAN_NUMA_ENABLED` 参数添加单个 numa 节点编号。

要在具有第 3 层支持的 HA 中成功部署 cn-series-as-a-kubernetes-CNF，请执行以下操作：

- 在 HA 中，每个 Kubernetes 节点至少要有三个接口：管理（默认）、HA2 和数据接口。
- 对于 L3 模式下的 CN 系列防火墙，至少要有两个接口：管理（默认）和数据接口。
- 修改新的网络附件定义 YAML 文件，进行以下更改：
  - 在工作进程节点上运行以下命令，从虚拟机管理程序接口检索 **`pciBusID`** 值：

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

    例如：

    ```
    00:05.0 Ethernet controller:Amazon.com, Inc.Elastic Network
     Adapter (ENA)
    ```

    ```
    00:06.0 Ethernet controller:Amazon.com, Inc.Elastic Network
     Adapter (ENA)
    ```

    ```
    00:07.0 Ethernet controller:Amazon.com, Inc.Elastic Network
     Adapter (ENA)
    ```

    ```
    00:08.0 Ethernet controller:Amazon.com, Inc.Elastic Network
     Adapter (ENA)
    ```

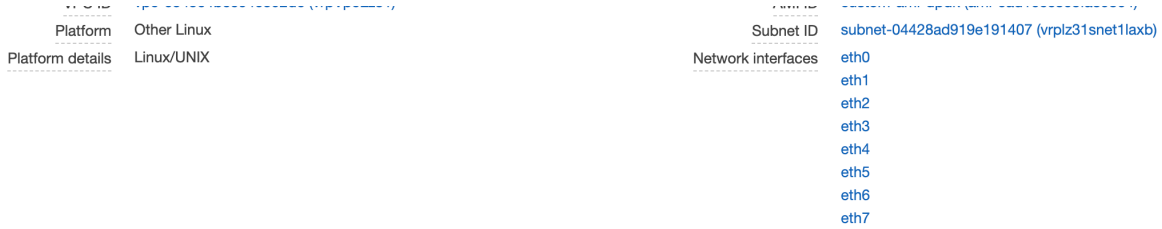
    ```
    00:09.0 Ethernet controller:Amazon.com, Inc.Elastic Network
     Adapter (ENA)
    ```

    ```
    00:0a.0 Ethernet controller:Amazon.com, Inc.Elastic Network
     Adapter (ENA)
    ```

    ```
    00:0b.0 Ethernet controller:Amazon.com, Inc.Elastic Network
     Adapter (ENA)
    ```

    ```
    00:0c.0 Ethernet controller:Amazon.com, Inc.Elastic Network
     Adapter (ENA)
    ```

    PCI 顺序与 AWS EC2 UI 上显示的 eth 接口的顺序相同



将上面检索到的 **PCIbusID** 值添加到以下网络定义文件中：

```
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
```

- 从 AWS 控制台上的对应节点实例获取 HA2 接口的静态 IP 地址，并将其添加到 `net-attach-def-ha2-0.yaml` 和 `net-attach-def-ha2-1.yaml` 文件的 `address` 参数中。

STEP 4 | 部署 CN-MGMT StatefulSet。

默认情况下，部署管理平面作为提供容错功能的 StatefulSet。只能将一个防火墙 CN-NGFW Pod 连接到 CN-MGMT StatefulSet。

1. （仅对于静态配置的 PV 为必需）为 CN-MGMT StatefulSet 部署持久卷 (PV)。

   1. 创建目录以匹配 pan-cn-pv-local.yaml 文件中定义的本地卷名称。

      在至少 2 个工作节点上需要六 (6) 个目录。登录将在其中部署 CN-MGMT StatefulSet 的每个工作节点，以创建目录。例如，要创建名为 /mnt/pan-local1 到 /mnt/pan-local6 的目录，请使用以下命令：

      ```
      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。

      在 `nodeaffinity` 下匹配主机名，并验证是否已修改在 `spec.local.path` 中创建的上述目录，然后部署文件以创建新的存储类 pan-local-storage 和本地 PV。

2. 验证是否已修改 pan-cn-mgmt-configmap 和 pan-cn-mgmt YAML 文件。
3. 使用 Kubectl 运行 yaml 文件。

   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

   如果您之前未完成为集群身份验证创建服务帐户，则必须运行 pan-mgmt-serviceaccount.yaml。
4. 验证 CN-MGMT Pod 是否启动。

   大约需要 5 至 6 分钟的时间。

   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 | 在 k8s-CNF 模式下部署 CN-NGFW。

1. 验证您是否已按照步骤 3 中的详细说明修改了 YAML 文件。

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

   您要确保已安装 *multus DaemonSet* 和创建网络附件定义文件。*pan-cn-ngfw-configmap-0.yaml* 和 *pan-cn-ngfw-configmap-1.yaml file* 文件中 *PAN_SERVICE_NAME* 的参数值应分别与 *pan-cn-mgmt-0.yaml* 和 *pan-cn-mgmt-1.yaml* 文件中的服务名称参数值匹配。

   对于 *HA* 支持，建议在不同的工作进程节点上部署 *DP Pod*。您可以通过 ***yaml nodeSelector*** 字段或启用 *Pod* 反关联来确保这一点。

   要启用 *HA* 支持，您要确保以下 *YAML* 文件中的 *PAN_HA_SUPPORT* 参数值为 ***true***：

   pan-cn-mgmt-configmap-0.yaml

   pan-cn-mgmt-configmap-1.yaml

   对于 *DP Pod* 的数据接口，应根据需要将 *CNI* 和接口资源添加到 *DP YAML* 文件中。例如：

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

   要启用 *DPDK* 支持，您要确保 *pan-cn-ngfw-configmap-0.yaml* 和 *pan-cn-ngfw-configmap-1.yaml* 文件中的 *PAN_DATA_MODE* 参数值为 ***dpdk***。

   此外，*HUGEPAGE_MEMORY_REQUEST* 参数值应与 *pan-cn-ngfw-0.yaml* 和 *pan-cn-ngfw-1.yaml* 文件中的大页面内存请求匹配。

   有关详细信息，请参阅在 CN 系列防火墙上配置 DPDK。

2. 使用 Kubectl apply 运行 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 运行 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. 确认已部署 CN-NGFW Pod。

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

STEP 6 | 部署 CN-NGFW Pod。请执行以下操作：

1. 验证是否已按 PAN-CN-NGFW-CONFIGMAP-0、PAN-CN-NGFW-CONFIGMAP-1、PAN-CN-NGFW-0 和 PAN-CN-NGFW-1 中的详细说明修改 YAML 文件。

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

2. 使用 Kubectl apply 运行 pan-cn-ngfw-configmap.yaml 文件。

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

3. 使用 Kubectl apply 运行 pan-cn-ngfw.yaml 文件。

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

4. 确认已部署 CN-NGFW Pod。

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

STEP 7 | 验证能否在 Kubernetes 集群上看到 CN-MGMT 和 CN-NGFW。运行以下命令：

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

以独立模式部署 Kubernetes CNF L3

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 图表部署 CN 系列的 Helm 3.6 or above version client

您可以在 Kubernetes 环境中以 L3 独立模式将 CN 系列防火墙部署为容器网络功能 (CNF)。

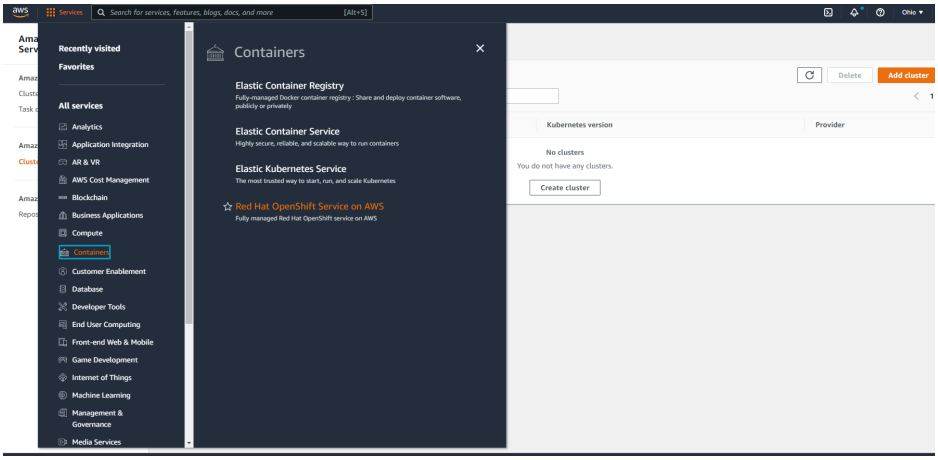
现在，CN 系列支持通过 vRouter 的流量，其中静态路由配置为将流量重定向到防火墙的数据平面接口。对于相反方向，流量将使用具有 IPv4 IP 地址的基于 L3 策略的路由 (PBR) 重定向到同一防火墙。K8s 环境中接口的 IP 地址通常使用 DHCP 通过 CNI 进行编程。

要在 L3 独立模式下部署 Kubernetes CNF，请执行以下操作：

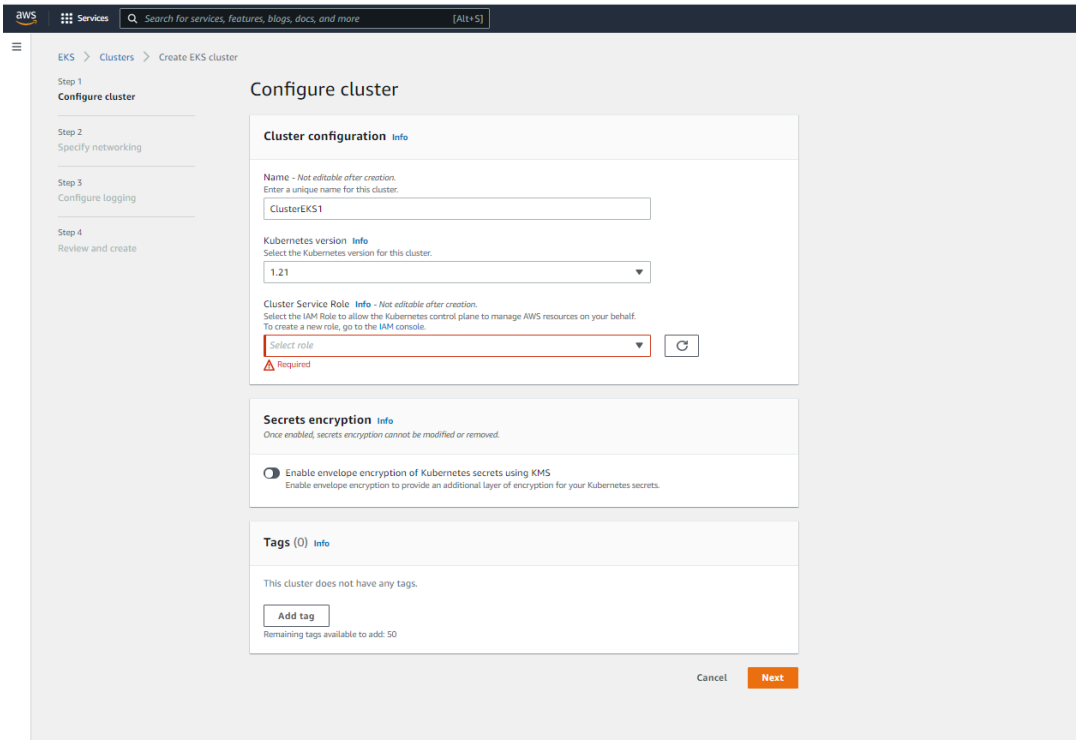
STEP 1 | 设置 Kubernetes 集群。

要在 AWS EKS 中创建集群，请执行以下操作：

1. 单击 **Services**（服务）导航菜单，转到 **Containers**（容器）->**Elastic Kubernetes Service**（**Elastic Kubernetes** 服务）。



2. 单击 **Create Cluster**（创建集群）。
3. 填写所需的详细信息，然后单击 **Create**（创建）。



1. 验证集群是否有足够的资源。确保该群集具有 CN 系列先决条件资源，以便支持防火墙。

   kubectl get nodes

   kubectl describe node <node-name>

   查看命令输出中容量标题下的信息，以了解指定节点上可用的 CPU 和内存。

CPU、内存和磁盘存储分配将取决于您的需求。请参阅 CN 系列的性能和扩展性

确保有以下信息：

- 收集端点 IP 地址，用于在 Panorama 上设置 API 服务器。Panorama 使用此 IP 地址连接到 Kubernetes 集群。
- 从 Panorama 收集模板堆栈名称、设备组名称、Panorama IP 地址和可选的日志收集器组名称。
- 收集授权代码以及自动注册 PIN ID 和值。
- 将映像下载到的容器映像存储库的位置。

STEP 2 | （可选）如果已在 Kubernetes 插件中为 Panorama 配置自定义证书，则必须通过执行以下命令来创建证书机密。不要更改 ca.crt 中的文件名。pan-cn-mgmt-0.yaml 和 pan-cn-ngfw-0.yaml 中的自定义证书数量是可选的。

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

STEP 3 | 编辑 YAML 文件以提供部署 CN 系列防火墙所需的详细信息。

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

您要替换 YAML 文件中的映像路径，以包括私有注册表的路径并提供所需的参数。有关详细信息，请参阅CN 系列部署 YAML 文件中的可编辑参数。

以下默认值在 pan-cn-mgmt-configmap-0.yaml 文件中定义。

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
```

您可以为 CPU 固定添加 numa 选项。将 `PAN_NUMA_ENABLED` 参数的单个 numa 节点编号添加到 pan-cn-ngfw-configmap-0.yaml 文件中。

要成功部署具有第 3 层支持的 CN-Series-as-a-kubernetes-CNF，请执行以下操作：

- 每个 Kubernetes 节点至少要有三个接口：管理（默认）、HA2 链路和数据接口。
- 对于 L3 模式下的 CN 系列防火墙，至少要有两个接口：管理（默认）和数据接口。
- 修改新的网络附件定义 YAML 文件，进行以下更改：
  - 在工作进程节点上运行以下命令，从虚拟机管理程序接口检索 **`pciBusID`** 值：

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

例如：

```
00:05.0 Ethernet controller:Amazon.com, Inc.Elastic Network
 Adapter (ENA)
```

```
00:06.0 Ethernet controller:Amazon.com, Inc.Elastic Network
 Adapter (ENA)
```

```
00:07.0 Ethernet controller:Amazon.com, Inc.Elastic Network
 Adapter (ENA)
```

```
00:08.0 Ethernet controller:Amazon.com, Inc.Elastic Network
 Adapter (ENA)
```

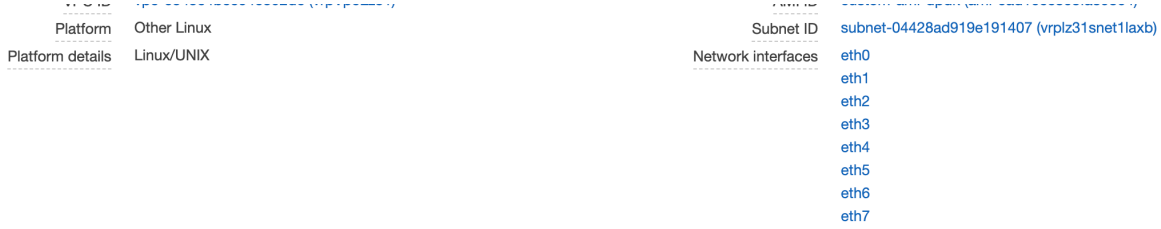
```
00:09.0 Ethernet controller:Amazon.com, Inc.Elastic Network
 Adapter (ENA)
```

```
00:0a.0 Ethernet controller:Amazon.com, Inc.Elastic Network
 Adapter (ENA)
```

```
00:0b.0 Ethernet controller:Amazon.com, Inc.Elastic Network
 Adapter (ENA)
```

```
00:0c.0 Ethernet controller:Amazon.com, Inc.Elastic Network
 Adapter (ENA)
```

PCI 顺序与 AWS EC2 UI 上显示的 eth 接口的顺序相同



将上面检索到的 **PCIbusID** 值添加到以下网络定义文件中：

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

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

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

STEP 4 | 部署 CN-MGMT StatefulSet。

默认情况下，部署管理平面作为提供容错功能的 StatefulSet。只能将一个防火墙 CN-NGFW Pod 连接到 CN-MGMT StatefulSet。

1. （仅对于静态配置的 PV 为必需）为 CN-MGMT StatefulSet 部署持久卷 (PV)。
   1. 创建目录以匹配 pan-cn-pv-local.yaml 文件中定义的本地卷名称。

      在至少 2 个工作节点上需要六 (6) 个目录。登录将在其中部署 CN-MGMT StatefulSet 的每个工作节点，以创建目录。例如，要创建名为 /mnt/pan-local1 到 /mnt/pan-local6 的目录，请使用以下命令：

      ```
      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。

      在 `nodeaffinity` 下匹配主机名，并验证是否已修改在 `spec.local.path` 中创建的上述目录，然后部署文件以创建新的存储类 pan-local-storage 和本地 PV。
2. 验证是否已修改 pan-cn-mgmt-configmap 和 pan-cn-mgmt YAML 文件。
3. 使用 Kubectl 运行 yaml 文件。

   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

   如果您之前未完成为集群身份验证创建服务帐户，则必须运行 pan-mgmt-serviceaccount.yaml。
4. 验证 CN-MGMT Pod 是否启动。

   大约需要 5 至 6 分钟的时间。

   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 | 在 k8s-CNF 模式下部署 CN-NGFW。

1. 验证您是否已按照步骤 3 中的详细说明修改了 YAML 文件。

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

   您要确保已安装 *multus DaemonSet* 和创建网络附件定义文件。*pan-cn-ngfw-configmap-0.yaml* 文件中 *PAN_SERVICE_NAME* 参数值应与 *pan-cn-mgmt-0.yaml* 文件中的服务名称参数值匹配。

   对于 *CN-NFGW Pod* 的数据接口，应根据需要将 *CNI* 和接口资源添加到 *CN-NFGW YAML* 文件中。例如：

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

   要启用 *DPDK* 支持，应确保 *pan-cn-ngfw-configmap-0.yaml* 文件中的 *PAN_DATA_MODE* 参数值为 ***dpdk***。

   此外，*HUGEPAGE_MEMORY_REQUEST* 参数值应与 *pan-cn-ngfw-0.yaml* 文件中的大页面内存请求匹配。

   有关详细信息，请参阅在 CN 系列防火墙上配置 DPDK。

2. 使用 Kubectl apply 运行 pan-cn-ngfw-configmap-0.yaml 文件。

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

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

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

4. 确认已部署 CN-NGFW Pod。

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

STEP 6 | 部署 CN-NGFW Pod。请执行以下操作：

1. 验证是否已按 PAN-CN-NGFW-CONFIGMAP-0 和 PAN-CN-NGFW-0 中的详细说明修改 YAML 文件。

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

2. 使用 Kubectl apply 运行 pan-cn-ngfw-configmap.yaml 文件。

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

3. 使用 Kubectl apply 运行 pan-cn-ngfw.yaml 文件。

   kubectl apply -f pan-cn-ngfw.yaml

4. 确认已部署 CN-NGFW Pod。

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

STEP 7 | 验证能否在 Kubernetes 集群上看到 CN-MGMT 和 CN-NGFW。运行以下命令：

```
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 系列防火墙

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

使用 Kubernetes 编排可轻松部署 CN 系列防火墙，从而简化网络安全与持续集成/持续开发 (CI/CD) 流程的集成。CN 系列防火墙的持续管理集中在 Panorama™ 网络安全管理中 — 与所有 Palo Alto Networks 防火墙相同的管理控制台 — 为网络安全团队提供了单一管理平台来管理其组织的整体网络安全态势。

本章包括以下部分：

CN 系列部署清单

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 图表部署 CN 系列的 Helm 3.6 or above version client

要部署 CN 系列防火墙，您必须完成以下任务：

- ❑ 如果您还没有执行此操作，授予 CN 系列防火墙许可证 — 请生成授权代码，并在准备好部署 CN 系列防火墙后，将其放在手边。
- ❑ 查看 CN 系列先决条件 — 在开始部署之前，请确保了解部署 CN 系列防火墙所需满足的系统要求。
- ❑ 准备组件。
  - 在 Panorama 上生成 VM 身份验证密钥。
  - （可选）在 CN 系列防火墙上安装设备证书。
  - 创建用于集群身份验证的服务帐户。
  - 部署 Panorama — 您必须使用 Panorama 来配置、部署和管理 CN 系列防火墙部署。有关部署和设置 Panorama 设备的更多信息，请参阅设置 Panorama。
  - 为 CN 系列安装 Kubernetes 插件。
  - 获取用于 CN 系列部署的映像和文件 — 访问 Palo Alto Networks 存储库以下载 Docker 文件和 GitHub，从而获取在 Kubernetes 环境中部署 CN 系列防火墙所需的 yaml 文件。
- ❑ 部署 CN 系列防火墙。
  - 编辑 HELM 图表，使其适合您的部署 — 或者，在部署 CN 系列防火墙之前，您也可以编辑 yaml 文件并查看CN 系列部署 YAML 文件中的可编辑参数。为了成功部署 CN 系列防火墙，您必须修改 yaml 文件中设置的许多参数。
  - 将 CN 系列防火墙部署为 Kubernetes 服务（推荐的部署模式）.
  - 将 CN 系列防火墙部署为守护进程.
  - （可选）如果将 CN 系列防火墙部署为 Kubernetes 服务，您可以在 CN 系列上启用水平 Pod 自动缩放。利用水平 Pod 自动缩放 (HPA)，您的 CN 系列防火墙部署可以与 Kubernetes 环境一起动态自动缩放。
  - 如果要在 OpenShift 环境中部署 CN 系列，请参阅在 Openshift 上部署 CN 系列防火墙。
  - 如果您使用 CN 系列防火墙保护 5G 流量，请参阅使用 CN 系列防火墙保护 5G。

- 配置 Panorama 以保护 Kubernetes 部署 — 部署 CN 系列防火墙后，使用 Panorama 配置启用流量实施的安全策略并将这些策略推送到防火墙。

使用（推荐）和不使用 Helm 图表部署 CN 系列防火墙

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

Helm 存储库包含使用 适用于 Kubernetes 的 Helm Packet Manager 部署 Palo Alto Networks CN 系列容器化防火墙的图表和模板。

您可以从 GitHub下载 CN 系列 Helm 图表。

- 准备使用 Helm 图表和模板
- 使用 HELM 图表部署 CN 系列防火墙（推荐）
- 通过 YAML 文件部署 CN 系列防火墙

准备使用 Helm 图表和模板

安装所需的软件。这些说明列出了最低版本，但是除非已指定上限，否则您可以在同一系列中安装更高版本。

STEP 1 | 部署 CN 系列防火墙 10.1.x、10.2.x、11.0.x 或 11.1.x 容器映像。

STEP 2 | 安装 1.16 - 1.25 之间的 Kubernetes 版本，然后创建一个 Kubernetes 集群。有关您的环境支持的 kubernetes 版本的更多信息，请参阅 CN 系列部署支持的环境。

STEP 3 | 在 Kubernetes 集群和用于保护集群安全的 CN 系列防火墙可以访问的位置部署 Panorama。

1. 确保 Panorama PAN-OS 版本为 10.x.x 或更高版本。
2. 为 Panorama 版本 1.0.x 或 2.0.x 安装 Kubernetes 插件。

STEP 4 | 安装 Helm 客户端版本 3.6.0 或更高版本。

继续阅读使用 HELM 图表部署 CN 系列防火墙（推荐）

或通过 YAML 文件部署 CN 系列防火墙。

使用 HELM 图表部署 CN 系列防火墙（推荐）

按照此过程克隆存储库并从本地环境进行部署。

STEP 1 | 在 Panorama 上生成 VM 身份验证密钥

STEP 2 | 从 GitHub 克隆存储库。

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

STEP 3 | 切换到克隆存储库的本地目录。例如：

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

STEP 4 | 切换到部署的子目录。

- 使用目录 `helm_cnv1` 将 CN 系列部署为守护程序集
- 使用目录 `helm_cnv2` 将 CN 系列部署为服务。
- 使用目录 `helm_cnv3` 将 CN 系列部署为 cnf。

STEP 5 | 下载 `plugin-serviceaccount.yaml` 的服务帐户 YAML 并应用 yaml。该服务帐户可启用 Panorama 要求对 GKE 集群进行身份验证所需的权限，从而检索 Kubernetes 标签和资源信息。默认情况下，将该服务帐户命名为 `pan-plugin-user`。运行以下命令，部署 `plugin-serviceaccount.yaml` 文件：

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

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

查看与该服务帐户相关联的密钥。

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

创建一个包含密钥的凭据文件（本例中命名为 `cred.json`）并保存此文件。您需要将此文件上传到 Panorama，以设置用于监控为 CN 系列防火墙安装 Kubernetes 插件中的集群的 Kubernetes 插件。

在 ***Openshift*** 上，您必须在部署 *Helm* 图表之前为每个 *Openshift* 命名空间文件手动部署 *`pan-cni-net-attach-def.yaml`*。

STEP 6 | 编辑 `values.yaml` 文件以输入配置信息。以下值来自 `helm_cnv1` 子目录。

```
# The K8s environment # Valid deployTo tags are: [gke|eks|aks||
native] # Valid multus tags are : [enable|disable] Keep the multus
 as enable for openshift and native deployments. cluster: deployTo:
 eks multus: disable
```

```
# 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 | 查看渲染的 YAML 文件。

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

STEP 8 | 对 Helm 图表执行 lint 检查。

```
helm lint helm_cnv1/
```

STEP 9 | 部署 HELM 图表。

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

卸载 *HELM* 图表时不会删除持久卷声明。您必须确保事先清除这些声明，以使 *HELM* 安装工作正常进行。

有关 HELM 的更多信息，请参阅 HELM 经典版：Kubernetes 包管理器。

通过 YAML 文件部署 CN 系列防火墙

要在不克隆存储库的情况下进行部署，请将存储库添加到 Helm 客户端。

STEP 1 | 在 Panorama 上生成 VM 身份验证密钥

STEP 2 | 下载 `plugin-serviceaccount.yaml` 的服务帐户 YAML 并应用 yaml。该服务帐户可启用 Panorama 要求对 GKE 集群进行身份验证所需的权限，从而检索 Kubernetes 标签和资源信息。默认情况下，将该服务帐户命名为 `pan-plugin-user`。运行以下命令，部署 `plugin-serviceaccount.yaml` 文件：

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

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

查看与该服务帐户相关联的密钥。

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

创建一个包含密钥的凭据文件（本例中命名为 `cred.json`）并保存此文件。您需要将此文件上传到 Panorama，以设置用于监控为 CN 系列防火墙安装 Kubernetes 插件中的集群的 Kubernetes 插件。

在 ***Openshift*** 上，您必须在部署 *Helm* 图表之前为每个 *Openshift* 命名空间文件手动部署 *`pan-cni-net-attach-def.yaml`*。

STEP 3 | 将 CN 系列存储库添加到本地 Helm 客户端。

在一行中输入以下命令：

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

```
“CN 系列”已添加到您的存储库
```

STEP 4 | 确认存储库已添加到 Helm 客户端。

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

STEP 5 | 选择 Kubernetes 集群。

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

STEP 6 | 使用 Helm chart 存储库进行部署。编辑以下命令以包含您的配置信息。

```
$ 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"
```

卸载 *HELM* 图表时不会删除持久卷声明。您必须确保事先清除这些声明，以使 *HELM* 安装工作正常进行。

使用 Terraform 模板部署 CN 系列防火墙

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • Terraform 0.13.0 或更高版本

CN 系列部署存储库包含部署 GKE、EKS 或 AKS 集群的 Terraform 计划。这些计划可确保集群节点大小和容器网络接口 (CNI) 支持集群内的 CN 系列防火墙部署。存储库还提供了一个 CN 系列防火墙部署计划和一个示例PHP Guestbook 应用程序，您可以使用防火墙保护这些应用程序。

此过程包含以下可选工作流：

- 准备使用 Helm 图表和模板
- 部署示例应用程序
- 使用 Terraform 部署 CN 系列防火墙
- 为 Panorama 配置 Kubernetes 插件

部署示例应用程序

Palo Alto Networks GitHub 存储库包含一个社区支持的示例应用程序，其中包含一个名为 `guestbook.yml` 的 Kubernetes 清单文件。

此文件部署了一个利用 Redis 后端的简单 PHP Guestbook Web 应用程序。

STEP 1 | 在 Palo Alto Networks GitHub 存储库 cn-series-deploy 目录下，切换到 sample-application 目录。

```
$ cd sample-application
```

STEP 2 | 部署 Guestbook 应用程序。

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

STEP 3 | 验证应用程序 Pod 是否已部署并处于“正在运行”状态，接着处于“就绪”状态。

```
$ 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 | 列出服务以确定 Web 前端的公共 IP 地址。

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

现在，您可以在 Panorama 上配置动态地址组和安全规则，以保护 Guestbook 应用程序的安全。

继续使用 Terraform 部署 CN 系列防火墙。

使用 Terraform 部署 CN 系列防火墙

使用 Terraform 部署 CN 系列防火墙。

STEP 1 | 使用本地 `cn-series\tfvars` 创建名为 `terraform.tfvars` 的文件，然后添加以下变量及其关联值。

```
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 | 验证 Terraform 计划。

```
$ terraform init
```

STEP 3 | 验证 Terraform 计划。

```
$ terraform plan
```

STEP 4 | 应用 Terraform 计划。

```
$ terraform apply
```

STEP 5 | 验证 Pod 是否已部署且处于就绪状态，同时且状态为“正在运行”。

```
$ 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
```

您已准备好为 Panorama 配置 Kubernetes 插件。

为 Panorama 配置 Kubernetes 插件

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama

使用 Kubernetes Panorama 插件将标签传播到 Panorama 设备组。

您可以使用 Kubernetes 插件完成 Panorama 和 Kubernetes API 的集成。该插件将学习新标签并将其传播到 Panorama 设备组。这些标签包括 Kubernetes 标签、服务、命名空间和其他可以定义动态地址组匹配标准的元数据。

如果集群凭据文件大小大于 *32KB*，则在 *Panorama Kubernetes* 插件上导入凭据文件时您将收到错误消息。错误消息显示作为错误原因的文件大小。

如果集群在 `ca.crt` 捆绑包中有多个 *CA* 证书，则 *Kubernetes* 插件只需要顶级 *CA* 证书。必须确保仅保留顶级 *CA* 证书，并从凭据文件中删除所有其他 *CA* 证书和 `service.crt`。然后您就可以使用更新后的凭据文件了。

此过程假定您已安装准备使用 Helm 图表和模板 中列出的支持软件。

STEP 1 | 从 Kubernetes 主节点检索 pan-plugin-user 服务帐户凭据。

每行输入一个命令：

```
$ 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 插件中创建集群定义。

使用 Terraform 输出中显示的 Kubernetes 主地址和位于 `~/Downloads/pan-plugin-user.json` 的 JSON 凭据文件。

定义要从 Kubernetes API 导入的标签。

STEP 3 | 在 Panorama Kubernetes 插件中创建通知组定义。

此定义用于将从 Kubernetes API 学习的标签传播到 Panorama 设备组。

执行以下步骤，以在 Panorama Kubernetes 插件中创建通知组：

1. 选择 **Panorama** > **Plugins**（插件）> **Kubernetes** > **Setup**（设置）> **Notify Groups**（通知组），然后选择 **Add**（添加）。

2. 为通知组输入一个最大长度为 31 个字符的 **Name**（名称）。
3. 如果您要共享除为集群创建的外部标签（默认）之外的内部标签，请选择 **Enable sharing internal tags with Device Groups**（启用与设备组共享内部标签）。
4. 选择要向其注册标签的设备组。

5. 单击 **Ok**（确定）。

STEP 4 | 在 Panorama 插件中创建监控定义。

使用在前面步骤中创建的集群和通知组定义。

STEP 5 | 提交到 Panorama。

STEP 6 | 要确认 API 连接和 MP 容器注册，请转到“监控定义”并单击“详细状态”和“集群 MP”。

现在，您已准备好部署应用程序并使用 CN 系列防火墙来提供保护。

使用 Rancher Orchestration 部署 CN 系列防火墙

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama

您现在可以使用 Rancher 编排和 PAN OS 10.1 将 CN 系列防火墙部署为 Kubernetes 服务。Rancher 是一个开源容器编排平台，可用于部署 CN 系列防火墙。

对于支持 Rancher 集群的 CN 系列防火墙部署，您的 Panorama 实例必须有 16 个 vCPU、32G 内存和额外的 2TB 磁盘。Panorama 将以一种便于从 CN 系列防火墙部署收集日志的模式进行部署。

在本地 Rancher Kubernetes 集群中部署 CN 系列防火墙时，请执行以下操作：

- 确保提供使用 CN 系列防火墙保护 Kubernetes 集群所需的组件。
- 确保 Kubernetes 集群满足最低系统要求。有关更多信息，请参阅 CN 系列系统要求。
- 执行 使用 Rancher Orchestration 部署 CN 系列防火墙。
-
- 修改 Rancher 集群选项 YAML 文件
- 为 CN 系列防火墙安装 Kubernetes 插件。
- 授予 CN 系列防火墙许可证。
- Rancher 上的 将 CN 系列防火墙部署为 Kubernetes 服务（推荐的部署模式）。

Rancher 集群部署

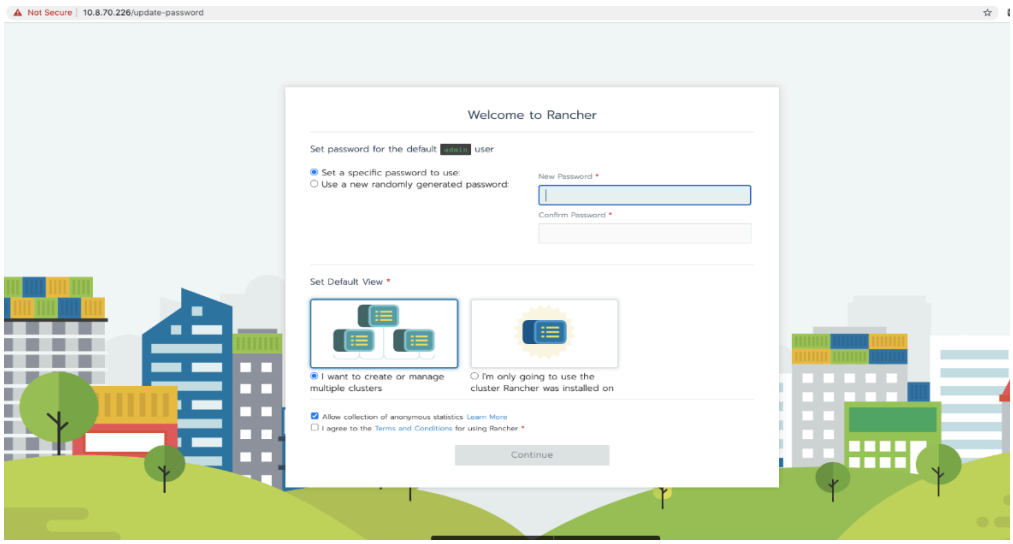
您可以按以下两个步骤部署 Rancher：

1. 准备一台具有支持的 Linux 发行版和 4GB 内存的 Linux 主机。在主机上安装支持的 Docker 版本。
2. 启动服务器。

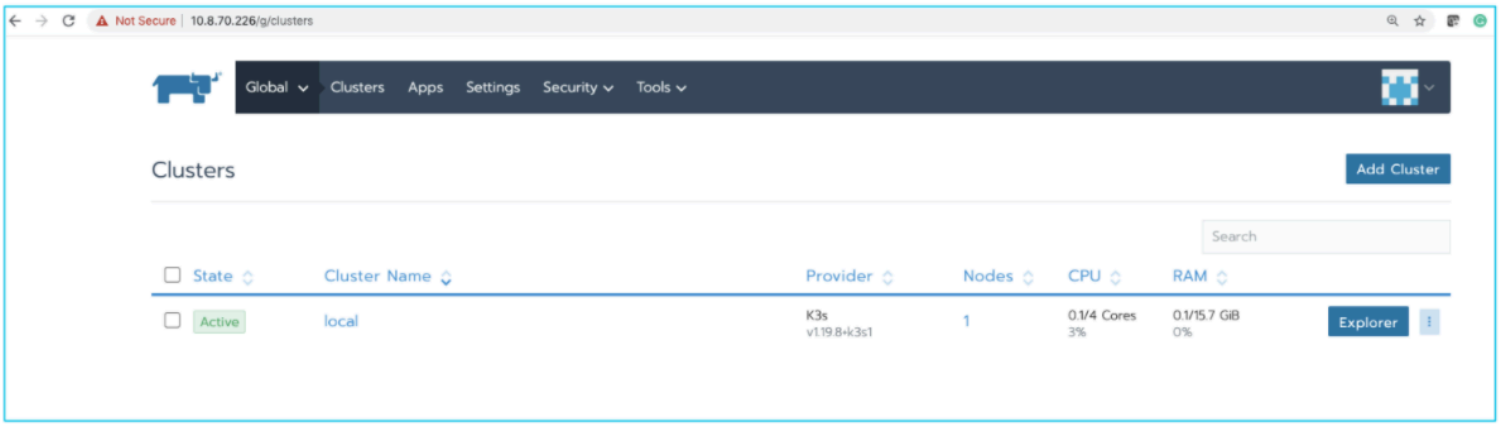
   要安装和运行 Rancher，请在您的主机上运行以下 Docker 命令：

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

部署成功后，您可以访问 Rancher 服务器 UI 并为管理员用户设置密码。要访问 Rancher 服务器 UI，请打开浏览器并转到安装容器的主机名或地址。您将根据向导设置第一个集群。



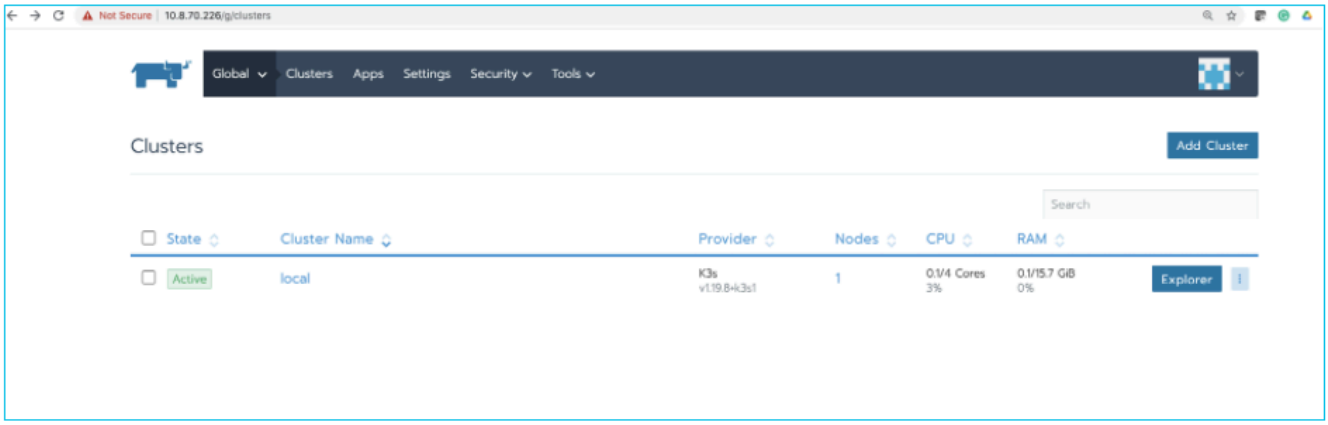
创建管理员用户后，将创建一个本地集群，如下所示：



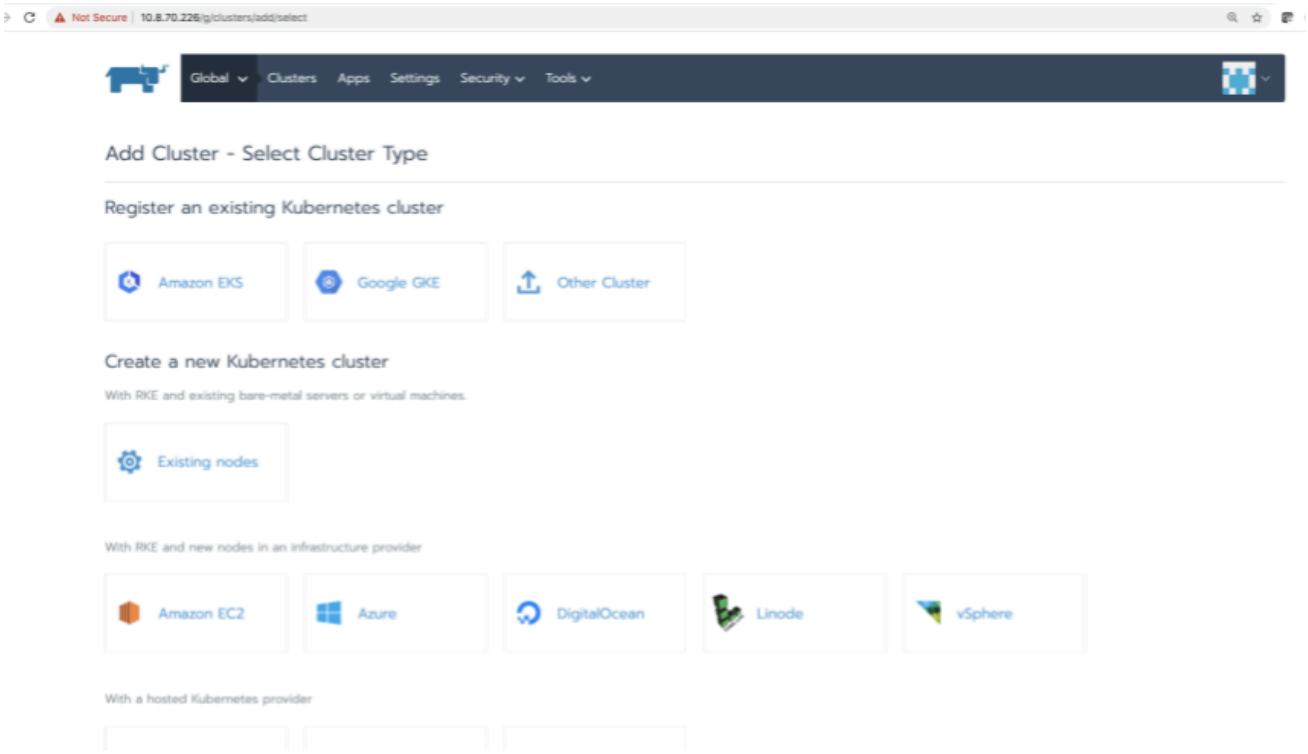
在 Rancher 集群上设置 Master 和 Worker 节点

在 Rancher UI上 创建本地集群，设置主节点和工作节点后，执行以下操作：

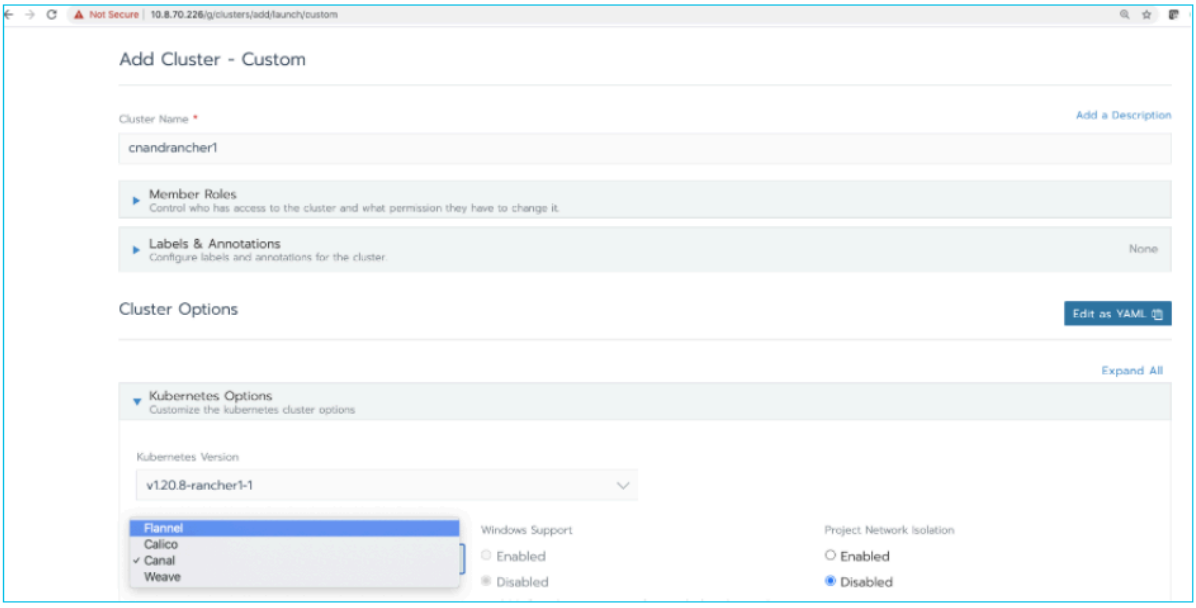
1. 进入 Rancher UI，然后点击 **Add Cluster**（添加集群）。



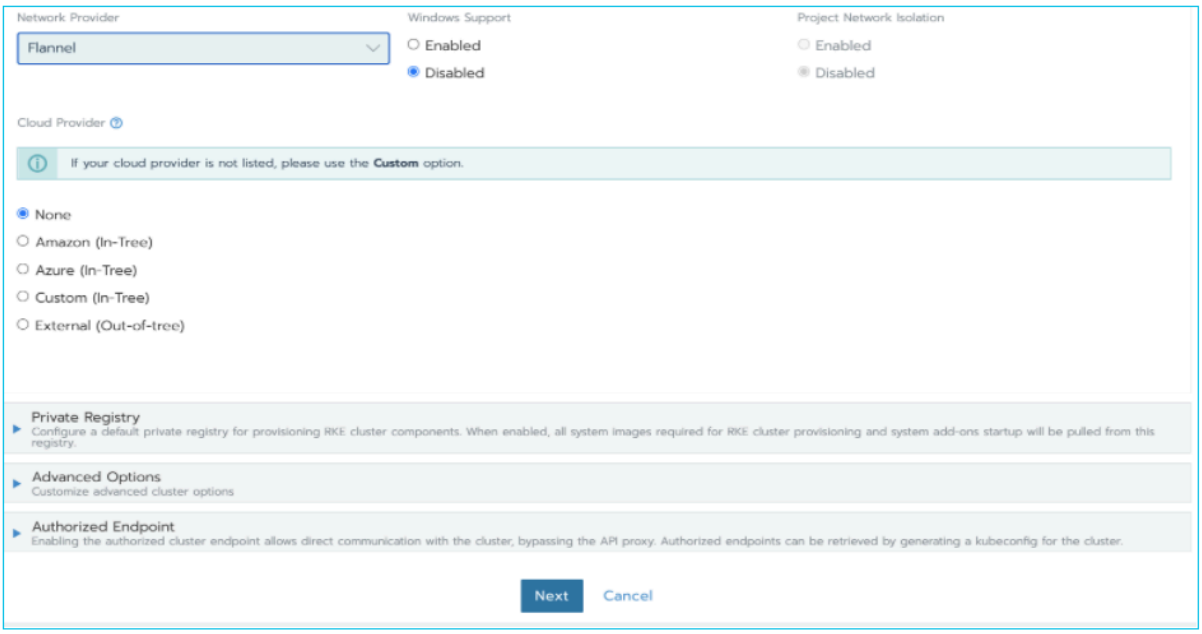
2. 单击 **Existing nodes**（现有节点）。



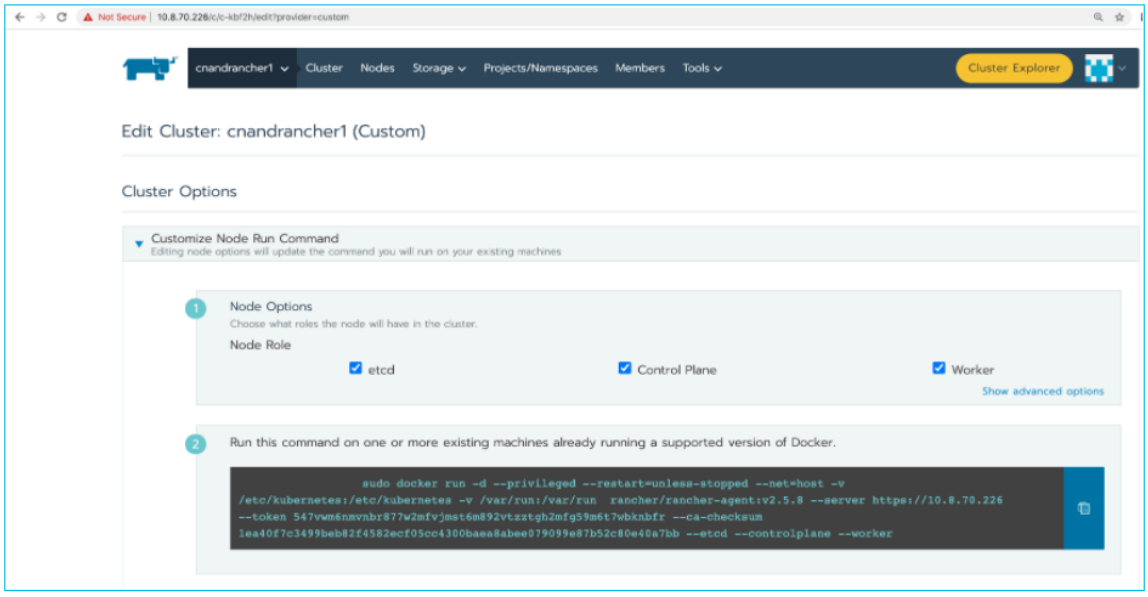
3. 输入 **Cluster name**（集群名称），然后从 **Network provider**（网络提供商）下拉列表中选择 Flannel。



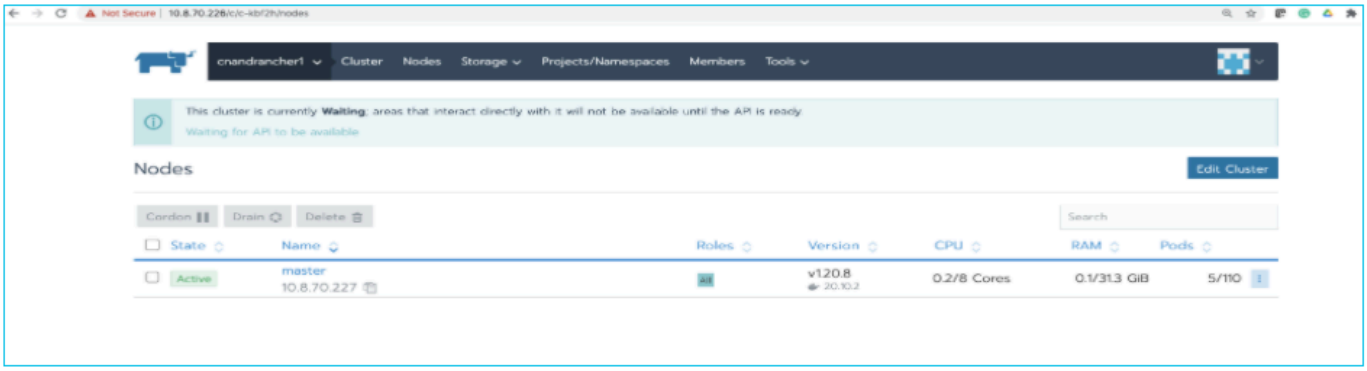
4. 保留所有其他字段的默认值，然后单击 **Next**（下一步）。



5. 在“节点”选项下，选择所有三个 **Node Role**（节点角色）选项，然后使用 SSH 在主节点上运行指定命令。



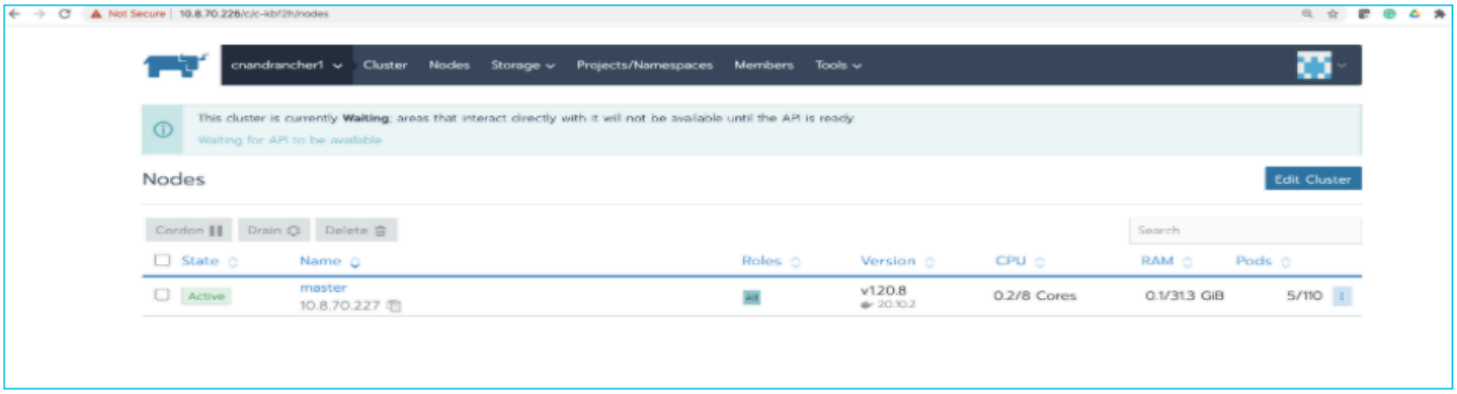
6. 验证主节点是否添加成功。



7. SSH 到每个 Worker 节点，并运行以下命令：

```
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
```

在一个 Master 节点和两个 Worker 节点上成功运行命令后，您将看到 Rancher 集群已准备就绪，如下所示：



修改 Rancher 集群选项 YAML 文件

在部署 CN 系列防火墙之前，必须按如下所述修改集群选项 YAML 文件。

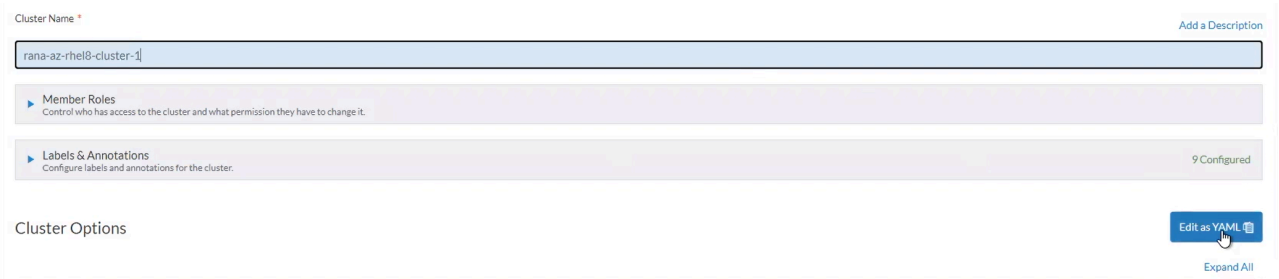
具有 *Rancher 2.5* 或更高版本（具有 *k8s 1.20.5*）支持的 *CN* 系列防火墙。

STEP 1 | 使用之前创建的管理员凭据登录 Rancher 门户。

STEP 2 | 单击导航菜单，然后选择集群管理。

STEP 3 | 找到要修改的集群，单击垂直省略号菜单，然后选择 **Edit Config**（编辑配置）。

STEP 4 | 单击 **Edit as YAML**（作为 **YAML** 编辑）。



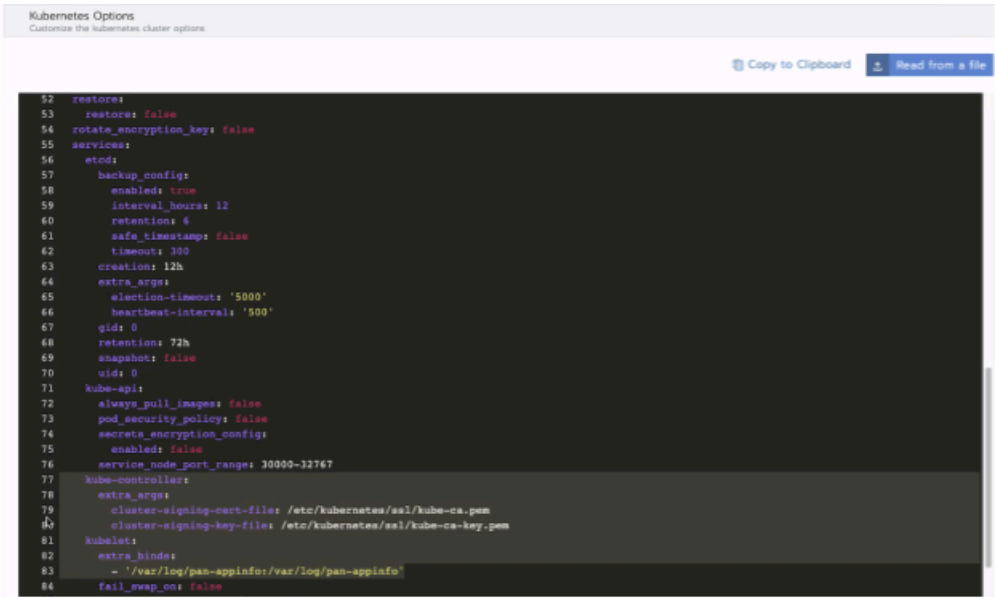
有关不同版本的 *Rancher*，请参阅 Rancher 文档。

STEP 5 | 在现有 YAML 文件的 **Services** 部分添加以下行。

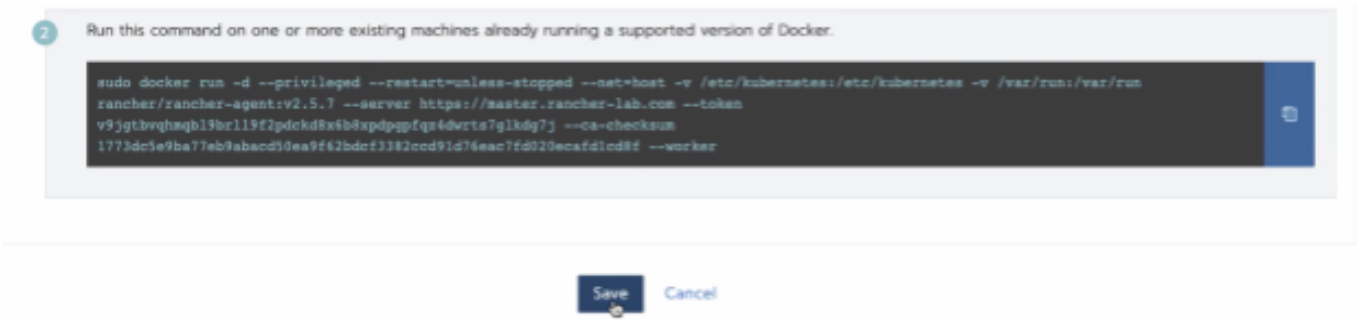
```
kube-controller: extra_args: cluster-signing-cert-file: "/etc/
kubernetes/ssl/kube-ca.pem" cluster-signing-key-file: "/etc/
kubernetes/ssl/kube-ca-key.pem"
```

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

如果使用的存储路径 *'/mnt'*，则要确保在 *extra_binds* 下修改存储路径。



STEP 6 | 单击保存并等待群集升级变为活动状态，然后部署 CN 系列防火墙。



CN 系列部署 YAML 文件中的可编辑参数

YAML 文件包括多个可编辑参数，下表列出了必须修改才能成功部署 CN 系列防火墙的参数。

- 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	
高级路由（Kubernetes 3.0.0 部署需要） `PAN_ADVANCED_ROUTING:” true”`	如果将高级路由与 Kubernetes 3.0.0 插件一起使用，则必须先在 PAN-OS 中启用它，然后在模板堆栈上手动配置。启用后提交并推送配置。有关详细信息，请参阅高级设置。
Panorama IP 地址 `PAN_PANORAMA_IP:`	包括 CN-MGMT Pod 将连接到的 Panorama IP 地址。如果您已在高可用性 (HA) 配置中配置 Panorama 管理服务器，请提供主要-活动 Panorama 的 IP 地址。 在 **Dashboard**（指示板） > **General Information**（常规信息）中可找到 Panorama IP 地址。
设备组名称 `PAN_DEVICE_GROUP:`	指定您希望为其分配 CN-NGFW Pod 的设备组名称。在 Panorama 中，您可以将相同策略推送到由一对 CN-MGMT Pod 托管（或属于 PAN-SERVICE-NAME）的所有 CN-NGFW Pod。 您可以在 **Panorama** > **Device Groups**（设备组）中找到设备组名称。
模板堆栈名称 `PAN_TEMPLATE_STACK:`	用于配置使防火墙 (CN-NGFW Pod) 能够在网络上运行的设置。

PAN-CN-MGMT-CONFIGMAP	
	您可以在 **Panorama** > 模板上找到模板堆栈名称。
日志收集器组名称 `PAN_PANORAMA_CGNAME:`	为 CN-NGFW 防火墙上生成的日志启用日志存储。如果不使用日志收集器组，则不会保存防火墙日志。 您可以在 **Panorama** 收集器组中找到 > **Collector Groups**（收集器组）名称。
（可选） `#CLUSTER_NAME:`	指定集群名称。CN-MGMT Pod 的主机名由 `PAN-CN-MGMT.yaml` 中定义的 StatefulSet 名称和此可选 CLUSTER_NAME 组合而成。如果在同一 Panorama 设备上管理多个集群，则使用此主机名可标识与不同集群相关联的 Pod。最佳做法是，在此处和在 Panorama 上的 Kubernetes 插件中使用同一名称。
（可选）Panorama HA 对等 IP 地址 `#PAN_PANORAMA_IP2:`	在高可用性设置中配置的 Panorama 对等（被动-次要）的 IP 地址。验证 PAN_PANORAMA_IP 是否是主要-活动 Panorama 的 IP 地址。 您可以在 **Panorama** > **High Availability**（高可用性）> **Setup**（设置）Panorama HA 对等体 IP 地址。
（对于 GTP，该参数为必需）GTP 安全 `#PAN_GTP_ENABLED: "true"`	在 CN 系列防火墙上为 GTP 安全启用此参数。启用 GTP 后，您可以在防火墙上使用 Panorama 配置 GTP 安全并监控 GTP 流量。
（如果主要 CNI 不使用 Jumbo 帧，则该参数对于 Jumbo 帧支持为必需）Jumbo 帧模式 `#PAN_JUMBO_FRAME_ENABLED: "true"`	CN-MGMT Pod 在启动过程中使用 eth0 MTU 自动检测是否启用 Jumbo 帧模式。因此，如果次要 CNI 使用 Jumbo 帧，而主要 CNI 不使用 Jumbo 帧，则必须定义 `PAN_JUMBO_FRAME_ENABLED:"True"` 才能在 VM 系列防火墙上启用 Jumbo 帧模式。 您必须在部署 CN-MGMT StatefulSet 之前进行此更改。
（对于灵活的系统资源分配，该参数为必需）	如果您需要更高的吞吐量，并且希望配置更多内存以满足部署需求，请使用此参数定义内存值。

PAN-CN-MGMT-CONFIGMAP	
• CN 系列即 DaemonSet #PAN_NGFW_MEMORY:"42Gi" • CN 系列即 K8s 服务 #PAN_NGFW_MEMORY:"6.5Gi" #PAN_NGFW_MEMORY:"42Gi" 对于 *5G* 本机安全，建议使用 *48Gi*。	• CN 系列即 DaemonSet 小容量不超过 42Gi，大容量大于 42Gi。 • CN 系列即 K8s 服务 小容量小于 6.5Gi，中等容量介于 6.5Gi 到42Gi 之间，大容量大于 42Gi。 此更改还要求在 *pan-cn-ngfw.yaml* 文件中分配相同或更大的内存。
（可选）AF-XDP #PAN_DATA_MODE: “next-gen”	启用地址系列 Express 数据路径 (AF-XDP) 需要此参数。 AF-XDP 是基于 eBPF 的套接字，为了提高有效吞吐量，它针对适用于云原生服务的高性能数据包处理进行了优化。这需要内核版本 5.4 或更高版本。此外，不支持巨型模式；EKS 无法使用此参数，因为默认情况下会启用巨型模式。 此外，PAN-CN-NGFW中还需要特权模式。
（启用 HPA 必需） （AKS 和 GKE）#HPA_NAME （仅限 EKS）#PAN_NAMESPACE_EKS （仅限 AKS）#PAN_INSTRUMENTATION_KEY	在 CN 系列防火墙即服务上启用水平 Pod 自动缩放 (HPA) 需要多个参数。 • 对于每个环境，必须提供一个唯一的名称来标识每个命名空间或每个租户的 HPA 资源。 • 对于 AKS 部署，必须提供 Azure 应用程序洞察工具密钥。

下列默认值已在 *pan-cn-mgmt-configmap.yaml* 文件中定义。

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

使用下列默认值可利用这些文件进行快速概念验证。如果要修改这些配置（例如，要部署多对具备容错功能的 PAN-MGMT Pod，以最多管理 30 个 PAN-NGFW Pod），则必须修改 `pan-mgmt-svc` 以使用另一个服务名称。修改这些默认值后，必须在其他 YAML 文件中更新相应的引用，以匹配在此文件中定义的值。

PAN-CN-MGMT-SECRET

PAN-CN-MGMT-SECRET	
VM 身份验证密钥 `PAN_PANORAMA_AUTH_KEY:`	允许 Panorama 对防火墙进行身份验证，以便可以将每个防火墙添加为托管设备。部署生命周期必须提供 VM 身份验证密钥。如果连接请求中没有有效密钥，则 CN 系列防火墙将无法注册到 Panorama。 请参阅为 CN 系列防火墙安装 Kubernetes 插件。

CN 系列的设备证书
`CN-SERIES-AUTO-REGISTRATION-PIN-ID`
`CN-SERIES-AUTO-REGISTRATION-PIN-VALUE`
	防火墙需要设备证书才能获取所有网站许可证授权，并安全访问 Palo Alto Networks 提供的云服务。在 Palo Alto Networks CSP 上生成 PIN ID 和 PIN 值，并在 PIN 到期前使用。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 的以下附加字段已被注释掉，并且也不是必填字段：*`"certificate.paloaltonetworks.com"`*
	请参阅在 CN 系列防火墙上安装设备证书。

PAN-CN-MGMT

PAN-CN-MGMT	
CN-MGMT 防火墙的 Init 容器映像的映像路径 `initContainers:  - name: pan-mgmt-init  image: <your-private-registry-image-path>`	Init 容器将生成证书，这些证书可用于保护 CN-MGMT Pod 的实例之间，以及 CN-MGMT Pod 和 CN-NGFW Pod 之间的通信。 编辑映像路径，以指向您已向其上传 CN-MGMT 容器的 Docker 映像的位置。

PAN-CN-MGMT	
CN-MGMT 映像容器的映像路径： `initContainers: - name: pan-mgmt image: <your-private-registry-image-path>`	编辑映像路径，以指向您已向其上传 CN-MGMT 容器的 Docker 映像的位置。
CN-MGMT 防火墙的主机名 `kind:StatefulSet metadata: name: pan-mgmt-sts`	CN-MGMT 防火墙的主机名由 StatefulSet 名称和可能在 `pan-cn-mgmt-configmap.yaml` 中定义的可选集群名称组合而成。 由于 StatefulSet 名称为 pan-mgmt-sts，并且未定义集群名称，因此 CN-MGMT Pod 的默认主机名为 pan-mgmt-sts-0 和 pan-mgmt-sts-1。 如果主机名长度超过 *30* 个字符，则会将其截断为 *30* 个字符。
（如果您已为灵活的系统资源分配定义内存，则该参数为必需）	如果您已分配大于或等于 40G 的内存值 (`#PAN_NGFW_MEMORY:"40Gi"`，在 `pan-cn-mgmt-configmap.yaml` 中），则对于 CPU 和内存，请确保 `request` 和 `limit` 中的值相同，从而在以下代码中实现更高的容量利用率： `containers: resources: requests: # configurable based on desired logging, capacities cpu:"4" memory:"16.0Gi" limits: cpu:"4" memory:"16.0Gi"` 对于 5G 本机安全，建议值为 CPU=4，内存=16Gi。
（仅适用于本地或自托管 Native Kubernetes 部署） `storageClassName: local`	对于自托管部署，默认配置为“storageClassName: local”。 如果集群已动态配置持久卷 (PV)，则必须修改“storageClassName: local”以匹配 storageClass；或者，如果使用 DefaultStorageClass，则删除这几行。 如果集群未动态配置 PV，则集群管理员可以使用提供的 `pan_cn_pv_local.yaml` 创建静态 PV，其中包含 2 组少量 PV，每个 PAN-CN-MGMT statefulSet Pod 各一组。您可以修改

PAN-CN-MGMT	
	pan_cn_pv_local.yaml 以匹配设置中的持久卷，然后在部署 PAN-CN-MGMT.yaml 之前进行部署。

PAN-CN-NGFW-CONFIGMAP

除非需要更改以下内容，否则无需修改任何 PAN 值：

- PAN_SERVICE_NAME: pan-mgmt-svc

  服务名称应与您在 PAN-CN-MGMT-CONFIGMAP 中定义的名称匹配。

- FAILOVER_MODE: failopen

  您可以将其更改为 failclose。仅当 CN-NGFW 无法获取许可证时，此更改才生效。

  - 在 failopen 模式下，防火墙将接收数据包，并将其发送出去而不进行检查。转换为 failopen 模式可能会导致内部重新启动并短暂中断流量。
  - 在 failclose 模式下，防火墙将丢弃收到的所有数据包。同时，failclose 模式还可能会关闭 CN-NFGW 并释放分配的插槽，以让许可的其他 CN-NFGW 使用该插槽。

- CPU 定位 — 在 pan-cn-ngfw-configmap.yaml 文件中，将禁用 CPU 定位和超线程。除非在 Palo Alto Networks 支持人员的指导下，否则请勿切换此设置为专用物理内核（而非具有超线程的逻辑内核）启用 CPU 定位。

  ```
  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: <your-private-registry-image-path>`	编辑映像路径，以指向您已向其上传 CN-NGFW 容器的 Docker 映像的位置。
（如果您已为灵活的系统资源分配定义内存，则该参数为必需）	如果您已分配大于或等于 40G 的内存值 (#PAN_NGFW_MEMORY:"40Gi"，在 pan-cn-mgmt-configmap.yaml 中），请确保用于 CPU 和内存的 request 和 limit 中的值相同，从而在以下代码中实现有保证的 QoS `containers: resources: requests: #configurable based`

PAN-CN-NGFW	
	`on desired throughput, number of running pods cpu:"1" memory:"40.0Gi" limits: cpu:"1" memory:"40.0Gi"` 对于 5G 本机安全，建议值为 CPU=12，内存=48Gi。

注意：
• 以下注释可标识 PAN-NGFW DaemonSet：
`paloaltonetworks.com/app: pan-ngfw-ds`
请勿修改此值。
• 以下注释可标识防火墙名称 (“pan-fw”)：
`paloaltonetworks.com/firewall: pan-fw`
在 `pan-cni-configmap.yaml` 文件中，该防火墙名称必须在以下注释中完全匹配：`cni_network_config: “firewall”`
该注释应在用于部署每个应用程序 Pod 的应用程序 yaml 文件中完全匹配。
	每个节点上的 CN-NGFW Pod 可保护具有注释的应用程序 Pod 和命名空间：
	`paloaltonetworks.com/firewall: pan-fw`
	保持此注释不变。

（可选）AF-XDP `imagePullPolicy:Always securityContext: capabilities: #add: ["NET_ADMIN","NET_RAW","NET_BROADCAST","NET_BIND_SERVICE"] add: ["ALL"] privileged: true resources:`	您必须在左侧显示的部分中添加 `privileged: true`。启用地址系列 Express 数据路径 (AF-XDP) 需要此参数。 您还必须在PAN-CN-MGMT-CONFIGMAP中启用 AF-XDP。

PAN-CNI-CONFIGMAP

这些参数是可选的。

PAN-CNI-CONFIGMAP	
应用程序 Pod 可能所属的防火墙名称列表： `"firewall": [ "pan-fw" ]`	尽管不需要进行任何修改，但如果更改 `pan-cn-ngfw.yaml` 中的 `paloaltonetworks.com/firewall:`

PAN-CNI-CONFIGMAP	
	pan-fw 注释，则必须替换 "firewall": ["pan-fw"] 中的值以进行匹配。
"exclude_namespaces": []	尽管不需要进行任何修改，但如果要排除特定命名空间，请将其添加到“**exclude_namespaces"**，以便忽略该命名空间中的应用程序 Pod 注释，并且不会将流量重向到 CN-NGFW Pod 进行检查。
"security_namespaces": ["kube-system"]	在 security_namespaces 中添加已在其中部署 CN-NGFW DaemonSet 的命名空间。默认命名空间为 kube-system。

“interfaces”	在您要从中将流量重定向到 CN-NGFW Pod 进行检查的应用程序 Pod 中添加接口。默认情况下，仅检查 eth0 流量，并且您可以将其他接口添加为以逗号分隔的字符串列表，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	
	除此之外，您还必须将 *pan-cni* 附加到应用程序 *Pod* 中的 *k8s.v1.cni.cncf.io/networks* 注释。 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` 目前，*CN* 系列不支持 *DPDK*，并且不允许应用程序 *Pod* 使用 *DPDK*。如果应用程序无法自动调整为非 *DPDK* 模式，则可能需要修改应用程序 *Pod*。
（仅 CN 系列即 Kubernetes 服务） “dpservicename” “dpservicenamespace”	当 CN 系列作为服务部署时，需要使用 dpservicename 和 dpservicenamespace。默认情况下，dpservicename 是 “pan-ngfw-svc”，dpservicenamespace 是 “kube-system”。

PAN-CNI

PAN-CNI	
在每个节点上具有 CNI 二进制文件和 CNI 网络配置文件的 PAN-CNI 容器映像的映像路径。 `containers: name: install-pan-cni image: <your-private-registry-image-path>`	编辑映像路径，以指向您已向其上传 PAN-CNI 容器的 Docker 映像的位置。

PAN-CNI-MULTUS

如果在 Kubernetes 的自托管或本机实施中使用 Multus CNI（例如，与 VMware TKG+ 一起使用），请使用 `pan-cni-multus.yaml` 文件，而不是 `pan-cni.yaml` 文件。

使用 CN 系列防火墙保护 5G

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

要查看和控制 Kubernetes 上私营企业和移动运营商网络中 5G 移动分组核心部署的 5G 流量，请查看以下各节以了解支持的环境，以及如何修改 YAML 文件以在 CN 系列防火墙上充分利用 GTP 安全和 5G 本机安全。在部署 CN 系列防火墙时，除启用这些功能之外，您还必须为 GTP 安全和/或 SCTP 安全启用 Panorama。

容器运行时	Docker CRI-O Containerd
Kubernetes 版本	1.17 至 1.27
云提供商托管 Kubernetes	• AWS EKS（1.17 至 1.27，适用于 CN 系列即守护进程集和 CN 系列即服务部署模式。） • AWS EKS（1.17 至 1.22，适用于 CN 系列即 CNF 部署模式。） • AWS EKS（1.22 至 1.27，适用于 CN 系列即 CN 集群部署。） • AWS Outpost 上的 AWS（1.17 至 1.25） *AWS Outpost* 上 *EKS* 的 *CN* 系列不支持 *SR-IOV* 或*Multus*。 • Azure AKS（1.17 至 1.27） 在 *Azure AKS* 中，*PAN-OS 11.0.2* 是支持 *kubernetes 1.25* 及以上版本所需的最低版本。

	• GCP GKE（1.17 至 1.27） 包括 *GKE Dataplane V2*。 • OCI OKE (1.23)
客户托管 Kubernetes	在公共云或本地数据中心上。 确保此表中已列出 Kubernetes 版本、CNI 类型和 Host VM OS 版本。 VMware TKG+ 版本 1.1.2 • 基础架构平台 — vSphere 7.0 • Kubernetes Host VM OS — Photon OS
Kubernetes 主机虚拟机	操作系统 • Ubuntu 16.04 • Ubuntu 18.04 • Ubuntu-22.04 • RHEL/Centos 7.3 及更高版本 • CoreOS 21XX、22XX • Container-Optimized OS
	Linux 内核版本： • 4.18 或更高版本（仅限 K8s 服务模式） • 5.4 或更新版本需要启用 AF_XDP 模式。有关详细信息，请参见 CN 系列部署 YAML文件中的可编辑参数。
	Linux Kernel Netfilter：Iptables
CNI 插件	CNI Spec 0.3 及更高版本： • AWS-VPC • Azure • Calico • Flannel • Weave • 适用于 Openshift、OpenshiftSDN

	• 作为 DaemonSet，CN 系列防火墙支持以下内容。
	  • Multus
	  • Bridge
	  • SR-IOV
	  • Macvlan

OpenShift	• 版本 4.2、4.4、4.5、4.6、4.7、4.8、4.9、4.10、4.11、4.12 和 4.13。 *OpenShift 4.7* 在 *CN* 系列上仅作为 *DaemonSet*。 *PAN-OS 11.0.2* 是支持 *4.12* 及以上版本所需的最低版本。 • AWS 上的 OpenShift

容器运行时	版本
CN 系列防火墙	PAN-OS 10.0.3 或更高版本
Kubernetes 插件	1.0.1 或更高版本
Panorama	10.0.0 或更高版本
以下是 YAML 文件中用于部署 CN 系列防火墙的所有可编辑参数列表：有关详细信息，请参阅 CN 系列部署 YAML 文件中的可编辑参数和 CN 系列核心构建块。	
启用 GTP	部署 CN-MGMT StatefulSet 之前，在 pan-cn-mgmt-configmap.yaml 文件中，请执行以下设置：`PAN_GTP_ENABLED :"True"`。
启用 Jumbo 帧模式	部署 CN-MGMT StatefulSet 之前，在 pan-cn-mgmt-configmap.yaml 文件中，请执行以下设置：`PAN_JUMBO_FRAME_ENABLED:"True"`。 CN-MGMT Pod 在启动过程中使用“eth0”MTU 自动检测是否启用 Jumbo 帧模式。因此，如果次要 CNI 使用 Jumbo 帧，而主要 CNI 不使用 Jumbo 帧，则必须定义 `PAN_JUMBO_FRAME_ENABLED:"True"` 才能在 CN 系列防火墙上启用 Jumbo 帧模式。

容器运行时	版本
	目前，*CN* 系列不支持 *DPDK*，并且不允许应用程序 *Pod* 使用 *DPDK*。如果应用程序无法自动调整为非 *DPDK* 模式，则可能需要修改应用程序 *Pod*。
启用系统资源灵活性	如果您需要更高的吞吐量，并且希望在 pan-cn-mgmt-configmap.yaml 中配置更多内存以满足部署需求，请执行以下设置：PAN_NGFW_MEMORY="48Gi" 对于模板 *(Helm)*，可使用与分配给 *CN-NGFW Pod* 相同的变量。启用更大的内存占用量时，*CN-MGMT StatefulSet* 仅支持一个 *CN-NGFW Pod*。
为 5G 配置 vCPU 内存	CN-MGMT Pod（pan-cn-mgmt.yaml 文件中）和 NGFW Pod（pan-cn-ngfw.yaml 文件中）的建议配置是 CPU 和内存在“request”和“limit”中具有相同的值，以实现有保证的 QoS。 对于 CN-MGMT Pod，建议值为 CPU=4，内存=16Gi。要控制 CN-MGMT Pod 的位置（例如在与部署 CN-NGFW Pod 相同或不同的节点上），请使用 k8s 中的节点选择器功能。
	对于 CN-NGFW Pod，建议值为 CPU=12，内存=48Gi。要控制 CN-NGFW Pod 的位置（例如在与部署 CN-NGFW Pod 相同或不同的节点上），请使用 k8s 中的节点选择器功能。
选择 CNI yaml 文件	Multus CNI 可用作调用其他 CNI 插件的元插件。在 OpenShift 环境中，默认启用 Multus，因此可使用 `pan-cni.yaml` 文件。在支持 Multus 但为可选的其他环境（例如，自托管（本机）环境）中，请使用 `pan-cni-multus.yaml` 文件，而不是 `pan-cni.yaml` 文件。

在继续部署 CN 系列防火墙之前，还需要查看 CN 系列防火墙系统要求。

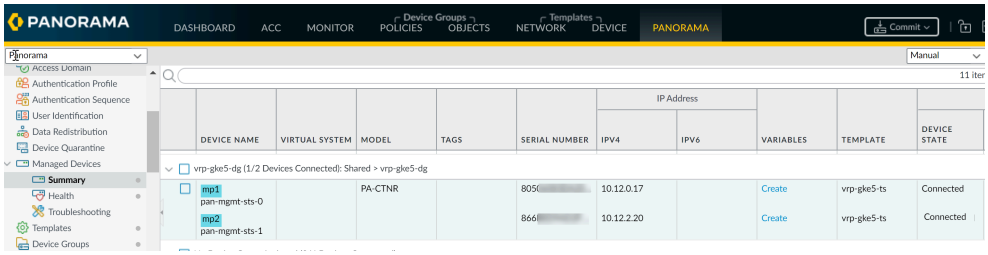
配置 Panorama 以保护 Kubernetes 部署

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 图表部署 CN 系列的 Helm 3.6 or above version client

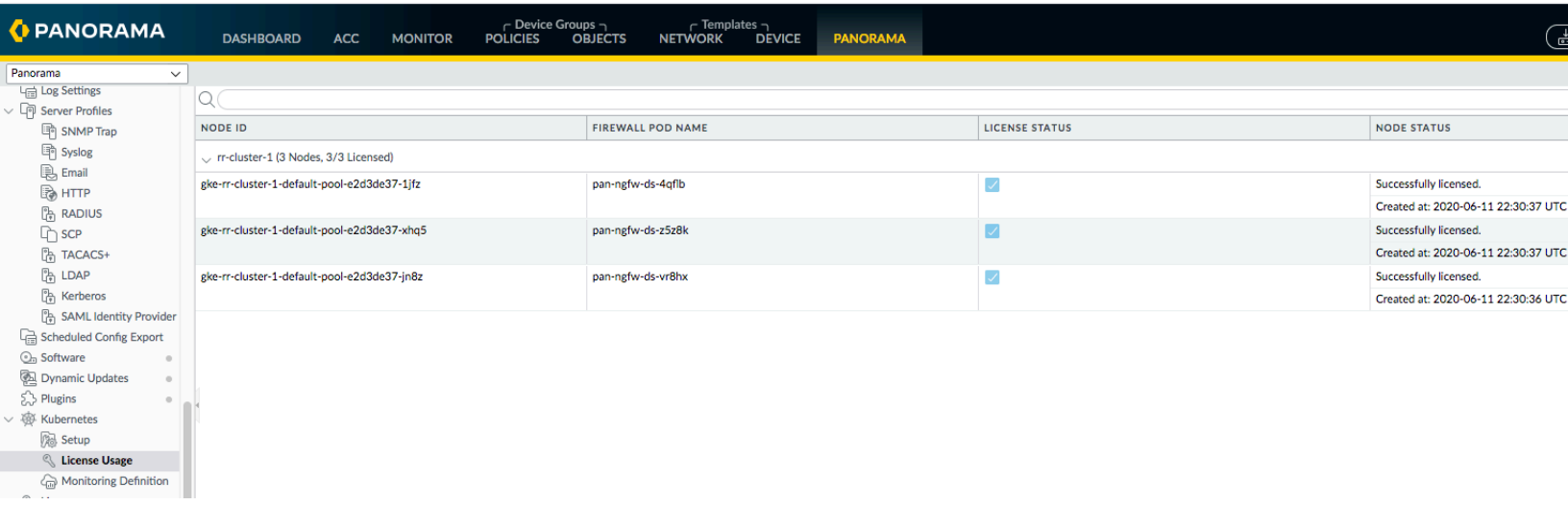
为 CN 系列安装 Kubernetes 插件并部署 CN 系列防火墙后，要监控 Kubernetes 集群和配置启用流量实施的安全策略，您需要完成以下任务

STEP 1 | 验证是否已在 Panorama 上注册 CN-MGMT Pod 且已许可 CN-NGFW Pod。

1. 选择 **Panorama** > **Managed Devices**（托管设备） > **Summary**（摘要）。



2. 选择 **Panorama** > **Plugins**（插件） > **Kubernetes** > **License Usage**（许可证使用情况），以验证是否已为集群中的每个节点分配许可证令牌。



STEP 2 | 创建日志转发配置文件以将日志转发到 Panorama。

该配置文件可定义将在防火墙上生成的不同日志的目标。

1. 从 **Device Group**（设备组）下拉列表中选择为 k8s 部署创建的设备组。
2. 选择 **Objects**（对象）> **Log Forwarding**（日志转发），然后单击 **Add**（添加）。
3. 输入 **Name**（名称）以标识配置文件。如果您希望为新安全规则和区域自动分配配置文件，请输入 **`default`**（默认值）。如果您不需要默认配置文件，或者您希望覆盖现有默认配置文件，请输入一个 **Name**（名称），在将配置文件分配到安全规则时，帮助您标识配置文件。
4. **Add**（添加）要转发的日志类型。
5. 单击 **OK**（确定）。

STEP 3 | 配置 Kubernetes 插件以将标签推送到指定设备组。

您必须添加监控定义，其中包括 Kubernetes 集群的名称，Panorama 将从该集群中检索预定义标签和可选通知组。

如果 *CN* 系列部署在 *kube-system* 以外的命名空间中，则需要通知组。

通知组是接收标签更新的设备组列表。对于 Kubernetes 插件，通知组应包括集群外部的防火墙（这意味着它们不属于与您从中收集属性的 Kubernetes 集群相同的设备组）。

由于您在用于部署 CN 防火墙的 YAML 文件中指定设备组名称，因此 Kubernetes 插件将自动获悉集群内部的所有设备组，并且默认会自动将所有预定义标签推送到这些设备组。

Kubernetes 插件使用 Kubernetes Secret 动态了解每个集群中的设备组。每次部署 CN-MGMT StatefulSet 时，都会将 Kubernetes Secret 发布到 Kubernetes API 服务器，并且 Panorama 会在下一个监控间隔中进行了解。

1. 设置 Kubernetes 插件以监控集群。
2. 添加通知组。添加通知组，并选择接收与 Kubernetes 集群相关的标签的设备组。
    1. 选择 **Panorama** > **Plugins**（插件） > **Kubernetes** > **Setup**（设置） > **Notify Groups**（通知组），然后选择 **Add**（添加）。
    2. 为通知组输入一个最大长度为 31 个字符的 **Name**（名称）。
    3. 如果您要共享除为集群创建的外部标签（默认）之外的内部标签，请选择 **Enable sharing internal tags with Device Groups**（启用与设备组共享内部标签）。
    4. 选择要向其注册标签的设备组。

        对于所选通知组，Panorama 仅推送外部标签。

        外部标签是可从集群外部访问的所有标签，例如为集群 IP 地址的外部服务 IP 地址和端口，所有节点和节点端口的外部 IP 地址，以及端口或节点端口的外部负载均衡器 IP 地址生成的标签。

        内部标签包括有关内部集群 IP 地址、Pod IP 地址、节点和节点端口的详细信息。

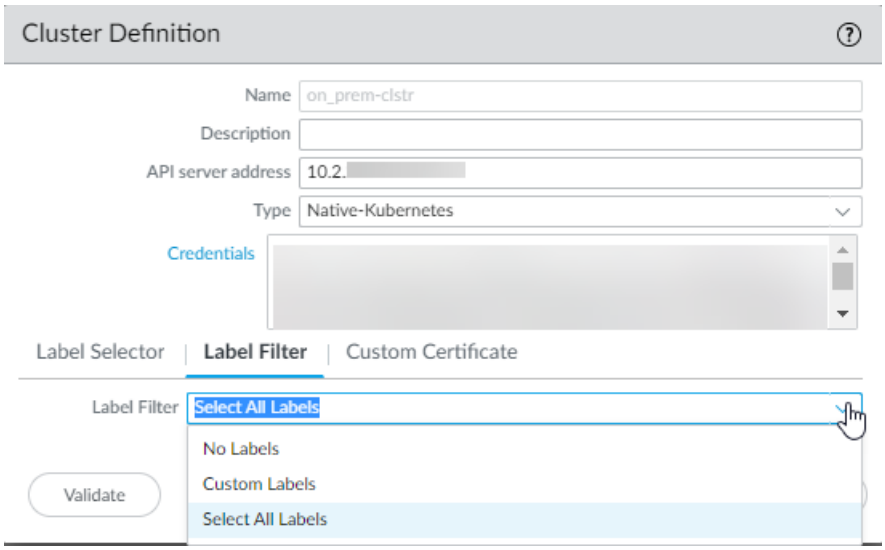
        默认情况下，Panorama 会将所发现的所有标签（根据所选的标签筛选器）到与集群相关联的设备组，如用于部署 CN-MGMT Pod 的 YAML 文件中所定义。
3. 为每个集群添加监控定义。
    1. 选择 **PanoramaPlugins** > **Kubernetes** > **Monitoring Definition**（监控定义），然后选择 **Add**（添加）。
    2. 为监控定义输入一个 **Name**（名称）。
    3. 选择要监控的 **Cluster**（集群）。
    4. （可选）选择要向其发送 IP 地址到标签映射的信息的 **Notify Group**（通知组）。

        默认情况下，标签将与集群中的所有 CN-NFGW Pod 共享。
    5. 单击 **OK**（确定）保存更改。

4. **Commit**（提交）到 Panorama。

STEP 4 | （可选）设置 Kubernetes 插件以从应用程序 YAML 文件中检索用户定义的标签。

1. 选择 **Panorama Plugins**（**Panorama** 插件） > **Kubernetes** > **Setup**（设置） > **Cluster**（集群），然后从列表中选择集群定义。
2. 从以下选项中选择标签筛选器：



1. **No Labels**（无标签）— 不为 Kubernetes 标签创建任何标签。
2. **Custom Labels**（自定义标签）— 仅为感兴趣的标签创建标签。

   要使用自定义标签，必须先在 Kubernetes 环境中为 YAML 文件添加注释，然后使用以下任意组合为相应的 IP 地址生成自定义标签：

   指定命名空间、键和值。可以使用 * 替代全部输入。当所有三个输入均有效时，该插件将增加标签数量。

   指定命名空间和键，为该命名空间中的所有匹配键创建标签。

   仅指定命名空间，为该命名空间中的每个标签创建标签。
3. **Select All Labels**（选择所有标签）— 为所有 Kubernetes 标签（包括所有自定义标签）创建标签。

3. 添加标签选择器表达式。

   标签选择器与 Kubernetes 集群中的指定标签进行匹配，并将与标签相关联的 IP 地址映射到单个标签。有关支持的前缀列表，请参阅Kubernetes 属性的 IP 地址到标签映射。

   对于每个标签选择器，Panorama 将会生成一个标签，该标签可用作动态地址组中的匹配条件，并用于实施安全策略：

   1. 标签前缀 — 每个标签结尾的短语，有助于轻松标识标签。例如，标签选择器 k8s.cl_<clustername>.<selector-name>，在与选择器匹配的所有 ClusterIP 和所有 Pod IP

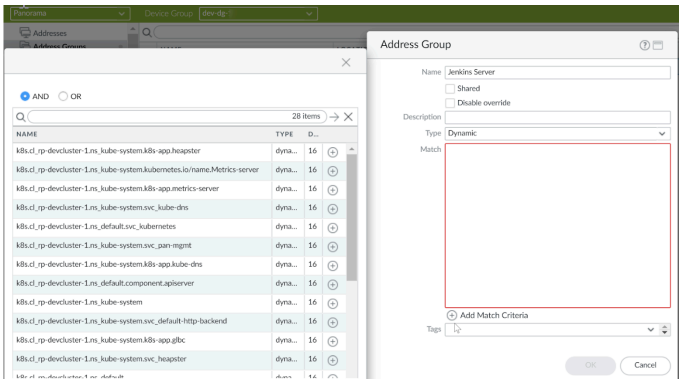
地址上匹配。这些标签可能在所有命名空间中，也可能在特定命名空间中，具体取决于所配置的内容。

2. 命名空间 — * 表示所有命名空间，或输入命名空间的名称。
3. 标签选择器筛选器 — Kubernetes 插件支持标签键和标签值的基于集合和基于等式的选择器。支持基于等式的以下选择器 — key = value；key ==value；key != value，例如 app = redis。您还可以在表达式中将多个选择器指定为以逗号分隔的列表，例如 app == web，tier != backend。支持基于集合的以下选择器 — key in (value1,value2)、key notin (value1, value2)、key、!key，例如 tier notin (frontend, backend)。
4. 应用于 — 应用此标签的资源类型为服务、Pod、全部。

STEP 5 | 设置动态地址组。

1. 选择用于管理 CN-NGFW Pod 的设备组。
2. 选择 **Object**（对象） > **Address Groups**（地址组）。
3. 单击 **Add**（添加），然后为地址组输入 **Name**（名称）和 **Description**（说明）。
4. 选择 **Dynamic**（动态）作为 **Type**（类型）。

STEP 6 | 单击 **Add Match Criteria**（添加匹配条件），并选择 **AND** 或 **OR** 运算符，然后选择要筛选或匹配的属性。

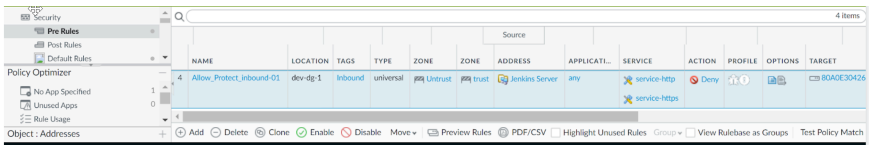


STEP 7 | 单击 **OK**（确定）和 **Commit on Panorama**（在 **Panorama** 上提交）。

使用 **more...**（更多...）链接查看与对象相关联的 IP 地址，在本例中为集群中的 Jenkins 服务器。

STEP 8 | 创建用于流量实施的安全策略规则。

1. 选择 **Policies**（策略） > **Security**（安全）。
2. 单击 **Add**（添加），然后为策略输入 **Name**（名称）和 **Description**（说明）。
3. 添加 **Sources Zone**（源区域）来指定产生流量的区域。
4. 添加流量于其中终止的 **Destination Zone**（目标区域）。
5. 对于 **Destination Address**（目标地址），选择刚创建的动态地址组。
6. 为流量指定操作 — **Deny**（拒绝），并可选地将默认安全配置文件附加至规则。
7. 选择 **Actions**（操作）选项卡，并选择您创建的 **Log Forwarding profile**（日志转发配置文件）。
8. 单击 **Commit**（提交）。



您还可以为命名空间中的东向西流量应用安全策略。例如，如果在一个名为 staging cluster 的集群中有两个命名空间：stage-ns 和 db-ns，其中已在 stage-NS 中为投票应用程序部署前端 Pod，而 Redis 后端 Pod 在 DB-NS 命名空间中运行。当将此集群添加到 Panorama 上的 Kubernetes 插件进行监控时，它会检索标签元数据以创建标签。您可以使用这些标签实施安全策略规则。要执行此操作，您需要

- 确保使用 paloaltonetworks.com/firewall: pan-fw 为用于部署前端和后端应用程序的命名空间或 YAML 文件添加注释。
- 为前端和后端 Pod 创建动态地址组。

  您必须在与集群关联的设备组中配置动态地址组，然后先选择前端服务器的标签。接下来，重复此过程可为后端服务器创建另一个动态地址组
- 添加安全策略规则以允许 Redis 应用程序从前端 Pod 到后端 Pod 的流量。

  来源是前端服务器的动态地址组，目标是后端服务器的动态地址组，操作为允许。

Kubernetes 属性的 IP 地址到标签映射

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

Panorama 上的 Kubernetes 插件可为 Kubernetes 集群中的预定义标签、Pod 和服务的用户定义的标签和服务对象创建标签。

该插件可为以下 Kubernetes 对象创建标签：

- Pod 类：ReplicaSets、DaemonSets、StatefulSets
- 服务类型：ClusterIP、NodePort、LoadBalancer
- 服务对象：端口、目标端口、节点端口和 Pod 接口

默认情况下，Panorama 上的 Kubernetes 插件从您正在 Panorama 上监控的每个 Kubernetes 集群中检索以下预定义标签，并以下面列出的格式创建标签。然后，您可以将这些标签用作动态地址组中的匹配条件，并对与每个标签相关联的基础 IP 地址实施安全策略。

每个标签的最大长度为 *127* 个字符。如果标签超过最大字符数，则会被截断。如果两个截断的标签相同，则会在标签中添加唯一哈希值来进行区分。

您可以使用 Kubernetes 插件将 Kubernetes 集群中部署的 Pod、节点、命名空间和服务的 IP 地址到标签映射分发到物理或 VM 系列防火墙，即使您在该集群中未部署 CN 系列防火墙。

预定义标签	**Panorama** 上的标签格式	收集的 **IP** 地址
DaemonSet	k8s.cl_<cluster-name>.ns_<namespace>.ds_<pod-name>	Pod IP 地址
ReplicaSet	k8s.cl_<cluster-name>.ns_<namespace>.rs_<pod-name>	Pod IP 地址
StatefulSet	k8s.cl_<cluster-name>.ns_<namespace>.ss_<pod-name>	Pod IP 地址
服务	k8s.cl_<cluster-name>.ns_<namespace>.svc_<svc-name>	集群 IP 地址 Pod IP 地址
外部服务	k8s.cl_<cluster-name>.ns_<namespace>.exsvc_<svc-name>	外部服务 IP 地址 负载均衡器 IP 地址
节点	k8s.cl_<cluster-name>.nodes	所有节点的专用 IP 地址
外部节点	k8s.cl_<cluster-name>.ex_nodes	所有节点的公用 IP 地址
命名空间	k8s.cl_<cluster-name>.ns_<namespace>	命名空间中所有集群 IP 地址 命名空间中所有 Pod IP 地址
接口	• k8s.cl_<cluster-name>.ns_<namespace>.ds_<daemonset-name>.if_<interface>	您的部署中每个容器上所有接口的所有 IP 地址。

预定义标签	**Panorama** 上的标签格式	收集的 **IP** 地址
	• k8s.cl_<cluster-name>.ns_<namespace>.rs_<replicaset-name>.if_<interface> • k8s.cl_<cluster-name>.ns_<namespace>.ss_<statefulset-name>.if_<interface>	

如果您使用标签组织 Kuberbetes 集群中的 Pod 和服务，则 Panorama 上的 Kubernetes 插件可查询这些标签并创建标签。支持以下用户定义的标签：

用户定义的标签	**Panorama** 上的标签格式	收集的 **IP** 地址
标签	k8s.cl_<cluster-name>.ns_<namespace>.<label-key>.<label-value>	该命名空间中与指定标签匹配的所有集群 IP 地址。 该命名空间中与指定标签匹配的所有 Pod IP 地址。
标签选择器	k8s.cl_<cluster-name>.<selector-name>	与指定选择器匹配的所有集群 IP 地址。 与指定选择器匹配的所有 Pod IP 地址。

标签选择器将根据 Kubernetes 集群中的 Pod 和服务与指定标签进行匹配，并将与标签相关联的 IP 地址映射到单个标签。Kubernetes 插件支持标签键和标签值的基于集合和基于等式的选择器。

支持以下基于等式的选择器：

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

您还可以在表达式中将多个选择器指定为以逗号分隔的列表。例如：

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

支持以下基于集合的选择器：

- key in (value1, value2)
- key notin (value1, value2), for example, tier notin (frontend, backend)
- key
- !key

对于监控的服务对象，该插件使用以下命名方案为端口、targetPort 和 nodePort 服务对象生成端口：

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

哈希可确保即使在 k8s 集群之间有重叠的命名空间和服务名称，服务对象也是唯一。

启用带标签的 VLAN 流量检查

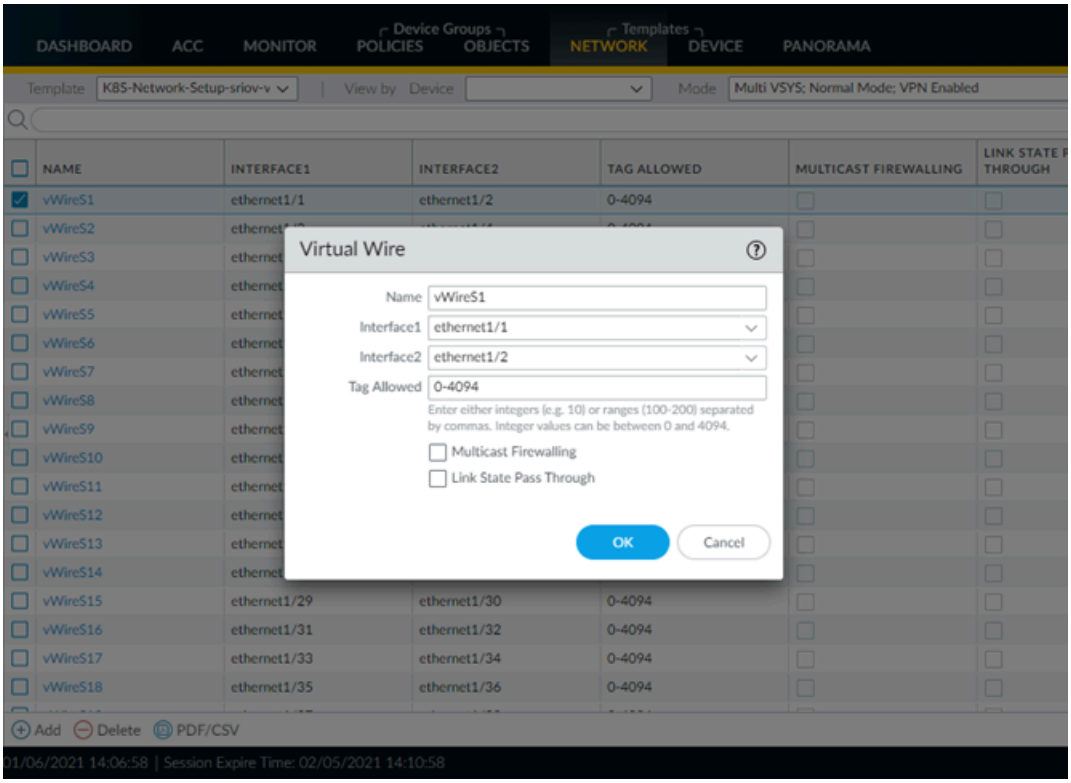
在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

完成以下过程，启用 CN 系列防火墙以检查带标签的 VLAN 流量。要检查带标签的 VLAN 流量，必须在 Panorama 上更新所有 Virtual Wire 的配置以允许所有 VLAN 标签。然后，您必须为应用程序 Pod YAML 文件添加注释，以将 VLAN 标签分配给应用程序 Pod 接口。此注释会告诉 CN-NGFW 哪些标签应用于通过防火墙发送的数据包。

不支持双 *VLAN* 标签。

STEP 1 | 在 CN-NGFW 的所有接口上启用所有 VLAN。

1. 登录到 Panorama。
2. 选择 **Network**（网络）> **Virtual Wire**。
3. 从 **Template**（模板）下拉列表中选择 **K8S-Network-Setup** 模板。
4. 选择第一个 Virtual Wire。
5. 将 **Tag Allowed**（允许的标签）设置为 0-4094。
6. 对于每个 Virtual Wire，请重复此过程。
7. **Commit**（提交）更改。



STEP 2 | 在应用程序 Pod YAML 文件后面附加以下注释，以在每个接口上应用静态 VLAN ID。

每个接口仅支持一个 *VLAN* 标签。

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 } ]'
```

启用 IPVLAN

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.1.x or above Container Images • 运行 PAN-OS 10.1.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

IPVLAN 是虚拟网络设备的驱动程序，可在容器化环境中用于访问主机网络。在 L2 模式下，无论在主机网络中创建的 IPVLAN 设备数量是多少，IPVLAN 都会向外部网络公开一个 MAC 地址。所有逻辑 IP 接口都使用相同的 MAC 地址。这样可以避免在父网卡上使用杂乱模式，并防止对 NIC 或交换机造成潜在的 MAC 限制。

现在，您可以在 CN 系列防火墙中使用 IPVLAN，但有以下限制。

- 需要 PAN-OS 10.1.2 及更高版本
- 仅限 IPv4
- 仅限 L2 模式
- 每个接口一个 IP 地址
- 如果您使用的是 Multus，请部署 **pan-cni-multus.yaml**，而不是 **pan-cni.yaml**。此外，您必须在部署了 Multus 应用程序 Pod 的每个命名空间中部署 pan-cni-net-attach-def.yaml。

同一台主机（共享同一父接口）中的 *IPVLAN* 子接口通信不起作用。

您必须注释掉应用程序 Pod yaml 文件才能启用 IPVLAN；任何 CN 系列 yaml 文件启用 IPVLAN 都不需要更改。以下是 IPVLAN 的网络附件定义的示例。请注意，模式设置为“**l2**”。CN 系列防火墙仅支持 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" } ] } }'
```

卸载 Panorama 上的 Kubernetes 插件

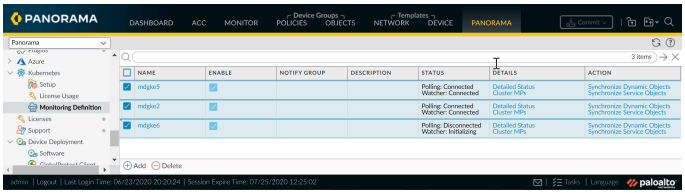
使用以下工作流以卸载 Panorama 上的 Kubernetes 插件，以便您可以将所有令牌成功返回到 Palo Alto Networks 许可服务器，然后清除授权代码。通过此工作流程，您可以确保在其他 Panorama 上使用令牌。如果您已在高可用性配置中部署 Panorama 管理服务器，则必须在活动-主要 Panorama 上完成相关步骤，然后移动到被动-主要 Panorama 对等。

STEP 1 | 如果已在高可用性配置中部署，请登录到活动-主要 Panorama 对等。

1. 从插件中删除所有集群配置。

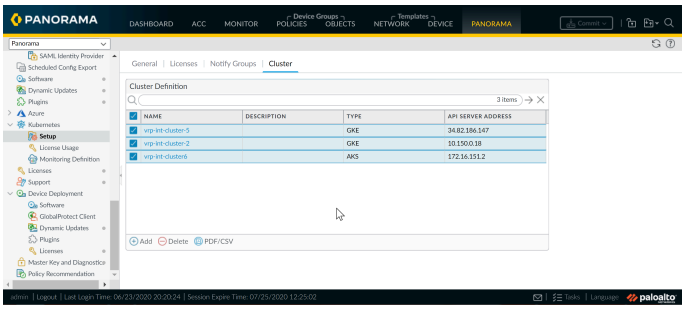
   1. 删除监控定义。

      选择 **Plugins**（插件） > **Kubernetes** > **Monitoring Definition**（监控定义），选择监控定义，然后单击 **Delete**（删除）。

   2. 删除 Kubernetes 集群定义。

      选择 **Plugins**（插件） > **Kubernetes** > **Set up**（设置） > **Cluster**（集群），选择集群定义，然后单击 **Delete**（删除）。

2. 在 Panorama 上提交更改。

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

3. 确认所使用的令牌计数为零。

   要确认已返回所有令牌，请返回许可服务器。

4. 执行明确的授权代码，并确保许可证列授权代码为“无”。

5. 删除配置并提交更改。

   1. 选择 **Plugins**（插件）并找到所安装的 Kubernetes 插件版本，然后 **Remove Config**（删除配置）。

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

6. 卸载 Kubernetes 插件。

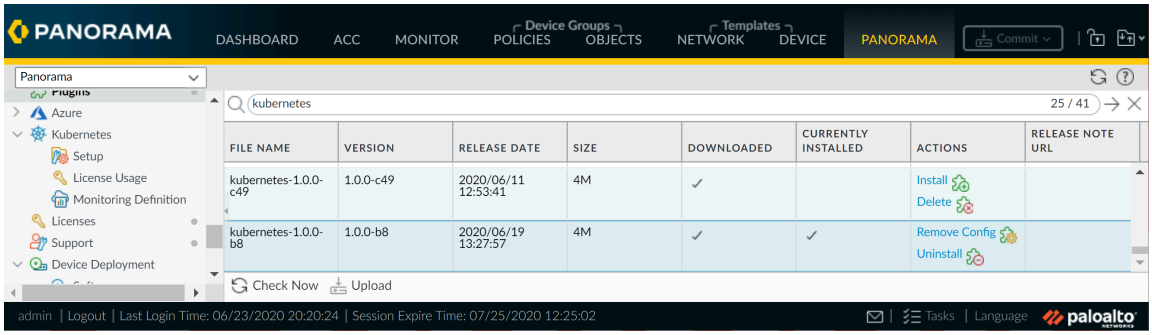
7. 挂起主动 Panorama 对端设备。

   选择 **Panorama** > **High Availability**（高可用性），然后单击 Operational Commands（操作命令）部分中的 **Suspend local Panorama**（挂起本地 **Panorama**）链接。

STEP 2 | 登录到其他 Panorama 对等。

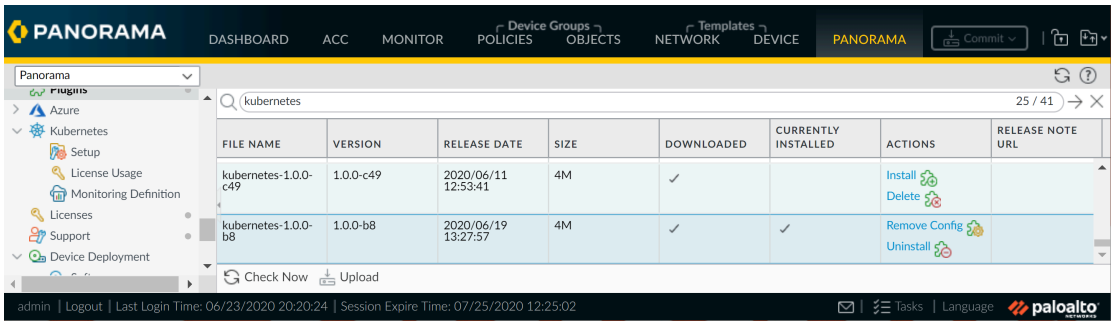
现在，该对等为活动-次要对等。

1. 选择 **Plugins**（插件）并找到所安装的 Kubernetes 插件版本，然后 **Remove Config**（删除配置）。



2. 卸载插件。

   1. 选择 **Plugins**（插件）并找到安装的 Kubernetes 插件版本，然后单击 **Uninstall**（卸载）。



   2. 确认已成功卸载。

在 Panorama 上清除 CN 系列防火墙的授权代码

仅当在清除授权代码之前删除插件配置并提交更改后才使用下面列出的解决方法。您可以使用此解决方法将令牌释放回到许可服务器，以便可以在其他 Panorama 设备上使用。

STEP 1 | 1.添加新插件用户并提交更改。

1. 选择 **Panorama** > **Administrators**（管理员）。
2. **Add**（添加）一个名为 **__kubernetes** 的新用户。
3. **Commit**（提交） > **Commit to Panorama**（提交到 **Panorama**）。

STEP 2 | 在 Panorama 上清除授权代码。

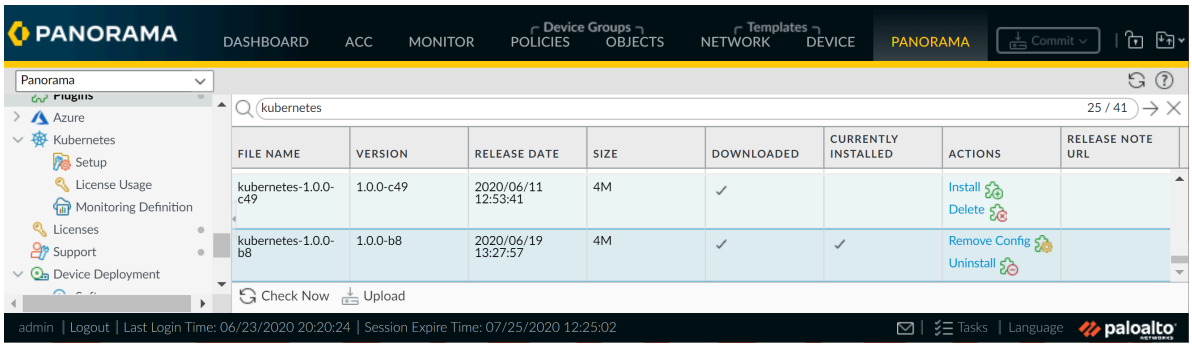
1. 选择 **Panorama** > **Plugins**（插件） > **Kubernetes** > **Setup**（设置） > **Licenses**（许可证）。
2. 选择 **Activate/update using authorization code**（使用授权代码激活/更新），然后 **Clear Auth Code**（清除授权代码）。
3. 确认许可证列将显示授权代码为 **None**（无）。

STEP 3 | 删除在步骤 1 中创建的插件用户 **__kubernetes**。

STEP 4 | 提交更改。

STEP 5 | 卸载插件。

1. 选择 **Plugins**（插件）并找到安装的 Kubernetes 插件版本，然后单击 **Uninstall**（卸载）。



2. 确认已成功卸载。

CN 系列不支持的功能

除非下文另有说明，否则 PAN-OS 支持的以下功能不适用于 CN 系列：

功能	DaemonSet	K8s 服务	CNF 模式	HSF 模式
身份验证	否	否	否	否
将日志转发至 Cortex Data Lake	否	否	否	否
企业 DLP	否	否	否	否
非 vWire 接口	否	否	是	是
IoT Security	否	否	否	否
IPv6	是	否	是	否
NAT	否	否	是	否
基于策略的转发	否	否	是	否
QoS	否	否	否	否
SD-WAN	否	否	否	否
User-ID	否	否	是	否
WildFire Inline ML	否	否	否	否
SaaS 内联	否	否	否	否
IPSec	否	否	否	否
隧道内容检测	否	否	否	否

CN 系列防火墙的高可用性和 DPDK 支持

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

高可用性 (HA) 是一种配置，在该配置中，两个防火墙结合成组，且其配置保持同步，从而防止网络上出现单点故障。防火墙对等端之间的检测信号连接可以确保当某个对等端关闭时提供无缝故障转移。在由两台设备组成的集群中设置防火墙可以提供冗余，并且可以确保业务连续性。

本章包括以下部分：

CN 系列防火墙即 Kubernetes CNF 的高可用性支持

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

高可用性 (HA) 是一种配置，在该配置中，两个防火墙结合成组，且其配置保持同步，从而防止网络上出现单点故障。防火墙对等端之间的检测信号连接可以确保当某个对等端关闭时提供无缝故障转移。在由两台设备组成的集群中设置防火墙可以提供冗余，并且可以确保业务连续性。

现在，您可以在 HA 中部署 CN-series-as-a-kubernetes-CNF。该部署模式仅支持具有会话和配置同步功能的主动/被动 HA。

在 HA 中部署 CN-Series-as-a-Kubernetes CNF 时，主动节点和被动节点将分别有两个 PAN-CN-MGMT-CONFIGMAP、PAN-CN-MGMT 和 PAN-CN-NGFW YAML 文件。

要成功将 CN 系列防火墙部署为 HA 中具有第 3 层支持的 Kubernetes CNF，请执行以下操作：

- 在 HA 中，每个 Kubernetes 节点至少要有三个接口：管理（默认）、HA2 接口和数据接口。
- 对于 L3 模式下的 CN 系列防火墙，至少要有两个接口：管理（默认）和数据接口。

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		

- 修改新的网络附件定义 YAML 文件，进行以下更改：
  - 确保以下 YAML 文件中的 PAN_HA_SUPPORT 参数值为 **true**：

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

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

  - 从运行以下命令的虚拟机管理程序接口检索 **pciBusID** 值：

    ```
    ethtool -i interface name
    ```

    将上面检索到的 **PCIbusID** 值添加到以下网络定义文件中：

    ```
    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
    ```

  - 从 AWS 控制台上的对应节点实例获取 HA2 接口的静态 IP 地址，并将其添加到 `net-attach-def-ha2-0.yaml` 和 `net-attach-def-ha2-1.yaml` 文件的 `address` 参数中。

  如果您使用 **Advanced Routing**（高级路由），请考虑以 CNF 模式部署的 CN 系列防火墙仅在 EKS 和本地环境中受支持。如果您使用带有 Kubernetes 3.0.0 插件的 **Advanced Routing**（高级模式），则必须在模板堆栈上手动配置它；在文件 `pan-cn-mgmt-console.yaml` 中，设置标志 `PAN_ADVANCED_ROUTING:”true”`。

AWS 上 VM 系列防火墙的高可用性

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

现在，您可以在 HA 中部署 CN-Series-as-a-Kubernetes-CNF。该部署模式仅支持具有会话和配置同步功能的主动/被动 HA。

AWS 环境不支持在具有 *IPV6* 的 *HA* 中部署 *CN* 系列即 *Kubernetes CNF*。

为确保冗余，您可采用主动/被动高可用性 (HA) 配置方式在 AWS 中部署 CN 系列防火墙。主动对等将与相同配置的被动对等持续同步其配置和会话信息。上述两个设备之间的脉动连接可确保在主动设备关闭时实现故障转移。您可以通过辅助 IP 移动在 HA 的 AWS EKS 上部署 CN 系列防火墙。

为确保面向 Internet 的应用程序的所有流量都通过防火墙，您可以配置 AWS 入口路由。AWS 入口路由功能可让您将路由表与 AWS Internet 网关相关联，并添加路由规则以通过 CN 系列防火墙重定向应用程序流量。该重定向可确保所有 Internet 流量都通过防火墙，而无需重新配置应用程序端点。

二次移动

当主动对等出现故障时，被动对等会检测到此故障并变成主动对等。此外，还会触发对 AWS 基础架构的 API 调用，以将配置的辅助 IP 地址从出现故障的对等的数据平面接口移动到自身。另外，AWS 更新路由表以确保将流量定向到主动防火墙实例。这两个操作可确保故障转移后恢复入站和出站流量会话。该选项允许您利用 DPDK 来提高 CN 系列防火墙实例的性能。

HA 的 IAM 角色

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

AWS 要求所有 API 请求必须使用由其发布的凭证进行加密签名。要为部署为 HA 对的 CN 系列防火墙启用 API 权限，必须创建一个策略，然后将该策略附加到 AWS 身份识别及访问管理 (IAM) 服务中的角色。相关角色必须在启动时附加到 CN 系列防火墙。该策略授予 IAM 角色权限，使其能够在触发故障转移时启动将接口或辅助 IP 地址从主动对等移动到被动对等所需的 API 操作。

有关创建策略的详细说明，请参阅关于创建客户托管策略的 AWS 文档。有关创建 IAM 角色、定义哪些帐户/AWS 服务可承担此角色、定义承担此角色时应用程序可使用哪些 API 操作和资源的详细说明，请参阅关于《Amazon EC2 的 IAM 角色》的 AWS 文档。

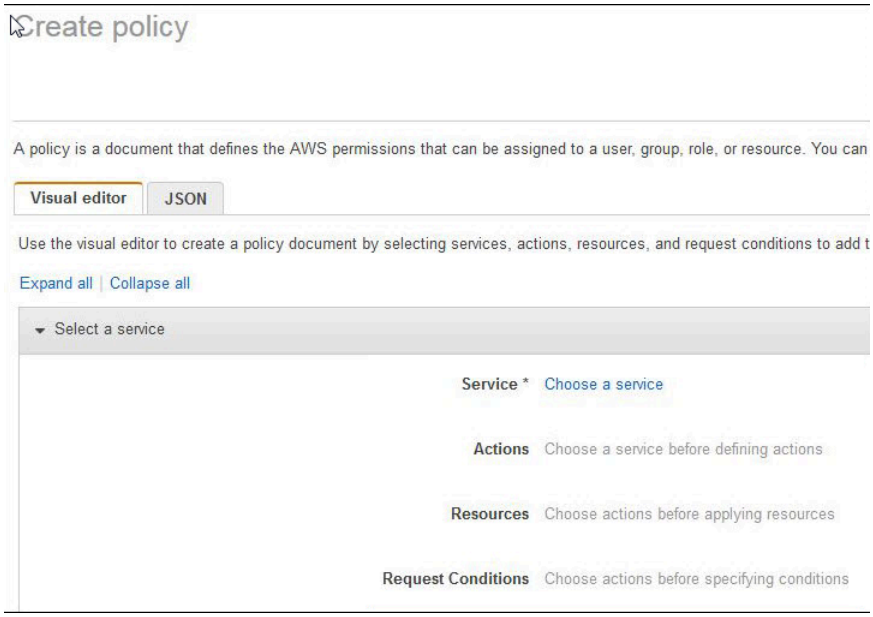
在 AWS 控制台中配置的 IAM 策略必须具备针对以下操作和资源的权限（基本要求）：

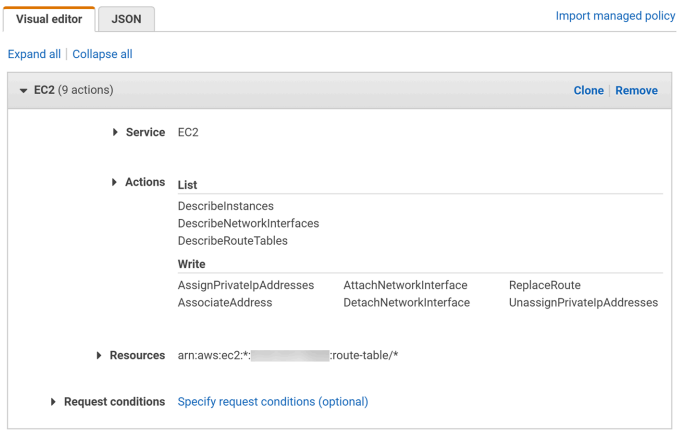
要启用 HA，您需要以下 IAM 操作、权限和资源。

IAM 操作、权限或资源	说明	辅助 IP 地址移动
AttachNetworkInterface	用于允许将 ENI 附加到实例。	✓
DescribeNetworkInterfaces	用于提取 ENI 参数以将接口附加到实例。	✓
DetachNetworkInterface	用于允许从 EC2 实例分离 ENI。	✓
DescribeInstances	用于允许在 VPC 中的 EC2 实例上获取信息。	✓
AssociateAddress	用于允许将与辅助 IP 地址关联的公共 IP 地址从被动接口移动到主动接口。	✓
AssignPrivateIpAddresses	用于允许将辅助 IP 地址和关联的公共 IP 地址分配至被动对等上的接口。	✓
DescribeRouteTables	获取检索与 CN 系列防火墙实例关联的所有路由表的权限。	✓
ReplaceRoute	用于允许更新 AWS 路由表条目。	✓
GetPolicyVersion	用于允许检索 AWS 策略版本信息。	✓
GetPolicy	用于允许检索 AWS 策略信息。	✓
ListAttachedRolePolicies	用于允许检索附加到指定 IAM 角色的所有托管策略列表。	✓
ListRolePolicies	用于允许检索指定 IAM 角色中嵌入的内联策略名称列表。	✓
GetRolePolicy	用于允许检索指定 IAM 角色中嵌入的指定内联策略。	✓

IAM 操作、权限或资源	说明	辅助 IP 地址移动
策略	用于允许访问 IAM 策略 Amazon 资源名称 (ARN)。	✓
角色	用于允许访问 IAM 角色 ARN。	✓
route-table	用于允许访问路由表 Amazon 资源名称 (ARN)，以便在发生故障转移后进行更新。	✓
通配符 (*)	在 ARN 字段中，使用 * 作为通配符。	✓

以下屏幕截图显示了针对辅助 IP HA 的上述 IAM 角色进行的访问管理设置：





辅助 IP 地址移动 HA 所需的最低权限为：{"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 链接

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

HA 对中的设备使用 HA 链接同步数据和维护状态信息。在 AWS 上，CN 系列防火墙使用以下端口：

- 控制链路 — HA1 链路用于交换呼叫消息、检测信号和 HA 状态信息，以及路由的管理平面同步。此链接还用于同步主动或被动设备与其对等端上的配置更改。

  使用管理端口进行 HA1 链接操作。使用 TCP 端口 28769 和 28260 进行明文通信；使用端口 28 进行加密通信 (SSH over TCP)。
- 数据链接 — HA2 链接用于在 HA 对中的设备之间同步会话、转发表、IPSec 安全关联和 ARP 表。HA2 链接上的数据流始终是单向的（HA2 保持活动状态除外）；它从主动设备流动到被动设备。

  必须将 Ethernet1/1 分配为 HA2 链接；这是在 HA 中在 AWS 上部署 CN 系列防火墙所必需的。可以将 HA 数据链接配置为使用 IP（协议号 99）或 UDP（端口 29281）进行传输。

AWS 上的 CN 系列防火墙不支持 HA1 或 HA2 的备份链路。

检测信号轮询和呼叫消息

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

防火墙使用呼叫消息和检测信号来验证对等设备是否有响应、是否可操作。呼叫消息以配置的呼叫间隔从一个对等端发送到另一个对等端，以验证设备的状态。检测信号是通过控制链接对 HA 对等

端进行的 ICMP ping 操作，对等端响应 ping 操作以确定该设备已连接并且有响应。有关触发故障转移的 HA 计时器的详细信息，请参阅 高可用性计时器。（CN 系列防火墙与 PA-5200 系列防火墙的 HA 计时器相同）。

设备优先级和抢先

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

可以为 HA 对中的设备分配设备优先级值，以在故障转移时指示优先选择哪台设备来承担主动角色并管理通信。如果您需要使用 HA 对中的特定设备来主动保护通信安全，则必须在两个防火墙上启用抢先行为，并为每台设备分配一个设备优先级值。具有较低数值，从而具有较高优先级的设备将被指定为主动设备，管理网络上的所有通信。另一台设备处于被动状态，它将同步主动设备的配置和状态信息，以便随时准备在主动设备发生故障时转换为主动状态。

在首次部署期间，较低的数值会变成活动状态。如果首先部署较高的数值并禁用抢先行为，则较高的数值将变成活动状态。

建议不要在 *AWS* 上的 *CN* 系列防火墙中为 *HA* 执行抢先行为。

默认情况下，防火墙上禁用抢先。启用后，抢先行为将允许具有较高优先级（较低数值）的防火墙在从故障中修复后恢复为主动角色。当发生抢先行为时，该事件会记录在系统日志中。

要添加优先级，应确保在 `pan-cn-mgmt-configmap-0.yaml` 和 `pan-cn-mgmt-configmap-1.yaml` 文件中将参数值 `PAN_HA_PRIORITY` 设置为数值。

例如：

```
PAN_HA_PRIORITY:“10”
```

高可用性计时器

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • 使用 Helm 部署 CN 系列的 Helm 3.6 or above version client

高可用性 (HA) 计时器用于检测防火墙故障和触发故障转移。要降低配置高可用性计时器的复杂性，您可以从以下三个配置文件中进行选择：**Recommended**（建议）、**Aggressive**（积极）和 **Advanced**（高级）。对于特定的防火墙平台，这些配置文件会自动填写最佳高可用性计时器值，从而更快地执行高可用性部署。

对于典型的故障转移计时器设置，使用 **Recommended**（建议）的配置文件；对于更快的故障转移计时器设置，使用 **Aggressive**（积极）的配置文件。**Advanced**（高级）配置文件可用于自定义适合您的网络需求的计时器值。

AWS 上 **CN** 系列的 **HA** 计时器	建议/积极配置文件的默认值
提升持有时间	2000/500 ms
呼叫间隔	8000/8000 ms
检测信号间隔	2000/1000 ms
最大翻动数	3/3
抢先持有时间	1/1 min
监控失败持续时间	0/0 ms
额外主设备持续时间	500/500 ms

使用辅助 IP 在 AWS EKS 上配置主动/被动 HA

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama

完成以下过程，以部署新的 CN 系列防火墙作为具有辅助 IP 地址的 HA 对。

STEP 1 | 为 HA 对部署 CN 系列防火墙之前，请确保执行以下操作：

- 在相同的 AWS 可用性区域中部署两个 HA 对等。请参阅HA 的 IAM 角色。
- 部署实例时，创建一个 IAM 角色并将其分配给运行 CN 系列防火墙的工作进程节点。
- 主动和被动防火墙必须至少各有三个接口 — 管理接口、HA2 接口和数据接口。

  默认情况下，管理接口将用作 HA1 接口。
- 在与集群相同的可用区中的 AWS 上创建网络接口。在 `eni` 上添加标记，使其不受 AWS 管理且可由 multus 使用：

  `node.k8s.amazonaws.com/no_manage:True`
- 验证是否已适当定义网络和安全组件。
  - 启用与互联网的通信。默认 VPC 包括 Internet 网关，如果在默认子网中安装 CN 系列防火墙，则该防火墙必须能够访问 Internet。
  - 创建子网。子网是分配给可以在其中启动 EC2 实例的 VPC 的 IP 地址范围的分段。CN 系列防火墙必须属于公共子网，以便将其配置为可访问 Internet。
  - 创建一个包含防火墙数据接口的数据安全组。此外，将安全配置为允许所有流量，因此，安全由防火墙强制执行。在故障转移过程中维护现有会话需要执行此操作。
  - 将路由添加到专用子网的路由表，以确保可以通过 VPC 中的子网和安全组路由流量（如适用）。

在 *EKS* 上部署 *CN* 系列防火墙时，如果 *`http-put-response-hop-limit`* 值设置为默认值 *1*，则 *IMDSv2* 令牌检索将失败。启用 *IMDSv2* 时，您必须确保将跃点限制值设置为 *3* 或更大。

例如：

运行以下命令：

```
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 系列防火墙。

1. 在每个 HA 对等上将 ethernet 1/1 接口配置为 HA2 接口。
   1. 打开 Amazon EC2 控制台。
   2. 选择 Network Interface（网络接口），然后选择您的网络接口。
   3. 选择 **Actions**（操作） > **Manage IP Addresses**（管理 **IP** 地址）。
   4. 将字段留空以允许 AWS 动态分配 IP 地址，或输入 CN 系列防火墙的子网范围内的 IP 地址。这将为 HA2 接口分配一个辅助 IP。
   5. 单击 **Yes**（是）和 **Update**（更新）。
   6. 选择 **Actions**（操作） > **Change Source/Dest.Check**（更改源/目标检查），然后选择 **Disable**（禁用）。
   7. 在第二个（被动）HA 对等上重复此过程。
2. 将辅助 IP 地址添加到第一个（主动）HA 对等上的数据平面接口。
   1. 选择 **Network Interface**（网络接口），然后选择您的网络接口。
   2. 选择 **Actions**（操作） > **Manage IP Addresses**（管理 **IP** 地址） > **IPv4 Addresses**（**IPv4** 地址） > **Assign new IP**（分配新的 **IP** 地址）。
   3. 将字段留空以允许 AWS 动态分配 IP 地址，或输入 CN 系列防火墙的子网范围内的 IP 地址。
   4. 单击 **Yes**（是）和 **Update**（更新）。
3. 将辅助弹性（公共）IP 地址与主动对等的不可信接口相关联。
   1. 选择 **Elastic IPs**（弹性 **IP** 地址），然后选择要关联的弹性 IP 地址。
   2. 选择 **Actions**（操作） > **Associate Elastic IP**（关联弹性 **IP** 地址）。
   3. 在 **Resource Type**（资源类型）下，选择 **Network Interface**（网络接口）。
   4. 选择要与弹性 IP 地址相关联的网络接口。
   5. 单击 **Associate**（关联）。
4. 对于出站流量检查，在子网路由表中添加一个条目，以将下一个跃点设置为防火墙可信接口。
   1. 选择 **VPC** > **Route Tables**（路由表）。
   2. 选择您的子网路由表。
   3. 选择 **Actions**（操作） > **Edit routes**（编辑路由） > **Add route**（添加路由）。
   4. 输入 **Destination**（目标）CIDR 块或 IP 地址。
   5. 对于 **Target**（目标），输入防火墙可信接口的网络接口。
   6. 单击 **Save routes**（保存路由）。
5. 要使用 AWS 入口路由，请创建一个路由表，并将 Internet 网关与其相关联。然后添加一个条目，并将下一个跃点设置为主动防火墙不可信接口。
   1. 选择 **Route Tables**（路由表） > **Create route table**（创建路由表）。

2. （可选）输入路由表的描述性 **Name tag**（名称标签）。
3. 单击 **Create**（创建）。
4. 单击路由表，然后选择 **Actions**（操作） > **Edit edge associations**（编辑边缘关联）。
5. 选择 **Internet gateways**（**Internet** 网关），然后选择 VPC Internet 网关。
6. 单击 **Save**（保存）。
7. 单击路由表，然后选择 **Actions**（操作） > **Edit routes**（编辑路由）。
8. 对于 **Target**（目标），选择 **Network Interface**（网络接口），然后选择主动防火墙的不可信接口。
9. 单击 **Save routes**（保存路由）。

STEP 3 | 启用 HA。

要启用 HA 支持，要确保以下 YAML 文件中的 PAN_HA_SUPPORT 参数值为 true：

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

系统会自动配置对等 HA1 IP 地址。

STEP 4 | 从 AWS 控制台上的对应节点实例获取 HA2 接口的静态 IP 地址，并将其添加到 `net-attach def-ha2-0.yaml` 和 `net-attach-def-ha2-1.yaml` 文件的 address 参数中。

（可选）修改 **HA2 Keep-alive**（**HA2** 保持活动状态）数据包的 **Threshold**（阈值）。在默认情况下，将启用 **HA2 Keep-alive**（**HA2** 保持活动状态）以监控对等之间的 HA2 数据链接。如果发生故障且超过此阈值（默认情况下为 10000 ms），则会执行已定义的操作。在发生 HA2 保持活动状态故障时，一条关键的系统日志消息将随之生成。

您可以在 *HA* 对的两台设备或仅在一台设备上配置 ***HA2 keep-alive***（***HA2*** 保持活动状态）选项。如果您仅在一台设备上启用此选项，则仅此一台设备会发送“保持活动状态”消息。

STEP 5 | 验证防火墙在主动/被动 HA 中配对。

1. 访问两个防火墙上的 **Dashboard**（仪表盘），并查看高可用性小部件。
2. 在主动 HA 对等上，单击 **Sync to peer**（同步到对等）。
3. 确认防火墙已配对并同步。
   - 在被动防火墙上：本地防火墙的状态应显示 **Passive**（被动），并且 **Running Config**（运行配置）应显示为 Synchronized（已同步）。
   - 在主动防火墙上：本地防火墙的状态应显示 **Active**（主动），并且 **Running Config**（运行配置）应显示为 Synchronized（已同步）。
4. 在防火墙命令行界面中，执行以下命令：
   - 验证故障转移就绪情况：

     ```
     show plugins vmw_series aws ha state
     ```
   - 显示辅助 IP 映射：

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


在 CN 系列防火墙上配置 DPDK

在何处可以使用？	需要提供什么？
• CN-Series 部署	• CN-Series 10.2.x or above Container Images • 运行 PAN-OS 10.2.x 或更高版本的 Panorama • Helm 3.6 or above version client

数据平面开发套件 (DPDK) 为数据平面应用中的快速数据包处理提供了一个简单的框架。

DPDK 模式仅在 *CN* 系列防火墙即 *Kubernetes* 容器网络功能 *(CNF)* 上受支持。

DPDK 模式下不支持 *DHCP IPAM*。

系统要求

要运行 DPDK 应用程序，必须在目标计算机上进行以下自定义。

- 内核配置 — 在主机操作系统内核中启用 HUGETLBFS 选项。
- **KNI** 和 **UIO/VFIO** — 在主机操作系统内核中插入 KNI 和 UIO/VFIO。

- 大页面

1. 保留大页面

   - 运行时，在 Pod 开始之前保留大页面。将所需的大页数添加到与特定页面大小 (KB) 对应的 `/sys/kernel/` 目录下的 `nr_hugepages` 文件中。例如，对于对单节点系统，如果需要 2M 页中的 1024 页，则使用以下命令。

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

   - 在启动期间保留大页面。例如，要将 4G 内存的大页面保留为四个 1G 页面，需将以下选项传递给内核。

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

2. 将大页面与 **DPDK** 结合使用 — 为大页面创建装载点，因为 PanOS 10.2 使用 DPDK 辅助进程。

   下面是创建大小为 1 GB 的大页面以供 DPDK 使用的示例命令。

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

3. 使用以下命令启用大页面后，在主机上重新启动 kubelet 服务。

   ```
   sudo systemctl restart kubelet
   ```

4. 检查 `/sys/fs/cgroup/hugetlb/kubepods.slice/hugetlb.2MB.limit_in_bytes` 以确保大小与大页面大小匹配。如果大小与大页面大小不匹配，请使用以下命令更新大小。

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

   在 *Pod* 中，应用程序可以分配和使用预先分配的多个大小的大页面。应用程序使用资源名称 *hugepages-<size>* 通过容器级别资源需求消耗大页面。例如，*hugepages-2Mi* 或 *hugepages-1Gi*。

   与 *CPU* 或内存不同，大页面不支持过度提交。

启用特权模式可访问主机设备空间。要列出网络设备并将其绑定到容器，请将 */sys* 装载到容器，以便 *DPDK* 可以访问目录中的文件。

下面是在 *DPDK* 上启用大页面的代码片段。

```
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** 固定 — 多个 DPDK 进程无法在同一内核上运行，因为它会导致内存池缓存损坏以及其他问题。辅助进程固定到不同的内核。使用 *configmap* 中的 CPU 固定选项来控制辅助进程。
- **Config** 和 **Pod** 更改
  - 启用 `PAN_DATA_MODE: "dpdk"` in `pan-cn-ngfw-configmap-0.yaml` and `pan-cn-ngfw-configmap-1.yaml`。

    DPDK 不是 *CN-Series-as-a-kubernetes-CNF* 的默认模式。
  - 将 `#HUGEPAGE_MEMORY_REQUEST` 参数与 `pan-cn-ngfw-configmap-0.yaml` 和 `pan-cn-ngfw-configmap-1.yaml` 中的大页面内存请求相匹配。

    如果大页面内存不可用，则默认为 *MMAP*。

有关详细信息，请参阅 DPDK 系统要求。

您可以在本地工作进程节点和 AWS EKS 集群上设置 DPDK

- 在本地工作进程节点上设置 DPDK
- 在 AWS EKS 上设置 DPDK

在本地工作进程节点上设置 DPDK

STEP 1 | 安装以下依赖项：

在要设置 DPDK 的工作节点上运行所有命令。

- 对于 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
```

- 对于 Ubuntu 操作系统：

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

STEP 2 | 安装依赖项后：

- 从 https://fast.dpdk.org/rel/ 下载 DPDK tar 文件。请参阅 DPDK 文档，获取编译步骤。

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

- 解压文件。

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

- 编译文件。编译后的文件将位于 `x86_64-native-linuxapp-gcc` 子文件夹中

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

STEP 3 | 在运行时以统计或动态方式插入已编译的内核模块 (modprobe/insmod)。有关更多信息，请参阅内核模块。

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

在 *Ubuntu* 上，如果您看到错误 — *`insmod`*：错误：无法插入模块 *`igb_uio.ko`*，请先插入 *`uio`* 模块。

```
modprobe uio
```

STEP 4 | 在启动过程中，使用特定于分发的方式插入模块。或者，您可以创建一个服务，在每次系统启动时运行 modprobe/insmod 命令。

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

STEP 5 | 激活并挂载大小为 2048K 的 2M 大页面。

您也可以使用步骤 4 中的服务脚本激活大页面。

```
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 | 创建 VM 的快照以供将来使用。

在 AWS EKS 上设置 DPDK

在 AWS EKS 上，每个 Pod 都有一个由 Amazon VPC CNI 插件分配的网络接口。使用 Multus，您可以创建具有多个接口的 Pod。

STEP 1 | 如果没有，请创建一个 AWS 帐户。

STEP 2 | 使用自定义 AMI 创建 EKS 集群。有关更多信息，请参阅创建 Amazon EKS 集群。

STEP 3 | 修改 VPC 和节点设置。有关更多信息，请参阅 AWS EKS 文档。

STEP 4 | (Multus)将多个 ENI 添加到 EKS 节点，并加载 KNI 和 UIO 驱动程序。

- 使用以下标记将多个 ENI 添加到 EKS 节点。

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

  检测到该标记时，Multus CNI 即可使用该接口。有关更多信息，请参阅 AWS 文档。

- 在 AWS CLI 中运行以下命令。

  ```
  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
  ```

- （如果未使用自定义 AMI）在工作进程节点上启用大页面。

  ```
  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
  ```