paloalto TECHDOCS

Modos de implementación del cortafuegos CN-Series

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

Inicio rápido: implementación de cortafuegos CN-Series..... 5

Modos de implementación de cortafuegos CN-Series..... 7

Implementar el cortafuegos CN-Series como servicio de Kubernetes (modo de implementación recomendado)..... 8

Habilitación del escalado automático de pod horizontal en la serie CN-Series..... 13

Implementación del cortafuegos CN-Series como un DaemonSet..... 19

Implementación del cortafuegos CN-Series como una CNF para Kubernetes..... 26

Implementar la CNF L3 para Kubernetes en modo independiente..... 40

Implementación de los cortafuegos CN-Series..... 49

Lista de comprobación de la implementación de la serie CN-Series..... 50

Implementación del cortafuegos CN-Series con (recomendado) y sin el gráfico de Helm..... 52

Prepárese para utilizar los gráficos y las plantillas de Helm..... 52

Implementación del cortafuegos CN-Series usando el gráfico HELM (recomendado)..... 52

Implementación del cortafuegos CN-Series a través de los archivos YAML..... 54

Implementación del cortafuegos CN-Series con plantillas Terraform..... 57

Implementación de una aplicación de ejemplo..... 57

Implementación de un cortafuegos CN-Series con Terraform..... 58

Configuración del complemento de Kubernetes para Panorama..... 59

Implementación del cortafuegos CN-Series con la orquestación de Rancher..... 61

Implementación de clústeres de Rancher..... 61

Configuración del nodo maestro y de trabajo en el clúster de Rancher..... 62

Modificación del archivo YAML de opciones de clúster de Rancher..... 66

Parámetros editables en archivos YAML de implementación de CN-Series..... 68

Protección de 5G con el cortafuegos CN-Series..... 78

Configuración de Panorama para proteger una implementación de Kubernetes..... 83

Asignación de dirección IP a etiqueta de atributos de Kubernetes..... 88

Habilitación de la inspección del tráfico de VLAN etiquetado..... 92

Habilitación IPVLAN..... 94

Desinstalación del complemento de Kubernetes en Panorama..... 95

Borre el código de autorización para los cortafuegos CN-Series en Panorama..... 97

Funciones no compatibles con el modelo CN-Series..... 99

Alta disponibilidad y soporte DPDK para cortafuegos CN-Series..... 101

Soporte de alta disponibilidad para el cortafuegos CN-Series como CNF para Kubernetes..... 102

Alta disponibilidad para el cortafuegos CN-Series en AWS..... 104

Funciones de IAM para HA....104
Enlaces de HA....107
Sondeos de heartbeat y mensajes de saludo....107
Prioridad y preferencia de dispositivos....108
Temporizadores de HA....108
Configuración de HA activa/pasiva en AWS EKS mediante una IP secundaria....109
Configurar DPDK en el cortafuegos CN-Series....114
Configurar DPDK en nodos de trabajo locales....117
Configurar DPDK en AWS EKS....118

Inicio rápido: implementación de cortafuegos CN-Series

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de CN-Series utilizando Helm

Comience con la CN-Seriesimplementación utilizando los siguientes pasos:

1. Inicie sesión en la cuenta CSP y Active créditos.
2. Cree un perfil de implementación.
3. Instale un certificado de dispositivo en el cortafuegos VM-Series
4. Instale el complemento de Kubernetes y configure Panorama para el modelo CN-Series
5. Descargue los archivos de implementación de la CN-Series desde el repositorio de Palo Alto Networks GitHub. Obtenga los archivos de la carpeta Native-k8s para su uso con las implementaciones nativas de Kubernetes locales o en la nube
6. Implemente el CN-Series con o sin el repositorio de Gráficos HELM.

   *Se recomienda implementar el cortafuegos CN-Series utilizando el **gráfico de HELM**.*

7. Configuración de Panorama para proteger una implementación de Kubernetes

Puede elegir implementar su cortafuegos CN-Series en los siguientes modos de implementación:

- Implementar el cortafuegos CN-Series como servicio de Kubernetes (modo de implementación recomendado)- El cortafuegos CN-Series se implementa en un modelo de implementación de clúster. Este modo de implementación utiliza capacidades de escalado automático, mejora la utilización, reduce los costes y aumenta la escala con modelos de implementación basados en kubernetes nativos.
- Implementación del cortafuegos CN-Series como un DaemonSet- El cortafuegos CN-Series se implementa en un modelo de implementación distribuida. Este modo de implementación es más adecuado cuando tiene un número menor de nodos para proteger por entorno.
- Implementación del cortafuegos CN-Series como una CNF para Kubernetes- Este modo de implementación protege las cargas de trabajo tanto de contenedores como de no contenedores. Puede implementar como una implementación de capa 3 independiente.

Modos de implementación de cortafuegos CN-Series

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

Después de revisar los Componentes básicos del CN-Series y la visión general de alto nivel del flujo de trabajo en Protección de las cargas de trabajo de Kubernetes con el cortafuegos CN-Series, puede empezar a implementar los cortafuegos CN-Series para proteger el tráfico entre contenedores dentro del mismo clúster, así como entre contenedores y otros tipos de cargas de trabajo, como máquinas virtuales y servidores sin sistema operativo.

Si está en el entorno OpenShift, consulte y para proteger el tráfico 5G, consulte Protección de 5G con el cortafuegos CN-Series.

Necesita herramientas estándar de Kubernetes, como kubectl o Helm, para implementar y gestionar sus clústeres, aplicaciones y servicios de cortafuegos de Kubernetes. Panorama no está diseñado para ser un orquestador para la implementación y gestión de clústeres de Kubernetes. Los proveedores gestionados de Kubernetes proporcionan las plantillas para la gestión de clústeres. Palo Alto Networks proporciona plantillas compatibles con la comunidad para implementar CN-Series con Helm *y* Terraform.

- Implementar el cortafuegos CN-Series como servicio de Kubernetes (modo de implementación recomendado)
- Implementación del cortafuegos CN-Series como un DaemonSet
- Implementación del cortafuegos CN-Series como una CNF para Kubernetes
- Implementar la CNF L3 para Kubernetes en modo independiente

Antes de pasar de implementar CN-Series como un DaemonSet a CN-Series como servicio o viceversa, debe eliminar y volver a aplicar `plugin-serviceaccount.yaml`.

- *Cuando implementa el CN-Series como un DaemonSet,* `pan-plugin-cluster-mode-secret` *no debe existir.*
- *Cuando implementa el CN-Series como un servicio de Kubernetes,* `pan-plugin-cluster-mode-secret` *debe estar presente.*

Implementar el cortafuegos CN-Series como servicio de Kubernetes (modo de implementación recomendado)

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con gráfico de Helm

Complete el siguiente procedimiento para implementar el cortafuegos CN-Series como un servicio de Kubernetes.

Antes de comenzar, asegúrese de que la versión del archivo YAML de la serie CN-Series sea compatible con la versión de PAN-OS.

- PAN-OS 10.1.2 o posterior requiere YAML 2.0.2
- PAN-OS 10.1.0 y 10.1.1 requieren YAML 2.0.0 o 2.0.1

STEP 1 | Configure su clúster de Kubernetes.

1. Verifique que el clúster tenga los recursos adecuados. Asegúrese de que el clúster tenga los recursos de Requisitos previos del modelo CN-Series compatibles con el cortafuegos:

   ```
   kubectl get nodes
   ```

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

   Vea la información bajo el encabezado Capacidad en la salida del comando para ver la CPU y la memoria disponibles en el nodo indicado.

   La asignación de CPU, memoria y almacenamiento en disco dependerá de sus necesidades. Consulte Rendimiento y ampliación de la CN-Series

   Asegúrese de tener la siguiente información:

   - Recopile la dirección IP del endpoint para configurar el servidor API en Panorama. Panorama utiliza esta dirección IP para conectarse a su clúster de Kubernetes.
   - Recopile el nombre de la pila de plantillas, el nombre del grupo de dispositivos, la dirección IP de Panorama y, opcionalmente, el Nombre del grupo del recopilador de registros de Panorama.
   - Recopile la Clave de autenticación de VM y el valor y el ID del PIN de registro automático.
   - La ubicación del repositorio de imágenes del contenedor en el que descargó las imágenes.

STEP 2 | (opcional) Si configuró un certificado personalizado en el complemento de Kubernetes para Panorama, debe crear el secreto del certificado ejecutando el siguiente comando. No cambie el

nombre del archivo de ca.crt. El volumen de certificados personalizados en pan-cn-mgmt.yaml y pan-cn-ngfw.yaml es opcional.

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

STEP 3 | Edite los archivos YAML para proporcionar los detalles necesarios para implementar los cortafuegos CN-Series.

Debe reemplazar la ruta de la imagen en los archivos YAML para incluir la ruta a su registro privado y proporcionar los parámetros requeridos. Consulte Parámetros editables en archivos YAML de implementación de CN-Series para obtener más detalles.

STEP 4 | (CN-Series para EKS en AWS Outpost únicamente) Actualice la clase de almacenamiento. Para ser compatible con la CN-Series implementada en AWS Outpost, debe utilizar el controlador de almacenamiento aws-ebs-csi-driver, que garantiza que Outpost extrae los volúmenes de Outpost durante la creación dinámica de volúmenes persistentes (PV).

1. Aplicar el siguiente yaml.

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

2. Verifique que el controlador ebs-sc se esté ejecutando.

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

3. Actualice pan-cn-storage-class.yaml para que coincida con el ejemplo siguiente.

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

4. Añada **`storageClassName: ebs-sc`** a pan-cn-mgmt.yaml en las ubicaciones que se muestran a continuación.

   ```
   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 | Si usa el ajuste de escalado automático en su entorno de Kubernetes, consulte Escalado automático de pod horizontal antes de continuar.

STEP 6 | Implemente el servicio CN-NGFW.

1. Verifique que ha creado la cuenta de servicio con pan-cni-serviceaccount.yaml.

   Consulte Creación de cuentas de servicio para la autenticación de clúster.

2. Utilice Kubectl para ejecutar pan-cni-configmap.yaml.

   kubectl apply -f pan-cni-configmap.yaml

3. Utilice kubectl para ejecutar pan-cn-ngfw-svc.yaml.

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

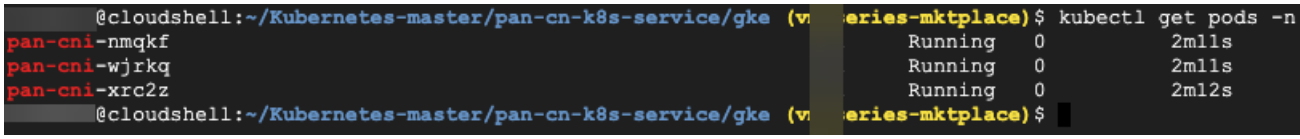
   Este yaml debe implementarse antes de pan-cni.yaml.

4. Utilice Kubectl para ejecutar pan-cni.yaml.

   kubectl apply -f pan-cni.yaml

5. Verifique que haya modificado los archivos pan-cni-configmap y pan-cni YAML.
6. Ejecute el siguiente comando y verifique que su resultado sea similar al siguiente ejemplo.

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



STEP 7 | Implemente el StatefulSet de CN-MGMT.

De forma predeterminada, el plano de gestión se implementa como StatefulSet que proporciona tolerancia a fallos. Es posible conectar hasta 30 pods CN-NGFW de cortafuegos a un StatefulSet de CN-MGMT.

1. (Requerido solo para los volúmenes persistentes aprovisionados estáticamente) Implemente los volúmenes persistentes (PV) para el StatefulSet de CN-MGMT.

   1. Cree los directorios para que coincidan con los nombres de los volúmenes locales definidos en pan-cn-pv-local.yaml.

   Necesita seis (6) directorios en al menos 2 nodos de trabajo. Inicie sesión en cada nodo de trabajo en el que se implementará el StatefulSet de CN-MGMT para crear los directorios. Por ejemplo, para crear directorios llamados / mnt / pan-local1 a / mnt / pan-local6, use el comando:

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

   2. Modifique pan-cn-pv-local.yaml.

   Haga coincidir el nombre de host en `nodeaffinity` y verifique que haya modificado los directorios que creó anteriormente en `spec.local.path`, luego implemente el archivo

para crear una nueva clase de almacenamiento pan-local-storage y volúmenes persistentes locales.

2. Verifique que haya modificado los archivos pan-cn-mgmt-configmap y pan-cn-mgmt YAML.

   Ejemplo de pan-cn-mgmt-configmap de EKS.

```
 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
```

   Ejemplo de pan-cn-mgmt.yaml

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

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

3. Utilice Kubectl para ejecutar archivos pan-cni.yaml.

   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

   Debe ejecutar pan-mgmt-serviceaccount.yaml, solo si no ha completado previamente la Creación de cuentas de servicio para la autenticación de clúster.

4. Verifique que los pods CN-MGMT estén activos.

   Tarda unos 5 y 6 minutos.

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

STEP 8 | Implemente los pods CN-NGFW.

1. Compruebe que haya modificado los archivos YAML como se detalla en PAN-CN-NGFW-CONFIGMAP y PAN-CN-NGFW.

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

2. Utilice Kubectl apply para ejecutar pan-cn-ngfw-configmap.yaml.

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

3. Utilice Kubectl apply para ejecutar pan-cn-ngfw.yaml.

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

4. Compruebe que los pods CN-NGFW se están ejecutando.

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

STEP 9 | Habilitación del escalado automático de pod horizontal en la serie CN-Series.

STEP 10 | Compruebe que puede ver CN-MGMT, CN-NGFW y PAN-CNI en el clúster de Kubernetes.

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

STEP 11 | Anote el yaml de la aplicación o el espacio de nombres para que el tráfico de sus nuevos pods se redirija al cortafuegos.

Debe añadir la siguiente anotación para redirigir el tráfico al CN-NGFW para su inspección:

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

Por ejemplo, para todos los nuevos pods en el espacio de nombres "predeterminado":

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

En algunas plataformas, los pods de aplicaciones pueden iniciarse cuando pan-cni no está activo en la cadena del complemento CNI. Para evitar estos escenarios, debe especificar los volúmenes como se muestra aquí en el pod de aplicación YAML.

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


STEP 12 | (opcional) Cierto tráfico puede evitar el cortafuegos basado en el recurso personalizado *PortInfo*:

1. Aplicar YAML de definición de recurso personalizado de PortInfo

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

2. Utilice pan-cn-ngfw-port-cr.yaml como ejemplo para crear un recurso personalizado PortInfo con protocolos y puertos que desea evitar. Es solo en la dirección de salida desde la perspectiva del pod de aplicación y es compatible con TCP y UDP, hasta 10 puertos individuales (sin rangos de puertos).

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

3. Aplique su YAML de recurso personalizado de PortInfo.

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

4. Anote el pod de aplicación, además de la anotación pan-fw. La anotación debe estar presente en el momento del inicio del pod de aplicación.

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

STEP 13 | Implemente su aplicación en el clúster.

Habilitación del escalado automático de pod horizontal en la serie CN-Series

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con gráfico de Helm

El escalador automático horizontal de pods (HPA) es un recurso de Kubernetes disponible en todos los entornos de nube que amplía automáticamente el número de pods CN-MGMT y CN-NGFW en una implementación basada en métricas supervisadas. HPA utiliza dos métricas estándar en todos los entornos de nube: utilización de CPU y memoria, así como métricas personalizadas específicas para cada entorno de nube. Como tal, cada nube requiere archivos yaml específicos para habilitar HPA en AKS, EKS y GKE.

HPA utiliza un adaptador de métricas específico de la nube para recuperar datos de métricas de un adaptador de supervisión en el entorno de nube, como CloudWatch en EKS, para determinar cuándo ampliar o reducir en función de los umbrales que se definan. Debe modificar los archivos yaml necesarios

para establecer el número mínimo y máximo de réplicas, los umbrales para cada métrica y qué métrica se utiliza en el escalado automático de los cortafuegos.

En PAN OS 10.1, si utiliza el escalado HPA de pods CN-MGMT, puede escalar muchos pods CN-MGMT sin ningún pod DP conectado a ellos. Se recomienda crear un número máximo de réplicas de pods CN-MGMT para evitar el escalado innecesario.

Entorno en la nube	Métricas		Valor medio
AKS, EKS y GKE	CN-MGMT	panloggingrate	Recuento de logs
		pandataplaneslots	Recuento de ranuras del plano de datos
	CN-NFGW	dataplanecpuutilizationpct	Porcentaje de utilización de CPU de CN-NGFW
		dataplanepacketbufferutilization	Porcentaje de utilización del búfer de paquetes de CN-NGFW
		pansessionactive	Número de sesiones activas en el CN-NGFW
		pansessionutilization	Porcentaje de utilización de la sesión
		pansessionsslproxyutilization	Porcentaje de utilización del proxy SSL de sesión
		panthroughput	Rendimiento en kbps
		panpacketrate	Tasa de paquetes en paquetes por segundo (pps)
		panconnectionspersecond	Conexiones por segundo

En el ejemplo a continuación se encuentra el archivo pan-cn-hpa-dp.yaml para EKS. En este ejemplo se utiliza el porcentaje de utilización de CPU del plano de datos para el escalado automático de los pods de CN-NGFW. Cuando esté al 25 por ciento, el clúster se ampliará. Si la utilización de la CPU alcanza el 50 por ciento, el clúster implementará un pod adicional. Si la utilización de la CPU alcanzara el 75 por ciento, el clúster implementaría dos pods adicionales. Esto se determina dividiendo la métrica total por el umbral métrico y, a continuación, implementando suficientes pods para reducir la métrica al umbral de configuración en todos los pods CN-NGFW en el clúster. Sin embargo, el clúster no implementará más pods CN-NGFW que las maxReplicas. Si más de una métrica supera el umbral al mismo tiempo, el clúster implementará el número necesario de pods para abordar la métrica más alta.

De forma predeterminada, el adaptador HPA sondea el adaptador de métricas cada 15 segundos. Si las métricas que ha especificado superan el umbral configurado durante 60 segundos, el clúster implementará

un pod CN-NGFW adicional. A continuación, el clúster espera 300 segundos (cinco minutos) antes de decidir si se requieren o no pods CN-NGFW adicionales. De forma predeterminada, se implementa un pod a la vez. A continuación, el clúster comprueba la métrica (en este caso, la utilización de la CPU) después de 300 segundos. Si la utilización se ha reducido a un nivel en el que ya no se necesita un pod, el clúster eliminará un pod. A continuación, el clúster esperará 60 segundos adicionales antes de decidir si se puede quitar otro pod.

Todos los valores que se muestran a continuación y para cualquier métrica se pueden modificar para que se adapten mejor a su implementación.

```
tipo: 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 | Implemente una instancia de Azure Application Insights en su clúster. Debe proporcionar la clave de instrumentación de Azure Application Insights necesaria y la clave de API de ID de aplicación de Azure Application Insight como secreto K8s.

STEP 2 | Descargue los archivos yaml HPA específicos de AKS desde el repositorio de GitHub de Palo Alto Networks.

STEP 3 | Si su CN-MGMT se implementa en un espacio de nombres personalizado, actualice pan-cn-adapater.yaml con el espacio de nombres personalizado. El espacio de nombres predeterminado es **kube-system**.

STEP 4 | Si aún no lo ha hecho, actualice los parámetros de HPA en el **`pan-cn-mgmt-configmap.yaml`**específico de AKS.

```
  #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 | Edite el archivo **`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 | Añada el nombre hpa que creó anteriormente a los lugares apropiados en **pan-cn-custommetrics.yaml**.

STEP 7 | Modifique **pan-cn-hpa-dp.yaml** y **pan-cn-hpa-mp.yaml**.

1. Introduzca el número mínimo y máximo de réplicas.
2. (Opcional) Cambie los valores de frecuencia de ampliación o reducción para adaptarlos a su implementación. Si no cambia estos valores, se utilizan los valores predeterminados.
3. Copie la siguiente sección para cada métrica que desee utilizar para el escalado.

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

4. Cambie el nombre de la métrica que desea usar y establezca el **averageValue** en el umbral descrito en la tabla anterior. Si no cambia estos valores, se utilizan los valores predeterminados.
5. Guarde los cambios.

STEP 8 | Implemente los archivos yaml de HPA. Los archivos deben implementarse en el orden que se describe a continuación.

1. Utilice Kubectl para ejecutar pan-cn-hpa-secret.yaml

   kubectl apply -f pan-cn-hpa-secret.yaml
2. Utilice Kubectl para ejecutar pan-cn-adapter.yaml

   kubectl apply -f pan-cn-adapter.yaml
3. Utilice Kubectl para ejecutar pan-cn-custommetrics.yaml

   kubectl apply -f pan-cn-custommetrics.yaml
4. Utilice Kubectl para ejecutar pan-cn-hpa-dp.yaml

   kubectl apply -f pan-cn-hpa-dp.yaml
5. Utilice Kubectl para ejecutar pan-cn-hpa-mp.yaml

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

STEP 9 | Compruebe la implementación.

- Utilice kubectl para comprobar el pod del adaptador de métricas personalizadas en el espacio de nombres de métricas personalizadas.

  kubectl get pods -n custom-metrics
- Utilice kubectl para comprobar el recurso HPA.

  kubectl get hpa -n kube-system

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

EKS

STEP 1 | Implemente el adaptador de métricas de Amazon CloudWatch para Kubernetes en su clúster de CN-Series como servicio. Debe permitir que CloudWatch tenga acceso completo a las dos funciones de IAM asociadas a los pods y clústeres de Kubernetes. Para publicar las métricas personalizadas en CloudWatch, la función de los nodos de trabajo debe tener la política gestionada de AWS **CloudWatchAgentServerPolicy** para que la HPA pueda recuperarlas.

STEP 2 | Descargue los archivos yaml de HPA específicos de EKS del Repositorio de GitHub de Palo Alto Networks.

STEP 3 | Si su CN-MGMT se implementa en un espacio de nombres personalizado, actualice pan-cn-adapater.yaml con el espacio de nombres personalizado. El espacio de nombres predeterminado es **kube-system**.

STEP 4 | Modifique **`pan-cn-hpa-dp.yaml`** y **`pan-cn-hpa-mp.yaml`**.

1. Introduzca el número mínimo y máximo de réplicas.
2. (Opcional) Cambie los valores de frecuencia de ampliación o reducción para adaptarlos a su implementación. Si no cambia estos valores, se utilizan los valores predeterminados.
3. Copie la siguiente sección para cada métrica que desee utilizar para el escalado.

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

4. Cambie el nombre de la métrica que desea usar y establezca el **`averageValue`** en el umbral descrito en la tabla anterior. Si no cambia estos valores, se utilizan los valores predeterminados.
5. Guarde los cambios.

STEP 5 | Implemente los archivos yaml de HPA. Los archivos deben implementarse en el orden que se describe a continuación.

1. Utilice Kubectl para ejecutar pan-cn-adapter.yaml

   `kubectl apply -f pan-cn-adapter.yaml`
2. Utilice Kubectl para ejecutar pan-cn-externalmetrics.yaml

   `kubectl apply -f pan-cn-externalmetrics.yaml`
3. Utilice Kubectl para ejecutar pan-cn-hpa-dp.yaml

   `kubectl apply -f pan-cn-hpa-dp.yaml`
4. Utilice Kubectl para ejecutar pan-cn-hpa-mp.yaml

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

STEP 6 | Compruebe la implementación.

- Utilice kubectl para comprobar el pod del adaptador de métricas personalizadas en el espacio de nombres de métricas personalizadas.

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

- Utilice kubectl para comprobar el recurso HPA.

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

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


Implementación del cortafuegos CN-Series como un DaemonSet

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con gráfico de Helm

Complete el siguiente procedimiento para implementar el cortafuegos CN-Series como Daemonset.

Antes de comenzar, asegúrese de que la versión del archivo YAML de la serie CN-Series sea compatible con la versión de PAN-OS.

- PAN-OS 10.1.2 o posterior requiere YAML 2.0.2
- PAN-OS 10.1.0 y 10.1.1 requieren YAML 2.0.0 o 2.0.1

STEP 1 | Configure su clúster de Kubernetes.

1. Verifique que el clúster tenga los recursos adecuados. Asegúrese de que el clúster tenga los recursos de Requisitos previos del modelo CN-Series compatibles con el cortafuegos:

   ```
   kubectl get nodes
   ```

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

   Vea la información bajo el encabezado Capacidad en la salida del comando para ver la CPU y la memoria disponibles en el nodo indicado.

   La asignación de CPU, memoria y almacenamiento en disco dependerá de sus necesidades. Consulte Rendimiento y ampliación de la CN-Series

   Asegúrese de tener la siguiente información:

   - Recopile la dirección IP del endpoint para configurar el servidor API en Panorama. Panorama utiliza esta dirección IP para conectarse a su clúster de Kubernetes.
   - Recopile el nombre de la pila de plantillas, el nombre del grupo de dispositivos, la dirección IP de Panorama y, opcionalmente, el Nombre del grupo del recopilador de registros de Panorama.
   - Recopile el código de autorización y el ID y valor del ID de PIN de registro automático.
   - La ubicación del repositorio de imágenes del contenedor en el que descargó las imágenes.

STEP 2 | (opcional) Si configuró un certificado personalizado en el complemento de Kubernetes para Panorama, debe crear el secreto del certificado ejecutando el siguiente comando. No cambie el

nombre del archivo de ca.crt. El volumen de certificados personalizados en pan-cn-mgmt.yaml y pan-cn-ngfw.yaml es opcional.

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

STEP 3 | Edite los archivos YAML para proporcionar los detalles necesarios para implementar los cortafuegos CN-Series.

Debe reemplazar la ruta de la imagen en los archivos YAML para incluir la ruta a su registro privado y proporcionar los parámetros requeridos. Consulte Parámetros editables en archivos YAML de implementación de CN-Series para obtener más detalles.

STEP 4 | Implementación de CNI DaemonSet.

El contenedor CNI se implementa como un DaemonSet (un pod por nodo) y crea dos interfaces en el pod CN-NGFW para cada aplicación implementada en el nodo. Cuando utilice los comandos kubectl para ejecutar los archivos pan-cni YAML, se convierte en parte de la cadena de servicios en cada nodo.

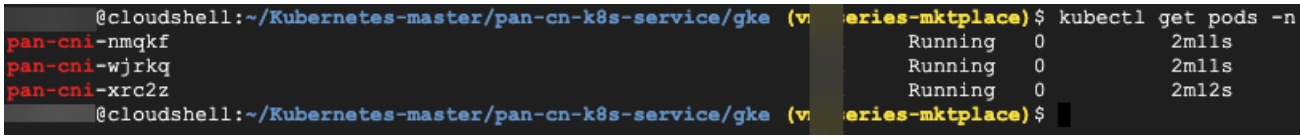
1. Verifique que ha creado la cuenta de servicio con pan-cni-serviceaccount.yaml.

   Consulte Creación de cuentas de servicio para la autenticación de clúster.
2. Utilice Kubectl para ejecutar pan-cni-configmap.yaml.

   ```
   kubectl apply -f pan-cni-configmap.yaml
   ```
3. Utilice Kubectl para ejecutar pan-cni.yaml.

   ```
   kubectl apply -f pan-cni.yaml
   ```
4. Verifique que haya modificado los archivos pan-cni-configmap y pan-cni YAML.
5. Ejecute el siguiente comando y verifique que su resultado sea similar al siguiente ejemplo.

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



STEP 5 | (CN-Series para EKS en AWS Outpost únicamente) Actualice la clase de almacenamiento. Para ser compatible con la CN-Series implementada en AWS Outpost, debe utilizar el controlador de

almacenamiento aws-ebs-csi-driver, que garantiza que Outpost extrae los volúmenes de Outpost durante la creación dinámica de volúmenes persistentes (PV).

1. Aplicar el siguiente yaml.

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

2. Verifique que el controlador ebs-sc se esté ejecutando.

   kubectl -n kube-system get pods

3. Actualice pan-cn-storage-class.yaml para que coincida con el ejemplo siguiente.

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

4. Añada **storageClassName: ebs-sc** a pan-cn-mgmt.yaml en las ubicaciones que se muestran a continuación.

```
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 | Implemente el StatefulSet de CN-MGMT.

De forma predeterminada, el plano de gestión se implementa como StatefulSet que proporciona tolerancia a fallos. Es posible conectar hasta 30 pods CN-NGFW de cortafuegos a un StatefulSet de CN-MGMT.

1. (Requerido solo para los volúmenes persistentes aprovisionados estáticamente) Implemente los volúmenes persistentes (PV) para el StatefulSet de CN-MGMT.

   1. Cree los directorios para que coincidan con los nombres de los volúmenes locales definidos en pan-cn-pv-local.yaml.

   Necesita seis (6) directorios en al menos 2 nodos de trabajo. Inicie sesión en cada nodo de trabajo en el que se implementará el StatefulSet de CN-MGMT para crear los directorios.

Por ejemplo, para crear directorios llamados / mnt / pan-local1 a / mnt / pan-local6, use el comando:

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

2. Modifique pan-cn-pv-local.yaml.

Haga coincidir el nombre de host en `nodeaffinity` y verifique que haya modificado los directorios que creó anteriormente en `spec.local.path`, luego implemente el archivo para crear una nueva clase de almacenamiento pan-local-storage y volúmenes persistentes locales.

2. Verifique que haya modificado los archivos pan-cn-mgmt-configmap y pan-cn-mgmt YAML.

Ejemplo de pan-cn-mgmt-configmap de EKS.

```
 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
```

Ejemplo de pan-cn-mgmt.yaml

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

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

3. Utilice Kubectl para ejecutar archivos pan-cni.yaml.

   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

   Debe ejecutar pan-mgmt-serviceaccount.yaml, solo si no ha completado previamente la Creación de cuentas de servicio para la autenticación de clúster.

4. Verifique que los pods CN-MGMT estén activos.

   Tarda unos 5 y 6 minutos.

   Utilice **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 | Implemente los pods CN-NGFW.

De forma predeterminada, el pod CN-NGFW del plano de datos del cortafuegos se implementa como un DaemonSet. Una instancia del pod CN-NFGW puede proteger el tráfico de hasta 30 pods de aplicaciones en un nodo.

1. Compruebe que haya modificado los archivos YAML como se detalla en PAN-CN-NGFW-CONFIGMAP y PAN-CN-NGFW.

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

2. Utilice Kubectl apply para ejecutar pan-cn-ngfw-configmap.yaml.

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

3. Utilice Kubectl apply para ejecutar pan-cn-ngfw.yaml.

   kubectl apply -f pan-cn-ngfw.yaml

4. Compruebe que todos los pods CN-NGFW se están ejecutando. (uno por nodo en su clúster)

   Este es un resultado de muestra de un clúster local de 4 nodos.

   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 | Compruebe que puede ver CN-MGMT, CN-NGFW y PAN-CNI en el clúster de Kubernetes.

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 | Anote el yaml de la aplicación o el espacio de nombres para que el tráfico de sus nuevos pods se redirija al cortafuegos.

Debe añadir la siguiente anotación para redirigir el tráfico al CN-NGFW para su inspección:

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

Por ejemplo, para todos los nuevos pods en el espacio de nombres "predeterminado":

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

En algunas plataformas, los pods de aplicaciones pueden iniciarse cuando pan-cni no está activo en la cadena del complemento CNI. Para evitar estos escenarios, debe especificar los volúmenes como se muestra aquí en el pod de aplicación YAML.

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

STEP 10 | Implemente su aplicación en el clúster.

Implementación del cortafuegos CN-Series como una CNF para Kubernetes

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con gráfico de Helm

Ahora puede implementar la CN-Series como una función de red de contenedores (CNF) en su entorno de Kubernetes.

El modo de implementación CN-Series como un daemonset y CN-Series como un servicio de kubernetes proporciona una implementación de seguridad automatizada y aprovecha las capacidades de escalado automático de Kubernetes. Sin embargo, estos modos de implementación tienen opciones de inserción limitadas y no admiten la aceleración de E/S. Además, limitan el rendimiento alcanzable para los pods de aplicaciones que requieren inspección y utilizan múltiples interfaces de red.

La implementación de CN-Series con un CNF de kubernetes resuelve estos desafíos para el tráfico que usa Service Function Chaining (SFC) a través de entidades externas, como el enrutamiento nativo del proveedor de la nube, vRouters y conmutadores Top of Rack (TOR). El modo de implementación CN-Series como un CNF de kubernetes no afecta a los pods de aplicaciones.

Complete el siguiente procedimiento para implementar el CN-Series como un CNF de kubernetes.

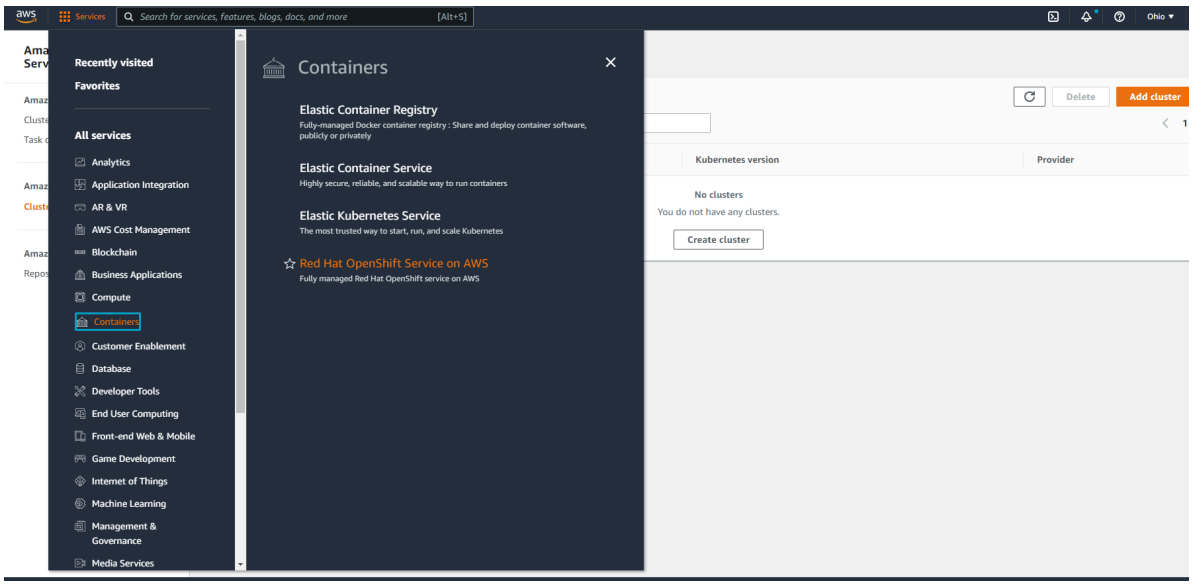
Antes de comenzar, asegúrese de que la versión del archivo YAML de la serie CN-Series sea compatible con la versión de PAN-OS:

PAN-OS 10.2.0 o posterior requiere YAML 3.0.0

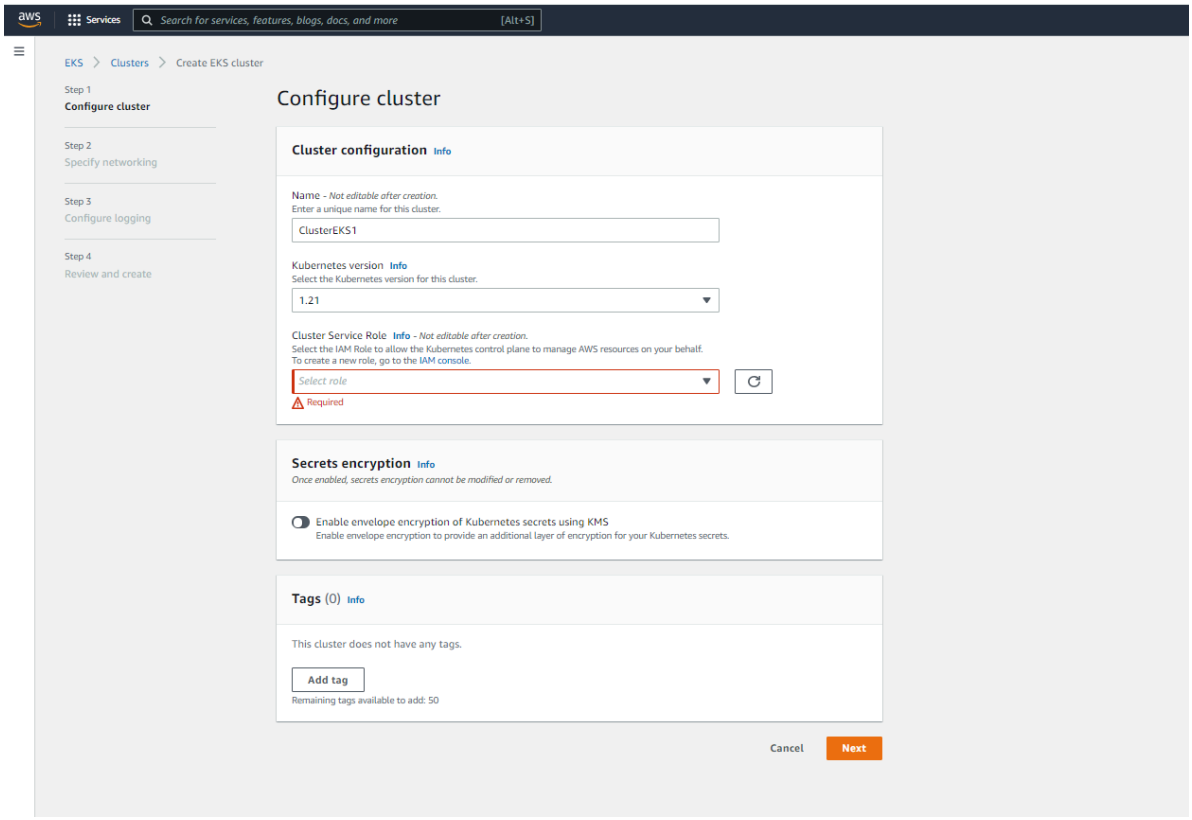
STEP 1 | Configure su clúster de Kubernetes. Para obtener más información, consulte Creación de un clúster de Amazon EKS y Múltiples interfaces de red para pods.

Para crear un clúster en AWS EKS, haga lo siguiente:

1. Haga clic en el menú de navegación de **Servicios**, vaya a **Contenedores**->**Elastic Kubernetes Service**.



2. Haga clic en **Create Cluster (Crear clúster)**.
3. Rellene los detalles necesarios y, a continuación, haga clic en **Create (Crear)**.



1. Verifique que el clúster tenga los recursos adecuados. Asegúrese de que el clúster tenga los recursos de Requisitos previos del modelo CN-Series compatibles con el cortafuegos:

   ```
   kubectl get nodes
   ```

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

   Vea la información bajo el encabezado Capacidad en la salida del comando para ver la CPU y la memoria disponibles en el nodo indicado.

   La asignación de CPU, memoria y almacenamiento en disco dependerá de sus necesidades. Consulte Rendimiento y ampliación de la CN-Series

   Asegúrese de tener la siguiente información:

   - Recopile la dirección IP del endpoint para configurar el servidor API en Panorama. Panorama utiliza esta dirección IP para conectarse a su clúster de Kubernetes.
   - Recopile el nombre de la pila de plantillas, el nombre del grupo de dispositivos, la dirección IP de Panorama y, opcionalmente, el Nombre del grupo del recopilador de registros de Panorama.
   - Recopile el código de autorización y el ID y valor del ID de PIN de registro automático.
   - La ubicación del repositorio de imágenes del contenedor en el que descargó las imágenes.

STEP 2 | (opcional) Si configuró un certificado personalizado en el complemento de Kubernetes para Panorama, debe crear el secreto del certificado ejecutando el siguiente comando. No cambie el nombre del archivo de ca.crt. El volumen de los certificados personalizados en pan-cn-mgmt-0.yaml, pan-cn-mgmt-1.yaml, pan-cn-ngfw-0.yaml y pan-cn-ngfw.yaml-1 es opcional.

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

STEP 3 | Edite los archivos YAML para proporcionar los detalles necesarios para implementar los cortafuegos CN-Series.

Deberá sustituir la ruta de la imagen en los archivos YAML para incluir la ruta a su registro privado y proporcionar los parámetros requeridos. Consulte Parámetros editables en archivos YAML de implementación de CN-Series para obtener más detalles.

CN-Series-as-a-kubernetes-CNF en HA admite solo HA activa/pasiva con sincronización de sesión y configuración.

Cuando implemente CN-Series-as-a-kubernetes-CNF en HA, habrá dos archivos PAN-CN-MGMT-CONFIGMAP, PAN-CN-MGMT y PAN-CN-NGFW YAML, cada uno para nodos activos y pasivos. de la siguiente manera:

- 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

Los siguientes valores predeterminados se definen en los archivos pan-cn-mgmt-configmap-0.yaml y pan-cn-mgmt-configmap-1.yaml.

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

```
metadata:
```

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

```
espacio de nombre: 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
```

```
espacio de nombre: kube-system
```

```
data:
```

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

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

Puede añadir la opción numa para la fijación de CPU. Añada el número de nodo numa único para el parámetro `PAN_NUMA_ENABLED` en los archivos pan-cn-ngfw-configmap-0.yaml y pan-cn-ngfw-configmap-1.yaml.

Para implementar con éxito CN-Series-as-a-kubernetes-CNF en HA con soporte de capa 3:

- En HA, cada nodo de Kubernetes debe tener al menos tres interfaces: Gestión (predeterminada), HA2 e interfaz de datos.
- Para el cortafuegos CN-Series en modo L3, debe haber al menos dos interfaces: Gestión (por defecto) y la interfaz de datos.
- Modifique los nuevos archivos YAML de definición de datos adjuntos de red con los siguientes cambios:
  - En el nodo de trabajo, recupere el valor **`pciBusID`** de la interfaz del hipervisor ejecutando el siguiente comando:

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

Por ejemplo:

```
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)
```

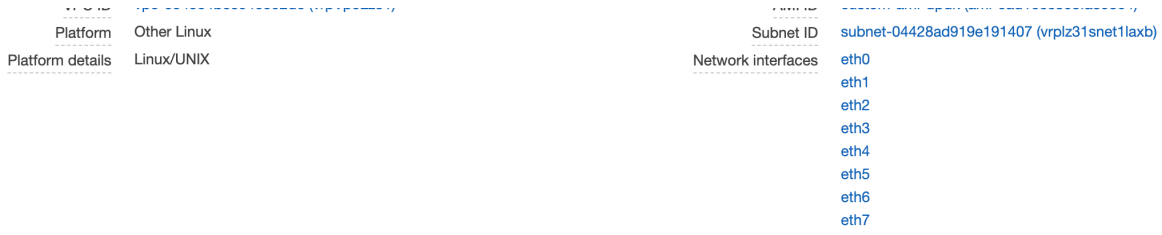
```
00:09.0 Controlador Ethernet: Amazon.com, Inc. Elastic Network
 Adapter (ENA)
```

```
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)
```

la ordenación de PCI es la misma que la ordenación de las interfaces eth como se muestra en la interfaz de usuario EC2 de AWS



Añada el valor de **`pciBusID`** recuperado anteriormente a los siguientes archivos de definición de red:

```
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
```

- Recupere la dirección IP estática de la interfaz HA2 de la instancia del nodo correspondiente en la consola de AWS y añádala al parámetro de `dirección` del archivo `net-attach-def-ha2-0.yaml` y `net-attach-def-ha2-1.yaml`.

STEP 4 | Implemente el StatefulSet de CN-MGMT.

De forma predeterminada, el plano de gestión se implementa como StatefulSet que proporciona tolerancia a fallos. Solo se puede conectar un pod del cortafuegos CN-NGFW a un StatefulSet de CN-MGMT.

1. (Requerido solo para los volúmenes persistentes aprovisionados estáticamente) Implemente los volúmenes persistentes (PV) para el StatefulSet de CN-MGMT.

   1. Cree los directorios para que coincidan con los nombres de los volúmenes locales definidos en pan-cn-pv-local.yaml.

   Necesita seis (6) directorios en al menos 2 nodos de trabajo. Inicie sesión en cada nodo de trabajo en el que se implementará el StatefulSet de CN-MGMT para crear los directorios. Por ejemplo, para crear directorios llamados / mnt / pan-local1 a / mnt / pan-local6, use el comando:

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

   2. Modifique pan-cn-pv-local.yaml.

   Haga coincidir el nombre de host en `nodeaffinity` y verifique que haya modificado los directorios que creó anteriormente en `spec.local.path`, luego implemente el archivo

para crear una nueva clase de almacenamiento pan-local-storage y volúmenes persistentes locales.

2. Verifique que haya modificado los archivos pan-cn-mgmt-configmap y pan-cn-mgmt YAML.
3. Utilice Kubectl para ejecutar archivos pan-cni.yaml.

   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

   Debe ejecutar pan-mgmt-serviceaccount.yaml, solo si no ha completado previamente la Creación de cuentas de servicio para la autenticación de clúster.
4. Verifique que los pods CN-MGMT estén activos.

   Tarda unos 5 y 6 minutos.

   Utilice **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 | Implemente el CN-NGFW en modo k8s-CNF.

1. Verifique que haya modificado los archivos YAML como se detalla en el Paso 3.

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

Debe asegurarse de que multus daemonset esté instalado y que se creen los archivos de definición de datos adjuntos de red. El valor del parámetro para `PAN_SERVICE_NAME` *en los archivos in pan-cn-ngfw-configmap-0.yaml y pan-cn-ngfw-configmap-1.yaml debe coincidir con el valor del parámetro del* `Nombre del servicio` *en los archivos pan-cn-mgmt-0.yaml y pan-cn-mgmt-1.yaml respectivamente.*

> *Para compatibilidad con alta disponibilidad, se recomienda implementar pods de DP en diferentes nodos de trabajo. Puede asegurarse de esto en el campo* ***nodeSelector de yaml*** *o activando antiafinidad del pod.*

Para habilitar la compatibilidad con HA, debe asegurarse de que el valor del parámetro `PAN_HA_SUPPORT` *sea* ***`verdadero`*** *en los siguientes archivos YAML:*

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

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

Para las interfaces de datos de los pods de DP, los CNI y los recursos de la interfaz deben añadirse a los archivos DP YAML según sea necesario. Por ejemplo:

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

Para habilitar la compatibilidad con DPDK, debe asegurarse de que el valor del parámetro `PAN_DATA_MODE` *sea* ***`dpdk`*** *en los archivos pan-cn-ngfw-configmap-0.yaml y pan-cn-ngfw-configmap-1.yaml.*

Además, el valor del parámetro `HUGEPAGE_MEMORY_REQUEST` *debe coincidir con la solicitud de memoria de hugepage en los archivos pan-cn-ngfw-0.yaml y pan-cn-ngfw-1.yaml.*

Para obtener más información, consulte Configurar DPDK en el cortafuegos CN-Series.

2. Utilice Kubectl apply para ejecutar pan-cn-ngfw-configmap-0.yaml y 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. Utilice Kubectl apply para ejecutar pan-cn-ngfw-0.yaml y pan-cn-ngfw-1.yaml.

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

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

4. Compruebe que los pods CN-NGFW se están ejecutando.

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

STEP 6 | Implemente los pods CN-NGFW. Haga lo siguiente:

1. Verifique que haya modificado los archivos YAML como se detalla en PAN-CN-NGFW-CONFIGMAP-0, PAN-CN-NGFW-CONFIGMAP-1, PAN-CN-NGFW-0 y PAN-CN-NGFW-1.

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

2. Utilice Kubectl apply para ejecutar pan-cn-ngfw-configmap.yaml.

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

3. Utilice Kubectl apply para ejecutar pan-cn-ngfw.yaml.

   kubectl apply -f pan-cn-ngfw.yaml

4. Compruebe que los pods CN-NGFW se están ejecutando.

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

STEP 7 | Verifique que pueda ver CN-MGMT y CN-NGFW en el clúster de Kubernetes. Ejecute el siguiente comando:

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


Implementar la CNF L3 para Kubernetes en modo independiente

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con gráfico de Helm

Puede implementar el cortafuegos CN-Series como una función de red de contenedores (CNF) en modo independiente L3 en su entorno de Kubernetes.

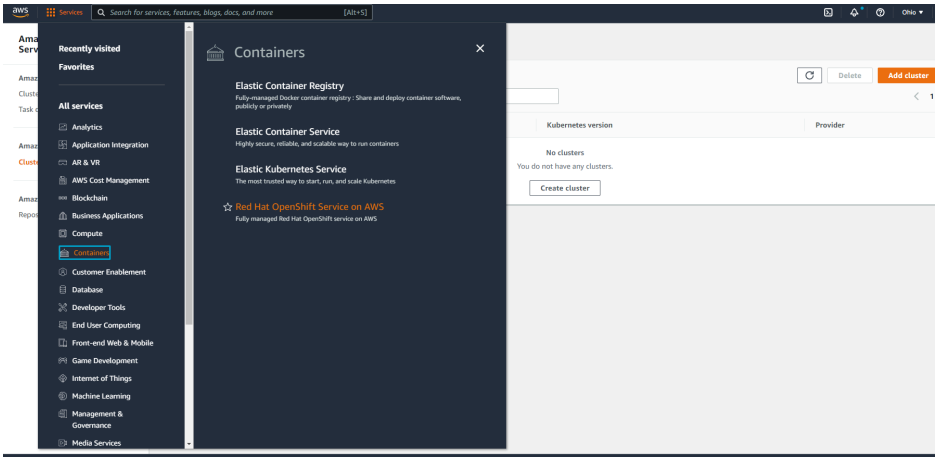
La cortafuegos CN-Series ahora admite el tráfico a través de un vRouter, donde las rutas estáticas se configuran para redirigir el tráfico a las interfaces del plano de datos del cortafuegos. Para la dirección inversa, el tráfico se redirige al mismo cortafuegos mediante el enrutamiento basado en políticas (PBR) L3 con direcciones IP IPv4. Las direcciones IP de las interfaces en el entorno K8 normalmente se programan a través de la CNI mediante DHCP.

Para implementar Kubernetes CNF en modo L3 independiente:

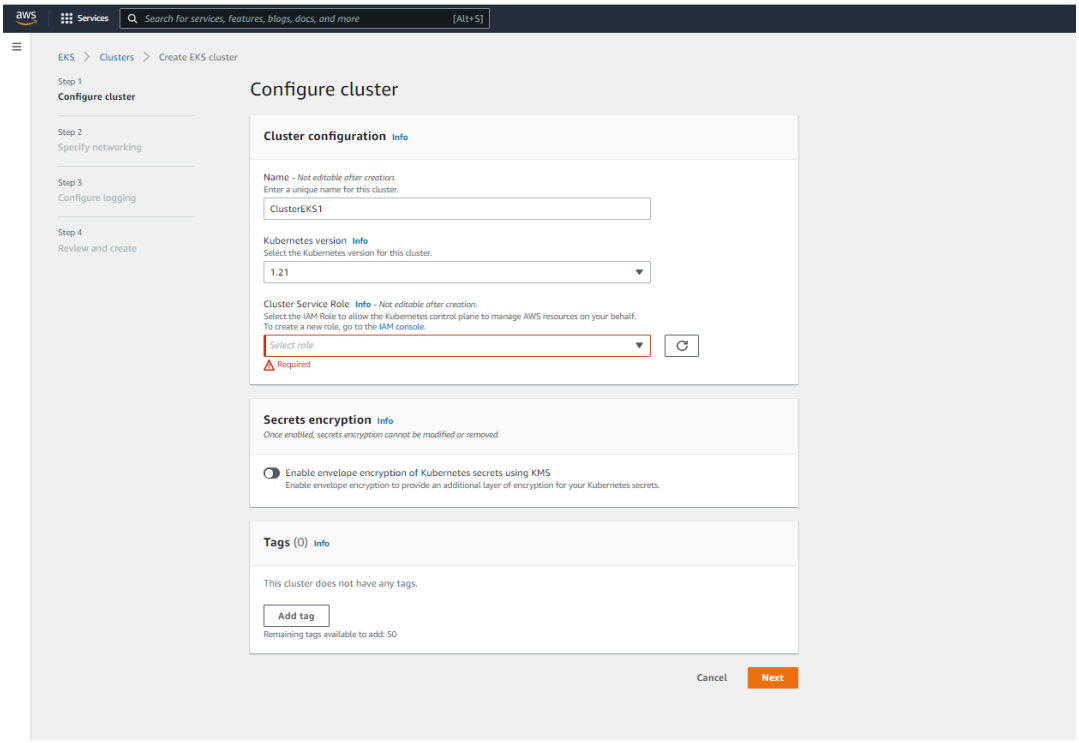
STEP 1 | Configure su clúster de Kubernetes.

Para crear un clúster en AWS EKS, haga lo siguiente:

1. Haga clic en el menú de navegación de **Servicios**, vaya a **Contenedores->Elastic Kubernetes Service**.



2. Haga clic en **Create Cluster (Crear clúster)**.
3. Rellene los detalles necesarios y, a continuación, haga clic en **Create (Crear)**.



1. Verifique que el clúster tenga los recursos adecuados. Asegúrese de que el clúster tenga los recursos de Requisitos previos del modelo CN-Series compatibles con el cortafuegos:

```
kubectl get nodes
```

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


Vea la información bajo el encabezado Capacidad en la salida del comando para ver la CPU y la memoria disponibles en el nodo indicado.

La asignación de CPU, memoria y almacenamiento en disco dependerá de sus necesidades. Consulte Rendimiento y ampliación de la CN-Series.

Asegúrese de tener la siguiente información:

- Recopile la dirección IP del endpoint para configurar el servidor API en Panorama. Panorama utiliza esta dirección IP para conectarse a su clúster de Kubernetes.
- Recopile el nombre de la pila de plantillas, el nombre del grupo de dispositivos, la dirección IP de Panorama y, opcionalmente, el Nombre del grupo del recopilador de registros de Panorama.
- Recopile el código de autorización y el ID y valor del ID de PIN de registro automático.
- La ubicación del repositorio de imágenes del contenedor en el que descargó las imágenes.

STEP 2 | Crear el secreto de certificado. (opcional) Si ha configurado un certificado personalizado en el complemento de Kubernetes para Panorama, debe crear el secreto del certificado ejecutando el siguiente comando. No cambie el nombre del archivo de ca.crt. El volumen de los certificados personalizados en pan-cn-mgmt-0.yaml y pan-cn-ngfw-0.yaml es opcional.

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

STEP 3 | Edite los archivos YAML para proporcionar los detalles necesarios para implementar los cortafuegos CN-Series.

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

Deberá sustituir la ruta de la imagen en los archivos YAML para incluir la ruta a su registro privado y proporcionar los parámetros requeridos. Consulte Parámetros editables en archivos YAML de implementación de CN-Series para obtener más detalles.

Los siguientes valores predeterminados se definen en el archivo pan-cn-mgmt-configmap-0.yaml.

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
```

Puede añadir la opción numa para la fijación de CPU. Añada el número de nodo numa único para el parámetro `PAN_NUMA_ENABLED` en el archivo pan-cn-ngfw-configmap-0.yaml.

Para implementar correctamente CN-Series-as-a-kubernetes-CNF con soporte de capa 3:

- Cada nodo de Kubernetes deberá tener al menos tres interfaces: Gestión (predeterminada), enlace HA2 e interfaz de datos.
- Para el cortafuegos CN-Series en modo L3, debe haber al menos dos interfaces: Interfaz de gestión (predeterminada) e interfaz de datos.
- Modifique los nuevos archivos YAML de definición de datos adjuntos de red con los siguientes cambios:
  - En el nodo de trabajo, recupere el valor **`pciBusID`** de la interfaz del hipervisor ejecutando el siguiente comando:

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

Por ejemplo:

```
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)
```

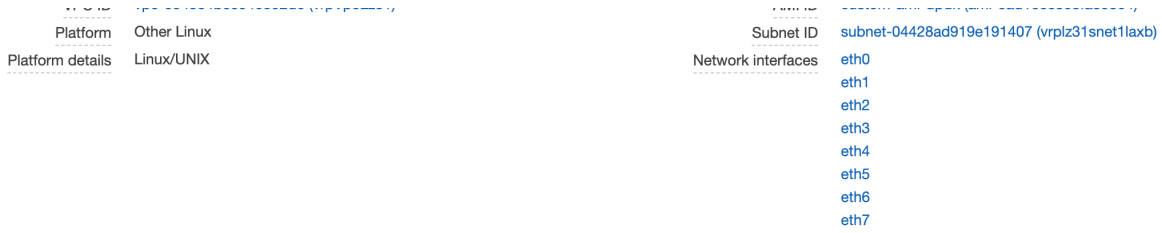
```
00:09.0 Controlador Ethernet: Amazon.com, Inc. Elastic Network
 Adapter (ENA)
```

```
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)
```

la ordenación de PCI es la misma que la ordenación de las interfaces eth como se muestra en la interfaz de usuario EC2 de AWS



Añada el valor de **pciBusID** recuperado anteriormente a los siguientes archivos de definición de red:

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

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

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

STEP 4 | Implemente el StatefulSet de CN-MGMT.

De forma predeterminada, el plano de gestión se implementa como StatefulSet que proporciona tolerancia a fallos. Solo se puede conectar un pod del cortafuegos CN-NGFW a un StatefulSet de CN-MGMT.

1. (Requerido solo para los volúmenes persistentes aprovisionados estáticamente) Implemente los volúmenes persistentes (PV) para el StatefulSet de CN-MGMT.
   1. Cree los directorios para que coincidan con los nombres de los volúmenes locales definidos en pan-cn-pv-local.yaml.

      Necesita seis (6) directorios en al menos 2 nodos de trabajo. Inicie sesión en cada nodo de trabajo en el que se implementará el StatefulSet de CN-MGMT para crear los directorios. Por ejemplo, para crear directorios llamados / mnt / pan-local1 a / mnt / pan-local6, use el comando:

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

   2. Modifique pan-cn-pv-local.yaml.

      Haga coincidir el nombre de host en `nodeaffinity` y verifique que haya modificado los directorios que creó anteriormente en `spec.local.path`, luego implemente el archivo para crear una nueva clase de almacenamiento pan-local-storage y volúmenes persistentes locales.
2. Verifique que haya modificado los archivos pan-cn-mgmt-configmap y pan-cn-mgmt YAML.
3. Utilice Kubectl para ejecutar archivos pan-cni.yaml.

   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

   Debe ejecutar pan-mgmt-serviceaccount.yaml, solo si no ha completado previamente la Creación de cuentas de servicio para la autenticación de clúster.
4. Verifique que los pods CN-MGMT estén activos.

   Tarda unos 5 y 6 minutos.

   Utilice **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 | Implemente el CN-NGFW en modo k8s-CNF.

1. Verifique que haya modificado los archivos YAML como se detalla en el Paso 3.

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

   Debe asegurarse de que multus daemonset esté instalado y que se creen los archivos de definición de datos adjuntos de red. El valor del parámetro para `PAN_SERVICE_NAME` *en el archivo pan-cn-ngfw-configmap-0.yaml debe coincidir con el valor del parámetro* `Nombre del servicio` *en el archivo pan-cn-mgmt-0.yaml.*

   Para las interfaces de datos para los pods CN-NFGW, los CNI y los recursos de interfaz se deben añadir a los archivos YAML de CN-NFGW según sea necesario. Por ejemplo:

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

   Para habilitar la compatibilidad con DPDK, debe asegurarse de que el valor del parámetro `PAN_DATA_MODE` *sea* **`dpdk`** *en el archivo pan-cn-ngfw-configmap-0.yaml.*

   Además, el valor del parámetro `HUGEPAGE_MEMORY_REQUEST` *debe coincidir con la solicitud de memoria de hugepage en el archivo pan-cn-ngfw-0.yaml.*

   Para obtener más información, consulte Configurar DPDK en el cortafuegos CN-Series.

2. Utilice Kubectl apply para ejecutar pan-cn-ngfw-configmap-0.yaml.

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

3. Utilice Kubectl apply para ejecutar pan-cn-ngfw-0.yaml y pan-cn-ngfw-1.yaml.

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

4. Compruebe que los pods CN-NGFW se están ejecutando.

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

STEP 6 | Implemente los pods CN-NGFW. Haga lo siguiente:

1. Compruebe que haya modificado los archivos YAML como se detalla en PAN-CN-NGFW-CONFIGMAP-0 y PAN-CN-NGFW-0.

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

2. Utilice Kubectl apply para ejecutar pan-cn-ngfw-configmap.yaml.

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

3. Utilice Kubectl apply para ejecutar pan-cn-ngfw.yaml.

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

4. Compruebe que los pods CN-NGFW se están ejecutando.

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

STEP 7 | Verifique que pueda ver CN-MGMT y CN-NGFW en el clúster de Kubernetes. Ejecute el siguiente comando:

```
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
```

Implementación de los cortafuegos CN-Series

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

Los cortafuegos CN-Series son fáciles de implementar utilizando la orquestación de Kubernetes para simplificar la integración de la seguridad de la red en los procesos de integración y desarrollo continuos (CI/CD). La gestión continua de los cortafuegos CN-Series está centralizada en la gestión de seguridad de red de Panorama™, la misma consola de gestión de todos los cortafuegos de Palo Alto Networks, lo que proporciona a los equipos de seguridad de red un único panel para gestionar la postura general de seguridad de la red de sus organizaciones.

En este capítulo se abordan las siguientes secciones:

- Lista de comprobación de la implementación de la serie CN-Series
- Implementación del cortafuegos CN-Series con (recomendado) y sin el gráfico de Helm
- Implementación del cortafuegos CN-Series con plantillas Terraform
- Implementación del cortafuegos CN-Series con la orquestación de Rancher
- Funciones no compatibles con el modelo CN-Series

Lista de comprobación de la implementación de la serie CN-Series

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con gráfico de Helm

Para implementar el cortafuegos CN-Series, debe completar las siguientes tareas:

- ☐ Si aún no lo ha hecho, otorgar la licencia al cortafuegos CN-Series, genere su código de autorización y téngalo a mano cuando esté listo para implementar el cortafuegos CN-Series.
- ☐ Revise los Requisitos previos de la CN-Series: antes de comenzar su implementación, asegúrese de comprender los requisitos del sistema necesarios para implementar el cortafuegos CN-Series.
- ☐ Prepare los componentes.
  - Genere una clave de autenticación de VM en Panorama.
  - (Opcional) puede Instalar un certificado de dispositivo en el cortafuegos CN-Series.
  - Creación de cuentas de servicio para la autenticación de clúster
  - Implementar Panorama: debe usar Panorama para configurar, implementar y administrar la implementación de su cortafuegos CN-Series. Para obtener más información sobre cómo implementar y configurar un dispositivo Panorama, consulte Set Up Panorama (Configurar Panorama).
  - Instale el complemento Kubernetes para la CN-Series.
  - Obtenga las imágenes y archivos para la implementación de CN-Series: acceda al Repositorio de Palo Alto Networks para descargar los archivos de Docker y GitHub para obtener los archivos yaml necesarios para implementar el cortafuegos CN-Series en su entorno de Kubernetes.
- ☐ Implementación de los cortafuegos CN-Series.
  - Edite el gráfico HELM para que se ajuste a su implementación: de forma alternativa, también puede editar los archivos yaml y revisar el Parámetros editables en archivos YAML de implementación de CN-Series antes de implementar el cortafuegos CN-Series. Muchos de los parámetros establecidos en los archivos yaml deben modificarse para implementar correctamente el cortafuegos CN-Series.
  - Implementar el cortafuegos CN-Series como servicio de Kubernetes (modo de implementación recomendado).
  - Implementación del cortafuegos CN-Series como un DaemonSet.
  - (Opcional) Si está implementando su cortafuegos CN-Series como un servicio de Kubernetes, puede Habilitación del escalado automático de pod horizontal en la serie CN-Series. El ajuste del escalado

horizontal automático de pods (HPA) permite que la implementación de su cortafuegos CN-Series se amplíe automáticamente de forma dinámica junto con su entorno de Kubernetes.

- Si va a implementar su CN-Series en un entorno OpenShift, consulte Implementación del cortafuegos CN-Series en Openshift.
- Si está protegiendo el tráfico 5G con su cortafuegos CN-Series, consulte Protección de 5G con el cortafuegos CN-Series.

- Configuración de Panorama para proteger una implementación de Kubernetes—después de implementar su cortafuegos CN-Series, use Panorama para configurar las políticas de seguridad que permitan la aplicación del tráfico y envíe esas políticas al cortafuegos.

Implementación del cortafuegos CN-Series con (recomendado) y sin el gráfico de Helm

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

El repositorio de Helm contiene gráficos y plantillas para implementar el cortafuegos en contenedor de la serie CN-Series de Palo Alto Networks utilizando Helm Packet Manager para Kubernetes.

Puede descargar gráficos de Helm para CN-Series de GitHub.

- Prepárese para utilizar los gráficos y las plantillas de Helm
- Implementación del cortafuegos CN-Series usando el gráfico HELM (recomendado)
- Implementación del cortafuegos CN-Series a través de los archivos YAML

Prepárese para utilizar los gráficos y las plantillas de Helm

Instale el software necesario. Estas instrucciones enumeran las versiones mínimas, pero puede instalar una versión posterior de la misma familia a menos que se especifique un límite superior.

STEP 1 | Implemente imágenes de contenedor del cortafuegos CN-Series 10.1.x, 10.2.x, 11.0.x, o 11.1.x.

STEP 2 | Instale una versión de Kubernetes entre 1.16 y 1.25 y cree un clúster de Kubernetes. Para obtener más información sobre la versión compatible con kubernetes para sus entornos, consulte Entornos compatibles con la implementación de CN-Series.

STEP 3 | Implemente Panorama en una ubicación a la que se pueda acceder desde el clúster de Kubernetes y el cortafuegos CN-Series que utiliza para proteger el clúster.

1. Asegúrese de que la versión PAN-OS de Panorama sea 10.xx o posterior.
2. Instale el complemento de Kubernetes para Panorama versión 1.0.xo 2.0.x.

STEP 4 | Instale la versión 3.6.0 o posterior del cliente de Helm.

Continúe a .Implementación del cortafuegos CN-Series usando el gráfico HELM (recomendado)

o Implementación del cortafuegos CN-Series a través de los archivos YAML.

Implementación del cortafuegos CN-Series usando el gráfico HELM (recomendado)

Utilice este procedimiento para clonar el repositorio e implementarlo desde su entorno local.

STEP 1 | Genere la clave de autenticación de VM en Panorama.

STEP 2 | Clone el repositorio de GitHub.

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

STEP 3 | Cambie a un directorio local para el repositorio clonado. Por ejemplo:

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

STEP 4 | Cambie al subdirectorio para su implementación.

- Utilice el directorio `helm_cnv1` para implementar el CN-Series como un conjunto de daemon
- Utilice el directorio `helm_cnv2` para implementarCN-Series como servicio.
- Utilice el directorio `helm_cnv3` para implementar CN-Series como cnf.

STEP 5 | Descargue la cuenta de servicio YAML para el `plugin-serviceaccount.yaml` y aplique el yaml. La cuenta de servicio habilita los permisos que Panorama requiere para autenticarse en el clúster para recuperar las etiquetas de Kubernetes y la información de recursos. Esta cuenta de servicio se denomina `pan-plugin-user` de forma predeterminada. Ejecute el siguiente comando para implementar el archivo `plugin-serviceaccount.yaml`:

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

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

Para ver los secretos asociados con esta cuenta de servicio.

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

Cree el archivo de credenciales, denominado `cred.json` en este ejemplo, que incluye los secretos y guarde este archivo. Es necesario cargar este archivo a Panorama para configurar el complemento Kubernetes para supervisar los clústeres en Instalación del complemento de Kubernetes para cortafuegos CN-series.

*En **Openshift**, debe implementar manualmente el `pan-cni-net-attach-def.yaml` para cada archivo del espacio de nombres Openshift antes de implementar los gráficos Helm.*

STEP 6 | Edite el archivo `values.yaml` para introducir su información de configuración. Los siguientes valores son del subdirectorio `helm_cnv1`.

```
# 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 | Vea los archivos YAML reproducidos.

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

STEP 8 | Realice una comprobación de lint en los gráficos de helm.

```
helm lint helm_cnv1/
```

STEP 9 | Implemente los gráficos HELM.

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

Las reclamaciones de volumen persistente no se eliminan cuando se desinstala un gráfico HELM. Debe asegurarse de que borra estas reclamaciones de antemano para que la instalación de HELM funcione.

Para obtener más información, consulte HELM Classic: Un gestor de paquetes de Kubernetes.

Implementación del cortafuegos CN-Series a través de los archivos YAML

Para implementar sin clonar el repositorio, añada el repositorio a su cliente Helm.

STEP 1 | Genere la clave de autenticación de VM en Panorama.

STEP 2 | Descargue la cuenta de servicio YAML para el `plugin-serviceaccount.yaml` y aplique el yaml. La cuenta de servicio habilita los permisos que Panorama requiere para autenticarse en el clúster para recuperar las etiquetas de Kubernetes y la información de recursos. Esta cuenta de servicio se denomina `pan-plugin-user` de forma predeterminada. Ejecute el siguiente comando para implementar el archivo `plugin-serviceaccount.yaml`:

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

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

Para ver los secretos asociados con esta cuenta de servicio.

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

Cree el archivo de credenciales, denominado `cred.json` en este ejemplo, que incluye los secretos y guarde este archivo. Es necesario cargar este archivo a Panorama para configurar el complemento

Kubernetes para supervisar los clústeres en Instalación del complemento de Kubernetes para cortafuegos CN-series.

*En **Openshift**, debe implementar manualmente el `pan-cni-net-attach-def.yaml` para cada archivo del espacio de nombres Openshift antes de implementar los gráficos Helm.*

STEP 3 | Añada el repositorio CN-Series a su cliente Helm local.

Introduzca este comando en una sola línea:

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

```
Se ha añadido "cn-series" a sus repositorios
```

STEP 4 | Confirme que el repositorio se haya añadido a su cliente de Helm.

```
$ helm search repo serie cn
```

STEP 5 | Seleccione el clúster de Kubernetes.

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


STEP 6 | Implementar usando el repositorio de gráficos de Helm. Edite el siguiente comando para incluir su información de configuración.

```
$ 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"
```

Las reclamaciones de volumen persistente no se eliminan cuando se desinstala un gráfico HELM. Debe asegurarse de que borra estas reclamaciones de antemano para que la instalación de HELM funcione.

Implementación del cortafuegos CN-Series con plantillas Terraform

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Terraform 0.13.0 o superior

El repositorio de implementación de la serie CN-Series contiene planes de Terraform para implementar un clúster de GKE, EKS o AKS. Estos planes garantizan que el tamaño del nodo del clúster y las interfaces de red de contenedores (CNI) sean compatibles con una implementación del cortafuegos CN-Series dentro del clúster. El repositorio también proporciona un plan de implementación del cortafuegos CN-Series y una aplicación de libro de visitas PHP de ejemplo que puede proteger con el cortafuegos.

Este procedimiento tiene el siguiente flujo de trabajo opcional:

- Prepárese para utilizar los gráficos y las plantillas de Helm
- Implementación de una aplicación de ejemplo
- Implementación de un cortafuegos CN-Series con Terraform
- Configuración del complemento de Kubernetes para Panorama

Implementación de una aplicación de ejemplo

El repositorio de GitHub de Palo Alto Networks incluye una la aplicación de muestra compatible con la comunidad con un archivo de manifiesto de Kubernetes denominado `guestbook.yml`.

Este archivo implementa una sencilla aplicación web de libro de visitas PHP que utiliza un back-end de Redis.

STEP 1 | En el directorio cn-series-deploy del repositorio de GitHub de Palo Alto Networks, cambie al directorio sample-application.

```
$ cd sample-application
```

STEP 2 | Implemente la aplicación de libro de visitas.

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

STEP 3 | Compruebe que los pods de la aplicación se han implementado y alcanzan el estado En ejecución y, a continuación, Listo.

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

```
NOMBRE LISTO ESTADO REINICIOS EDAD 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 | Enumere los servicios para determinar la dirección IP pública del front-end web.

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

Ahora puede configurar los grupos de direcciones dinámicas y las reglas de seguridad en Panorama para proteger la aplicación del libro de visitas.

Continúe con la implementación del cortafuegos CN-Series mediante Terraform.

Implementación de un cortafuegos CN-Series con Terraform

Implementación del cortafuegos CN-Series con Terraform.

STEP 1 | Utilice su `cn-series\tfvars` local para crear un archivo denominado `terraform.tfvars`, y añada las siguientes variables y sus valores asociados.

```
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 | Validar el plan Terraform.

```
$ terraform init
```

STEP 3 | Validar el plan Terraform.

```
$ terraform plan
```

STEP 4 | Aplicar el plan Terraform.

```
$ terraform apply
```

STEP 5 | Compruebe que los pods se han implementado y están listos y el estado es En ejecución.

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

```
NOMBRE DEL ESPACIO DE NOMBRES LISTO ESTADO REINICIA LA EDAD ...
 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
```

Está listo para configurar el complemento de Kubernetes para Panorama.

Configuración del complemento de Kubernetes para Panorama

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores

Use el complemento de Kubernetes para Panorama para propagar etiquetas a grupos de dispositivos Panorama.

Puede usar el complemento Kubernetes para completar la integración de Panorama y la API de Kubernetes. El complemento aprende nuevas etiquetas y las propaga a grupos de dispositivos de Panorama. Estas etiquetas pueden incluir etiquetas de Kubernetes, servicios, espacios de nombres y otros metadatos a partir de los cuales se pueden definir criterios de coincidencia de grupos dinámicos de direcciones.

Si el tamaño del archivo de credenciales del clúster es superior a 32 KB, recibirá un mensaje de error al importar el archivo de credenciales al complemento de Kubernetes en Panorama. El mensaje de error muestra el tamaño del archivo como la causa del error.

Si el clúster tiene muchos certificados CA en el paquete `ca.crt`, el complemento Kubernetes solo requiere el certificado CA superior. Debe asegurarse de conservar solo el certificado CA superior y eliminar todos los demás certificados CA y `service.crt` del archivo de credenciales. A continuación, puede utilizar este archivo de credenciales actualizado.

Este procedimiento asume que ha instalado el software de apoyo enumerado en Prepárese para utilizar los gráficos y las plantillas de Helm.

STEP 1 | Recupere las credenciales de la cuenta de servicio pan-plugin-user del maestro o servidor de Kubernetes.

Introduzca cada comando como una línea única:

```
$ 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 | Cree una definición de clúster en el complemento de Kubernetes en Panorama.

Utilice la dirección del maestro de Kubernetes que se muestra en la salida de Terraform y el archivo de credenciales JSON ubicado en `~/Downloads/pan-plugin-user.json`.

Defina las etiquetas que desea importar desde la API de Kubernetes.

STEP 3 | Cree una definición de grupo de notificación en el complemento de Kubernetes en Panorama.

Esta definición se utiliza para propagar las etiquetas obtenidas de la API de Kubernetes a un grupo de dispositivos Panorama.

Realice los siguientes pasos para crear un grupo de notificaciones en el complemento de Kubernetes en Panorama:

1. Seleccione **Panorama** > **Plugins (Complementos)** > **Kubernetes** > **Setup (Configuración)** > **Notify Groups (Grupos de notificación)** y **Add (Añadir)**.

2. Introduzca un **Name (Nombre)** de hasta 31 caracteres para el grupo de notificación.
3. Seleccione **Habilitar el uso compartido de etiquetas internas con grupos de dispositivos** si desea compartir etiquetas internas además de las etiquetas externas (predeterminadas) creadas para el clúster.
4. Seleccione los grupos de dispositivos en los que desea registrar las etiquetas.

5. Haga clic en **Ok (Aceptar)**.

STEP 4 | Cree una definición de supervisión en el complemento de Panorama.

Utilice las definiciones de clúster y grupo de notificación creadas en los pasos anteriores.

STEP 5 | Confirmar en Panorama.

STEP 6 | Para confirmar la conectividad de API y los registros de contenedores MP, vaya a la Definición de supervisión y haga clic en Estado detallado y MPs de clúster.

Ahora está listo para implementar una aplicación y protegerla con el cortafuegos CN-Series.

Implementación del cortafuegos CN-Series con la orquestación de Rancher

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores

Ahora puede implementar el cortafuegos CN-Series como un servicio de Kubernetes mediante la orquestación de Rancher y PAN OS 10.1. Rancher es una plataforma de orquestación de contenedores de código abierto que puede utilizarse para implementar el cortafuegos CN-Series.

Para la implementación del cortafuegos CN-Series con compatibilidad con clústeres de Rancher, la instancia de Panorama debe tener 16 vCPUS, memoria de 32 G con un disco adicional de 2 TB. Panorama se implementará en un modo que facilitará la recopilación de logs de las implementaciones de cortafuegos CN-series.

Al implementar el cortafuegos CN-Series dentro del clúster de Kubernetes de Rancher local, haga lo siguiente:

- Asegúrese de que los componentes necesarios para proteger los clústeres de Kubernetes con el cortafuegos CN-Series estén disponibles.
- Asegúrese de que el clúster de Kubernetes cumple con los requisitos mínimos del sistema. Para obtener más información, consulte Requisitos del sistema CN-Series.
- Realizar el Implementación del cortafuegos CN-Series con la orquestación de Rancher
-
- Modificación del archivo YAML de opciones de clúster de Rancher
- Instalación del complemento de Kubernetes para el cortafuegos CN-Series.
- Autorización de licencia para el cortafuegos CN-Series
- Implementar el cortafuegos CN-Series como servicio de Kubernetes (modo de implementación recomendado) en Rancher

Implementación de clústeres de Rancher

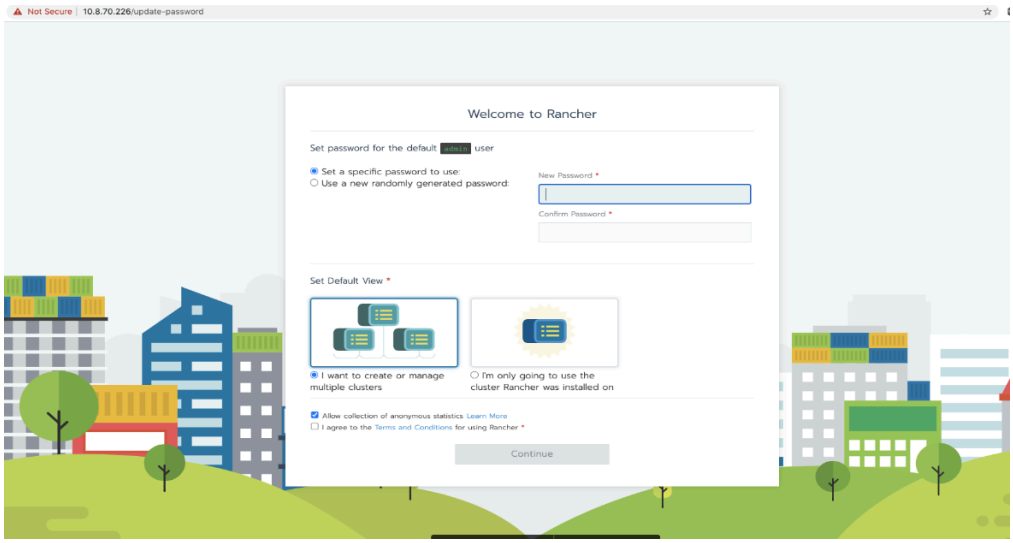
Puede implementar Rancher en los siguientes dos pasos:

1. Prepare un host Linux con distribución Linux compatible y 4 GB de memoria. Instale una versión compatible de Docker en el host.
2. Inicie el servidor.

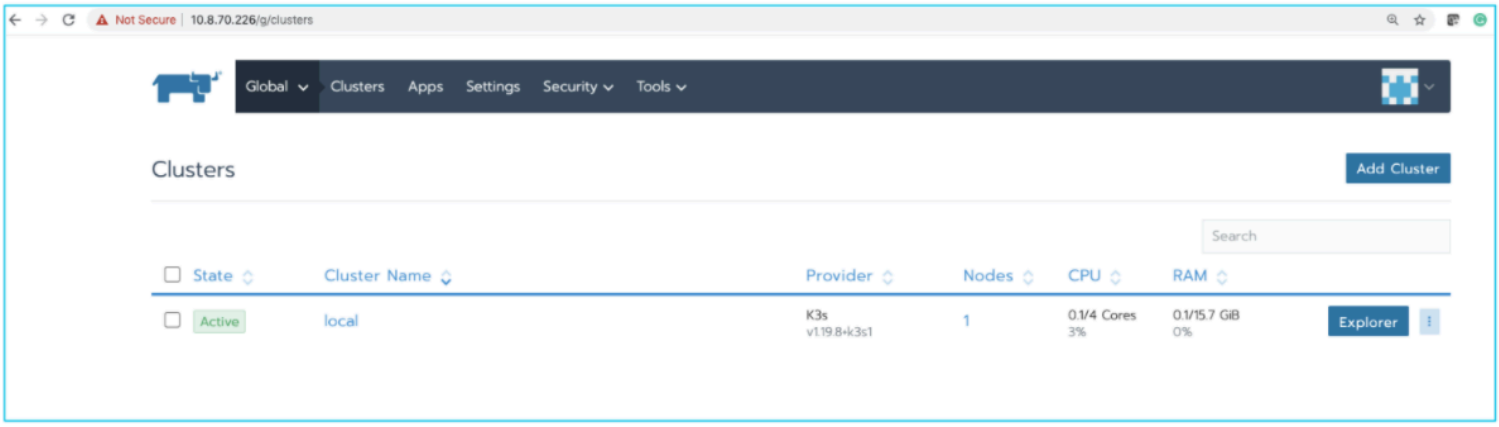
   Para instalar y ejecutar Rancher, ejecute el siguiente comando de Docker en el host:

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


Después de una implementación correcta, puede acceder a la interfaz de usuario del servidor de Rancher y configurar una contraseña para el usuario administrador. Para acceder a la interfaz de usuario del servidor de Rancher, abra un navegador y vaya al nombre de host o la dirección donde se instaló el contenedor. Se le guiará a través de la configuración de su primer clúster.



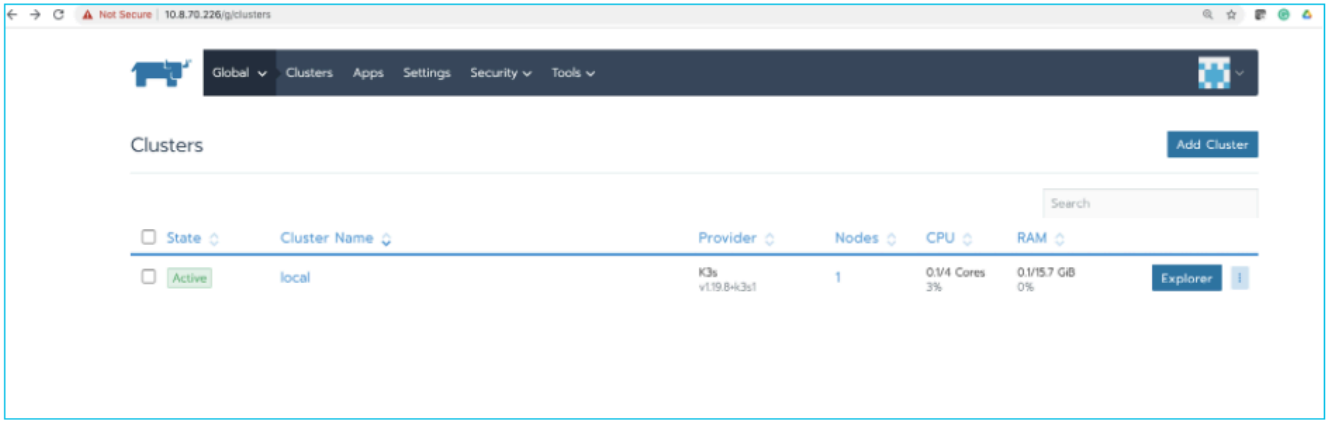
Tras la creación del usuario administrador, se creará un clúster local como se muestra a continuación:



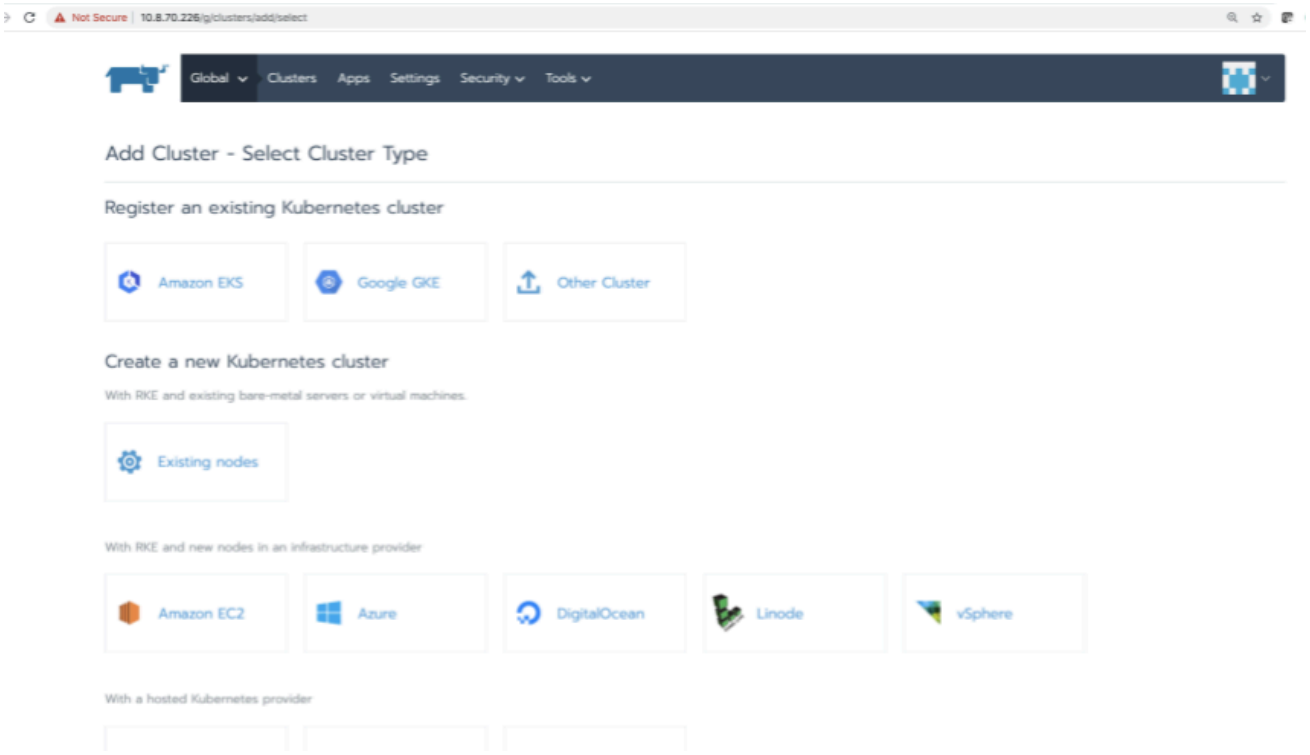
Configuración del nodo maestro y de trabajo en el clúster de Rancher

Después de crear un clúster local en la interfaz de usuario de Rancher, configure un nodo maestro y de trabajo, haga lo siguiente:

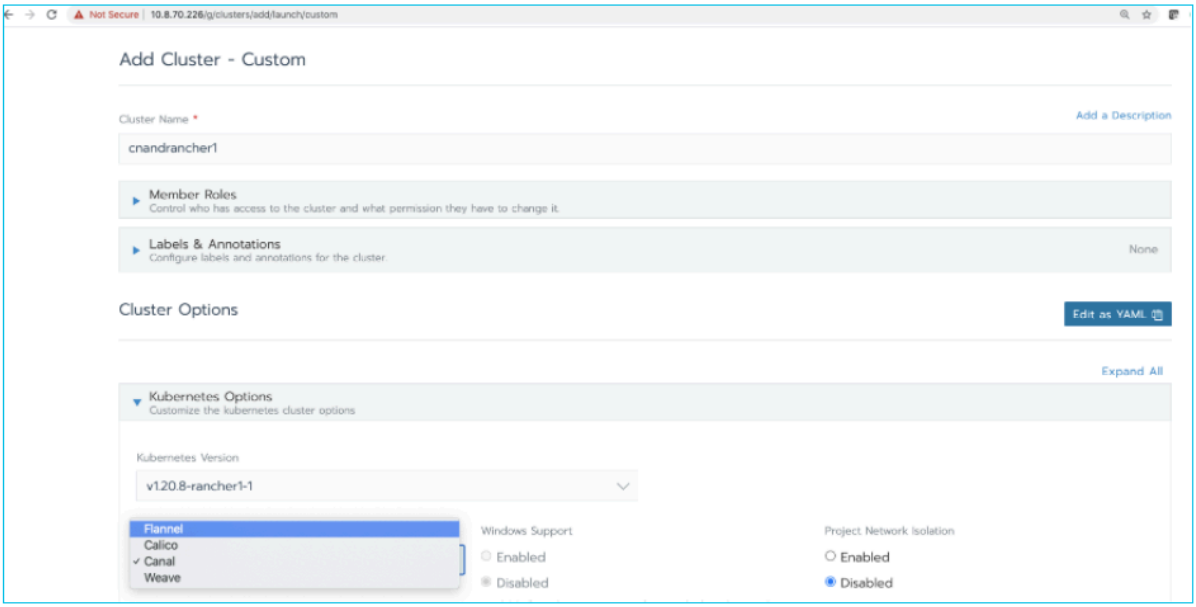
1. Vaya a la interfaz de usuario de Rancher y haga clic en **Add Cluster (Añadir clúster)**.



2. Haga clic en **Existing nodes (Nodos existentes)**.

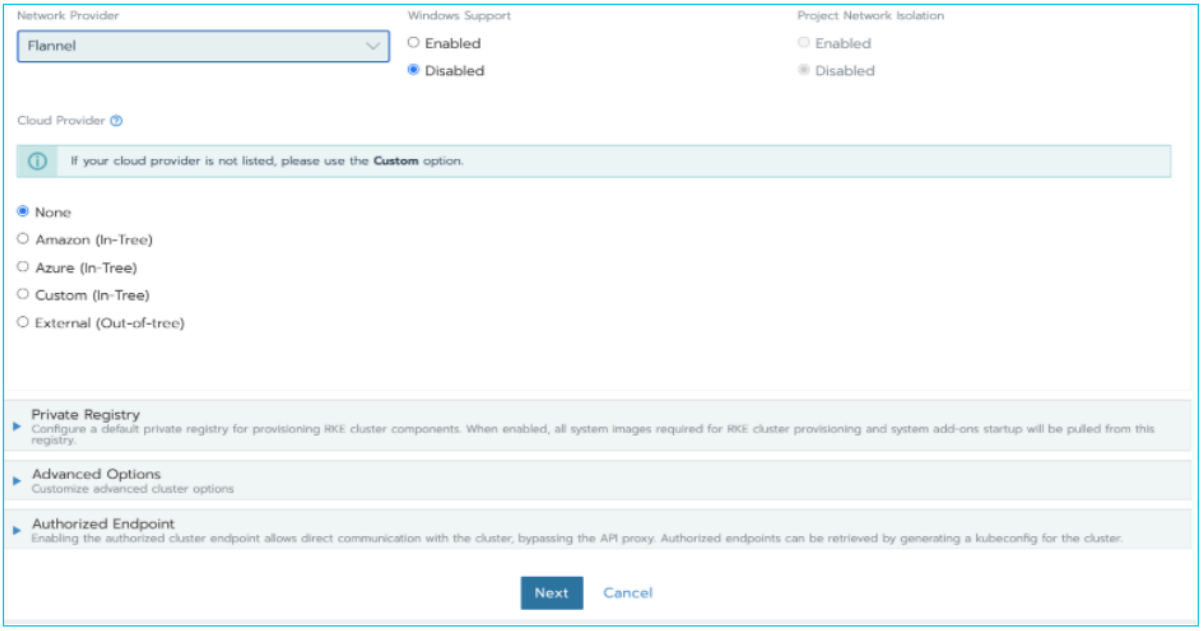


3. Escriba el **nombre del clúster** y, a continuación, seleccione Flannel en el menú desplegable **Proveedor de red**

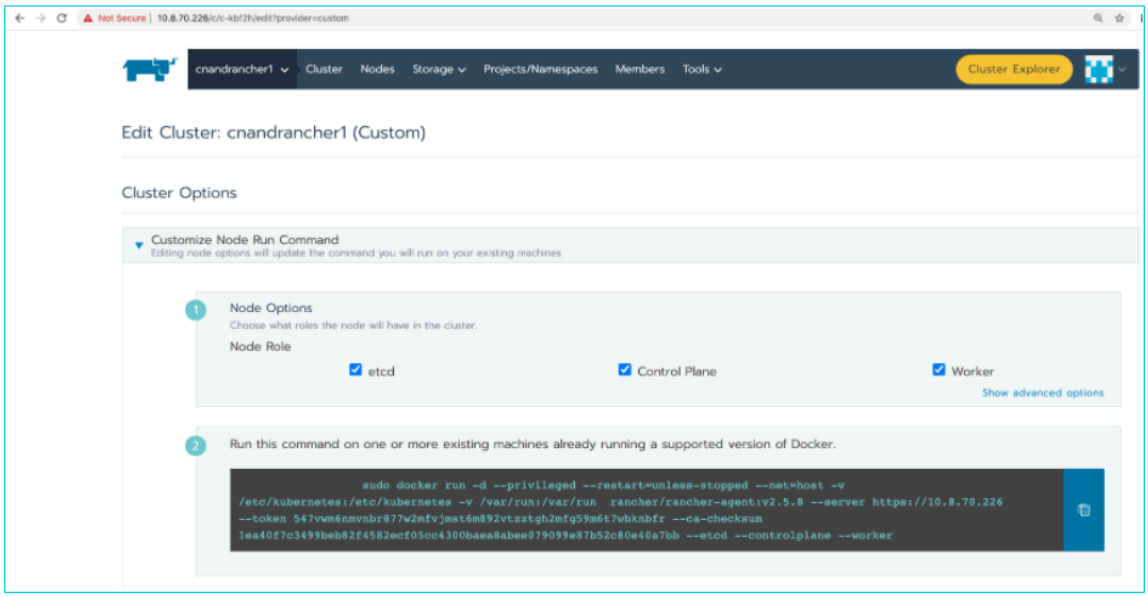


.

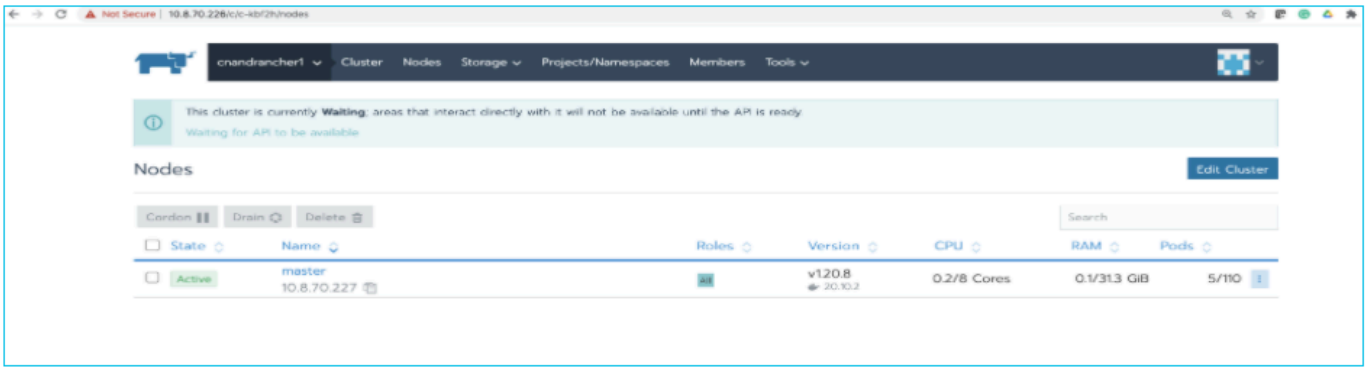
4. Conserve los valores predeterminados para todos los demás campos y, a continuación, haga clic en **Next (Siguiente)**.



5. En Opciones de nodo, seleccione las tres opciones de **Función de nodo** y, a continuación, ejecute el comando dado en el nodo maestro usando SSH.



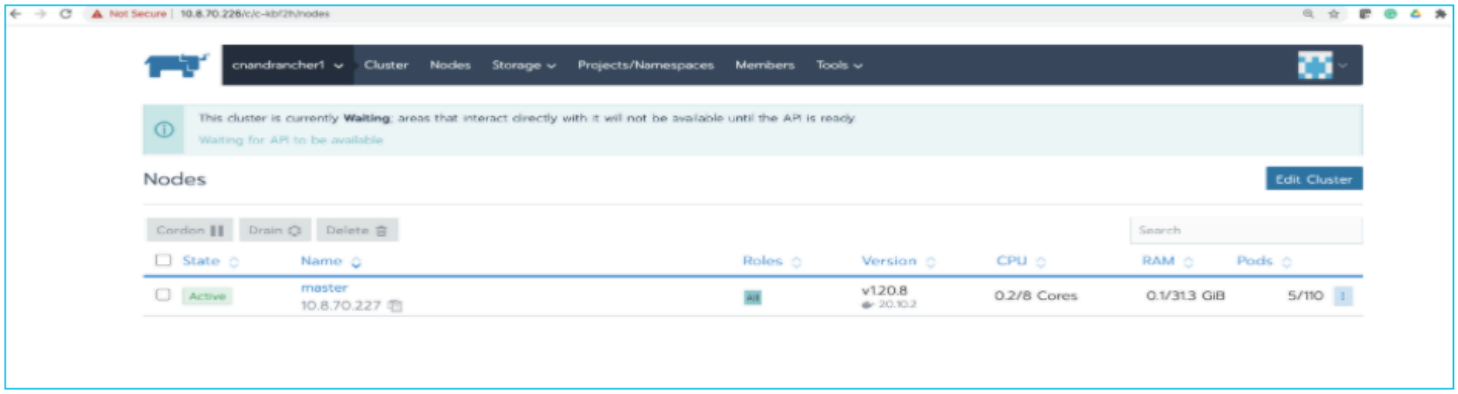
6. Compruebe que el nodo maestro se ha añadido correctamente.



7. SSH en cada nodo de trabajo y ejecute el siguiente comando:

```
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
```

Al ejecutar correctamente el comando en un nodo Maestro y dos nodos de Trabajo, verá que el clúster de Rancher está listo como se muestra a continuación:



Modificación del archivo YAML de opciones de clúster de Rancher

Antes de implementar el cortafuegos CN-Series, debe modificar el archivo YAML de opciones de clúster como se describe a continuación.

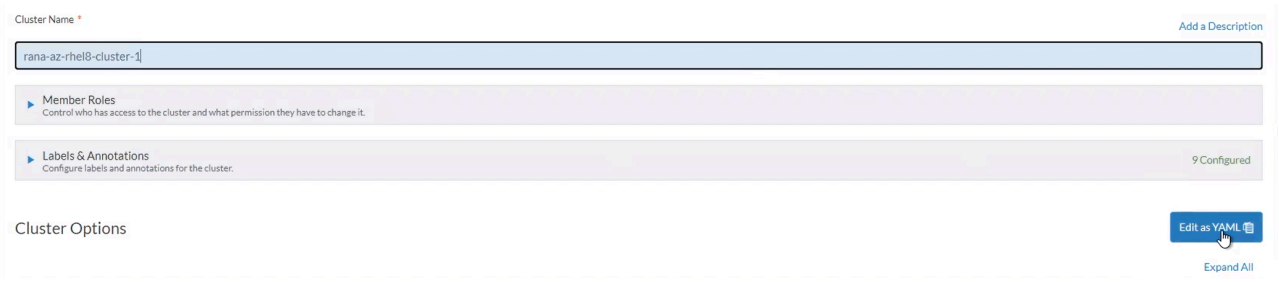
El cortafuegos de CN-Series con Rancher es compatible con Rancher 2.5 o posterior con k8s 1.20.5.

STEP 1 | Con las credenciales de administrador que creó anteriormente, inicie sesión en el portal de Rancher.

STEP 2 | Haga clic en el **Menú de navegación** y seleccione **Gestión de clústeres**.

STEP 3 | Busque el clúster que se va a modificar, haga clic en el menú de puntos suspensivos verticales y, a continuación, seleccione **Edit Config (Editar configuración)**.

STEP 4 | Haga clic en **Edit as YAML (Editar como YAML)**.



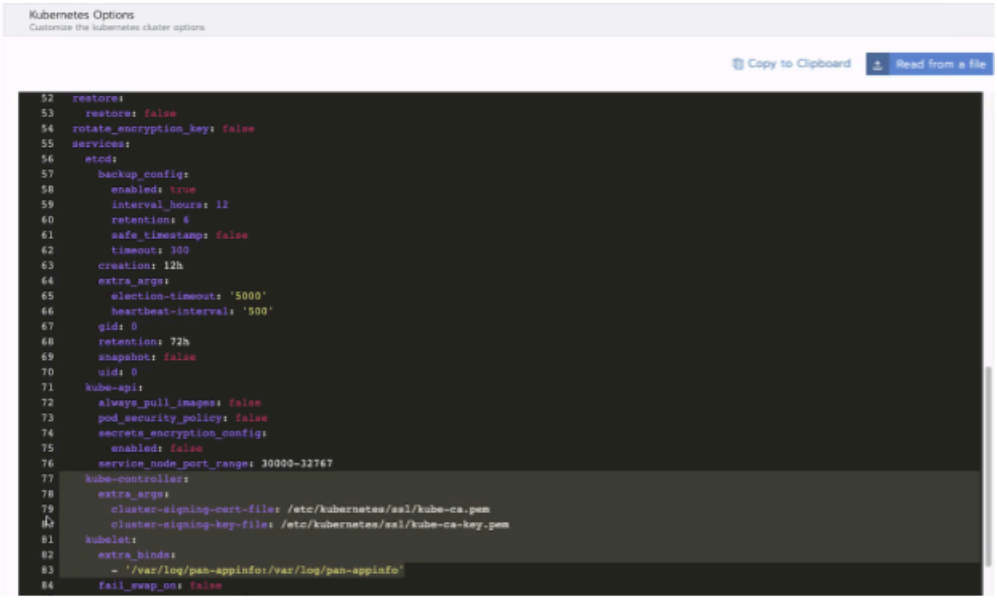
Para obtener diferentes versiones de Rancher, consulte la Documentación de Rancher.

STEP 5 | Añada las siguientes líneas en la sección **Services (Servicios)** en el archivo YAML existente.

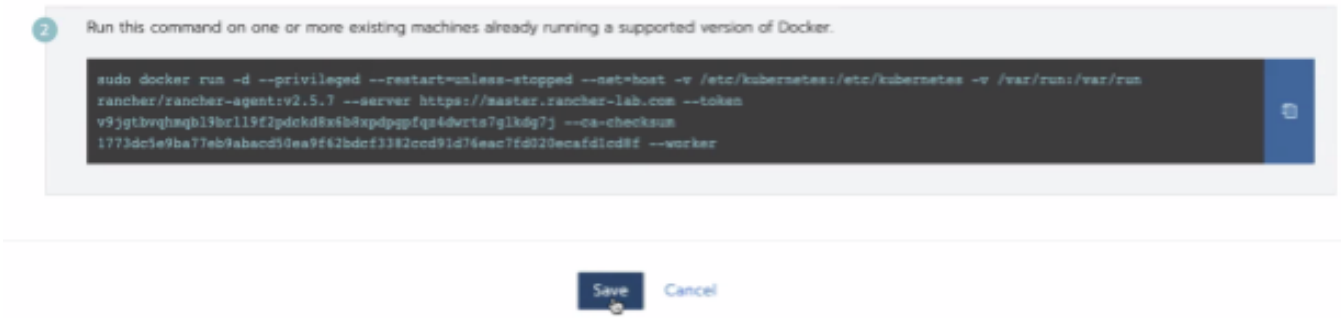
```
kube-controller: extra_args: cluster-signing-cert-file: "/etc/
kubernetes/ssl/kube-ca.pem" cluster-signing-key-file: "/etc/
kubernetes/ssl/kube-ca-key.pem"
```

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

Si está utilizando una ruta de almacenamiento distinta de `'/mnt'`, debe asegurarse de modificar la ruta de almacenamiento en `extra_binds`.



STEP 6 | Haga clic en **Save (Guardar)** y espere hasta que se active la actualización del clúster antes de implementar el cortafuegos CN-Series.



Parámetros editables en archivos YAML de implementación de CN-Series

Los archivos YAML incluyen varios parámetros editables, las siguientes tablas enumeran los que debe modificar para Implementación de los cortafuegos CN-Series correctamente.

- 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	
Enrutamiento avanzado (necesario para implementaciones de Kubernetes 3.0.0) `PAN_ADVANCED_ROUTING:"true"`	Si está utilizando el Enrutamiento avanzado con el complemento de Kubernetes 3.0.0, primero debe configurar habilitar en PAN-OS y luego manualmente en la pila de plantillas. Después de habilitarlo, confirme y envíe la configuración. Para obtener más información, consulte Enrutamiento avanzado.
Dirección IP de Panorama `PAN_PANORAMA_IP:`	Incluya la dirección IP de Panorama a la que se conectará el pod CN-MGMT. Si ha configurado sus servidores de administración de Panorama en una configuración de alta disponibilidad (HA), proporcione la dirección IP del Panorama activo principal. Puede localizar la dirección IP de Panorama en **Dashboard (Panel)** > **General Information (Información general)**.
Nombre del grupo de dispositivos `PAN_DEVICE_GROUP:`	Especifique el nombre del grupo de dispositivos al que desea asignar los pods CN-NGFW. Desde Panorama, enviará políticas idénticas a todos los pods CN-NGFW que son gestionados por un par de pods CN-MGMT (o que pertenecen a un PAN-SERVICE-NAME).

PAN-CN-MGMT-CONFIGMAP	
	Puede buscar el nombre del grupo de dispositivos en **Panorama** > **Device Groups (Grupos de dispositivos)**.
Nombre de la pila de plantillas `PAN_TEMPLATE_STACK:`	Le permite configurar los ajustes que permiten que los cortafuegos (pods CN-NGFW) operen en la red. Puede localizar el nombre de la pila de plantillas en **Panorama** > **Templates (Plantillas)**.
Nombre del grupo de recopiladores de logs `PAN_PANORAMA_CGNAME:`	Habilita el almacenamiento de logs para los logs generados en los cortafuegos CN-NGFW. Sin un grupo de recopiladores, los logs del cortafuegos no se guardan. Puede buscar el nombre del grupo de recopiladores en **Panorama** > **Collector Groups (Grupos de recopiladores)**.
(Opcional) `#CLUSTER_NAME:`	Especifique el nombre del clúster. El nombre de host del pod CN-MGMT combina el nombre StatefulSet definido en `PAN-CN-MGMT.yaml` y este CLUSTER_NAME opcional. Este nombre de host le permite identificar pods que están asociados con diferentes clústeres, si gestiona varios clústeres en el mismo dispositivo Panorama. Se recomienda usar el mismo nombre aquí y en el complemento de Kubernetes en Panorama.
(Opcional) Dirección IP de HA de Panorama `#PAN_PANORAMA_IP2:`	Dirección IP del par de Panorama (pasivo-secundario) que está configurado en una configuración de alta disponibilidad. Compruebe que PAN_PANORAMA_IP sea el del Panorama principal activo. Puede localizar la dirección IP del peer de HA de Panorama en **Panorama** > **High Availability (Alta disponibilidad)** > **Setup (Configuración)**.
(Requerido para GTP) Seguridad de GTP `#PAN_GTP_ENABLED: "true"`	Habilite este parámetro para la seguridad de GTP en el cortafuegos CN-Series. Después de habilitar GTP, puede usar Panorama para configurar la seguridad de GTP y supervisar el tráfico de GTP en el cortafuegos.
(Requerido para la compatibilidad con tramas gigantes, si el CNI principal no usa tramas gigantes) Modo de tramas gigantes	El pod CN-MGMT durante el arranque utiliza la MTU eth0 para detectar automáticamente si se habilita el modo de trama gigante. Por lo tanto, si su CNI secundario usa tramas gigantes, mientras que el CNI principal no lo hace, debe definir

PAN-CN-MGMT-CONFIGMAP	
`#PAN_JUMBO_FRAME_ENABLED: "true"`	`PAN_JUMBO_FRAME_ENABLED: "True"` se utiliza para habilitar el modo de trama gigante en el cortafuegos CN-Series. Debe realizar este cambio antes de implementar el StatefulSet de CN-MGMT.

(Requerido para la asignación flexible de recursos del sistema)
• CN-Series como un DaemonSet
`#PAN_NGFW_MEMORY: "42 Gi"`
• La serie CN-Series como servicio K8s
`#PAN_NGFW_MEMORY: "6,5 Gi"`
`#PAN_NGFW_MEMORY: "42 Gi"`
 	
Para seguridad nativa 5G, se recomienda 48Gi
	Si necesita una mayor capacidad y desea configurar más memoria para satisfacer sus necesidades de implementación, defina el valor de la memoria con este parámetro.
	• CN-Series como un DaemonSet
	La capacidad pequeña es de 42 Gi o menos y la gran capacidad está por encima de 42 Gi.
	• La serie CN-Series como servicio K8s
	La capacidad pequeña es inferior a 6,5 Gi, la capacidad mediana está entre 6,5 Gi y 42Gi, y la gran capacidad está por encima de 42 Gi.
	
	Este cambio también requiere la misma asignación de memoria o una mayor en `pan-cn-ngfw.yaml`.

(Opcional) AF-XDP `#PAN_DATA_MODE: “next-gen”`	Este parámetro es necesario para habilitar la ruta de datos eXpress de la familia de direcciones (AF-XDP). AF-XDP es un socket basado en eBPF que está optimizado para el procesamiento de paquetes de alto rendimiento adecuado para servicios nativos en la nube, para aumentar el rendimiento efectivo. Esto requiere la versión del kernel 5.4 o posterior. Además, el modo jumbo no es compatible; EKS no puede usar este parámetro porque el modo jumbo está habilitado de forma predeterminada. Además, se requiere el modo privilegiado en PAN-CN-NGFW.
(Necesario para habilitar HPA) (AKS y GKE) `#HPA_NAME` (Solo EKS) `#PAN_NAMESPACE_EKS` (Solo AKS) `#PAN_INSTRUMENTATION_KEY`	Se requieren varios parámetros para habilitar el Escalado automático de pod horizontal (HPA) en el cortafuegos CN-Series como servicio. • Para cada entorno, debe proporcionar un nombre único para identificar el recurso HPA por espacio de nombres o por inquilino.

PAN-CN-MGMT-CONFIGMAP	
	• Para la implementación de AKS, debe proporcionar una clave de instrumentación de Azure Application Insight.

Los siguientes valores predeterminados se definen en el archivo `pan-cn-mgmt-configmap.yaml`.

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

Estos valores predeterminados le permiten utilizar estos archivos para una prueba de concepto rápida. Si desea modificarlos, por ejemplo, para implementar más de un par tolerante a los fallos de pods PAN-MGMT que gestionan hasta 30 pods PAN-NGFW, debe modificar `pan-mgmt-svc` para usar otro nombre de servicio. Cuando modifica estos valores, debe actualizar las referencias correspondientes en los otros archivos YAML para que coincidan con los valores que defina en este archivo.

PAN-CN-MGMT-SECRET

PAN-CN-MGMT-SECRET	
Clave de auten. de VM `PAN_PANORAMA_AUTH_KEY:`	Permite que Panorama autentique los cortafuegos para que pueda añadir cada cortafuegos como dispositivo gestionado. La clave de autenticación de máquina virtual es necesaria durante todo el periodo de duración de la implementación. Sin una clave válida en la solicitud de conexión, el cortafuegos CN-Series no podrá registrarse con Panorama. Consulte Instalar el complemento de Kubernetes para el cortafuegos CN-Series.

Certificado de dispositivo para la serie CN-Series
`CN-SERIES-AUTO-REGISTRATION-PIN-ID`
`CN-SERIES-AUTO-REGISTRATION-PIN-VALUE`
	El cortafuegos requiere el certificado del dispositivo para obtener cualquier derecho de licencia del sitio y acceder de forma segura a los servicios en la nube de Palo Alto. Genere el ID de PIN y el valor del PIN en el CSP de Palo Alto Networks y utilice el PIN antes de que caduque. Por ejemplo:
	`CN-SERIES-AUTO-REGISTRATION-PIN-ID:`
	`"01cc5-0431-4d72-bb84-something”`
	`CN-SERIES-AUTO-REGISTRATION-PIN-VALUE:`
	`"12…….13e"`

PAN-CN-MGMT-SECRET	
	El siguiente campo adicional para CN-SERIES-AUTO-REGISTRATION-API-CSP está desactivado y no es obligatorio: `"certificate.paloaltonetworks.com"` Consulte Instalar un certificado de dispositivo en el cortafuegos CN-Series.

PAN-CN-MGMT

PAN-CN-MGMT	
Ruta de la imagen para la imagen del contenedor de inicialización para el cortafuegos CN-MGMT `initContainers: - name: pan-mgmt-init image: <your-private-registry-image-path>`	El contenedor de inicialización genera certificados que se utilizan para asegurar la comunicación entre instancias de pods CN-MGMT y entre pods CN-MGMT y pods CN-NGFW. Edite la ruta de la imagen para que apunte a la ubicación en la que cargó la imagen docker para el contenedor CN-MGMT.
Ruta de la imagen para los contenedores de imágenes CN-MGMT: `initContainers: - name: pan-mgmt image: <your-private-registry-image-path>`	Edite la ruta de la imagen para que apunte a la ubicación en la que cargó la imagen docker para el contenedor CN-MGMT.
Nombre de host del cortafuegos CN-MGMT `tipo: StatefulSet metadata: name: pan-mgmt-sts`	El nombre de host del cortafuegos CN-MGMT se obtiene combinando el nombre StatefulSet y el nombre del clúster opcional que puede haber definido en `pan-cn-mgmt-configmap.yaml`. El nombre de host predeterminado de los pods CN-MGMT es pan-mgmt-sts-0 y pan-mgmt-sts-1, porque el nombre de StatefulSet es pan-mgmt-sts y el nombre del clúster no está definido. *Si el nombre de host tiene más de 30 caracteres, el nombre se truncará a los 30 caracteres.*
(Obligatorio si ha definido la memoria para la asignación flexible de recursos del sistema)	Si ha asignado un valor de memoria mayor o igual a 40Gi para `#PAN_NGFW_MEMORY: "40Gi"`en el `pan-cn-mgmt-`

PAN-CN-MGMT	
	`configmap.yaml`, asegúrese de que tiene valores idénticos en la `solicitud` y el `límite` para la CPU y la memoria para lograr una mayor utilización de la capacidad en `containers: resources: requests: # configurable based on desired logging, capacities cpu: "4" memory: "16.0Gi" limits: cpu: "4" memory: "16.0Gi"` Para la seguridad nativa 5G, los valores recomendados son cpu=4, memoria=16Gi
(Solo para una implementación de Kubernetes nativa autogestionada o local) `storageClassName: local`	Para una implementación autogestionada, la configuración predeterminada tiene "storageClassName: local". Si su clúster ha aprovisionado dinámicamente Volúmenes persistentes (PV), debe modificar "storageClassName: local" para que coincida con esa storageClass o eliminar estas líneas si se utiliza DefaultStorageClass. Si su clúster no tiene volúmenes persistentes (PV) aprovisionados dinámicamente, el administrador del clúster puede crear PV estáticos con el `pan_cn_pv_local.yaml` proporcionado; que tiene 2 conjuntos de pocos PV, uno para cada uno de los pods de StatefulSet de PAN-CN-MGMT. Puede modificar `pan_cn_pv_local.yaml` para que coincida con los volúmenes de su configuración e implementarlo antes de implementar `PAN-CN-MGMT.yaml`.

PAN-CN-NGFW-CONFIGMAP

No necesita modificar ningún valor PAN a menos que necesite cambiar lo siguiente:

- `PAN_SERVICE_NAME: pan-mgmt-svc`

  El nombre del servicio deberá coincidir con lo definido en PAN-CN-MGMT-CONFIGMAP.

- `FAILOVER_MODE: failopen`

  Puede cambiar esto para failclose. Entra en vigor solo cuando CN-NGFW no obtiene una licencia.

  - En el modo de apertura por fallo, el cortafuegos recibirá el paquete y lo enviará sin inspeccionarlo. La transición al modo de apertura por fallo provoca un reinicio interno y una breve interrupción del tráfico.
  - En el modo de cierre por fallo, el cortafuegos descartará todos los paquetes que reciba. El modo de cierre por fallo también cierra el CN-NFGW y libera la ranura asignada para permitir que otros CN-NFGW con licencia usen esa ranura.
- Fijación de CPU: en `pan-cn-ngfw-configmap.yaml`, la fijación de CPU y el hyperthreading están deshabilitados. No alterne este ajuste para habilitar la fijación de CPU para núcleos físicos dedicados en lugar de núcleos lógicos con hyperthreading, a menos que la asistencia técnica de Palo Alto Networks le guíe.

  ```
  PAN_CPU_PINNING_ENABLED: "Verdadero"/"Falso"
  PAN_HYPERTHREADING_ENABLE: "Verdadero"/"Falso"
  ```

PAN-CN-NGFW

PAN-CN-NGFW	
Ruta de imagen para el contenedor de imagen de CN-NGFW imagen `containers: - name: pan-ngfw-container image: <your-private-registry-image-path>`	Edite la ruta de la imagen para que apunte a la ubicación en la que cargó la imagen docker para el contenedor de CN-NGFW.
(Obligatorio si ha definido la memoria para la asignación flexible de recursos del sistema)	Si ha asignado un valor de memoria mayor o igual a 40Gi para `#PAN_NGFW_MEMORY: "40Gi"`en el `pan-cn-mgmt-configmap.yaml`, asegúrese de que tiene valores idénticos en `request` y `limit` para la cpu y la memoria para lograr QoS garantizada en `containers: resources: requests: #configurable based on desired throughput, number of running pods cpu: "1" memory: "40.0Gi" limits: cpu: "1" memory: "40.0Gi"` Para la seguridad nativa 5G, los valores recomendados son cpu=12, memory=48Gi.

PAN-CN-NGFW	

Nota:
• La siguiente anotación identifica el daemonset de PAN-NGFW:
`paloaltonetworks.com/app: pan-ngfw-ds`
No modifique este valor.
• La siguiente anotación identifica el nombre del cortafuegos ("pan-fw"):
`paloaltonetworks.com/firewall: pan-fw`
En `pan-cni-configmap.yaml`, este nombre de cortafuegos debe coincidir exactamente con el `cni_network_config: "firewall"`
Y esta anotación debe coincidir exactamente en el yaml de la aplicación que usa para implementar cada pod de aplicación.
	El pod CN-NGFW en cada nodo protege los pods de aplicaciones y los espacios de nombres que tienen la anotación:
	`paloaltonetworks.com/firewall: pan-fw`
	Mantenga esta anotación como está.

(Opcional) AF-XDP `imagePullPolicy: Always securityContext: capabilities: #add: ["NET_ADMIN","NET_RAW","NET_BROADCAST","NET_BIND_SERVICE"] add: ["ALL"] privileged: true resources:`	Debe añadir `privileged: true` a la sección que se muestra a la izquierda. Este parámetro es necesario para habilitar la ruta de datos eXpress de la familia de direcciones (AF-XDP). También debe habilitar AF-XDP en PAN-CN-MGMT-CONFIGMAP.

PAN-CNI-CONFIGMAP

Estos parámetros son opcionales.

PAN-CNI-CONFIGMAP	
Lista de nombres de cortafuegos a los que podría pertenecer el pod de la aplicación: `"firewall": [ "pan-fw" ]`	Si bien no se requieren modificaciones, si cambia la anotación `paloaltonetworks.com/firewall: pan-fw` en el `pan-cn-ngfw.yaml` , debe reemplazar el valor en `"firewall": ["pan-fw"]` para que coincida.
`"exclude_namespaces": []`	Si bien no se requieren modificaciones, si desea excluir espacios de nombres específicos, añádalo

PAN-CNI-CONFIGMAP	
	a **"exclude_namespaces"**, de modo que la anotación del pod de aplicación en ese espacio de nombres se ignore y el tráfico no se redirija al pod CN-NGFW para su inspección.
"security_namespaces": ["kube-system"]	Añada los espacios de nombres en los que ha implementado el daemonset de CN-NGFW en security_namespaces. El espacio de nombres predeterminado es kube-system.

“interfaces”	Añada las interfaces en los pods de aplicaciones desde los que desea redirigir el tráfico al pod CN-NGFW para su inspección. De forma predeterminada, solo se inspecciona el tráfico eth0 y puede añadir interfaces adicionales como una lista de cadenas separadas por comas, p. ej. [“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"],`
	`}`
	Además de esto, también debe añadir `pan-cni` *a la anotación* `k8s.v1.cni.cncf.io/networks` *en el pod de la aplicación.*
	Por ejemplo:
	`metadata: name: testpod annotations: paloaltonetworks.com/firewall: pan-fw k8s.v1.cni.cncf.io/networks: sriov-net1, sriov-net2, macvlan-conf, pan-cni`

PAN-CNI-CONFIGMAP	
	Actualmente, el modelo CN-Series no es compatible con DPDK y no permite que el pod de aplicaciones use DPDK. Es posible que deba modificar el pod de la aplicación si la aplicación no se ajusta automáticamente al modo no DPDK.
(CN-Series como un servicio de Kubernetes únicamente) “dpservicename” “dpservicenamespace”	Cuando la serie CN-Series se implementa como un servicio, se requieren `dpservicename` y `dpservicenamespace`. De forma predeterminada, `dpservicename` es `"pan-ngfw-svc"` y `dpservicenamespace` es `"kube-system"`.

PAN-CNI

PAN-CNI	
Ruta de la imagen para la imagen del contenedor PAN-CNI que tiene los archivos binarios CNI y el archivo de configuración de red CNI en cada nodo. `containers: name: install-pan-cni image: <your-private-registry-image-path>`	Edite la ruta de la imagen para que apunte a la ubicación en la que cargó la imagen docker para el contenedor de PAN-CNI.

PAN-CNI-MULTUS

Si utiliza Multus CNI en una implementación nativa o autogestionada de Kubernetes, como VMware TKG +, utilice el `pan-cni-multus.yaml` en lugar del `pan-cni.yaml`.

Protección de 5G con el cortafuegos CN-Series

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

Para obtener visibilidad y control del tráfico 5G para empresas privadas y las implementaciones de núcleos de paquetes móviles 5G en redes de operadores móviles en Kubernetes, revise las siguientes secciones para conocer los entornos compatibles y cómo modificar los archivos YAML para desbloquear la Seguridad GTP y la Seguridad nativa de 5G en el cortafuegos CN-Series. Además de habilitar estas capacidades, cuando implementa el cortafuegos CN-series, también debe habilitar Panorama para la seguridad GTP y/o Seguridad SCTP.

Tiempo de ejecución del contenedor	Docker CRI-O Containerd
Versión de Kubernetes	1.17 a 1.27
Kubernetes gestionado por el proveedor de la nube	• AWS EKS (1.17 a 1.27 para CN-Series como un daemonset demonio y CN-Series como un modo de servicio de implementación.) • AWS EKS (1.17 a 1.22 para la CN-Series como modo de implementación CNF). • AWS EKS (1.22 a 1.27 para la implementación de la CN-Series como un clúster CN). • EKS en AWS Outpost (1.17 a 1.25) *CN-Series para EKS en AWS Outpost no es compatible con SR-IOV ni Multus.* • Azure AKS (1.17 a 1.27) *En Azure AKS, PAN-OS 11.0.2 es la versión mínima necesaria compatible con kubernetes 1.25 y superiores.*

	• GCP GKE (1.17 a 1.27)  *Incluye GKE Dataplane V2.* • OCI OKE (1.23)
Kubernetes gestionado por el cliente	En la nube pública o en el centro de datos local. Asegúrese de que la versión de Kubernetes, los tipos de CNI y las versiones de Host VM OS sean según se enumeran en esta tabla. VMware TKG+ versión 1.1.2 • Plataforma de infraestructura: vSphere 7.0 • SO de VM de host de Kubernetes—Photon OS
VM de host de Kubernetes	Sistema operativo: • Ubuntu 16.04 • Ubuntu 18.04 • Ubuntu-22.04 • RHEL / Centos 7.3 y posterior • CoreOS 21XX, 22XX • SO optimizado para contenedores
	Versión del kernel de Linux: • 4.18 o posterior (solo modo de servicio K8s) • 5.4 o posterior necesario para habilitar el modo AF_XDP. Consulte Parámetros editables en archivos YAML de la implementación CN-Series para obtener más información.
	Filtro de red del kernel de Linux: Iptables
Complementos CNI	CNI Spec 0.3 y posterior: • AWS-VPC • Azure • Calico • Flannel • Weave • Para Openshift, OpenshiftSDN

	• Lo siguiente se admite en el cortafuegos CN-Series como un DaemonSet.
	  • Multus
	  • Bridge
	  • SR-IOV
	  • Macvlan

OpenShift	• Versiones 4.2, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 y 4.13 *OpenShift 4.7 está clasificado en la serie CN-Series como DaemonSet únicamente.* *PAN-OS 11.0.2 es la versión mínima requerida compatible con 4.12 y superior.* • OpenShift en AWS

Tiempo de ejecución del contenedor	Versión
Cortafuegos CN-Series	PAN-OS 10.0.3 o versiones posteriores
Complemento de Kubernetes	1.0.1 o posterior
Panorama	10.0.0 o posterior
La siguiente es una lista de todos los parámetros editables en un archivo YAML que utiliza para implementar el cortafuegos CN-Series: Para obtener más detalles, consulte Parámetros editables en archivos YAML de la implementación CN-Series y los Componentes básicos del modelo CN-Series.	
Habilitar GTP	En el conjunto pan-cn-mgmt-configmap.yaml — `PAN_GTP_ENABLED : "True"`, antes de implementar el StatefulSet de CN-MGMT.
Habilitar el modo de trama gigante	En el conjunto pan-cn-mgmt-configmap.yaml: `PAN_JUMBO_FRAME_ENABLED: "True"`, antes de implementar el StatefulSet de CN-MGMT. El pod CN-MGMT durante el arranque utiliza la MTU "eth0" para detectar automáticamente si se habilita el modo de trama gigante. Por lo tanto, si su CNI secundario usa tramas gigantes, mientras que el CNI principal no lo hace, debe definir `PAN_JUMBO_FRAME_ENABLED: "True"` se utiliza para habilitar el modo de trama gigante en el cortafuegos CN-Series.

Tiempo de ejecución del contenedor	Versión
	Actualmente, el modelo CN-Series no es compatible con DPDK y no permite que el pod de aplicaciones use DPDK. Es posible que deba modificar el pod de la aplicación si la aplicación no se ajusta automáticamente al modo no DPDK.
Habilitar la flexibilidad de los recursos del sistema	Si necesita un mayor rendimiento y desea configurar más memoria para satisfacer sus necesidades de implementación en el conjunto pan-cn-mgmt-configmap.yaml: PAN_NGFW_MEMORY="48Gi" *Para crear plantillas (Helm), puede tomar la misma variable que la que se asigna al pod CN-NGFW. Cuando se habilita una huella de memoria más grande, el StatefulSet de CN-MGMT solo admite un pod CN-NGFW.*
Configurar vCPU, memoria para 5G	La configuración recomendada para los pods CN-MGMT (en pan-cn-mgmt.yaml) y los pods NGFW (en pan-cn-ngfw.yaml) es tener valores idénticos en "request" (solicitud) y "limit" (límite) para que la CPU y la memoria ofrezcan una QoS garantizada. Para los pods CN-MGMT, los valores recomendados son cpu = 4, memoria = 16Gi. Para controlar la ubicación de los pods CN-MGMT, por ejemplo, en el mismo o en diferentes nodos que donde se implementan los pods CN-NGFW, utilice la capacidad de selector de nodos en k8s.
	Para los pods CN-NGFW, los valores recomendados son cpu = 12, memoria = 48Gi. Para controlar la ubicación de los pods CN-NGFW, por ejemplo, en el mismo o en diferentes nodos que donde se implementan los pods CN-NGFW, utilice la capacidad de selector de nodos en k8s.
Seleccionar el archivo CNI yaml	Multus CNI funciona como un meta-complemento que llama a otros complementos CNI. En entornos OpenShift, Multus está habilitado de

Tiempo de ejecución del contenedor	Versión
	forma predeterminada, por lo que puede usar `pan-cni.yaml`. En otros entornos en los que se admite Multus pero es opcional, como los entornos autogestionados (nativos), utilice `pan-cni-multus.yaml` en lugar de `pan-cni.yaml`.

Revise también los Requisitos del sistema para el cortafuegos CN-Series antes de continuar implementando el cortafuegos CN-Series.

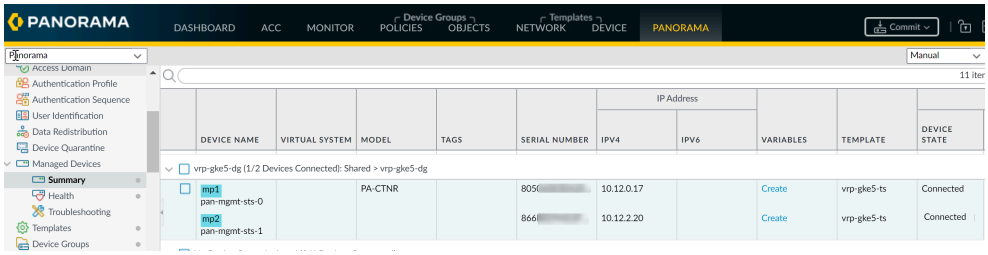
Configuración de Panorama para proteger una implementación de Kubernetes

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con gráfico de Helm

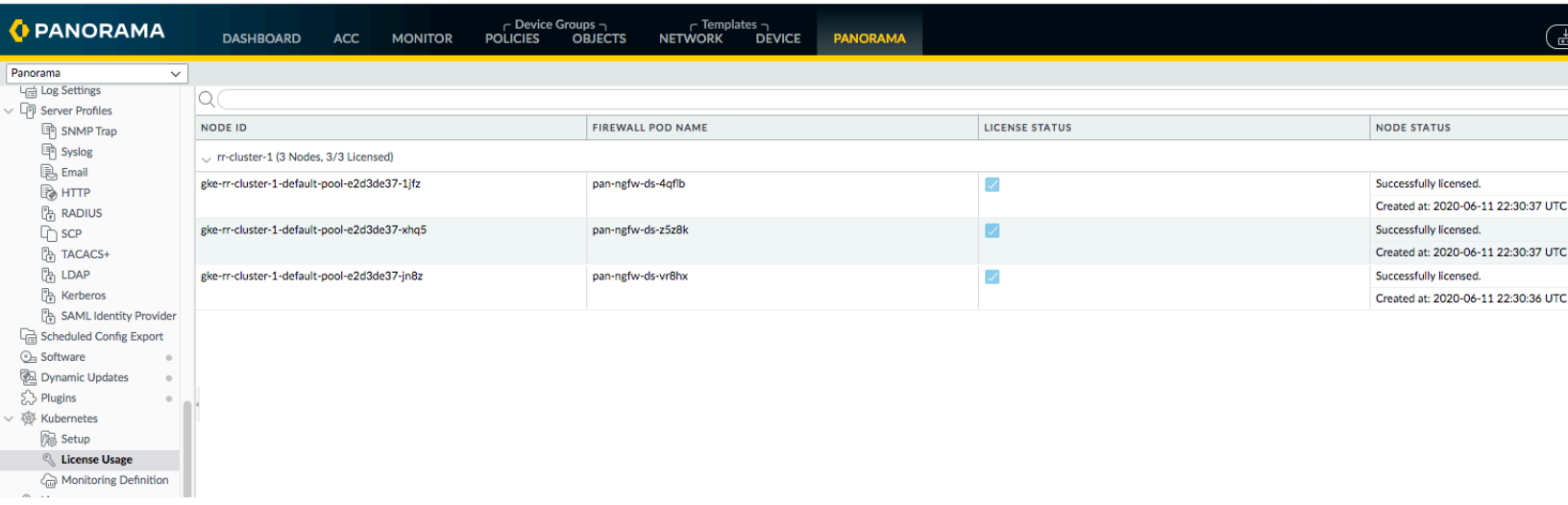
Después de la Instalación del complemento de Kubernetes para la CN-Series e implementación del cortafuegos CN-Series, para supervisar el clúster de Kubernetes y configurar las políticas de seguridad que habilitan la aplicación del tráfico, debe completar las siguientes tareas

STEP 1 | Verifique que los pods CN-MGMT estén registrados en Panorama y que los pods CN-NGFW cuentan con licencias.

1. Seleccione **Panorama** > **Managed Devices (Dispositivos gestionados)** > **Summary (Resumen)**.



2. Seleccione **Panorama** > **Plugins (Complementos)** > **Kubernetes** > **License Usage (Uso de licencia)** para comprobar que cada nodo dentro del clúster tiene un token asignado.



STEP 2 | Cree un perfil de reenvío de logs para reenviar los logs a Panorama.

El perfil define los destinos para los distintos logs que se generarán en el cortafuegos.

1. Seleccione el grupo de dispositivos que ha creado para la implementación de k8s en el menú desplegable **Device Group (Grupo de dispositivos)**.
2. Seleccione **Objects (Objetos)** > **Log Forwarding (Reenvío de logs)** y luego **Add (Añadir)**.
3. Introduzca un **Name (Nombre)** para identificar el perfil. Si desea asignar automáticamente el perfil a nuevas reglas y zonas de seguridad, escriba **`predeterminado`**. Si no desea un perfil predeterminado o desea anular un perfil predeterminado existente, introduzca un **Name (Nombre)** que le ayudará a identificar el perfil cuando lo asigne a las reglas de seguridad.
4. **Add (Añadir)** los tipos de log que se van a reenviar.
5. Haga clic en **OK (Aceptar)**.

STEP 3 | Configure el complemento de Kubernetes para enviar las etiquetas a los grupos de dispositivos especificados.

Debe añadir una definición de supervisión que incluya el nombre del clúster de Kubernetes del que Panorama recupera etiquetas predefinidas y, opcionalmente, un grupo de notificación.

Se requiere un grupo de notificación si la serie CN-Series se implementa en un espacio de nombres que no sea kube-system.

Un grupo de notificación es una lista de grupos de dispositivos que reciben actualizaciones de etiquetas. Para el complemento de Kubernetes, el grupo de notificación debe incluir cortafuegos externos al clúster (lo que significa que no pertenecen al mismo grupo de dispositivos que el clúster de Kubernetes del que está recopilando atributos).

Dado que especifica el nombre del grupo de dispositivos en los archivos YAML que se usan para implementar los cortafuegos CN-Series, el complemento de Kubernetes es consciente de forma automática de todos los grupos de dispositivos que son internos al clúster y automáticamente envía todas las etiquetas predefinidas a esos grupos de dispositivos de forma predeterminada.

El complemento de Kubernetes usa Kubernetes Secrets para conocer de forma dinámica los grupos de dispositivos dentro de cada clúster. Cada vez que implementa un StatefulSet de CN-MGMT, el

Secret (secreto) se publica en el servidor de API de Kubernetes y Panorama lo descubre en el siguiente intervalo de supervisión.

1. Configure el complemento de Kubernetes para supervisar los clústeres.
2. Añada grupos de notificación. Añada un grupo de notificaciones y seleccione los grupos de dispositivos que reciben las etiquetas relacionadas con el clúster de Kubernetes.
   1. Seleccione **Panorama** > **Plugins (Complementos)** > **Kubernetes** > **Setup (Configuración)** > **Notify Groups (Grupos de notificación)** y **Add (Añadir)**.
   2. Introduzca un **Name (Nombre)** de hasta 31 caracteres para el grupo de notificación.
   3. Seleccione **Enable sharing internal tags with Device Groups (Habilitar el uso compartido de etiquetas internas con grupos de dispositivos)** si desea compartir etiquetas internas además de las etiquetas externas (predeterminadas) creadas para el clúster.
   4. Seleccione los grupos de dispositivos en los que desea registrar las etiquetas.

      Para el Grupo de notificación que seleccione, Panorama solo inserta las etiquetas externas.

      Una etiqueta externa es cualquier etiqueta a la que se puede acceder desde fuera del clúster, como las etiquetas generadas para una dirección IP de servicio externo y un puerto para una dirección IP de clúster, una dirección IP externa para todos los nodos y puertos de nodo, y la dirección IP y el puerto o nodo de puerto del balanceador de carga externo.

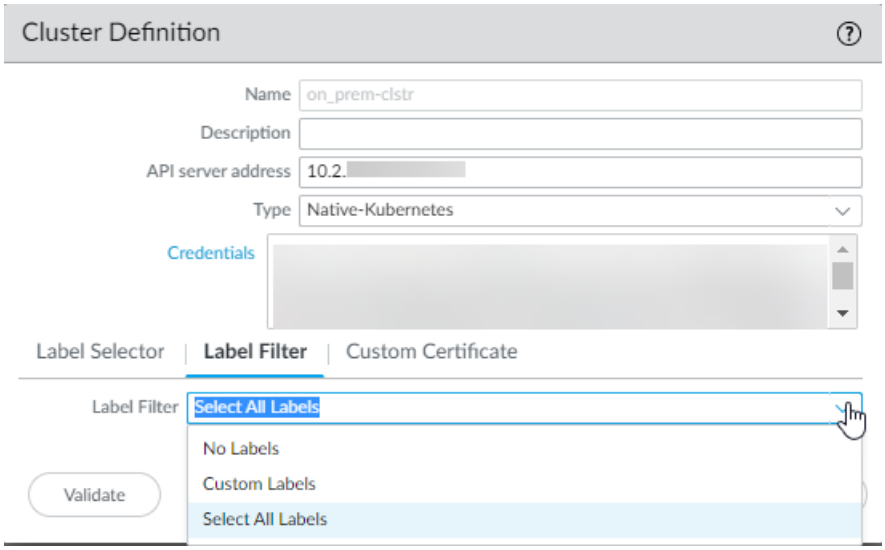
      Las etiquetas internas incluyen detalles sobre las direcciones IP del clúster interno, la dirección IP del pod, los nodos y los puertos de los nodos.

      De forma predeterminada, Panorama envía todas las etiquetas que descubre (según los filtros de etiquetas que seleccione) al grupo de dispositivos asociado con el clúster, tal como se define en el archivo YAML que utilizó para implementar los pods CN-MGMT.
3. Añada una definición de supervisión para cada clúster.
   1. Seleccione **PanoramaPlugins** > **Kubernetes** > **Monitoring Definition (Definición de supervisión)** y **Add (añadir)**.
   2. Introduzca un **Name (Nombre)** para la definición de supervisión.
   3. Seleccione el **Cluster (Clúster)** que desea supervisar.
   4. (Opcional) Seleccione un **Notify Group (Grupo de notificación)** al que desea enviar la información de asignación de dirección IP a etiqueta.

      De forma predeterminada, las etiquetas se comparten con todos los pods CN-NFGW dentro del clúster.
   5. Haga clic en **OK (Aceptar)** para guardar los cambios.
4. **Commit (Confirmar)** en Panorama.

STEP 4 | (Opcional) Configure el complemento de Kubernetes para recuperar etiquetas definidas por el usuario de los archivos YAML de su aplicación.

1. Seleccione **PanoramaPlugins** > **Kubernetes** > **Setup (Configuración)** > **Cluster (Clúster)** y seleccione la definición de clúster de la lista.
2. Seleccione el filtro de etiqueta de las siguientes opciones:



1. **No Labels (Sin etiquetas)**: no crea etiquetas para las etiquetas de Kubernetes.
2. **Custom Labels (Etiquetas personalizadas)**: crea etiquetas solo para las etiquetas que le interesan.

   Para utilizar etiquetas personalizadas, primero debe anotar los archivos YAML en su implementación de Kubernetes y luego usar cualquiera de las siguientes combinaciones para generar etiquetas personalizadas para las direcciones IP correspondientes:

   Especifique el espacio de nombres, la clave y el valor. Utilice * para todos. El complemento crea etiquetas cuando las tres entradas son válidas.

   Especifique el espacio de nombres y la clave para crear etiquetas para todas las claves coincidentes dentro de ese espacio de nombres.

   Especifique el espacio de nombres solo para crear una etiqueta para cada etiqueta dentro de ese espacio de nombres.
3. **Select All Labels (Seleccionar todas las etiquetas)**: cree etiquetas para todas las etiquetas de Kubernetes, incluidas las etiquetas personalizadas.

3. Añada una expresión de selector de etiquetas.

   El selector de etiquetas hace coincidir la etiqueta especificada dentro del clúster de Kubernetes y asigna las direcciones IP asociadas con la etiqueta a una única etiqueta. Para obtener una lista de prefijos soportados, consulte Asignación de dirección IP a etiqueta de atributos de Kubernetes.

   Para cada selector de etiquetas, Panorama genera una etiqueta que está disponible como criterio de coincidencia en grupos de direcciones dinámicas y le permite hacer cumplir las políticas de seguridad:

   1. **Prefijo de etiqueta**: frase con la que termina cada etiqueta para ayudarle a identificarla fácilmente. Por ejemplo, el selector de etiquetas k8s.cl_<clustername>.<selector-name>,

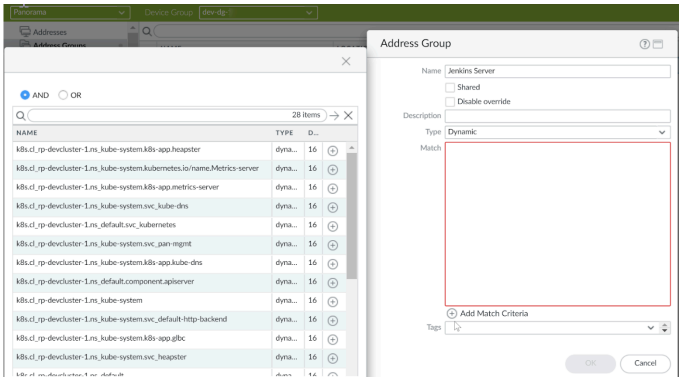
coincide con todas las IP del clúster que coinciden con el selector, todas las IP del pod que coinciden con el selector. Estas pueden estar en todos los espacios de nombres o en uno específico según lo que configure.

2. **Espacio de nombres**: * para todos los espacios de nombres, o introduzca un valor para el espacio de nombres.
3. **Filtro selector de etiquetas**: el complemento de Kubernetes admite selectores basados en igualdad para la clave de etiqueta y el valor de etiqueta. Los siguientes selectores basados en igualdad son compatibles: key = value; key ==value; key != value, por ejemplo, app = redis. También puede especificar varios selectores en una expresión como una lista separada por comas, como app == web, tier != backend. Los siguientes selectores basados en conjuntos son compatibles: key in (value1, value2), key notin (value1, value2), key, !key, por ejemplo, tier notin (frontend, backend).
4. **Aplicar en**: el tipo de recurso en el que se va a aplicar esto es Servicio, Pod, Todos.

STEP 5 | Configure los grupos de direcciones dinámicas.

1. Seleccione su Grupo de dispositivos para gestionar los pods CN-NGFW.
2. Seleccione **Objects (Objetos)** > **Address Groups (Grupos de direcciones)**.
3. Haga clic en **Add (Añadir)** e introduzca un **Name (Nombre)** y una **Description (Descripción)** para el grupo de direcciones.
4. Defina el **Type (Tipo)** como **Dynamic (Dinámico)**.

STEP 6 | Haga clic en **Add Match Criteria (Añadir criterios de coincidencia)** y seleccione el operador **AND** u **OR** y seleccione además los atributos por los que le gustaría filtrar o buscar coincidencias.

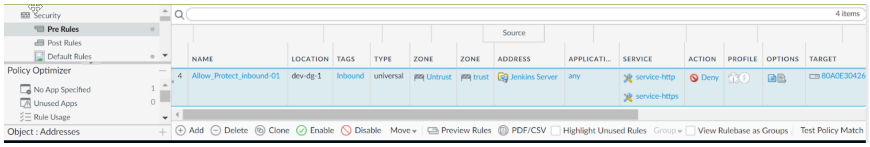


STEP 7 | Haga clic en **OK (Aceptar)** y en **Commit on Panorama (Confirmar en Panorama)**.

Utilice el enlace **more... (más...)** para ver las direcciones IP asociadas con el objeto, los servidores Jenkins en su clúster en este ejemplo.

STEP 8 | Cree reglas de políticas de seguridad para la aplicación del tráfico.

1. Seleccione **Policies (Políticas)** > **Security (Seguridad)**.
2. Haga clic en **Add (Añadir)** y escriba un **Name (Nombre)** y una **Description (Descripción)** para la política.
3. Añada la **Source Zone (Zona de origen)** para especificar la zona desde la que se origina el tráfico.
4. Añada la **Destination Zone (Zona de destino)** donde finaliza el tráfico.
5. Para la **Destination Address (Dirección de destino)**, seleccione el grupo de direcciones dinámicas que ha creado anteriormente.
6. Especifique la acción **Deny (Denegar)** para el tráfico y adjunte, de manera opcional, los perfiles de seguridad predeterminados a la regla.
7. Seleccione **Actions (Acciones)** y seleccione el perfil de **Log Forwarding (Reenvío de logs)** que creó.
8. Haga clic en **Commit (Confirmar)**.



También puede aplicar la política de seguridad para el tráfico de este a oeste dentro de los espacios de nombres. Si, por ejemplo, tiene dos espacios de nombres stage-ns y db-ns dentro de un clúster denominado clúster de preparación donde los pods frontend para una aplicación de votación se implementan en stage-NS y los pods backend de Redis se ejecutan en el espacio de nombres DB-NS. Cuando añade este clúster al complemento de Kubernetes en Panorama para su supervisión, recupera los metadatos de la etiqueta para crear etiquetas. Puede utilizar estas etiquetas para hacer cumplir las reglas de la política de seguridad. Para hacer esto, necesita

- asegúrese de que los archivos de espacio de nombres o YAML que usa para implementar las aplicaciones frontend y backend estén anotados con paloaltonetworks.com/cortafuegos: pan-fw.
- Cree el grupo de direcciones dinámicas para los pods de frontend y backend.

  Debe configurar los grupos de direcciones dinámicas en el grupo de dispositivos asociado al clúster y seleccionar primero las etiquetas para los servidores frontend. Luego, repita el proceso para crear otro grupo de direcciones dinámicas para los servidores backend.
- Agregue la regla de directiva de seguridad para permitir el tráfico de la aplicación Redis desde los pods frontend a los pods backend.

  La fuente es el grupo de direcciones dinámicas de los servidores frontend y el destino es el grupo de direcciones dinámicas para los servidores backend, y la acción está permitida.

Asignación de dirección IP a etiqueta de atributos de Kubernetes

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images

¿Dónde puedo usar esto?	¿Qué necesito?
	• Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

El complemento de Kubernetes en Panorama crea etiquetas para etiquetas predefinidas en sus clústeres de Kubernetes, etiquetas definidas por el usuario para pods y servicios, y objetos de servicio.

El complemento crea etiquetas para los siguientes objetos de Kubernetes:

- Clases de pods: ReplicaSets, DaemonSets, StatefulSets
- Tipos de servicios: ClusterIP, NodePort, LoadBalancer
- Objetos de servicio: puerto, targetPort, nodePort e interfaces de pod

De forma predeterminada, el complemento de Kubernetes en Panorama recupera las siguientes etiquetas predefinidas de cada clúster de Kubernetes que está supervisando en Panorama y crea etiquetas en el formato que se indica a continuación. A continuación, puede usar estas etiquetas como criterios de coincidencia en los grupos de direcciones dinámicas y hacer cumplir la política de seguridad para las direcciones IP subyacentes asociadas con cada etiqueta.

La longitud máxima de cada etiqueta es de 127 caracteres. Las etiquetas se truncan si superan el número máximo de caracteres. Si dos etiquetas truncadas son iguales, se añade un hash único a las etiquetas para diferenciarlas entre sí.

Puede usar el complemento de Kubernetes para distribuir la asignación de dirección IP a etiqueta para pods, nodos, espacios de nombres y servicios implementados dentro del clúster de Kubernetes a cortafuegos físicos o VM-Series, incluso si no ha implementado un cortafuegos CN-Series en ese grupo.

Etiquetas predefinidas	Formato de etiqueta en Panorama	Dirección IP recopilada
DaemonSet	k8s.cl_<cluster-name>.ns_<namespace>.ds_<pod-name>	Direcciones IP del pod
ReplicaSet	k8s.cl_<cluster-name>.ns_<namespace>.rs_<pod-name>	Direcciones IP del pod
StatefulSet	k8s.cl_<cluster-name>.ns_<namespace>.ss_<pod-name>	Direcciones IP del pod
Servicio	k8s.cl_<cluster-name>.ns_<namespace>.svc_<svc-name>	Direcciones IP del clúster Direcciones IP del pod
Servicio externo	k8s.cl_<cluster-name>.ns_<namespace>.exsvc_<svc-name>	Direcciones IP del servicio externo Direcciones IP de LoadBalancer

Etiquetas predefinidas	Formato de etiqueta en Panorama	Dirección IP recopilada
Nodos	k8s.cl_<cluster-name>.nodes	Direcciones IP privadas de todos los nodos
Nodos externos	k8s.cl_<cluster-name>.ex_nodes	Direcciones IP públicas de todos los nodos
Nombre de espacio	k8s.cl_<cluster-name>.ns_<namespace>	Direcciones IP de todos los clústeres en el nombre de espacio Direcciones IP de todos los pods en el nombre de espacio
Interface (Interfaz)	• k8s.cl_<cluster-name>.ns_<namespace>.ds_<daemonset-name>.if_<interface> • k8s.cl_<cluster-name>.ns_<namespace>.rs_<replicaset-name>.if_<interface> • k8s.cl_<cluster-name>.ns_<namespace>.ss_<statefulset-name>.if_<interface>	Todas las direcciones IP de todas las interfaces de cada pod de la implementación.

Si utiliza etiquetas para organizar los pods y los servicios dentro del clúster de Kubernetes, el complemento de Kubernetes en Panorama puede consultar estas etiquetas y crear etiquetas para usted. Las siguientes etiquetas definidas por usuarios son compatibles:

Etiquetas definidas por el usuario	Formato de etiqueta en Panorama	Dirección IP recopilada
Label	k8s.cl_<cluster-name>.ns_<namespace>.<label-key>.<label-value>	Todas las direcciones IP del clúster en ese espacio de nombre que coincide con la etiqueta específica. Todas las direcciones IP del pod en el espacio de nombre que coincide con la etiqueta específica.
Selector de etiqueta	k8s.cl_<cluster-name>.<selector-name>	Todas las direcciones IP del clúster que coinciden con el selector específico.

Etiquetas definidas por el usuario	Formato de etiqueta en Panorama	Dirección IP recopilada
		Todas las direcciones IP del pod que coinciden con el selector específico.

El selector de etiquetas hace coincidir la etiqueta especificada con los pods y servicios dentro del clúster de Kubernetes y asigna las direcciones IP asociadas con la etiqueta a una única etiqueta. El complemento de Kubernetes admite selectores basados en conjuntos y basados en igualdad para la clave de etiqueta y el valor de etiqueta.

Los siguientes selectores basados en igualdad son compatibles:

- `key = value; key ==`
- `value; key != value, por ejemplo, app = redis`

También puede especificar varios selectores en una expresión como una lista separada por comas. Por ejemplo:

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

Los siguientes selectores basados en conjuntos son compatibles:

- key in (value1, value2)
- key notin (value1, value2), por ejemplo, tier notin (frontend, backend)
- key
- !key

Para los objetos de servicio supervisados, el complemento genera puertos para los objetos de servicio puerto, targetPort y nodePort utilizando el siguiente esquema de nomenclatura:

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

El hash garantiza que, incluso si tiene espacios de nombres y nombres de servicios superpuestos en los clústeres de k8s, los objetos de servicio son únicos.

Habilitación de la inspección del tráfico de VLAN etiquetado

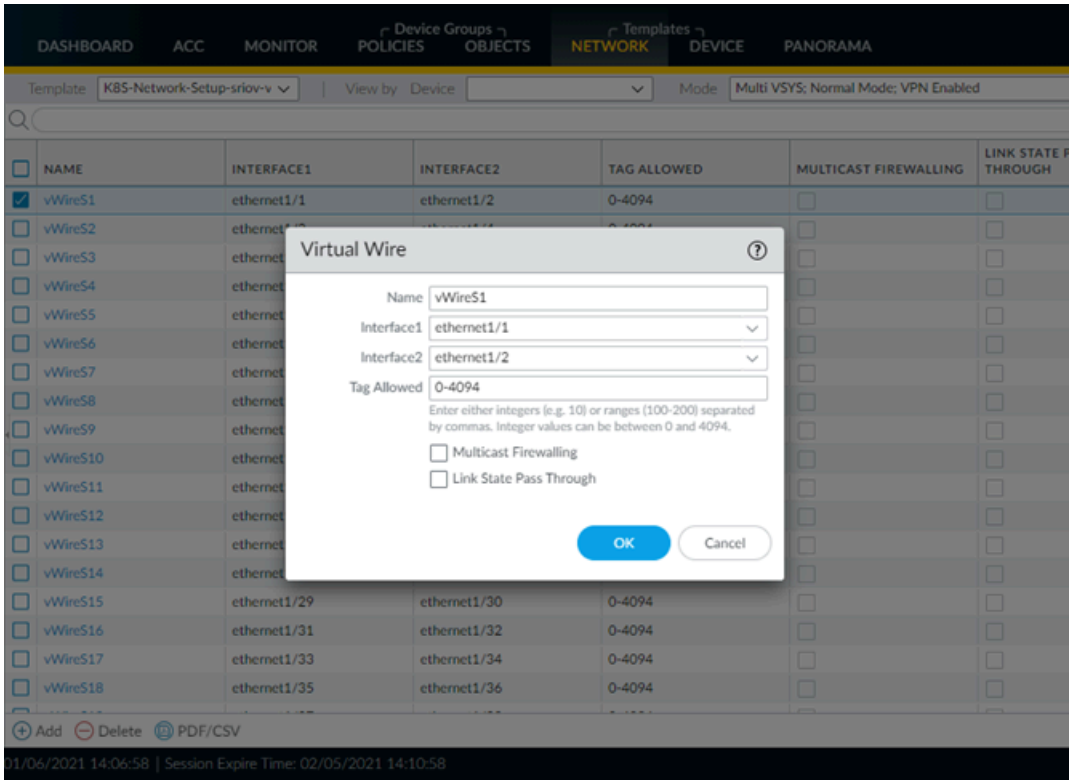
¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

Complete el siguiente procedimiento para permitir que el cortafuegos CN-Series inspeccione el tráfico de VLAN etiquetado. Para inspeccionar el tráfico etiquetado de VLAN, debe actualizar la configuración de todos los cables virtuales en Panorama, para permitir todas las etiquetas de VLAN. Luego, debe anotar el archivo YAML del pod de la aplicación para asignar etiquetas VLAN a las interfaces del pod de la aplicación. Esta anotación le dice al CN-NGFW qué etiquetas se aplican a los paquetes que se envían a través del cortafuegos.

El etiquetado de VLAN doble no es compatible.

STEP 1 | Habilite todas las VLAN en todas las interfaces de CN-NGFW.

1. Inicie sesión en Panorama.
2. Seleccione **Network (Red)** > **Virtual Wires (Cables virtuales)**.
3. Seleccione la plantilla **K8S-Network-Setup** del menú desplegable **Template (Plantilla)**.
4. Seleccione el primer cable virtual.
5. Establezca la **Tag Allowed (Etiqueta permitida)** en 0-4094.
6. Repita este procedimiento para cada cable virtual.
7. **Commit (Confirmar)** los cambios.



STEP 2 | Añada el archivo YAML del pod de la aplicación con las siguientes anotaciones para aplicar un ID de VLAN estático por interfaz.

Solo se admite una etiqueta VLAN por interfaz.

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 } ]'
```


Habilitación IPVLAN

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.1.x or above Container Images • Panorama con PAN-OS 10.1.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

IPVLAN es un controlador para un dispositivo de red virtual que se puede utilizar en un entorno en contenedores para acceder a la red de host. En el modo L2, IPVLAN muestra una dirección MAC única a la red externa, independientemente del número de dispositivos IPVLAN creados dentro de la red de host. Todas las interfaces IP lógicas utilizan la misma dirección MAC. Esto le permite evitar el uso del modo promiscuo en la NIC principal y evita posibles limitaciones de MAC en la NIC o el conmutador.

Ahora puede utilizar IPVLAN con el cortafuegos CN-Series con las siguientes limitaciones.

- Requiere PAN-OS 10.1.2 y posterior
- Solo IPv4
- Solo modo L2
- Una dirección IP por interfaz
- Si está utilizando Multus, implemente **`pan-cni-multus.yaml`** en lugar de **`pan-cni.yaml`**. Además, debe implementar pan-cni-net-attach-def.yaml en cada espacio de nombres donde se implementen los pods de aplicación de Multus.

La comunicación de la interfaz secundaria IPVLAN en el mismo host (que comparte la misma interfaz principal) no funciona.

Debe anotar los archivos yaml del pod de la aplicación para habilitar IPVLAN; no se requieren cambios en ningún archivo yaml del CN-Series para habilitar IPVLAN. El siguiente es un ejemplo de una definición de conexión de red para IPVLAN. Tenga en cuenta que el modo está establecido en **`"l2"`**. El cortafuegos CN-Series solo admite el modo L2.

```
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" } ] } }'
```

Desinstalación del complemento de Kubernetes en Panorama

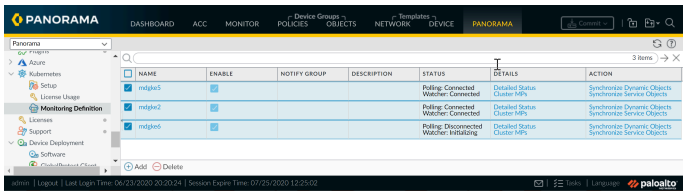
Utilice el siguiente flujo de trabajo para desinstalar el complemento de Kubernetes en Panorama, de forma que pueda devolver correctamente todos los tokens a los servidores de licencias de Palo Alto Networks y luego borrar el código de autenticación. Este flujo de trabajo le permite asegurarse de que los tokens estén disponibles para su uso en otro Panorama. Si ha implementado su servidor de gestión de Panorama en una configuración de alta disponibilidad, debe completar los pasos en el Panorama activo-principal antes de pasar al par pasivo- principal de Panorama.

STEP 1 | Inicie sesión en su par de Panorama activo-principal, si se ha implementado en una configuración de alta disponibilidad.

1. Elimine toda la configuración del clúster del complemento.

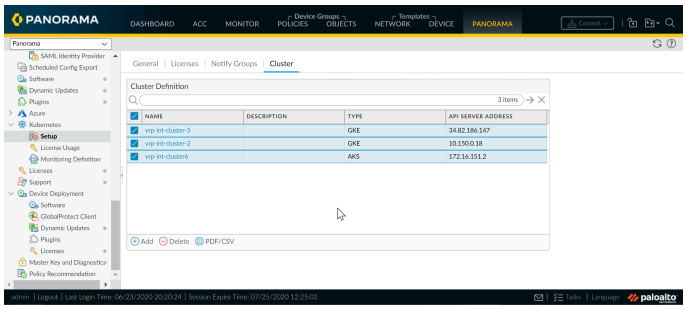
   1. Elimine la definición de supervisión.

   Seleccione **Plugins (Complementos)** > **Kubernetes** > **Monitoring Definition (Definición de supervisión)**, seleccione las definiciones de supervisión y **Delete (Eliminar)**.

   2. Elimina las definiciones de clúster de Kubernetes.

   Seleccione **Plugins (Complementos)** > **Kubernetes** > **Setup (Configuración)** > **Cluster (Clúster)**, seleccione las definiciones de clúster y luego **Delete (Eliminar)**.

2. Confirme los cambios realizados en Panorama.

   Commit (Confirmar) > **Commit to Panorama (Confirmar en Panorama)**.

3. Compruebe que el recuento de tokens usados sea cero.

   Para confirmar que todos los tokens se devuelven al servidor de licencias.

4. Realice un borrado de código de autorización y asegúrese de que el código de autorización de la columna de licencia sea Ninguno.

5. Elimine la configuración y confirme sus cambios.

   1. Seleccione **Plugins (Complementos)** y encuentre la versión del complemento de Kubernetes que ha instalado y seleccione **Remove config (Eliminar configuración)**.

   2. **Commit (Confirmar)** > **Commit to Panorama (Confirmar en Panorama)**.

6. Desinstale el complemento de Kubernetes.

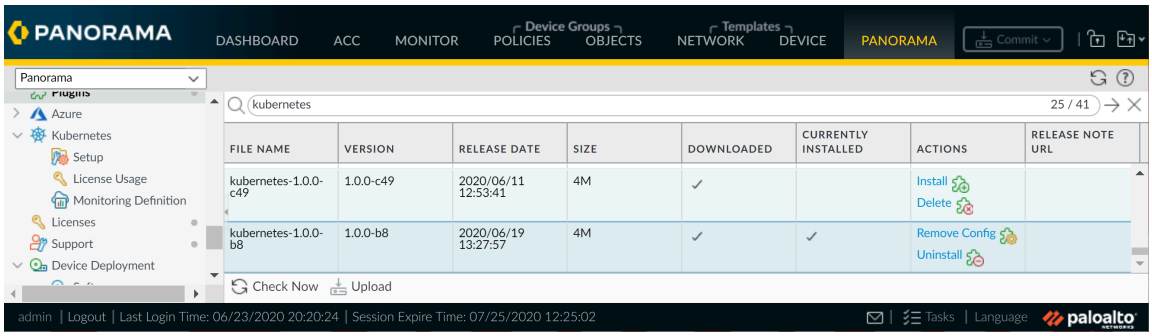
7. Suspenda el peer activo de Panorama.

   Seleccione **Panorama** > **High Availability (Alta disponibilidad)** y, a continuación, haga clic en el enlace **Suspend local Panorama (Suspender Panorama local)**, en la sección Comandos de operación.

STEP 2 | Inicie sesión en su otro par de Panorama.

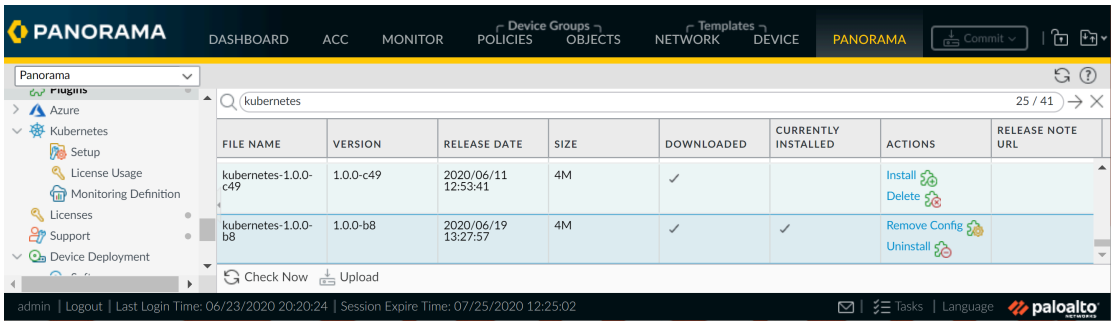
Ahora este par es el par activo-secundario.

1. Seleccione **Plugins (Complementos)** y encuentre la versión del complemento de Kubernetes que ha instalado y seleccione **Remove config (Eliminar configuración)**.



2. Desinstale el complemento.

   1. Seleccione **Plugins (Complementos)** y encuentre la versión del complemento de Kubernetes que ha instalado y proceda a **Uninstall (Desinstalar)**.



   2. Compruebe que la desinstalación se haya realizado correctamente.

Borre el código de autorización para los cortafuegos CN-Series en Panorama

Utilice la solución alternativa que se indica a continuación solo si eliminó la configuración del complemento y confirmó los cambios antes de borrar el código de autorización. Esta solución alternativa le permite devolver los tokens al servidor de licencias para que pueda usarlo en otro dispositivo Panorama.

STEP 1 | 1. Añada un nuevo usuario de complemento y confirme sus cambios.

1. Seleccione **Panorama** > **Administrators (Administradores)**.
2. **Add (Añadir)** un nuevo usuario llamado **`__kubernetes`**.
3. **Commit (Confirmar)** > **Commit to Panorama (Confirmar en Panorama)**.

STEP 2 | Borre el código de autorización en Panorama.

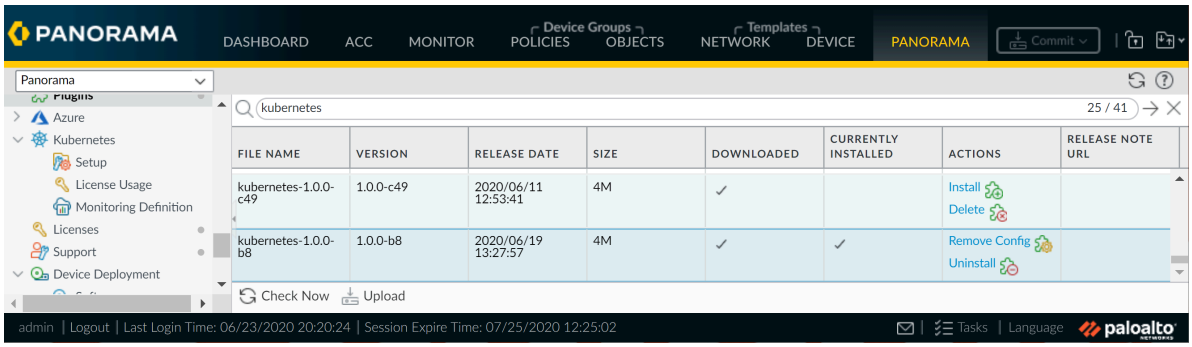
1. Seleccione **Panorama** > **Plugins (Complementos)** > **Kubernetes** > **Setup (Configuración)** > **Licenses (Licencias)**.
2. Seleccione **Activate/update using authorization code (Activar/actualizar utilizando el código de autorización)** y **Clear Auth Code (Borrar código de autorización)**.
3. Compruebe que la columna de licencia muestra el código de autorización **None (Ninguno)**.

STEP 3 | Elimine el usuario del complemento **__kubernetes** que creó en el Paso 1.

STEP 4 | Confirme los cambios.

STEP 5 | Desinstale el complemento.

1. Seleccione **Plugins (Complementos)** y encuentre la versión del complemento de Kubernetes que ha instalado y proceda a **Uninstall (Desinstalar)**.



2. Compruebe que la desinstalación se haya realizado correctamente.

Funciones no compatibles con el modelo CN-Series

Las siguientes capacidades compatibles con PAN-OS no están disponibles para la CN-Series a menos que se indique lo contrario a continuación:

Función	DaemonSet	Servicio K8s	Modo CNF	Modo HSF
Autenticación	No	No	No	No
Envío de logs a Cortex Data Lake	No	No	No	No
DLP empresarial	No	No	No	No
Interfaces que no son de vWire	No	No	**Sí**	**Sí**
IoT Security (Seguridad de IoT)	No	No	No	No
IPv6	**Sí**	No	**Sí**	No
NAT	No	No	**Sí**	No
Reenvío basado en políticas	No	No	**Sí**	No
QoS	No	No	No	No
SD-WAN	No	No	No	No
User-ID	No	No	**Sí**	No
WildFire Inline ML	No	No	No	No
SaaS en línea	No	No	No	No
IPSec	No	No	No	No
Inspección del contenido del túnel	No	No	No	No

Alta disponibilidad y soporte DPDK para cortafuegos CN-Series

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

La alta disponibilidad (HA) es una configuración en la que dos cortafuegos se colocan en un grupo y su configuración se sincroniza para prevenir el fallo de un único punto en su red. La conexión de heartbeat entre los peers del cortafuegos garantiza una conmutación por error sin problemas en el caso de que falle un peer. La configuración de los cortafuegos en un clúster de dos dispositivos proporciona redundancia y le permite garantizar la continuidad de la actividad empresarial.

En este capítulo se abordan las siguientes secciones:

- Soporte de alta disponibilidad para el cortafuegos CN-Series como CNF para Kubernetes
- Alta disponibilidad para el cortafuegos CN-Series en AWS
- Configurar DPDK en el cortafuegos CN-Series

Soporte de alta disponibilidad para el cortafuegos CN-Series como CNF para Kubernetes

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

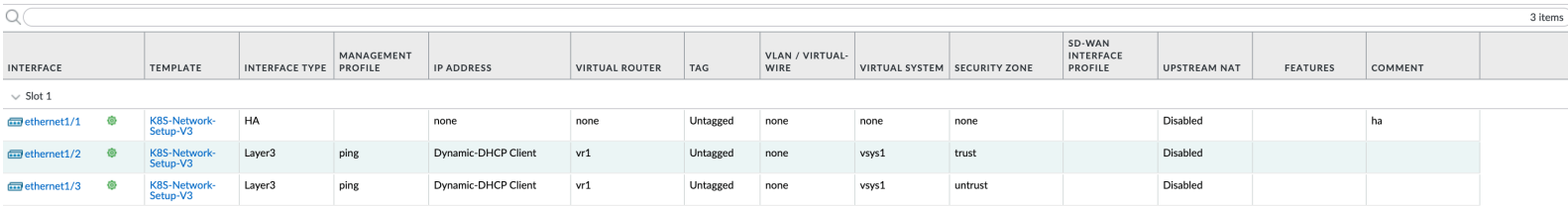
La alta disponibilidad (HA) es una configuración en la que dos cortafuegos se colocan en un grupo y su configuración se sincroniza para prevenir el fallo de un único punto en su red. La conexión de heartbeat entre los peers del cortafuegos garantiza una conmutación por error sin problemas en el caso de que falle un peer. La configuración de los cortafuegos en un clúster de dos dispositivos proporciona redundancia y le permite garantizar la continuidad de la actividad empresarial.

Ahora puede implementar CN-Series-as-a-Kubernetes-CNF en HA. Este modo de implementación solo admite alta disponibilidad activo/pasivo con sincronización de sesión y configuración.

Cuando implemente CN-Series-as-a-kubernetes-CNF en HA, habrá tres archivos PAN-CN-MGMT-CONFIGMAP, PAN-CN-MGMT y PAN-CN-NGFW YAML, cada uno para nodos activos y pasivos.

Para implementar correctamente el cortafuegos CN-Series como un CNF de Kubernetes en HA con soporte de capa 3:

- En HA, cada nodo de Kubernetes debe tener al menos tres interfaces: Gestión(predeterminada), interfaz HA2 e interfaz de datos.
- Para el cortafuegos CN-Series en modo L3, debe haber al menos dos interfaces: Gestión (por defecto) y la interfaz de datos.



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		

- Modifique los nuevos archivos YAML de definición de datos adjuntos de red con los siguientes cambios:
  - Asegúrese de que el valor del parámetro `PAN_HA_SUPPORT` sea **`verdadero`** en los siguientes archivos YAML:

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

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

  - Recupere el valor **`pciBusID`** de la interfaz del hipervisor ejecutando el siguiente comando:

    ```
    ethtool -i interface name
    ```

    Añada el valor de **`pciBusID`** recuperado anteriormente a los siguientes archivos de definición de red:

    ```
    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
    ```

  - Recupere la dirección IP estática de la interfaz HA2 de la instancia del nodo correspondiente en la consola de AWS y añádala al parámetro de `dirección` del archivo `net-attach-def-ha2-0.yaml` y `net-attach-def-ha2-1.yaml`.

Si está usando **Advanced Routing (Enrutamiento avanzado)**, tenga en cuenta que los cortafuegos CN-Series implementados en modo CNF solo son compatibles con entornos EKS e in situ. Si está utilizando **Advanced Routing (Enrutamiento avanzado)**con el complemento Kubernetes 3.0.0, debe configurarlo manualmente en la pila de plantillas; en el archivo `pan-cn-mgmt-console.yaml`, establezca el indicador `PAN_ADVANCED_ROUTING:”true”.`

Alta disponibilidad para el cortafuegos CN-Series en AWS

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

Ahora puede implementar CN-Series-as-a-Kubernetes-CNF en HA. Este modo de implementación solo admite alta disponibilidad activo/pasivo con sincronización de sesión y configuración.

La implementación CN-Series-as-a-Kubernetes CNF en HA con IPV6 no es compatible con el entorno AWS.

Para asegurar la redundancia, puede implementar cortafuegos CN-Series en AWS en una configuración de alta disponibilidad (HA) activo/pasivo. El peer activo sincroniza continuamente su información de sesión y configuración con el peer pasivo configurado de manera idéntica. Una conexión de latido entre los dos dispositivos garantiza la conmutación por error si falla el dispositivo activo. Puede implementar el cortafuegos CN-Series en AWS EKS en HA a través del movimiento de IP secundaria.

Para asegurarse de que todo el tráfico a las aplicaciones orientadas a Internet pasa a través del cortafuegos, puede configurar el enrutamiento de entrada de AWS. La capacidad de enrutamiento de entrada de AWS le permite asociar tablas de ruta con la puerta de enlace de Internet de AWS y añadir reglas de ruta para redirigir el tráfico de la aplicación a través del cortafuegos CN-Series. Esta redirección garantiza que todo el tráfico de Internet pase a través del cortafuegos sin tener que reconfigurar los endpoints de la aplicación.

Movimiento secundario

Cuando el par activo deja de funcionar, el par pasivo detecta este fallo y se activa. Además, activa las llamadas de la API a la infraestructura de AWS para mover las direcciones IP secundarias configuradas desde las interfaces del plano de datos del par que ha fallado a sí mismo. Además, AWS actualiza las tablas de rutas para garantizar que el tráfico se dirija a la instancia de cortafuegos activa. Estas dos operaciones garantizan que las sesiones de tráfico de entrada y saliente se restauren después de la conmutación por error. Esta opción le permite aprovechar DPDK para mejorar el rendimiento de sus instancias de cortafuegos CN-Series.

Funciones de IAM para HA

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores

¿Dónde puedo usar esto?	¿Qué necesito?
	• Helm 3.6 or above version client para la implementación de la CN-Series con Helm

AWS requiere que todas las solicitudes de API se firmen criptográficamente con las credenciales emitidas por ellos. Para habilitar los permisos de API para cortafuegos CN-Series que se implementarán como pareja HA, debe crear una política y vincular esa política a una función en el Servicio de gestión de identidad y acceso (IAM) de AWS. La función debe vincularse a los cortafuegos CN-Series al iniciarse. La política otorga a la función de IAM permisos para iniciar las acciones de API necesarias para mover interfaces o direcciones IP secundarias del par activo al par pasivo cuando se activa la conmutación por error.

Para instrucciones detalladas sobre la creación de políticas, consulte la documentación de AWS sobre la Creación de políticas gestionadas del cliente. Para obtener instrucciones detalladas sobre cómo crear una función de IAM, cómo definir qué cuentas o servicios de AWS pueden asumir la función, cómo definir qué acciones y recursos de API puede usar la aplicación luego de asumir la función, consulte la documentación de AWS sobre Funciones de IAM para Amazon EC2.

La función de IAM, que se configura en la consola de AWS, debe tener los permisos para las siguientes acciones y recursos (como mínimo):

Las siguientes acciones, permisos y recursos de IAM son necesarios para habilitar la HA.

Acción, permiso o recurso de IAM	Description (Descripción)	Movimiento de IP secundaria
AttachNetworkInterface	Para obtener permiso para adjuntar una ENI a una instancia.	✓
DescribeNetworkInterfaces	Para obtener los parámetros de ENI con el fin de adjuntar una interfaz a la instancia.	✓
DetachNetworkInterface	Para obtener permiso para desconectar la ENI de la instancia EC2.	✓
DescribeInstances	Para dar permiso para obtener información sobre las instancias de EC2 en la VPC.	✓
AssociateAddress	Para obtener permisos para mover direcciones IP públicas asociadas con las direcciones IP primarias de la interfaz pasiva a la activa.	✓
AssignPrivateIpAddresses	Para permisos para asignar direcciones IP secundarias y direcciones IP públicas asociadas a interfaces en el par pasivo.	✓
DescribeRouteTables	Para obtener permiso para recuperar todas las tablas de enrutamiento asociadas a las instancias del cortafuegos CN-Series.	✓

Acción, permiso o recurso de IAM	Description (Descripción)	Movimiento de IP secundaria
ReplaceRoute	Para obtener permisos para actualizar las entradas de la tabla de enrutamiento de AWS.	✓
GetPolicyVersion	Para obtener permiso para recuperar información sobre la versión de la política de AWS.	✓
GetPolicy	Para obtener permiso para recuperar información sobre políticas de AWS.	✓
ListAttachedRolePolicies	Para obtener permiso para recuperar la lista de todas las políticas administradas adjuntas a un rol de IAM específico.	✓
ListRolePolicies	Para obtener permiso para recuperar una lista de los nombres de las políticas en línea integradas en un rol de IAM específico.	✓
GetRolePolicy	Para obtener permiso para recuperar una política en línea especificada incrustada en una función de IAM especificada.	✓
política	Para obtener permiso para acceder al Nombre de recurso de Amazon (ARN) de la política de IAM.	✓
función	Para obtener permiso para acceder al ARN de las funciones de IAM.	✓
tabla de enrutamiento	Para obtener permiso para acceder a la tabla de enrutamiento Nombre de recurso de Amazon (ARN) para actualizarla tras una conmutación por error.	✓
Comodín (*)	En el campo ARN use * como comodín.	✓

La siguiente captura de pantalla muestra la configuración de gestión de acceso para la función de IAM descrita anteriormente para el HA de IP secundaria:

Los permisos mínimos que necesita para la HA de movimiento de la IP secundaria son:
{"Versión": "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/*"}]}

Enlaces de HA

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

Los cortafuegos en una pareja HA usan enlaces de HA para sincronizar datos y mantener la información de su estado. En AWS, el cortafuegos CN-Series utiliza los siguientes puertos:

- **Enlace de control**: El enlace de HA1 se utiliza para intercambiar saludos, heartbeats, e información de estado de HA, así como la sincronización del plano de gestión para el enrutamiento. Este enlace también se utiliza para sincronizar cambios de configuración en el dispositivo activo o pasivo con su peer.

  El puerto de gestión se utiliza para HA1. Puertos TCP 28769 y 28260 para una comunicación con texto claro; puerto 28 para una comunicación cifrada (SSH sobre TCP).
- **Enlace de datos**: El enlace de HA2 se utiliza para sincronizar sesiones, reenviar tablas, asociaciones de seguridad de IPSec y tablas de ARP entre dispositivos de un par de HA. El flujo de datos del enlace de HA2 siempre es unidireccional (excepto en la conexión persistente de HA2); fluye desde el dispositivo activo al dispositivo pasivo.

  Ethernet1/1 debe asignarse como el enlace HA2; esto es necesario para implementar el cortafuegos CN-Series en AWS en HA. El enlace de datos de HA puede configurarse para utilizar IP (número de protocolo 99) o UDP (puerto 29281) como transporte.

El cortafuegos CN-Series en AWS no admite los enlaces de backup para HA1 ni HA2.

Sondeos de heartbeat y mensajes de saludo

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

Los cortafuegos utilizan mensajes de saludo y latido para comprobar que el peer responde y está operativo. Los mensajes de saludo se envían desde un peer al otro en el *intervalo de saludo* configurado para verificar el estado del dispositivo. El heartbeat es un ping ICMP para el peer de HA a través del enlace de control y el peer responde al ping para establecer que los dispositivos están conectados y responden. Si desea información detallada sobre los temporizadores de HA que activan una conmutación, consulte

Temporizadores de HA. (Los temporizadores de HA para el cortafuegos CN-Series son los mismos que para los cortafuegos PA-5200 Series).

Prioridad y preferencia de dispositivos

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores • Helm 3.6 or above version client para la implementación de la CN-Series con Helm

A los dispositivos de un par de HA se les puede asignar un valor de *prioridad de dispositivo* para indicar una preferencia por el dispositivo que debería asumir el papel activo y gestionar el tráfico en una conmutación por error. Si necesita utilizar un dispositivo específico del par de HA para proteger de manera activa el tráfico, debe habilitar el comportamiento de preferencia en ambos cortafuegos y asignar un valor de prioridad de dispositivo para cada dispositivo. El dispositivo con el valor numérico más bajo y, por lo tanto, *mayor prioridad*, se designará como activo y gestionará todo el tráfico de la red. El otro dispositivo estará en un estado pasivo y sincronizará información de configuración y estado con el dispositivo activo, de manera que esté listo para pasar al estado activo en el caso de producirse un fallo.

El valor numérico más bajo se activa durante la primera implementación. Si el valor numérico más alto se implementa primero y la prioridad está deshabilitada, entonces el valor numérico más alto se activará.

No se recomienda el derecho a la preferencia para HA en el cortafuegos CN-Series en AWS.

De forma predeterminada, el derecho a la preferencia está deshabilitado en los cortafuegos. Cuando se habilita, el comportamiento de preferencia permite que el cortafuegos con la *mayor prioridad* (valor numérico más bajo) vuelva a estar activo cuando se recupere de un fallo. Cuando se produce una preferencia, el evento se registra en los logs del sistema.

Para añadir prioridad, debe asegurarse de que el valor del parámetro `PAN_HA_PRIORITY` se establezca en un valor numérico en los archivos `pan-cn-mgmt-configmap-0.yaml` y `pan-cn-mgmt-configmap-1.yaml`.

Por ejemplo:

```
PAN_HA_PRIORITY: “10”
```

Temporizadores de HA

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores

¿Dónde puedo usar esto?	¿Qué necesito?
	• Helm 3.6 or above version client para la implementación de la CN-Series con Helm

Los temporizadores de alta disponibilidad (HA) se utilizan para detectar un fallo de cortafuegos y activar una conmutación por error. Para reducir la complejidad al configurar temporizadores de HA, puede seleccionar uno de los tres perfiles: **Recommended (Recomendado)**, **Aggressive (Agresivo)** y **Advanced (Avanzado)**. Estos perfiles cumplimentan automáticamente los valores óptimos del temporizador de HA para la plataforma de cortafuegos específica con el fin de habilitar una implementación de HA más rápida.

Utilice el perfil **Recommended (Recomendado)** para ajustes comunes del temporizador de conmutación por error y el perfil **Aggressive (Agresivo)** para ajustes más rápidos del temporizador de conmutación por error. El perfil **Advanced (Avanzado)** le permite personalizar los valores del temporizador para que se adapten a sus requisitos de red.

Temporizador de HA en CN-Series en AWS	Valores predeterminados para los perfiles Recomendado/ Agresivo
Tiempo de espera de promoción	2000/500 ms
Intervalo de saludo	8000/8000 ms
Intervalo de heartbeat	2000/1000 ms
Número máximo de flaps	3/3
Tiempo de espera para ser preferente	1/1 min
Tiempo de espera activo tras fallo de supervisor	0/0 ms
Tiempo de espera ascendente principal adicional	500/500 ms

Configuración de HA activa/pasiva en AWS EKS mediante una IP secundaria

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores

Complete el siguiente procedimiento para implementar nuevos cortafuegos CN-Series como un par de HA con direcciones IP secundarias.

STEP 1 | Antes de implementar los cortafuegos CN-Series para su par HA, asegúrese de lo siguiente:

- Implemente los pareja de HA en la misma zona de disponibilidad de AWS. Consulte Funciones de IAM para HA.
- Cree una función de IAM y asígnelo a los nodos de trabajo que ejecutan los cortafuegos CN-Series al implementar las instancias.
- Los cortafuegos activos y pasivos deben tener al menos tres interfaces cada uno: interfaz de gestión, interfaz HA2 e interfaz de datos.

  De forma predeterminada, la interfaz de gestión se utilizará como interfaz HA1.
- Cree interfaces de red en AWS en la misma zona de disponibilidad que el clúster. Añada una etiqueta en el `eni` para que AWS no lo gestione y pueda ser utilizado por multus:

  `node.k8s.amazonaws.com/no_manage: Verdadero`
- Compruebe que los componentes de red y seguridad están definidos adecuadamente.
  - Habilite la comunicación con internet. La VPC predeterminada incluye una puerta de enlace de internet, y si instala el cortafuegos CN-Series en la subred predeterminada, tendrá acceso a internet.
  - Cree subredes. Las subredes son segmentos de intervalos de direcciones IP asignados a la VPC en las que puede iniciar las instancias de EC2. El cortafuegos CN-Series debe pertenecer a la subred pública para que se pueda configurar su acceso a internet.
  - Cree un grupo de seguridad de datos que incluya las interfaces de datos del cortafuegos. Además, configure la seguridad para permitir todo el tráfico, de modo que los cortafuegos apliquen la seguridad. Esto es necesario para mantener las sesiones existentes durante la conmutación por error.
  - Añada rutas a la tabla de enrutamiento para una subred privada para garantizar que el tráfico se puede dirigir a través de subredes y grupos de seguridad en la VPC, según corresponda.

Al implementar el cortafuegos CN-Series en EKS, se produce un error en la recuperación de tokens IMDSv2 si el valor `http-put-response-hop-limit` se establece en el valor predeterminado de 1. Debe asegurarse de que el valor límite de salto esté establecido en 3 o más cuando IMDSv2 esté habilitado.

Por ejemplo:

Ejecute el siguiente comando:

```
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 | Implementación de los cortafuegos CN-Series en EKS.

1. Configure ethernet 1/1 como la interfaz HA2 en cada par de HA.
   1. Abra la consola de Amazon EC2.
   2. Seleccione Interfaz de red y luego seleccione su interfaz de red.
   3. Seleccione **Actions (Acciones)** > **Manage IP Addresses (Gestionar direcciones IP).**
   4. Deje el campo en blanco para permitir que AWS asigne una dirección IP dinámicamente o introduzca una dirección IP dentro del rango de subred para el cortafuegos CN-Series. Esto asignará una IP secundaria a la interfaz HA2.
   5. Haga clic en **Yes (Sí)** y **Update (Actualizar)**.
   6. Seleccione **Actions (Acciones)** > **Cambiar origen/destino. Marque** y seleccione **Disable (Desactivar)**.
   7. Repita este proceso en el segundo par de HA (para que sea pasivo).
2. Añada una dirección IP secundaria a las interfaces de su plano de datos en el par de HA (para que sea activo).
   1. Seleccione **Network Interface (Interfaz de red)** y luego seleccione su interfaz de red.
   2. Seleccione **Actions (Acciones)** > **Manage IP Addresses (Gestionar direcciones IP)** > **IPv4 Addresses (Direcciones IPv4)** > **Assign new IP (Asignar nueva IP)**.
   3. Deje el campo en blanco para permitir que AWS asigne una dirección IP dinámicamente o introduzca una dirección IP dentro del rango de subred para el cortafuegos CN-Series.
   4. Haga clic en **Yes (Sí)** y **Update (Actualizar)**.
3. Asocie una dirección IP elástica secundaria (pública) con la interfaz no trust del par activo.
   1. Seleccione **Elastic IPs (IP elásticas)** y luego elija la Dirección de IP elástica para asociar.
   2. Seleccione **Actions (Acciones)** > **Associate Elastic IP (Asociar IP elástica)**.
   3. En **Resource Type (Tipo de recurso)**, seleccione **Network Interface (Interfaz de red)**.
   4. Seleccione la interfaz de red con la que asociar la Dirección IP elástica.
   5. Haga clic en **Associate (Asociar)**.
4. Para la inspección del tráfico saliente, añada una entrada a la tabla de enrutamiento de subred que establezca el siguiente salto como la interfaz trust del cortafuegos.
   1. Seleccione **VPC** > **Route Tables (Tablas de enrutamiento)**.
   2. Elija su tabla de enrutamiento de subred.
   3. Seleccione **Actions (Acciones)** > **Edit routes (Editar rutas)** > **Add route (Añadir ruta)**.
   4. Introduzca el bloque CIDR o la dirección IP de **Destination (Destino)**.
   5. Para **Target (Destino)**, introduzca la interfaz de red de la interfaz trust del cortafuegos.
   6. Haga clic en **Save routs (Guardar rutas)**.
5. Para utilizar AWS Ingress Routing, cree una tabla de enrutamiento y asóciela a la puerta de enlace de Internet. Luego añada una entrada con el siguiente salto configurado como la interfaz untrust del cortafuegos activo.
   1. Seleccione **Route Tables (Tablas de enrutamiento)** > **Create Route Table (Crear tabla de enrutamiento)**.

2. (Opcional) Introduzca una **Name tag (Etiqueta de nombre)** descriptiva para su tabla de enrutamiento.
3. Haga clic en **Create (Crear)**.
4. Haga clic en su tabla de enrutamiento y seleccione **Actions (Acciones)** > **Edit edge associations (Editar asociaciones perimetrales)**.
5. Seleccione **Internet gateways (Puertas de enlace de Internet)** y elija su puerta de enlace de Internet de VPC.
6. Haga clic en **Save (Guardar)**.
7. Haga clic en su tabla de enrutamiento y seleccione **Actions (Acciones)** > **Edit routes (Editar rutas)**.
8. Para el **Target (Destino)**, seleccione **Network Interface (Interfaz de red)** y elija la interfaz untrust del cortafuegos activo.
9. Haga clic en **Save routs (Guardar rutas)**.

STEP 3 | Habilitar HA.

Para habilitar la compatibilidad con alta disponibilidad, debe asegurarse de que el valor del parámetro PAN_HA_SUPPORT es verdadero en los siguientes archivos YAML:

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

La dirección IP HA1 del mismo nivel se configura automáticamente.

STEP 4 | Recupere la dirección IP estática de la interfaz HA2 de la instancia de nodo correspondiente en la consola de AWS y añádala al parámetro de dirección del archivo `net-attach def-ha2-0.yaml` y `net-attach-def-ha2-1.yaml`.

(Opcional) Modifique el **Threshold (Umbral)** para los paquetes **HA2 Keep-alive (Conexión persistente de HA2)**. De manera predeterminada, **HA2 Keep-alive (Conexión persistente de HA2)** Conexión persistente de HA2) está habilitada para la supervisión del enlace de datos de HA2 entre los peers. Si ocurre un fallo y se supera este umbral (el valor predeterminado es 10000 ms), se producirá la acción definida. Se generará un mensaje de log de sistema crítico cuando se produzca un fallo de conexión persistente de HA2.

*Puede configurar la opción **HA2 keep-alive (Conexión persistente de HA2)** en ambos dispositivos o solamente un dispositivo del par de HA. Si habilita esta opción en un solo dispositivo, solamente ese dispositivo enviará los mensajes de conexión persistente.*

STEP 5 | Compruebe que los cortafuegos estén emparejados en alta disponibilidad activo/pasivo.

1. Acceda al **Panel** de ambos dispositivos y visualice el widget de Alta disponibilidad.
2. En el par de HA activo, haga clic en **Sincronizar con el par**.
3. Confirme que los cortafuegos estén emparejados y sincronizados.
   - En el cortafuegos pasivo: el estado del cortafuegos local debería mostrar **Pasivo** y la **Config. en ejecución** debería mostrar como sincronizada.
   - En el cortafuegos activo: el estado del cortafuegos local debe mostrar **Activo** y la **Config. en ejecución** debe aparecer como Sincronizada.
4. Desde la interfaz de línea de comandos del cortafuegos, ejecute los siguientes comandos:
   - Para verificar la disposición de la conmutación por error:

     ```
     show plugins vmw_series aws ha state
     ```
   - Para mostrar la asignación de IP secundaria:

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


Configurar DPDK en el cortafuegos CN-Series

¿Dónde puedo usar esto?	¿Qué necesito?
• CN-Series Implementación	• CN-Series 10.2.x or above Container Images • Panorama con PAN-OS 10.2.x o versiones superiores • Helm 3.6 or above version client

Data Plane Development Kit (DPDK) proporciona un marco simple para el procesamiento rápido de paquetes en aplicaciones de plano de datos.

El modo DPDK solo es compatible con el cortafuegos CN-Series como función de red de contenedores (CNF) de Kubernetes.

DHCP IPAM no es compatible con el modo DPDK.

Requisitos del sistema

Para ejecutar una aplicación DPDK, debe realizar las siguientes personalizaciones en el equipo de destino.

- **Configuración del kernel**: habilite la opción HUGETLBFS en el kernel del sistema operativo host.
- **KNI y UIO/VFIO**:inserte KNI y UIO/VFIO en el kernel del sistema operativo host.
- **Hugepages**
  1. **Reservar hugepages**
     - Reserve hugepages durante el tiempo de ejecución, antes de que se inicien los pods. Añada el número de hugepages necesarias en el archivo `nr_hugepages` en el directorio `/sys/kernel/` correspondiente a un tamaño de página específico (en KB). Por ejemplo, utilice el siguiente comando para un sistema de un único nodo si se requieren 1024 de 2 millones de páginas.

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

     - Reserve hugepages durante el arranque. Por ejemplo, para reservar una hugepage de memoria 4G como cuatro páginas 1G, las siguientes opciones deberán pasarse al núcleo.

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

  2. **Usar hugepages con DPDK**: cree puntos de montaje para hugepages porque PanOS 10.2 utiliza procesos secundarios dpDK.

     El siguiente es un comando de ejemplo para crear hugepage de 1 GB de tamaño para el uso de DPDK.

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

3. Reinicie el servicio kubelet en el host después de habilitar hugepages usando el siguiente comando.

```
sudo systemctl restart kubelet
```

4. Compruebe `/sys/fs/cgroup/hugetlb/kubepods.slice/hugetlb.2MB.limit_in_bytes` para asegurarse de que el tamaño coincida con el tamaño de

página enorme. Si el tamaño no coincide con el tamaño de hugepage, actualice el tamaño mediante el siguiente comando.

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

En un pod, las aplicaciones pueden asignar y consumir hugepages preasignadas de múltiples tamaños. Las aplicaciones consumen hugepages a través de los requisitos de recursos a nivel de contenedor utilizando el nombre de recurso hugepages-<size>. Por ejemplo, hugepages-2Mi o hugepages-1Gi.

A diferencia de la CPU o la memoria, hugepages no admiten la confirmación excesiva.

El modo privilegiado está habilitado para acceder al espacio del dispositivo del host. Para enumerar y enlazar dispositivos de red a un contenedor, monte `/sys` en el contenedor para que DPDK pueda acceder a los archivos del directorio.

A continuación se muestra un fragmento de código para habilitar hugepages en DPDK.

```
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
```


- **Anclaje de NUMA y CPU**: varios procesos DPDK no se pueden ejecutar en el mismo núcleo porque causan daños en la caché del grupo de memoria, entre otros problemas. Los procesos secundarios están fijados a un núcleo diferente. Utilice las opciones de anclaje de CPU en *configmap* para controlar los procesos secundarios.
- **Configuración y cambios de pods**
  - Habilite `PAN_DATA_MODE: "dpdk"` en `pan-cn-ngfw-configmap-0.yaml` y `pan-cn-ngfw-configmap-1.yaml`.

    DPDK no es el modo predeterminado para CN-Series-as-a-kubernetes-CNF.
  - Haga coincidir el parámetro `#HUGEPAGE_MEMORY_REQUEST` con la solicitud de memoria hugepage en `pan-cn-ngfw-configmap-0.yaml` y `pan-cn-ngfw-configmap-1.yaml`.

    Si la memoria hugepage no está disponible, el valor predeterminado es MMAP.

Para obtener más información, consulte Requisitos del sistema de DPDK.

Puede configurar DPDK en nodos de trabajo locales y clúster de AWS EKS

- Configurar DPDK en nodos de trabajo locales
- Configurar DPDK en AWS EKS

Configurar DPDK en nodos de trabajo locales

STEP 1 | Instale las siguientes dependencias:

Ejecute todos los comandos del nodo de trabajo en el que desea configurar DPDK.

- Para CentOS:

```
yum groupinstall 'Herramientas de desarrollo' -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
```

- Para el sistema operativo Ubuntu:

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


STEP 2 | Después de instalar las dependencias:

- Descargue el archivo tar de DPDK de https://fast.dpdk.org/rel/. Consulte Documentación de DPDK para los pasos de compilación.

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

- Descomprima el archivo.

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

- Compile el archivo. El archivo compilado estará en la subcarpeta `x86_64-native-linuxapp-gcc`

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

STEP 3 | Inserte el módulo del kernel compilado de forma estática o dinámica durante el tiempo de ejecución (modprobe/insmod). Para obtener más información, consulte el módulo del kernel.

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

En Ubuntu, si ves el error:`insmod: ERROR: no se pudo insertar el módulo igb_uio.ko`, insertar el módulo `uio` primero.

```
modprobe uio
```

STEP 4 | Utilice formas específicas de distribución para insertar módulos durante el arranque. De forma alternativa, puede crear un servicio que ejecute los comandos modprobe/insmod cada vez que se inicie el sistema.

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

STEP 5 | Active y monte 2M hugepages de tamaño 2048K.

También puede activar hugepages utilizando el script de servicio en el paso 4.

```
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 | Cree una instantánea de la máquina virtual (VM) para uso futuro.

Configurar DPDK en AWS EKS

En AWS EKS, cada pod tiene una interfaz de red asignada por el complemento CNI de Amazon VPC. Con Multus, puede crear pods con varias interfaces.

STEP 1 | Cree una cuenta de AWS si aún no tiene una.

STEP 2 | Cree un clúster EKS utilizando la AMI personalizada. Para obtener más información, consulte Creación de un clúster de Amazon EKS.

STEP 3 | Modifique la configuración de VPC y nodos. Para obtener más información, consulte la Documentación de AWS EKS.

STEP 4 | (Multus)Añada varios ENI al nodo EKS y cargue los controladores KNI y UIO.

- Utilice la siguiente etiqueta para añadir varios ENI al nodo EKS.

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

  Cuando se detecta la etiqueta, Multus CNI puede utilizar la interfaz. Para obtener más información, consulte la Documentación de AWS.

- Ejecute los siguientes comandos en la AWS CLI.

  ```
  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
  ```

- (Si no está utilizando la AMI personalizada) Habilite hugepages en los nodos de trabajo.

  ```
  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
  ```