paloalto
TECHDOCS

Modes de déploiement du pare-feu 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

Démarrage rapide – Déploiement du pare-feu CN-Series

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series à l'aide de Helm

Commencez avec le déploiement CN-Series en procédant comme suit :

1. Connectez-vous au compte CSP et activez les crédits.
2. Créez un profil de déploiement.
3. Installez un certificat de périphérique sur le pare-feu CN-Series.
4. Installez le plug-in Kubernetes et configurez Panorama pour CN-Series.
5. Téléchargez les fichiers de déploiement CN-Series à partir du référentiel GitHub de Palo Alto Networks. Récupérez les fichiers du dossier Native-k8s pour les utiliser avec les déploiements natifs Kubernetes sur site ou dans le cloud.
6. Déployer le CN-Series avec ou sans le référentiel Graphiques HELM.

   *Il est recommandé de déployer le pare-feu CN-Series à l'aide du **graphique HELM**.*

7. Configurer Panorama pour sécuriser un déploiement Kubernetes

Vous pouvez choisir de déployer votre pare-feu CN-Series dans les modes de déploiement suivants :

- Déployer le pare-feu CN-Series en tant que service Kubernetes (mode de déploiement recommandé)- Le pare-feu CN-Series est déployé dans un modèle de déploiement en cluster. Ce mode de déploiement utilise des capacités de mise à l'échelle automatique, améliore l'utilisation, réduit les coûts et augmente l'évolutivité avec des modèles de déploiement natifs basés sur Kubernetes.
- Déployer le pare-feu CN-Series en tant que DaemonSet- Le pare-feu CN-Series est déployé dans un modèle de déploiement distribué. Ce mode de déploiement est plus adapté lorsque vous disposez d'un nombre réduit de nœuds à sécuriser par environnement.
- Déployer le pare-feu CN-Series en tant que Kubernetes CNF- Ce mode de déploiement protège à la fois les charges de travail conteneurisées et non conteneurisées. Vous pouvez déployer en tant que déploiement de couche 3 autonome.

Modes de déploiement des pare-feu CN-Series

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Après avoir examiné les blocs de construction principaux CN-Series et la présentation générale du flux de travail dans Sécuriser les charges de travail Kubernetes avec le pare-feu CN-Series , vous pouvez commencer à déployer les pare-feu CN-Series pour sécuriser le trafic entre les conteneurs au sein du même cluster, ainsi qu'entre les conteneurs et d'autres types de charges de travail tels que les machines virtuelles et les serveurs bare-metal.

Si vous êtes dans l'environnement OpenShift, consultez Sécuriser la 5G avec le pare-feu CN-Series et pour la sécurisation du trafic 5G, consultez .

Vous avez besoin d'outils Kubernetes standard tels que kubectl ou Helm pour déployer et gérer vos applications, vos services pare-feu et vos clusters Kubernetes. Panorama n'est pas conçu pour être utilisé comme orchestrateur pour le déploiement et la gestion de clusters Kubernetes. Les modèles pour la gestion des clusters sont fournis par les fournisseurs de Kubernetes gérés. Palo Alto Networks fournit des modèles pris en charge par la communauté pour le déploiement CN-Series avec Helm et Terraform.

- Déployer le pare-feu CN-Series en tant que service Kubernetes (mode de déploiement recommandé)
- Déployer le pare-feu CN-Series en tant que DaemonSet
- Déployer le pare-feu CN-Series en tant que Kubernetes CNF
- Déployer Kubernetes CNF L3 en mode autonome

Avant de passer du déploiement de CN-Series en tant que DaemonSet à CN-Series en tant que service ou vice versa, vous devez supprimer et réappliquer `plugin-serviceaccount.yaml`.

- *Lorsque vous déployez CN-Series en tant que DaemonSet,* `pan-plugin-cluster-mode-secret` *ne doit pas exister.*
- *Lorsque vous déployez CN-Series en tant que service Kubernetes,* `pan-plugin-cluster-mode-secret` *doit être présent.*

Déployer le pare-feu CN-Series en tant que service Kubernetes (mode de déploiement recommandé)

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec graphique Helm

Effectuez la procédure suivante pour déployer le pare-feu CN-Series en tant que service Kubernetes.

Avant de commencer, assurez-vous que la version du fichier YAML CN-Series est compatible avec la version PAN-OS.

- PAN-OS 10.1.2 ou version ultérieure nécessite YAML 2.0.2
- PAN-OS 10.1.0 et 10.1.1 nécessitent YAML 2.0.0 ou 2.0.1

STEP 1 | Configurez votre cluster Kubernetes.

1. Vérifiez que le cluster dispose des ressources adéquates. Assurez-vous que ce cluster dispose des conditions préalables de CN-Series pour prendre en charge le pare-feu :

   `kubectl get nodes`

   `kubectl describe node <node-name>`

   Affichez les informations sous l'en-tête Capacity (Capacité) dans la sortie de la commande pour voir le processeur et la mémoire disponibles sur le nœud spécifié.

   L'allocation du processeur, de la mémoire et du stockage sur disque dépendra de vos besoins. Voir Performances et mise à l'échelle de CN-Series.

   Assurez-vous d'avoir les informations suivantes :

   - Collectez l'adresse IP du terminal pour configurer le serveur API sur Panorama. Panorama utilise cette adresse IP pour se connecter à votre cluster Kubernetes.
   - Collectez le nom de la pile de modèles, le nom du groupe d'appareils, l'adresse IP Panorama et éventuellement le nom du groupe du collecteur de journaux à partir de Panorama.
   - Collectez la clé d'authentification VM et l'ID et la valeur du code PIN d'enregistrement automatique.
   - L'emplacement du conteneur d'images dans lequel vous avez téléchargé les images.

STEP 2 | (facultatif) Si vous avez configuré un certificat personnalisé dans le plug-in Kubernetes pour Panorama, vous devez créer le secret de certificat en exécutant la commande suivante. Ne modifiez pas le nom de fichier de ca.crt. Le volume des certificats personnalisés dans pan-cn-mgmt.yaml et pan-cn-ngfw.yaml est facultatif.

```
kubectl -n kube-system crée un secret générique custom-ca --from-file=ca.crt
```

STEP 3 | Modifiez les fichiers YAML afin de fournir les détails nécessaires au déploiement des pare-feu CN-Series.

Vous devez remplacer le chemin d'accès de l'image dans les fichiers YAML pour inclure le chemin d'accès à votre registre privé et fournir les paramètres requis. Pour plus d'informations, reportez-vous à la sectionParamètres modifiables dans les fichiers YAML de déploiement CN-Series.

STEP 4 | (CN-Series pour EKS sur AWS Outpost uniquement) Mettez à jour la classe de stockage. Pour prendre en charge CN-Series déployé sur AWS Outpost, vous devez utiliser le pilote de stockage aws-ebs-csi-driver, qui garantit qu'Outpost extrait les volumes d'Outpost lors de la création dynamique de volume persistant (PV).

1. Appliquez le yaml suivant.

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

2. Vérifiez que le contrôleur ebs-sc est en cours d'exécution.

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

3. Mettez à jour pan-cn-storage-class.yaml pour qu'il corresponde à l'exemple ci-dessous.

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

4. Ajoutez **`storageClassName: ebs-sc`** à pan-cn-mgmt.yaml aux emplacements indiqués ci-dessous.

   ```
   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 vous utilisez l'autoscaling dans votre environnement Kubernetes, consultez Horizontal Pod Autoscaling avant de continuer.

STEP 6 | Déployez le service CN-NGFW.

1. Vérifiez que vous avez créé le compte de service à l'aide du fichier pan-cni-serviceaccount.yaml.

   Voir Création de comptes de service pour l'authentification des clusters.

2. Utilisez Kubectl pour exécuter le fichier pan-cni-configmap.yaml.

   kubectl apply -f pan-cni-configmap.yaml

3. Utilisez kubectl pour exécuter le fichier pan-cn-ngfw-svc.yaml.

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

   Ce fichier yaml doit être déployé avant pan-cni.yaml.

4. Utilisez Kubectl pour exécuter le fichier pan-cni.yaml.

   kubectl apply -f pan-cni.yaml

5. Vérifiez que vous avez modifié les fichiers YAML pan-cni et pan-cni-configmap.
6. Exécutez la commande suivante et vérifiez que votre sortie est similaire à l'exemple suivant.

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

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

STEP 7 | Déployez le StatefulSet CN-MGMT.

Par défaut, le plan de gestion est déployé comme un StatefulSet qui garantie la tolérance aux pannes. Jusqu'à 30 pods pare-feu CN-NGFW peuvent se connecter à un StatefulSet CN-MGMT.

1. (Requis uniquement pour les PV provisionnés statiquement) Déployez les volumes persistants (PV) pour le StatefulSet CN-MGMT.

   1. Créez les répertoires qui correspondent aux noms des volumes locaux définis dans le fichier pan-cn-pv-local.yaml.

      Vous avez besoin de six (6) répertoires sur au moins 2 nœuds esclaves. Connectez-vous à chaque nœud esclave sur lequel le StatefulSet CN-MGMT sera déployé pour

créer les répertoires. Par exemple, pour créer des répertoires nommés /mnt/pan-local1 vers /mnt/pan-local6, utilisez la commande :

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

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

   Faites correspondre le nom d'hôte sous `nodeaffinity`, et vérifiez que vous avez modifié les répertoires que vous avez créés ci-dessus dans `spec.local.path` puis déployez le fichier pour créer une nouvelle storage class pan-local-storage et des PV locaux.

2. Vérifiez que vous avez modifié les fichiers YAML pan-cn-mgmt et pan-cn-mgmt-configmap.

   Exemple de pan-cn-mgmt-configmap de l'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
```

   Exemple 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. Utilisez Kubectl pour exécuter les fichiers 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

   Vous devez exécuter pan-mgmt-serviceaccount.yaml, uniquement si vous n'aviez pas déjà terminé la création de comptes de service pour l'authentification de cluster.
4. Vérifiez que les pods CN-MGMT sont activés.

   Cela prend environ 5-6 minutes.

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

STEP 8 | Déployez les pods CN-NGFW.

1. Vérifiez que vous avez modifié les fichiers YAML comme indiqué dans PAN-CN-NGFW-CONFIGMAP et PAN-CN-NGFW.

   containers: - name: pan-ngfw-container image: <your-private-registry-image-path>
2. Utilisez l'application Kubectl pour exécuter le fichier pan-cn-ngfw-configmap.yaml.

   kubectl apply -f pan-cn-ngfw-configmap.yaml
3. Utilisez l'application Kubectl pour exécuter le pan-cn-ngfw.yaml.

   kubectl apply -f pan-cn-ngfw.yaml
4. Vérifiez que les pods CN-NGFW sont en cours d'exécution.

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

STEP 9 | Activer la mise à l'échelle automatique du pod horizontal sur CN-Series.

STEP 10 | Vérifiez que vous pouvez voir CN-MGMT, le CN-NGFW et le PAN-CNI sur le cluster Kubernetes.

kubectl -n kube-system get pods

STEP 11 | Annotez l'application yaml ou l'espace de noms afin que le trafic de leurs nouveaux pods soit redirigé vers le pare-feu.

Vous devez ajouter l'annotation suivante pour rediriger le trafic vers le CN-NGFW pour inspection :

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

Par exemple, pour tous les nouveaux pods dans l'espace de noms « par défaut :

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

Sur certaines plateformes, les pods de l'application peuvent démarrer lorsque le pan-cni n'est pas actif dans la chaîne de plug-ins CNI. Pour éviter de tels scénarios, vous devez spécifier les volumes comme indiqué ici dans le pod d'application YAML.

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

STEP 12 | (facultatif) Certains trafics peuvent contourner le pare-feu en fonction de la ressource personnalisée *PortInfo* :

1. Appliquez la définition de ressource personnalisée PortInfo YAML.

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

2. Utilisez pan-cn-ngfw-port-cr.yaml comme exemple pour créer une ressource personnalisée PortInfo avec les protocoles et les ports que vous souhaitez contourner. Il est uniquement dans la direction sortante du point de vue du pod d'application et prend en charge TCP et UDP, jusqu'à 10 ports individuels (pas de plages de ports).

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

3. Appliquez votre ressource personnalisée PortInfo YAML.

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

4. Annotez le module d'application, en plus de l'annotation pan-fw. L'annotation doit être présente au moment du démarrage du pod d'application.

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

STEP 13 | Déployez votre application dans le cluster.

Activer la mise à l'échelle automatique du pod horizontal sur CN-Series

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec graphique Helm

Le module de mise à l'échelle automatique horizontal (HPA) est une ressource Kubernetes disponible dans tous les environnements cloud qui met automatiquement à l'échelle le nombre de pods CN-MGMT et CN-NGFW dans un déploiement basé sur des métriques surveillées. HPA utilise deux métriques standard dans tous les environnements cloud (utilisation du processeur et de la mémoire), ainsi que des métriques personnalisées spécifiques à chaque environnement cloud. En tant que tel, chaque cloud nécessite des fichiers yaml spécifiques pour activer HPA dans AKS, EKS et GKE.

HPA utilise un adaptateur de métriques spécifique au cloud pour récupérer les données de métriques d'un adaptateur de surveillance dans l'environnement cloud, tel que CloudWatch dans EKS, afin de déterminer quand augmenter ou réduire la mise à l'échelle en fonction des seuils que vous définissez. Vous devez modifier les fichiers yaml nécessaires pour définir le nombre minimum et maximal de réplicas, les seuils pour chaque mesure et les mesures utilisées dans la mise à l'échelle automatique de vos pare-feu.

Dans PAN OS 10.1, si vous utilisez la mise à l'échelle HPA des pods CN-MGMT, vous pouvez mettre à l'échelle de nombreux pods CN-MGMT sans qu'aucun pod DP ne leur soit connecté. Il est recommandé de créer un maximum de répliques des pods CN-MGMT pour éviter toute mise à l'échelle inutile.

Environnement Cloud	Métriques		Valeur moyenne
AKS, EKS et GKE	CN-MGMT	panloggingrate	Nombre de journaux
		pandataplaneslots	Nombre d'emplacements de plan de données
	CN-NFGW	dataplanecpuutilizationpct	Pourcentage d'utilisation du processeur CN-NGFW
		dataplanepacketbufferutilizationpct	Pourcentage d'utilisation de la mémoire tampon de paquets CN-NGFW

Environnement Cloud	Métriques		Valeur moyenne
		pansessionactive	Nombre de sessions actives sur le CN-NGFW
		pansessionutilization	Pourcentage d'utilisation de la session
		pansessionsslproxyutilization	Pourcentage d'utilisation du proxy SSL de session
		panthroughput	Débit en Kbit/s
		panpacketrate	Taux de paquets en paquets par seconde (pps)
		panconnectionspersecond	Connexions par seconde

Dans l'exemple ci-dessous se trouve le fichier pan-cn-hpa-dp.yaml pour EKS. Cet exemple utilise le pourcentage d'utilisation du processeur du plan de données pour mettre à l'échelle automatiquement les pods CN-NGFW. À 25 %, le cluster monte en échelle. Si l'utilisation du processeur atteint 50 %, le cluster déploie un pod supplémentaire. Si l'utilisation du processeur atteint 75 %, le cluster déploie deux pods supplémentaires. Ceci est déterminé en divisant la métrique totale par le seuil de métrique, puis en déployant suffisamment de pods pour ramener la métrique au seuil de configuration sur tous les pods CN-NGFW du cluster. Toutefois, le cluster ne déploiera pas plus de pods CN-NGFW que la valeur maxReplicas. Si plusieurs métriques dépassent le seuil en même temps, le cluster déploie le nombre nécessaire de pods pour traiter la métrique supérieure.

Par défaut, la carte HPA interroge l'adaptateur de mesures toutes les 15 secondes. Si les métriques que vous avez spécifiées dépassent le seuil configuré pendant 60 secondes, le cluster déploiera un pod CN-NGFW supplémentaire. Le cluster attend ensuite 300 secondes (cinq minutes) avant de décider si des pods CN-NGFW supplémentaires sont nécessaires ou non. Par défaut, un pod est déployé à la fois. Le cluster vérifie ensuite la métrique (dans ce cas, l'utilisation du processeur) après 300 secondes. Si l'utilisation est tombée à un niveau où un pod n'est plus nécessaire, le cluster supprime un pod. Le cluster attendra ensuite 60 secondes supplémentaires avant de décider si un autre pod peut être supprimé.

Toutes les valeurs indiquées ci-dessous et pour n'importe quelle métrique peuvent être modifiées pour s'adapter au mieux à votre déploiement.

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

```
to be ready - type: Percent value: 1 periodSeconds: 300 # assuming 5
mins for dp to be ready selectPolicy: Max metrics: - type: External
external: metric: name: dataplaneCpuUtilizationPct target: type:
Value value: 25
```

AKS

STEP 1 | Déployez une instance Azure Application Insights dans votre cluster. Vous devez fournir la clé d'instrumentation Azure Application Insights et la clé API Azure Application Insight APP ID requises en tant que secret K8s.

STEP 2 | Téléchargez les fichiers yaml HPA spécifiques à AKS à partir du référentiel GitHub de Palo Alto Networks.

STEP 3 | Si votre CN-MGMT est déployé dans un espace de noms personnalisé, mettez à jour pan-cn-adapater.yaml avec l'espace de noms personnalisé. L'espace de noms par défaut est **kube-system**.

STEP 4 | Si vous ne l'avez pas déjà fait, mettez à jour les paramètres HPA dans le fichier **`pan-cn-mgmt-configmap.yaml`** spécifique à 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 | Modifiez le fichier **`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 | Ajoutez le nom HPA que vous avez créé ci-dessus aux emplacements appropriés dans **`pan-cn-custommetrics.yaml`**.

STEP 7 | Modifiez **`pan-cn-hpa-dp.yaml`** et **`pan-cn-hpa-mp.yaml`**.

1. Entrez le nombre minimal et maximal de réplicas.
2. (Facultatif) Modifiez les valeurs de fréquence de mise à l'échelle et de montée en puissance en fonction de votre déploiement. Si vous ne modifiez pas ces valeurs, les valeurs par défaut sont utilisées.
3. Copiez la section suivante pour chaque métrique que vous souhaitez utiliser pour la mise à l'échelle.

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

4. Modifiez le nom de la métrique que vous souhaitez utiliser et définissez **`averageValue`** sur le seuil décrit dans le tableau ci-dessus. Si vous ne modifiez pas ces valeurs, les valeurs par défaut sont utilisées.
5. Enregistrez vos modifications.

STEP 8 | Déployez les fichiers yaml HPA. Les fichiers doivent être déployés dans l'ordre décrit ci-dessous.

1. Utiliser Kubectl pour exécuter le fichier pan-cn-hpa-secret.yaml

   `kubectl apply -f pan-cn-hpa-secret.yaml`
2. Utiliser Kubectl pour exécuter le fichier pan-cn-adapter.yaml

   `kubectl apply -f pan-cn-adapter.yaml`
3. Utiliser Kubectl pour exécuter le fichier pan-cn-custommetrics.yaml

   `kubectl apply -f pan-cn-custommetrics.yaml`
4. Utiliser Kubectl pour exécuter le fichier pan-cn-hpa-dp.yaml

   `kubectl apply -f pan-cn-hpa-dp.yaml`
5. Utiliser Kubectl pour exécuter le fichier pan-cn-hpa-mp.yaml

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

STEP 9 | Vérifiez votre déploiement.

- Utilisez kubectl pour vérifier que le pod d'adaptateur de mesures personnalisées dans l'espace de noms de mesures personnalisées.

  `kubectl get pods -n custom-metrics`
- Utilisez kubectl pour rechercher la ressource HPA.

  `kubectl get hpa -n kube-system`

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

EKS

STEP 1 | Déployez l'adaptateur Amazon CloudWatch Metrics pour Kubernetes dans votre cluster CN-Series en tant que service. Vous devez autoriser CloudWatch à accéder complètement aux deux rôles IAM associés à vos pods et clusters Kubernetes. Pour publier les métriques personnalisées dans CloudWatch, le rôle des nœuds esclaves doit disposer de la politique

gérée par AWS **CloudWatchAgentServerPolicy** afin que la fonction HPA puisse les récupérer.

STEP 2 | Téléchargez les fichiers yaml HPA spécifiques à EKS à partir du référentiel GitHub de Palo Alto Networks.

STEP 3 | Si votre CN-MGMT est déployé dans un espace de noms personnalisé, mettez à jour pan-cn-adapater.yaml avec l'espace de noms personnalisé. L'espace de noms par défaut est **kube-system**.

STEP 4 | Modifiez **`pan-cn-hpa-dp.yaml`** et **`pan-cn-hpa-mp.yaml`**.

1. Entrez le nombre minimal et maximal de réplicas.
2. (Facultatif) Modifiez les valeurs de fréquence de mise à l'échelle et de montée en puissance en fonction de votre déploiement. Si vous ne modifiez pas ces valeurs, les valeurs par défaut sont utilisées.
3. Copiez la section suivante pour chaque métrique que vous souhaitez utiliser pour la mise à l'échelle.

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

4. Modifiez le nom de la métrique que vous souhaitez utiliser et définissez **`averageValue`** sur le seuil décrit dans le tableau ci-dessus. Si vous ne modifiez pas ces valeurs, les valeurs par défaut sont utilisées.
5. Enregistrez vos modifications.

STEP 5 | Déployez les fichiers yaml HPA. Les fichiers doivent être déployés dans l'ordre décrit ci-dessous.

1. Utiliser Kubectl pour exécuter le fichier pan-cn-adapter.yaml

   `kubectl apply -f pan-cn-adapter.yaml`
2. Utiliser Kubectl pour exécuter le fichier pan-cn-externalmetrics.yaml

   `kubectl apply -f pan-cn-externalmetrics.yaml`
3. Utiliser Kubectl pour exécuter le fichier pan-cn-hpa-dp.yaml

   `kubectl apply -f pan-cn-hpa-dp.yaml`
4. Utiliser Kubectl pour exécuter le fichier pan-cn-hpa-mp.yaml

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

STEP 6 | Vérifiez votre déploiement.

- Utilisez kubectl pour vérifier que le pod d'adaptateur de mesures personnalisées dans l'espace de noms de mesures personnalisées.

  `kubectl get pods -n custom-metrics`
- Utilisez kubectl pour rechercher la ressource HPA.

  `kubectl get hpa -n kube-system`

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

Déployer le pare-feu CN-Series en tant que DaemonSet

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec graphique Helm

Effectuez la procédure suivante pour déployer le pare-feu CN-Series en tant que Daemonset.

Avant de commencer, assurez-vous que la version du fichier YAML CN-Series est compatible avec la version PAN-OS.

- PAN-OS 10.1.2 ou version ultérieure nécessite YAML 2.0.2
- PAN-OS 10.1.0 et 10.1.1 nécessitent YAML 2.0.0 ou 2.0.1

STEP 1 | Configurez votre cluster Kubernetes.

1. Vérifiez que le cluster dispose des ressources adéquates. Assurez-vous que ce cluster dispose des conditions préalables de CN-Series pour prendre en charge le pare-feu :

   ```
   kubectl get nodes
   ```

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

   Affichez les informations sous l'en-tête Capacity (Capacité) dans la sortie de la commande pour voir le processeur et la mémoire disponibles sur le nœud spécifié.

   L'allocation du processeur, de la mémoire et du stockage sur disque dépendra de vos besoins. Voir Performances et mise à l'échelle de CN-Series.

   Assurez-vous d'avoir les informations suivantes :

   - Collectez l'adresse IP du terminal pour configurer le serveur API sur Panorama. Panorama utilise cette adresse IP pour se connecter à votre cluster Kubernetes.
   - Collectez le nom de la pile de modèles, le nom du groupe d'appareils, l'adresse IP Panorama et éventuellement le nom du groupe du collecteur de journaux à partir de Panorama.
   - Collectez le code d'autorisation et l'ID et la valeur du code PIN d'enregistrement automatique.
   - L'emplacement du conteneur d'images dans lequel vous avez téléchargé les images.

STEP 2 | (facultatif) Si vous avez configuré un certificat personnalisé dans le plug-in Kubernetes pour Panorama, vous devez créer le secret de certificat en exécutant la commande suivante. Ne

modifiez pas le nom de fichier de ca.crt. Le volume des certificats personnalisés dans pan-cn-mgmt.yaml et pan-cn-ngfw.yaml est facultatif.

```
kubectl -n kube-system crée un secret générique custom-ca --from-file=ca.crt
```

STEP 3 | Modifiez les fichiers YAML afin de fournir les détails nécessaires au déploiement des pare-feu CN-Series.

Vous devez remplacer le chemin d'accès de l'image dans les fichiers YAML pour inclure le chemin d'accès à votre registre privé et fournir les paramètres requis. Pour plus d'informations, reportez-vous à la sectionParamètres modifiables dans les fichiers YAML de déploiement CN-Series.

STEP 4 | Déployez le DaemonSet CNI.

Le conteneur CNI est déployé comme un DaemonSet (un pod par nœud) et il crée deux interfaces sur le pod CN-NGFW pour chaque application déployée sur le nœud. Lorsque vous utilisez les commandes kubectl pour exécuter les fichiers YAML pan-cni, il devient une partie de la chaîne de service sur chaque nœud.

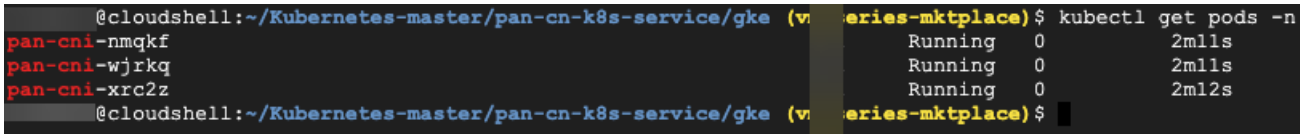
1. Vérifiez que vous avez créé le compte de service à l'aide du fichier pan-cni-serviceaccount.yaml.

   Voir Création de comptes de service pour l'authentification des clusters.
2. Utilisez Kubectl pour exécuter le fichier pan-cni-configmap.yaml.

   ```
   kubectl apply -f pan-cni-configmap.yaml
   ```
3. Utilisez Kubectl pour exécuter le fichier pan-cni.yaml.

   ```
   kubectl apply -f pan-cni.yaml
   ```
4. Vérifiez que vous avez modifié les fichiers YAML pan-cni et pan-cni-configmap.
5. Exécutez la commande suivante et vérifiez que votre sortie est similaire à l'exemple suivant.

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



STEP 5 | (CN-Series pour EKS sur AWS Outpost uniquement) Mettez à jour la classe de stockage. Pour prendre en charge CN-Series déployé sur AWS Outpost, vous devez utiliser le pilote de

stockage aws-ebs-csi-driver, qui garantit qu'Outpost extrait les volumes d'Outpost lors de la création dynamique de volume persistant (PV).

1. Appliquez le yaml suivant.

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

2. Vérifiez que le contrôleur ebs-sc est en cours d'exécution.

   kubectl -n kube-system get pods

3. Mettez à jour pan-cn-storage-class.yaml pour qu'il corresponde à l'exemple ci-dessous.

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

4. Ajoutez **storageClassName: ebs-sc** à pan-cn-mgmt.yaml aux emplacements indiqués ci-dessous.

```
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 | Déployez le StatefulSet CN-MGMT.

Par défaut, le plan de gestion est déployé comme un StatefulSet qui garantie la tolérance aux pannes. Jusqu'à 30 pods pare-feu CN-NGFW peuvent se connecter à un StatefulSet CN-MGMT.

1. (Requis uniquement pour les PV provisionnés statiquement) Déployez les volumes persistants (PV) pour le StatefulSet CN-MGMT.

   1. Créez les répertoires qui correspondent aux noms des volumes locaux définis dans le fichier pan-cn-pv-local.yaml.

      Vous avez besoin de six (6) répertoires sur au moins 2 nœuds esclaves. Connectez-vous à chaque nœud esclave sur lequel le StatefulSet CN-MGMT sera déployé pour

créer les répertoires. Par exemple, pour créer des répertoires nommés /mnt/pan-local1 vers /mnt/pan-local6, utilisez la commande :

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

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

   Faites correspondre le nom d'hôte sous `nodeaffinity`, et vérifiez que vous avez modifié les répertoires que vous avez créés ci-dessus dans `spec.local.path` puis déployez le fichier pour créer une nouvelle storage class pan-local-storage et des PV locaux.

2. Vérifiez que vous avez modifié les fichiers YAML pan-cn-mgmt et pan-cn-mgmt-configmap.

   Exemple de pan-cn-mgmt-configmap de l'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
```

   Exemple 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. Utilisez Kubectl pour exécuter les fichiers 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

   Vous devez exécuter pan-mgmt-serviceaccount.yaml, uniquement si vous n'aviez pas déjà terminé la création de comptes de service pour l'authentification de cluster.

4. Vérifiez que les pods CN-MGMT sont activés.

   Cela prend environ 5-6 minutes.

   Utilisez **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 | Déployez les pods CN-NGFW.

Par défaut, le pod CN-NGFW du plan de données du pare-feu est déployé comme un DaemonSet. Une instance du pod CN-NFGW peut sécuriser le trafic pour un maximum de 30 pods d'application sur un nœud.

1. Vérifiez que vous avez modifié les fichiers YAML comme indiqué dans PAN-CN-NGFW-CONFIGMAP et PAN-CN-NGFW.

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

2. Utilisez l'application Kubectl pour exécuter le fichier pan-cn-ngfw-configmap.yaml.

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

3. Utilisez l'application Kubectl pour exécuter le pan-cn-ngfw.yaml.

   kubectl apply -f pan-cn-ngfw.yaml

4. Vérifiez que tous les pods CN-NGFW sont en cours d'exécution (un par nœud dans votre cluster)

   Il s'agit d'un exemple de résultat provenant d'un cluster de 4 nœuds sur site.

   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 | Vérifiez que vous pouvez voir CN-MGMT, le CN-NGFW et le PAN-CNI sur le cluster 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 | Annotez l'application yaml ou l'espace de noms afin que le trafic de leurs nouveaux pods soit redirigé vers le pare-feu.

Vous devez ajouter l'annotation suivante pour rediriger le trafic vers le CN-NGFW pour inspection :

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

Par exemple, pour tous les nouveaux pods dans l'espace de noms « par défaut :

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

Sur certaines plateformes, les pods de l'application peuvent démarrer lorsque le pan-cni n'est pas actif dans la chaîne de plug-ins CNI. Pour éviter de tels scénarios, vous devez spécifier les volumes comme indiqué ici dans le pod d'application YAML.

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

STEP 10 | Déployez votre application dans le cluster.

Déployer le pare-feu CN-Series en tant que Kubernetes CNF

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec graphique Helm

Vous pouvez désormais déployer la série CN en tant que fonction réseau de conteneurs (CNF) dans votre environnement Kubernetes.

Le mode de déploiement CN-Series-as-a-daemonset et CN-Series-as-a-kubernetes-service fournit un déploiement de sécurité automatisé et tire parti des capacités de mise à l'échelle automatique de Kubernetes. Toutefois, ces modes de déploiement ont des options d'insertion limitées et ne prennent pas en charge l'accélération des E/S. En outre, ils limitent le débit réalisable pour les pods d'application qui nécessitent une inspection et utilisent plusieurs interfaces réseau.

Le déploiement CN-series-as-a-kubernetes-CNF résout ces problèmes pour le trafic qui utilise le Service Function Chaining (SFC) via des entités externes telles que le routage natif du fournisseur de cloud, les vRouters et les commutateurs Top of Rack (TOR). Le mode de déploiement CN-series-as-a-kubernetes-CNF n'a pas d'impact sur les pods d'application.

Effectuez la procédure suivante pour déployer le pare-feu CN-Series-as-a-kubernetes-CNF.

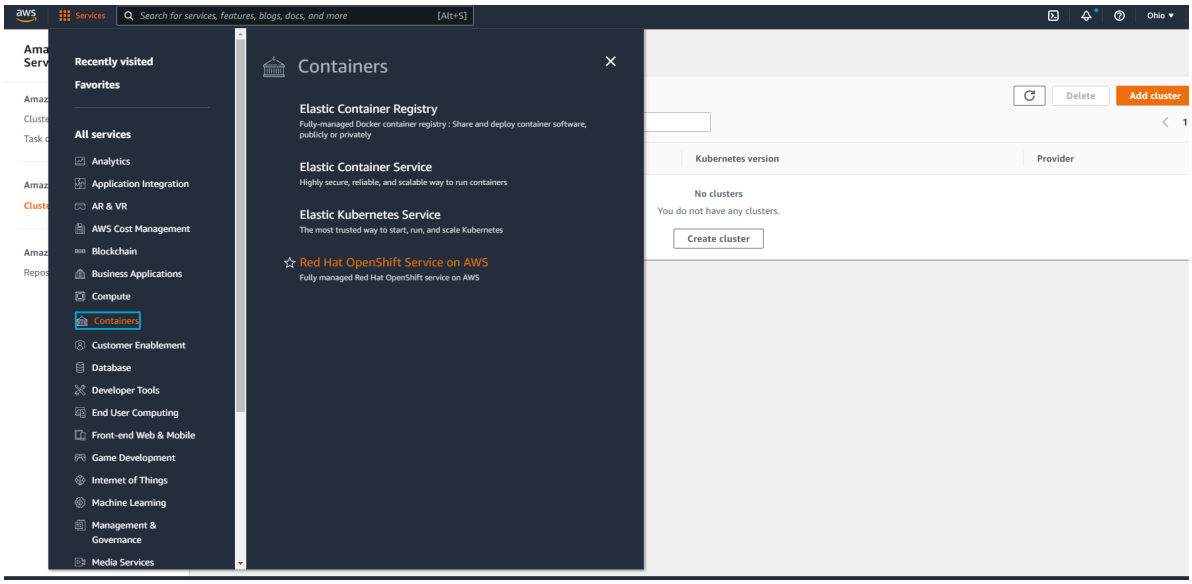
Avant de commencer, assurez-vous que la version du fichier YAML CN-Series est compatible avec la version de PAN-OS :

PAN-OS 10.2.0 ou version ultérieure nécessite YAML 3.0.0

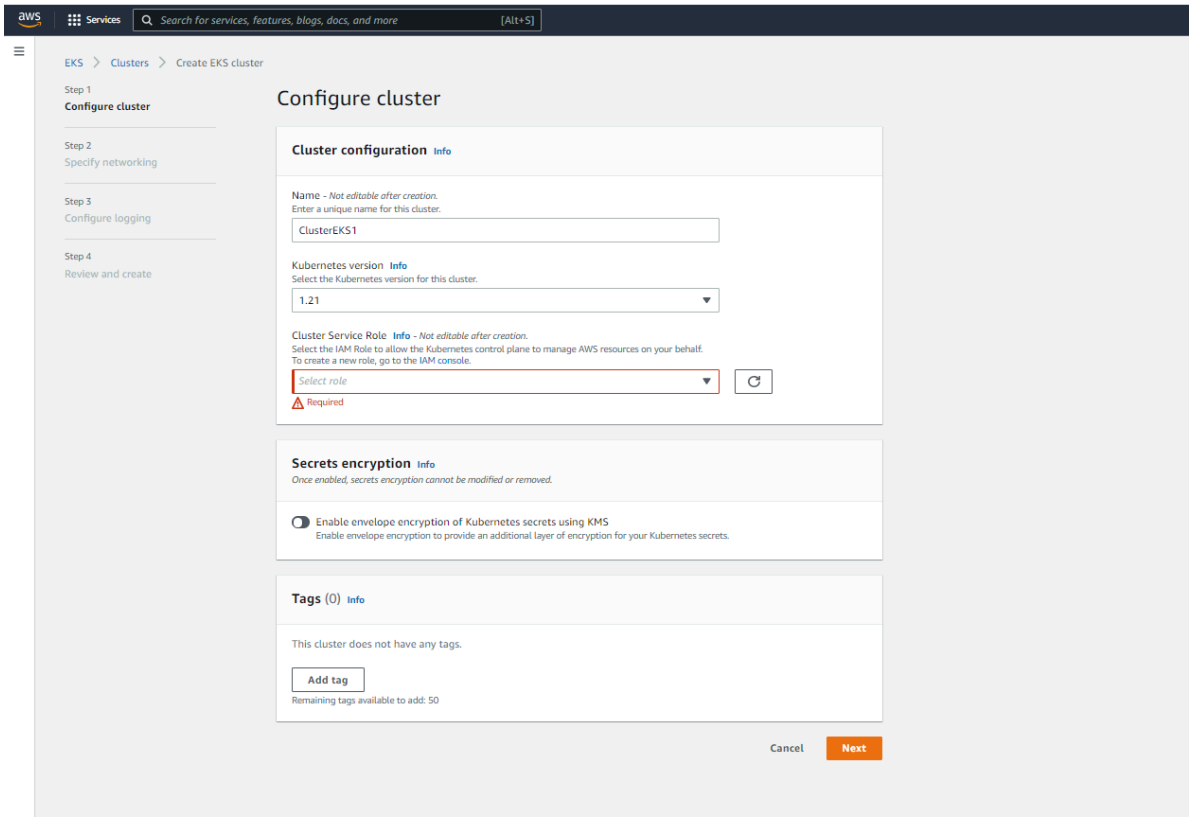
STEP 1 | Configurez votre cluster Kubernetes. Pour plus d'informations, consultez Création d'un cluster Amazon EKS et Interfaces réseau multiples pour les pods.

Pour créer un cluster dans AWS EKS, procédez comme suit :

1. Cliquez sur le menu de navigation **Services**, accédez à **Containers (Conteneurs)**->**Elastic Kubernetes Service (Service Elastic Kubernetes)**.



2. Cliquez sur **Create Cluster (Créer un cluster)**.
3. Renseignez les détails requis, puis cliquez sur **Create (Créer)**.



1. Vérifiez que le cluster dispose des ressources adéquates. Assurez-vous que ce cluster dispose des conditions préalables de CN-Series pour prendre en charge le pare-feu :

   kubectl get nodes

   kubectl describe node <node-name>

   Affichez les informations sous l'en-tête Capacity (Capacité) dans la sortie de la commande pour voir le processeur et la mémoire disponibles sur le nœud spécifié.

   L'allocation du processeur, de la mémoire et du stockage sur disque dépendra de vos besoins. Voir Performances et mise à l'échelle de CN-Series.

   Assurez-vous d'avoir les informations suivantes :

   - Collectez l'adresse IP du terminal pour configurer le serveur API sur Panorama. Panorama utilise cette adresse IP pour se connecter à votre cluster Kubernetes.
   - Collectez le nom de la pile de modèles, le nom du groupe d'appareils, l'adresse IP Panorama et éventuellement le nom du groupe du collecteur de journaux à partir de Panorama.
   - Collectez le code d'autorisation et l'ID et la valeur du code PIN d'enregistrement automatique.
   - L'emplacement du conteneur d'images dans lequel vous avez téléchargé les images.

STEP 2 | (facultatif) Si vous avez configuré un certificat personnalisé dans le plug-in Kubernetes pour Panorama, vous devez créer le secret de certificat en exécutant la commande suivante. Ne modifiez pas le nom de fichier de ca.crt. Le volume des certificats personnalisés dans pan-cn-mgmt-0.yaml, pan-cn-mgmt-1.yaml, pan-cn-ngfw-0.yaml et pan-cn-ngfw.yaml-1 est facultatif.

kubectl -n kube-system crée un secret générique custom-ca --from-file=ca.crt

STEP 3 | Modifiez les fichiers YAML afin de fournir les détails nécessaires au déploiement des pare-feu CN-Series.

Vous devez remplacer le chemin d'accès de l'image dans les fichiers YAML pour inclure le chemin d'accès à votre répertoire privé et fournir les paramètres requis. Pour plus

d'informations, reportez-vous à la sectionParamètres modifiables dans les fichiers YAML de déploiement CN-Series.

CN-Series-as-a-kubernetes-CNF en HA ne prend en charge que la HA active/passive avec synchronisation de session et de configuration.

Lorsque vous déployez CN-Series-as-a-kubernetes-CNF dans HA, il y aura deux fichiers YAML PAN-CN-MGMT-CONFIGMAP, PAN-CN-MGMT et PAN-CN-NGFW, chacun pour les nœuds actifs et passifs comme suit :

- 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

Les valeurs par défaut suivantes sont définies dans les fichiers pan-cn-mgmt-configmap-0.yaml et pan-cn-mgmt-configmap-1.yaml.

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

```
metadata:
```

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

```
namespace: kube-system
```

```
data:
```

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

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

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

```
metadata:
```

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

```
namespace: kube-system
```

```
data:
```

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

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


Vous pouvez ajouter l'option numa pour l'épinglage du processeur. Ajoutez le numéro de nœud numa unique pour le paramètre PAN_NUMA_ENABLED dans les fichiers pan-cn-ngfw-configmap-0.yaml et pan-cn-ngfw-configmap-1.yaml.

Pour déployer avec succès le CN-Series-as-a-kubernetes-CNF en HA avec prise en charge de la couche 3 :

- En HA, chaque nœud Kubernetes doit avoir au moins trois interfaces : Gestion (par défaut), HA2 et interface de données.
- Pour le pare-feu CN-Series en mode L3, il doit y avoir au moins deux interfaces : Gestion (par défaut) et interface de données.
- Modifiez les nouveaux fichiers YAML de définition de pièce jointe réseau avec les modifications suivantes :
  - Sur le workernode, récupérez la valeur **pciBusID** à partir de l'interface de l'hyperviseur en exécutant la commande suivante :

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

    Par exemple :

    ```
    Contrôleur Ethernet 00:05.0 : Amazon.com, Inc. Carte
     réseau Elastic (ENA)
    ```

    ```
    Contrôleur Ethernet 00:06.0 : Amazon.com, Inc. Carte
     réseau Elastic (ENA)
    ```

    ```
    Contrôleur Ethernet 00:07.0 : Amazon.com, Inc. Carte
     réseau Elastic (ENA)
    ```

    ```
    Contrôleur Ethernet 00:08.0 : Amazon.com, Inc. Carte
     réseau Elastic (ENA)
    ```

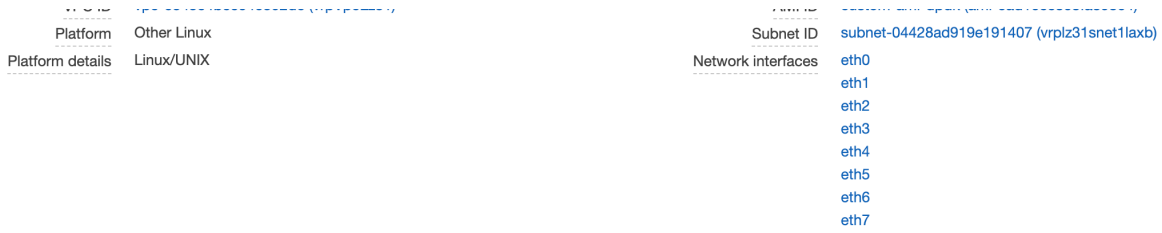
    ```
    Contrôleur Ethernet 00:09.0 : Amazon.com, Inc. Carte
     réseau Elastic (ENA)
    ```

    ```
    Contrôleur Ethernet 00:0a.0 : Amazon.com, Inc. Carte
     réseau Elastic (ENA)
    ```

    ```
    Contrôleur Ethernet 00:0b.0 : Amazon.com, Inc. Carte
     réseau Elastic (ENA)
    ```

    ```
    Contrôleur Ethernet 00:0c.0 : Amazon.com, Inc. Carte
     réseau Elastic (ENA)
    ```

    L'ordre PCI est identique à l'ordre des interfaces eth, comme indiqué sur l'interface utilisateur AWS EC2



Ajoutez la valeur **`pciBusID`** récupérée ci-dessus aux fichiers de définition de réseau suivants :

```
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
```

- Récupérez l'adresse IP statique de l'interface HA2 à partir de l'instance de nœud correspondante sur la console AWS et ajoutez-la au paramètre `d'adresse` du fichier `net-attach-def-ha2-0.yaml` et `net-attach-def-ha2-1.yaml`.

STEP 4 | Déployez le StatefulSet CN-MGMT.

Par défaut, le plan de gestion est déployé comme un StatefulSet qui garantie la tolérance aux pannes. Un seul pod de pare-feu CN-NGFW peut se connecter à un StatefulSet CN-MGMT.

1. (Requis uniquement pour les PV provisionnés statiquement) Déployez les volumes persistants (PV) pour le StatefulSet CN-MGMT.

   1. Créez les répertoires qui correspondent aux noms des volumes locaux définis dans le fichier pan-cn-pv-local.yaml.

   Vous avez besoin de six (6) répertoires sur au moins 2 nœuds esclaves. Connectez-vous à chaque nœud esclave sur lequel le StatefulSet CN-MGMT sera déployé pour créer les répertoires. Par exemple, pour créer des répertoires nommés /mnt/pan-local1 vers /mnt/pan-local6, utilisez la commande :

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

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

   Faites correspondre le nom d'hôte sous `nodeaffinity`, et vérifiez que vous avez modifié les répertoires que vous avez créés ci-dessus dans `spec.local.path`, puis

déployez le fichier pour créer une nouvelle storageclass pan-local-storage et des PV locaux.

2. Vérifiez que vous avez modifié les fichiers YAML pan-cn-mgmt et pan-cn-mgmt-configmap.
3. Utilisez Kubectl pour exécuter les fichiers 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

   Vous devez exécuter pan-mgmt-serviceaccount.yaml, uniquement si vous n'aviez pas déjà terminé la création de comptes de service pour l'authentification de cluster.
4. Vérifiez que les pods CN-MGMT sont activés.

   Cela prend environ 5-6 minutes.

   Utilisez **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 | Déployez le CN-NGFW en mode k8s-CNF.

1. Vérifiez que vous avez modifié les fichiers YAML comme indiqué à l'étape 3.

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

Vous devez vous assurer que le daemonset multus est installé et que les fichiers de définition des pièces jointes réseau sont créés. La valeur du paramètre `PAN_SERVICE_NAME` dans les fichiers pan-cn-ngfw-configmap-0.yaml et pan-cn-ngfw-configmap-1.yaml doit correspondre à la valeur du paramètre `Service Name (Nom du service)` dans les fichiers pan-cn-mgmt-0.yaml et pan-cn-mgmt-1.yaml respectivement.

*Pour la prise en charge HA, il est recommandé de déployer des pods DP sur différents nœuds de travail. Vous pouvez vous en assurer à partir du champ **yaml nodeSelector** ou en activant l'anti-affinité du pod.*

*Pour activer la prise en charge HA, vous devez vous assurer que la valeur du paramètre `PAN_HA_SUPPORT` est **`true`** dans les fichiers YAML suivants :*

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

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

Pour les interfaces de données pour les pods DP, les CNI et les ressources d'interface doivent être ajoutées aux fichiers DP YAML selon les besoins. Par exemple :

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

*Pour activer la prise en charge de DPDK, vous devez vous assurer que la valeur du paramètre `PAN_DATA_MODE` est **`dpdk`** dans le fichier pan-cn-ngfw-configmap-0.yaml et pan-cn-ngfw-configmap-1.yaml.*

En outre, la valeur du paramètre `HUGEPAGE_MEMORY_REQUEST` doit correspondre à la demande de mémoire de page énorme dans les fichiers pan-cn-ngfw-0.yaml et pan-cn-ngfw-1.yaml.

Pour plus d'informations, consultez Configurer DPDK sur le pare-feu CN-Series.

2. Utilisez Kubectl apply pour exécuter les pan-cn-ngfw-configmap-0.yaml et 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. Utilisez Kubectl apply pour exécuter les pan-cn-ngfw-0.yaml et pan-cn-ngfw-1.yaml.

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

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

4. Vérifiez que les pods CN-NGFW sont en cours d'exécution.

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

STEP 6 | Déployez les pods CN-NGFW. Procédez comme suit :

1. Vérifiez que vous avez modifié les fichiers YAML comme indiqué dans PAN-CN-NGFW-CONFIGMAP-0, PAN-CN-NGFW-CONFIGMAP-1, PAN-CN-NGFW-0 et PAN-CN-NGFW-1.

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

2. Utilisez l'application Kubectl pour exécuter le fichier pan-cn-ngfw-configmap.yaml.

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

3. Utilisez l'application Kubectl pour exécuter le pan-cn-ngfw.yaml.

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

4. Vérifiez que les pods CN-NGFW sont en cours d'exécution.

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

STEP 7 | Vérifiez que vous pouvez voir CN-MGMT et CN-NGFW sur le cluster Kubernetes. Exécutez la commande suivante :

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


Déployer Kubernetes CNF L3 en mode autonome

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec graphique Helm

Vous pouvez déployer le pare-feu CN-Series en tant que fonction réseau de conteneur (CNF) en mode autonome L3 dans votre environnement Kubernetes.

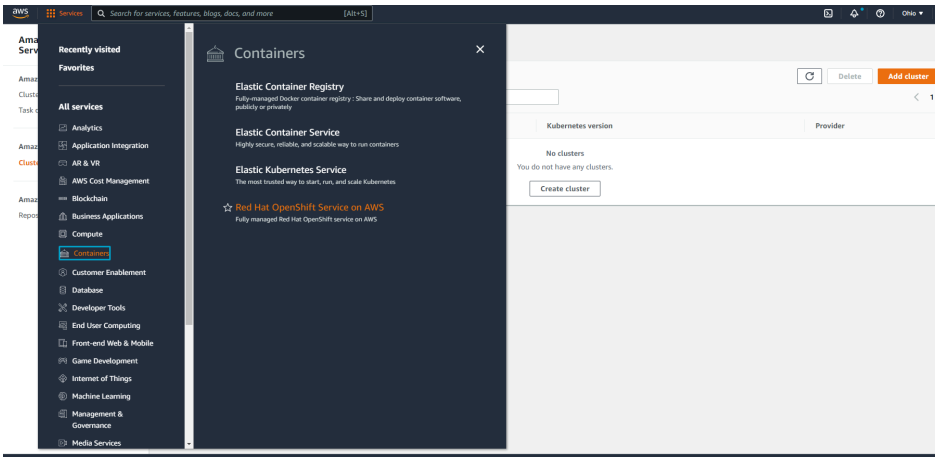
La série CN prend désormais en charge le trafic via un vRouter, où des itinéraires statiques sont configurés pour rediriger le trafic vers les interfaces de plan de données du pare-feu. Pour la direction inverse, le trafic est redirigé vers le même pare-feu à l'aide du routage basé sur la politique L3 (PBR) avec des adresses IP IPv4. Les adresses IP des interfaces dans l'environnement K8s sont généralement programmées via le CNI à l'aide de DHCP.

Pour déployer le Kubernetes CNF en mode autonome L3 :

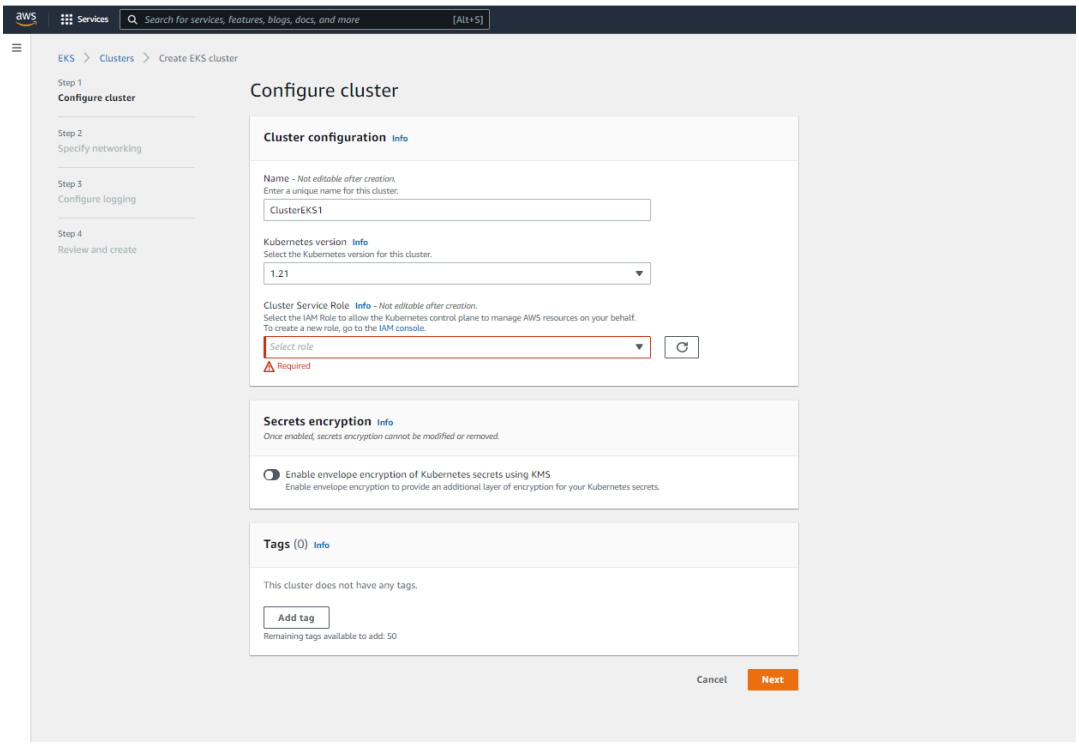
STEP 1 | Configurez votre cluster Kubernetes.

Pour créer un cluster dans AWS EKS, procédez comme suit :

1. Cliquez sur le menu de navigation **Services**, accédez à **Containers (Conteneurs)->Elastic Kubernetes Service (Service Elastic Kubernetes)**.



2. Cliquez sur **Create Cluster (Créer un cluster)**.
3. Renseignez les détails requis, puis cliquez sur **Create (Créer)**.



1. Vérifiez que le cluster dispose des ressources adéquates. Assurez-vous que ce cluster dispose des conditions préalables de CN-Series pour prendre en charge le pare-feu :

   `kubectl get nodes`

   `kubectl describe node <node-name>`

   Affichez les informations sous l'en-tête Capacity (Capacité) dans la sortie de la commande pour voir le processeur et la mémoire disponibles sur le nœud spécifié.

L'allocation du processeur, de la mémoire et du stockage sur disque dépendra de vos besoins. Voir Performances et mise à l'échelle de CN-Series.

Assurez-vous d'avoir les informations suivantes :

- Collectez l'adresse IP du terminal pour configurer le serveur API sur Panorama. Panorama utilise cette adresse IP pour se connecter à votre cluster Kubernetes.
- Collectez le nom de la pile de modèles, le nom du groupe d'appareils, l'adresse IP Panorama et éventuellement le nom du groupe du collecteur de journaux à partir de Panorama.
- Collectez le code d'autorisation et l'ID et la valeur du code PIN d'enregistrement automatique.
- L'emplacement du conteneur d'images dans lequel vous avez téléchargé les images.

STEP 2 | Créez un secret de certificat (facultatif) Si vous avez configuré un certificat personnalisé dans le plug-in Kubernetes pour Panorama, vous devez créer le secret de certificat en exécutant la commande suivante. Ne modifiez pas le nom de fichier de ca.crt. Le volume des certificats personnalisés dans pan-cn-mgmt-0.yaml et pan-cn-ngfw-0.yaml est facultatif.

```
kubectl -n kube-system crée un secret générique custom-ca --from-file=ca.crt
```

STEP 3 | Modifiez les fichiers YAML afin de fournir les détails nécessaires au déploiement des pare-feu CN-Series.

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

Vous devez remplacer le chemin d'accès de l'image dans les fichiers YAML pour inclure le chemin d'accès à votre répertoire privé et fournir les paramètres requis. Pour plus

d'informations, reportez-vous à la sectionParamètres modifiables dans les fichiers YAML de déploiement CN-Series.

Les valeurs par défaut suivantes sont définies dans le fichier 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
```

Vous pouvez ajouter l'option numa pour l'épinglage du processeur. Ajoutez le numéro de nœud numa unique pour le paramètre `PAN_NUMA_ENABLED` dans le fichier pan-cn-ngfw-configmap-0.yaml.

Pour déployer avec succès le CN-Series-as-a-kubernetes-CNF avec prise en charge de la couche 3 :

- Chaque nœud Kubernetes doit avoir au moins trois interfaces : Gestion (par défaut), lien HA2 et interface de données.
- Pour le pare-feu CN-Series en mode L3, il doit y avoir au moins deux interfaces : Gestion (par défaut) et interface de données.
- Modifiez les nouveaux fichiers YAML de définition de pièce jointe réseau avec les modifications suivantes :
  - Sur le workernode, récupérez la valeur **`pciBusID`** à partir de l'interface de l'hyperviseur en exécutant la commande suivante :

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

Par exemple :

```
Contrôleur Ethernet 00:05.0 : Amazon.com, Inc. Carte
 réseau Elastic (ENA)
```

```
Contrôleur Ethernet 00:06.0 : Amazon.com, Inc. Carte
 réseau Elastic (ENA)
```

```
Contrôleur Ethernet 00:07.0 : Amazon.com, Inc. Carte
 réseau Elastic (ENA)
```

```
Contrôleur Ethernet 00:08.0 : Amazon.com, Inc. Carte
 réseau Elastic (ENA)
```

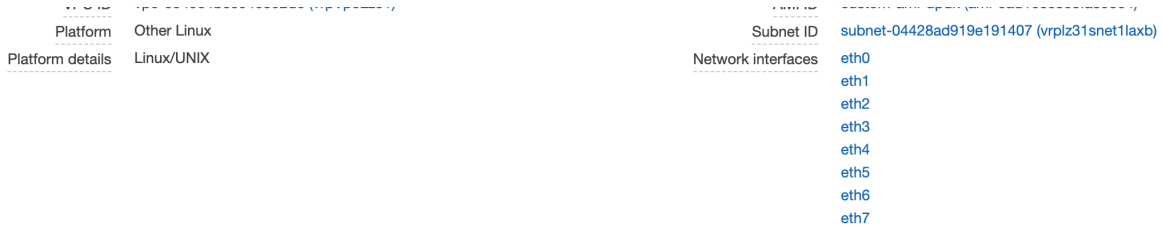
```
Contrôleur Ethernet 00:09.0 : Amazon.com, Inc. Carte
 réseau Elastic (ENA)
```

```
Contrôleur Ethernet 00:0a.0 : Amazon.com, Inc. Carte
 réseau Elastic (ENA)
```

```
Contrôleur Ethernet 00:0b.0 : Amazon.com, Inc. Carte
 réseau Elastic (ENA)
```

```
Contrôleur Ethernet 00:0c.0 : Amazon.com, Inc. Carte
 réseau Elastic (ENA)
```

L'ordre PCI est identique à l'ordre des interfaces eth, comme indiqué sur l'interface utilisateur AWS EC2



Ajoutez la valeur **`pciBusID`** récupérée ci-dessus aux fichiers de définition de réseau suivants :

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

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

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

STEP 4 | Déployez le StatefulSet CN-MGMT.

Par défaut, le plan de gestion est déployé comme un StatefulSet qui garantie la tolérance aux pannes. Un seul pod de pare-feu CN-NGFW peut se connecter à un StatefulSet CN-MGMT.

1. (Requis uniquement pour les PV provisionnés statiquement) Déployez les volumes persistants (PV) pour le StatefulSet CN-MGMT.
   1. Créez les répertoires qui correspondent aux noms des volumes locaux définis dans le fichier pan-cn-pv-local.yaml.

      Vous avez besoin de six (6) répertoires sur au moins 2 nœuds esclaves. Connectez-vous à chaque nœud esclave sur lequel le StatefulSet CN-MGMT sera déployé pour créer les répertoires. Par exemple, pour créer des répertoires nommés /mnt/pan-local1 vers /mnt/pan-local6, utilisez la commande :

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

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

      Faites correspondre le nom d'hôte sous `nodeaffinity`, et vérifiez que vous avez modifié les répertoires que vous avez créés ci-dessus dans `spec.local.path` puis déployez le fichier pour créer une nouvelle storage class pan-local-storage et des PV locaux.
2. Vérifiez que vous avez modifié les fichiers YAML pan-cn-mgmt et pan-cn-mgmt-configmap.
3. Utilisez Kubectl pour exécuter les fichiers 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

   Vous devez exécuter pan-mgmt-serviceaccount.yaml, uniquement si vous n'aviez pas déjà terminé la création de comptes de service pour l'authentification de cluster.
4. Vérifiez que les pods CN-MGMT sont activés.

   Cela prend environ 5-6 minutes.

   Utilisez **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 | Déployez le CN-NGFW en mode k8s-CNF.

1. Vérifiez que vous avez modifié les fichiers YAML comme indiqué à l'étape 3.

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

   Vous devez vous assurer que le daemonset multus est installé et que les fichiers de définition des pièces jointes réseau sont créés. La valeur du paramètre pour PAN_SERVICE_NAME dans le fichier pan-cn-ngfw-configmap-0.yaml doit correspondre à la valeur du paramètre `Service Name (Nom du service)` dans le fichier pan-cn-mgmt-0.yaml.

   Pour les interfaces de données des pods CN-NFGW, les CNI et les ressources d'interface doivent être ajoutées aux fichiers YAML CN-NFGW selon les besoins. Par exemple :

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

   *Pour activer la prise en charge de DPDK, vous devez vous assurer que la valeur du paramètre PAN_DATA_MODE est **dpdk** dans le fichier pan-cn-ngfw-configmap-0.yaml.*

   De plus, la valeur du paramètre HUGEPAGE_MEMORY_REQUEST doit correspondre à la demande de mémoire de page énorme dans le fichier pan-cn-ngfw-0.yaml.

   Pour plus d'informations, consultez Configurer DPDK sur le pare-feu CN-Series.

2. Utilisez Kubectl apply pour exécuter le fichier pan-cn-ngfw-configmap-0.yaml.

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

3. Utilisez Kubectl apply pour exécuter les pan-cn-ngfw-0.yaml et pan-cn-ngfw-1.yaml.

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

4. Vérifiez que les pods CN-NGFW sont en cours d'exécution.

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

STEP 6 | Déployez les pods CN-NGFW. Procédez comme suit :

1. Vérifiez que vous avez modifié les fichiers YAML comme indiqué dans PAN-CN-NGFW-CONFIGMAP-0 et PAN-CN-NGFW-0.

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

2. Utilisez l'application Kubectl pour exécuter le fichier pan-cn-ngfw-configmap.yaml.

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

3. Utilisez l'application Kubectl pour exécuter le pan-cn-ngfw.yaml.

   kubectl apply -f pan-cn-ngfw.yaml

4. Vérifiez que les pods CN-NGFW sont en cours d'exécution.

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

STEP 7 | Vérifiez que vous pouvez voir CN-MGMT et CN-NGFW sur le cluster Kubernetes. Exécutez la commande suivante :

```
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
```

Déployer les pare-feu CN-Series

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Les pare-feu CN-Series sont faciles à déployer à l'aide de l'orchestration Kubernetes pour simplifier l'intégration de la sécurité réseau dans les processus d'intégration continue/développement continu (CI/CD). La gestion continue des pare-feu CN-Series est centralisée dans la gestion de la sécurité réseau Panorama™ (la même console de gestion que tous les pare-feu Palo Alto Networks), offrant aux équipes de sécurité réseau une interface unique pour gérer la posture globale de sécurité réseau de leurs organisations.

Ce chapitre couvre les sections suivantes :

Liste de vérification du déploiement de CN-Series

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec graphique Helm

Pour déployer le pare-feu CN-Series, vous devez effectuer les tâches suivantes :

- ☐ Si vous ne l'avez pas déjà fait, obtenez une licence pour le pare-feu CN-Series : générez votre code d'autorisation et gardez-le à portée de main lorsque vous êtes prêt à déployer le pare-feu CN-Series.
- ☐ Consultez les conditions préalables de CN-Series. Avant de commencer votre déploiement, assurez-vous de bien comprendre la configuration système requise pour déployer le pare-feu CN-Series.
- ☐ Préparez les composants.
  - Générez une clé d'authentification VM sur Panorama.
  - (Facultatif) Installez un certificat de périphérique sur le pare-feu CN-Series.
  - Créez des comptes de service pour l'authentification des clusters.
  - Déployer Panorama – vous devez utiliser Panorama pour configurer, déployer et gérer le déploiement de votre pare-feu CN-Series. Pour plus d'informations sur le déploiement et la configuration d'un appareil Panorama, voir Configurer Panorama.
  - Installez le plug-in Kubernetes pour le CN-Series.
  - Obtenez les images et les fichiers pour le déploiement de CN-Series : accédez au référentiel Palo Alto Networks pour télécharger les fichiers Docker et GitHub pour obtenir les fichiers yaml requis afin de déployer le pare-feu CN-Series dans votre environnement Kubernetes.
- ☐ Déployez le pare-feu CN-Series.
  - Modifiez le graphique HELM pour l'adapter à votre déploiement. Vous pouvez également modifier les fichiers yaml et passer en revue le Paramètres modifiables dans les fichiers YAML de déploiement CN-Series avant de déployer le pare-feu CN-Series. De nombreux paramètres définis dans les fichiers yaml doivent être modifiés pour déployer correctement le pare-feu CN-Series.
  - Déployer le pare-feu CN-Series en tant que service Kubernetes (mode de déploiement recommandé).
  - Déployer le pare-feu CN-Series en tant que DaemonSet.
  - (Facultatif) Si vous déployez votre pare-feu CN-Series en tant que service Kubernetes, vous pouvez Activer la mise à l'échelle automatique du pod horizontal sur CN-Series. La mise à

l'échelle automatique horizontale des pods (HPA) permet à votre déploiement de pare-feu CN-Series de se mettre à l'échelle dynamiquement avec votre environnement Kubernetes.

- Si vous déployez votre CN-Series dans un environnement OpenShift, consultez Déployer le pare-feu CN-Series sur Openshift.
- Si vous sécurisez un trafic 5G avec votre pare-feu CN-Series, reportez-vous à la section Sécuriser la 5G avec le pare-feu CN-Series.

- ☐ Configurer Panorama pour sécuriser un déploiement Kubernetes – après avoir déployé votre pare-feu CN-Series, utilisez Panorama pour configurer des politiques de sécurité qui permettent l'application du trafic et transmettre ces politiques au pare-feu.

Déployer les pare-feu CN-Series avec (recommandé) et sans le graphique Helm

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Le référentiel Helm contient des graphiques et des modèles pour le déploiement du pare-feu conteneurisé CN-Series de Palo Alto Networks à l'aide du gestionnaire de paquets Helm pour Kubernetes.

Vous pouvez télécharger les graphiques Helm CN-Series sur GitHub.

- Préparation à l'utilisation des graphiques et modèles Helm
- Déployer le pare-feu CN-Series à l'aide du graphique HELM (recommandé)
- Déployer le pare-feu CN-Series dans les fichiers YAML

Préparation à l'utilisation des graphiques et modèles Helm

Installez le logiciel requis. Ces instructions répertorient les versions minimales, mais vous pouvez installer une version ultérieure dans la même famille, sauf si une limite supérieure est spécifiée.

STEP 1 | Déployez des images de conteneur de pare-feu CN-Series 10.1.x, 10.2.x, 11.0.x ou 11.1.x.

STEP 2 | Installez une version de Kubernetes entre 1.16 et 1.25 et créez un cluster Kubernetes. Pour plus d'informations sur la version prise en charge de Kubernetes pour vos environnements, voir Environnements pris en charge par le déploiement CN-Series.

STEP 3 | Déployez Panorama dans un emplacement accessible à partir du cluster Kubernetes et du pare-feu CN-Series que vous utilisez pour sécuriser le cluster.

1. Assurez-vous que la version de Panorama PAN-OS est 10.x.x ou ultérieure.
2. Installez le plugin Kubernetes pour Panorama version 1.0.x ou 2.0.x.

STEP 4 | Installez le client Helm version 3.6.0 ou ultérieure.

Passez à la section Déployer le pare-feu CN-Series à l'aide du graphique HELM (recommandé) ou Déployer le pare-feu CN-Series dans les fichiers YAML.

Déployer le pare-feu CN-Series à l'aide du graphique HELM (recommandé)

Utilisez cette procédure pour cloner le référentiel et le déployer à partir de votre environnement local.

STEP 1 | Générez la clé d'authentification VM sur Panorama.

STEP 2 | Clonez le référentiel à partir de GitHub.

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

STEP 3 | Passez à un répertoire local pour le référentiel cloné. Par exemple :

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

STEP 4 | Accédez au sous-répertoire de votre déploiement.

- Utilisez le répertoire `helm_cnv1` pour déployer CN-Series comme DaemonSet.
- Utilisez le répertoire `helm_cnv2` pour déployer CN-Series en tant que service.
- Utilisez le répertoire `helm_cnv3` pour déployer CN-Series en tant que cnf.

STEP 5 | Téléchargez le compte de service YAML pour le fichier `plugin-serviceaccount.yaml` et appliquez le yaml. Le compte de service permet d'obtenir les autorisations dont Panorama a besoin pour s'authentifier auprès du cluster afin de récupérer les étiquettes Kubernetes et les informations sur les ressources. Ce compte de service est nommé `pan-plugin-user` par défaut. Exécutez la commande suivante pour déployer le fichier `plugin-serviceaccount.yaml` :

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

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

Pour consulter les secrets associés à ce compte de service.

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

Créez le fichier d'identification contenant les secrets, appelé `cred.json` dans cet exemple, puis enregistrez ce fichier. Vous devez télécharger ce fichier dans Panorama pour configurer le plug-in Kubernetes pour la surveillance des clusters dans Installer le plug-in Kubernetes pour pare-feu CN-Series.

> *Sur **Openshift**, vous devez déployer manuellement le fichier `pan-cni-net-attach-def.yaml` pour chaque fichier d'espace de noms Openshift avant de déployer les graphiques Helm.*

STEP 6 | Modifiez le `fichier values.yaml` pour entrer vos informations de configuration. Les valeurs suivantes proviennent du sous-répertoire `helm_cnv1`.

```
# L'environnement K8s # Les étiquettes deployTo valides sont :
 [gke|eks|aks||native] # Les étiquettes multus valides sont :
 [enable|disable] Garder le multus comme activable pour les
```

```
 déploiements openshift et natifs. 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 | Affichez les fichiers YAML rendus.

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

STEP 8 | Effectuez une vérification pelucheuse sur les graphiques Helm.

```
helm lint helm_cnv1/
```

STEP 9 | Déployez les graphiques HELM.

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

Les demandes de volume persistantes ne sont pas supprimées lors de la désinstallation d'un graphique HELM. Vous devez vous assurer d'effacer ces demandes au préalable pour que l'installation HELM fonctionne.

Pour plus d'informations HELM, voir HELM Classic : un gestionnaire de paquets Kubernetes.

Déployer le pare-feu CN-Series dans les fichiers YAML

Pour déployer sans cloner le référentiel, ajoutez le référentiel à votre client Helm.

STEP 1 | Générez la clé d'authentification VM sur Panorama.

STEP 2 | Téléchargez le compte de service YAML pour le fichier `plugin-serviceaccount.yaml` et appliquez le yaml. Le compte de service permet d'obtenir les autorisations dont Panorama a besoin pour s'authentifier auprès du cluster afin de récupérer les étiquettes Kubernetes et les informations sur les ressources. Ce compte de service est nommé `pan-plugin-`

`user` par défaut. Exécutez la commande suivante pour déployer le fichier `plugin-serviceaccount.yaml` :

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

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

Pour consulter les secrets associés à ce compte de service.

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

Créez le fichier d'identification contenant les secrets, appelé `cred.json` dans cet exemple, puis enregistrez ce fichier. Vous devez télécharger ce fichier dans Panorama pour configurer le plug-in Kubernetes pour la surveillance des clusters dans Installer le plug-in Kubernetes pour pare-feu CN-Series.

*Sur **Openshift**, vous devez déployer manuellement le fichier `pan-cni-net-attach-def.yaml` pour chaque fichier d'espace de noms Openshift avant de déployer les graphiques Helm.*

STEP 3 | Ajoutez le référentiel CN-Series à votre client Helm local.

Entrez cette commande sur une seule ligne :

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

```
« cn-series » a été ajouté à vos référentiels
```

STEP 4 | Vérifiez que le référentiel a été ajouté à votre client Helm.

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

STEP 5 | Sélectionnez le cluster Kubernetes.

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


STEP 6 | Déployez à l'aide du référentiel de graphiques Helm. Modifiez la commande suivante pour inclure vos informations de configuration.

```
$ 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"
```

Les demandes de volume persistantes ne sont pas supprimées lors de la désinstallation d'un graphique HELM. Vous devez vous assurer d'effacer ces demandes au préalable pour que l'installation HELM fonctionne.

Déployer un pare-feu CN-Series avec des modèles Terraform

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Terraform 0.13.0 ou version supérieure

Le référentiel de déploiement CN-Series contient des plans Terraform pour déployer un cluster GKE, EKS ou AKS. Ces plans garantissent que le dimensionnement des nœuds du cluster et les interfaces réseau de conteneurs (CNI) prennent en charge un déploiement de pare-feu CN-Series au sein du cluster. Le référentiel fournit également un plan de déploiement de pare-feu CN-Series et un exemple d'application de livre d'or PHP que vous pouvez sécuriser avec le pare-feu.

Cette procédure a le workflow facultatif suivant :

- Préparation à l'utilisation des graphiques et modèles Helm
- Déployer un exemple d'application
- Déployer un pare-feu CN-Series avec Terraform
- Configurer le plug-in Kubernetes pour Panorama

Déployer un exemple d'application

Le référentiel GitHub de Palo Alto Networks comprend un exemple d'application pris en charge par la communauté avec un fichier manifeste Kubernetes nommé `guestbook.yml`.

Ce fichier déploie une application Web de livre d'or PHP simple qui utilise un backend Redis.

STEP 1 | Dans le répertoire cn-series-deploy du référentiel GitHub de Palo Alto Networks, apportez une modification au répertoire sample-application.

```
$ cd sample-application
```

STEP 2 | Déployez l'application de livre d'or.

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

STEP 3 | Vérifiez que les pods d'application ont été déployés et atteignent l'état En cours d'exécution, puis Prêt.

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

```
NAME READY STATUS RESTARTS AGE frontend-69859f6796-96bs7
 1/1 Running 0 111m frontend-69859f6796-k2k4z 1/1 Running
 0 53m frontend-69859f6796-zwwbg 1/1 Running 0 111m redis-
```

```
master-596696dd4-5l5qv 1/1 Running 0 53m redis-slave-6bb9896d48-
dwhw2 1/1 Running 0 53m redis-slave-6bb9896d48-nhqzh 1/1 Running 0
 111m
```

STEP 4 | Répertoriez les services pour déterminer l'adresse IP publique de l'interface Web.

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

Vous pouvez désormais configurer vos groupes d'adresses dynamiques et vos règles de sécurité sur Panorama pour sécuriser l'application de livre d'or.

Continuez à déployer le pare-feu CN-Series à l'aide de Terraform.

Déployer un pare-feu CN-Series avec Terraform

Déployez le pare-feu CN-Series à l'aide de Terraform.

STEP 1 | Utilisez votre `cn-series\tfvars` local pour créer un fichier nommé `terraform.tfvars`, et ajoutez les variables suivantes et leurs valeurs associées.

```
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 | Validez le plan Terraform.

```
$ terraform init
```

STEP 3 | Validez le plan Terraform.

```
$ terraform plan
```

STEP 4 | Appliquez le plan Terraform.

```
$ terraform apply
```

STEP 5 | Vérifiez que les pods ont été déployés et qu'ils sont prêts et que l'état est En cours d'exécution.

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

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

Vous êtes prêt à configurer le plug-in Kubernetes pour Panorama.

Configurer le plug-in Kubernetes pour Panorama

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure

Utilisez le plug-in Kubernetes pour Panorama pour propager des étiquettes à des groupes d'appareils Panorama.

Vous pouvez utiliser le plug-in Kubernetes pour compléter l'intégration de Panorama et de l'API Kubernetes. Le plug-in apprend de nouvelles étiquettes et les propage aux groupes d'appareils Panorama. Ces étiquettes peuvent inclure des étiquettes Kubernetes, des services, des espaces de noms et d'autres métadonnées à partir desquelles des critères de correspondance de groupe d'adresses dynamiques peuvent être définis.

Si la taille du fichier d'informations d'identification du cluster est supérieure à 32 Ko, vous recevrez un message d'erreur lors de l'importation du fichier d'informations d'identification sur le plug-in Panorama Kubernetes. Le message d'erreur affiche la taille du fichier comme cause de l'erreur.

Si le cluster possède de nombreux certificats CA dans le bundle `ca.crt`*, le plug-in Kubernetes nécessite uniquement le certificat CA supérieur. Vous devez vous assurer de conserver uniquement le certificat CA supérieur et de supprimer tous les autres certificats CA et* `service.crt` *du fichier d'informations d'identification. Vous pouvez ensuite utiliser ce fichier d'informations d'identification mis à jour.*

Cette procédure suppose que vous avez installé le logiciel de prise en charge répertorié dans Préparation à l'utilisation des graphiques et modèles Helm.

STEP 1 | Récupérez les informations d'identification du compte de service pan-plugin-user à partir du maître Kubernetes.

Entrez chaque commande sous la forme d'une seule ligne :

```
$ 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 | Créez une définition de cluster dans le plug-in Panorama Kubernetes.

Utilisez l'adresse principale Kubernetes affichée dans la sortie Terraform et le fichier d'informations d'identification JSON situé dans `~/Downloads/pan-plugin-user.json`.

Définissez les étiquettes que vous souhaitez importer à partir de l'API Kubernetes.

STEP 3 | Créez une définition de groupe de notification dans le plug-in Panorama Kubernetes.

Cette définition est utilisée pour propager les étiquettes apprises de l'API Kubernetes à un groupe d'appareils Panorama.

Effectuez les étapes suivantes pour créer un groupe de notification dans le plug-in Panorama Kubernetes :

1. Sélectionnez **Panorama** > **Plugins (Plug-ins)** > **Kubernetes** > **Setup (Configuration)** > **Notify Groups (Groupes de notification)**et **Add (Ajouter)**.

2. Donnez un **Name (Nom)** au groupe de notification (31 caractères au maximum).
3. Sélectionnez **Enable sharing internal tags with Device Groups** (Activer le partage d'étiquettes internes avec les groupes d'appareils) si vous souhaitez partager des étiquettes internes en plus des étiquettes externes (par défaut) créées pour le cluster.
4. Sélectionnez les groupes d'appareils sur lesquels vous souhaitez enregistrer les étiquettes.

5. Cliquez sur **Ok**.

STEP 4 | Créez une définition de surveillance dans le plug-in Panorama.

Utilisez les définitions de cluster et de groupe de notification créées dans les étapes précédentes.

STEP 5 | Validez sur Panorama).

STEP 6 | Pour confirmer la connectivité API et les enregistrements de conteneur MP, accédez à la définition de surveillance et cliquez sur Detailed Status (État détaillé) et Cluster MPs (MP de cluster).

Vous êtes maintenant prêt à déployer une application et à la sécuriser avec le pare-feu CN-Series.

Déployer le pare-feu CN-Series avec Rancher Orchestration

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure

Vous pouvez désormais déployer le pare-feu CN-Series en tant que service Kubernetes à l'aide de l'orchestration Rancher et de PAN OS 10.1. Rancher est une plate-forme d'orchestration de conteneurs open source que vous pouvez utiliser pour déployer le pare-feu CN-Series.

Pour le déploiement du pare-feu CN-Series avec prise en charge du cluster Rancher, votre instance Panorama doit avoir 16 vCPU, 32 Go de mémoire avec un disque supplémentaire de 2 To. Panorama sera déployé dans un mode qui facilitera la collecte des journaux des déploiements de pare-feu CN-Series.

Lors du déploiement du pare-feu CN-Series dans le cluster On-Premises Rancher Kubernetes, procédez comme suit :

- Assurez-vous que les composants requis pour sécuriser les clusters Kubernetes avec le pare-feu CN-Series sont disponibles.
- Assurez-vous que le cluster Kubernetes répond à la configuration minimale requise. Pour plus d'informations, reportez-vous à la section Configuration système requise pour CN-Series.
- Effectuez la procédure Déployer le pare-feu CN-Series avec Rancher Orchestration.
-
- Modifier le fichier YAML des options du cluster Rancher
- Installez le plug-in Kubernetes pour le pare-feu CN-Series.
- Obtenez la licence pour le pare-feu CN-Series.
- Déployer le pare-feu CN-Series en tant que service Kubernetes (mode de déploiement recommandé) sur Rancher.

Déploiement du cluster Rancher

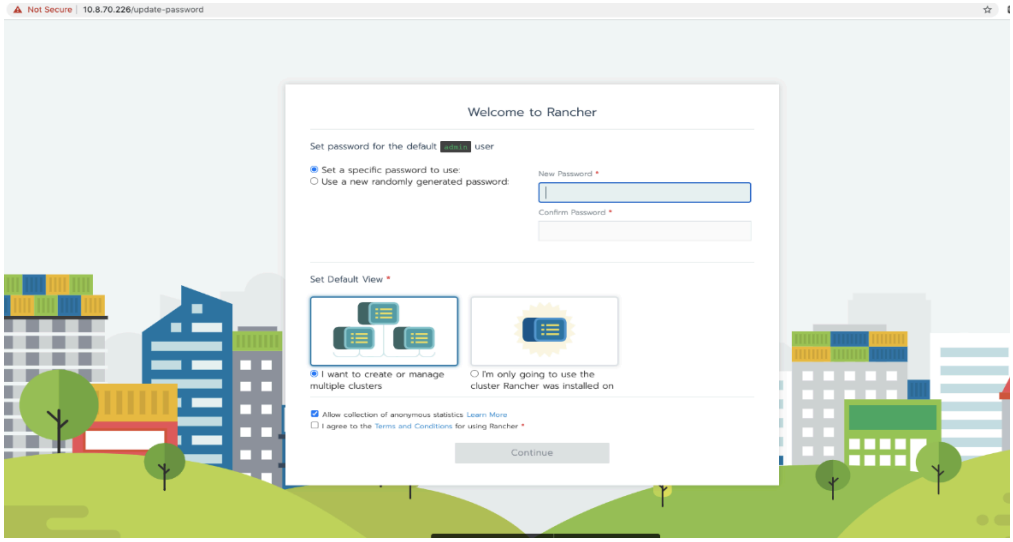
Vous pouvez déployer Rancher en deux étapes :

1. Préparez un hôte Linux avec une distribution Linux prise en charge et 4 Go de mémoire. Installez une version prise en charge de Docker sur l'hôte.
2. Démarrez le serveur.

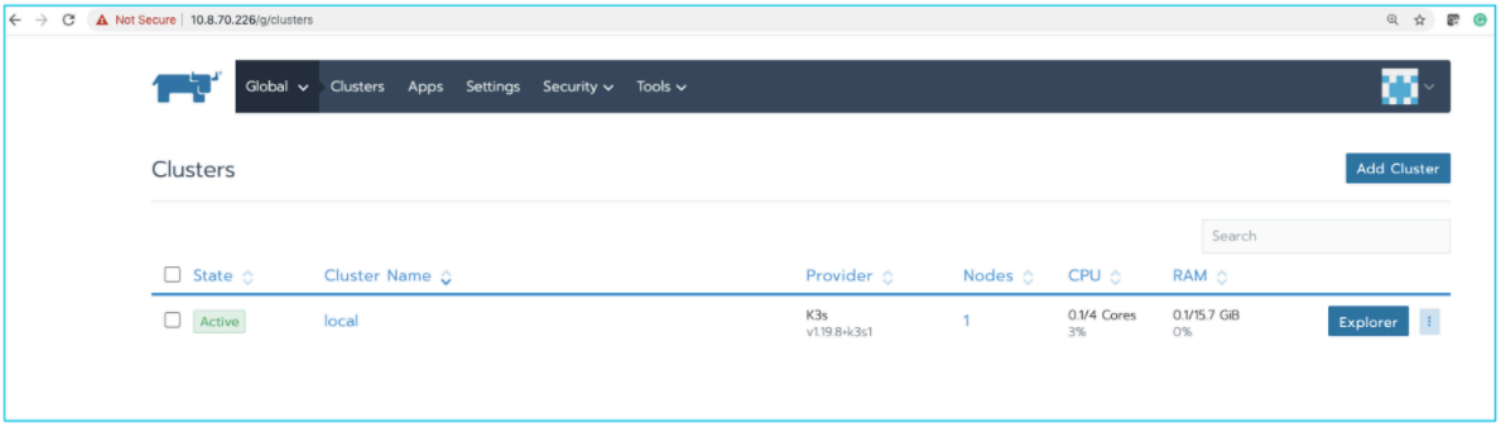
   Pour installer et exécuter Rancher, exécutez la commande Docker suivante sur votre hôte :

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

Après un déploiement réussi, vous pouvez accéder à l'interface utilisateur du serveur Rancher et configurer un mot de passe pour l'utilisateur admin. Pour accéder à l'interface utilisateur du serveur Rancher, ouvrez un navigateur et accédez au nom d'hôte ou à l'adresse où le conteneur a été installé. Vous serez guidé tout au long de la configuration de votre premier cluster.



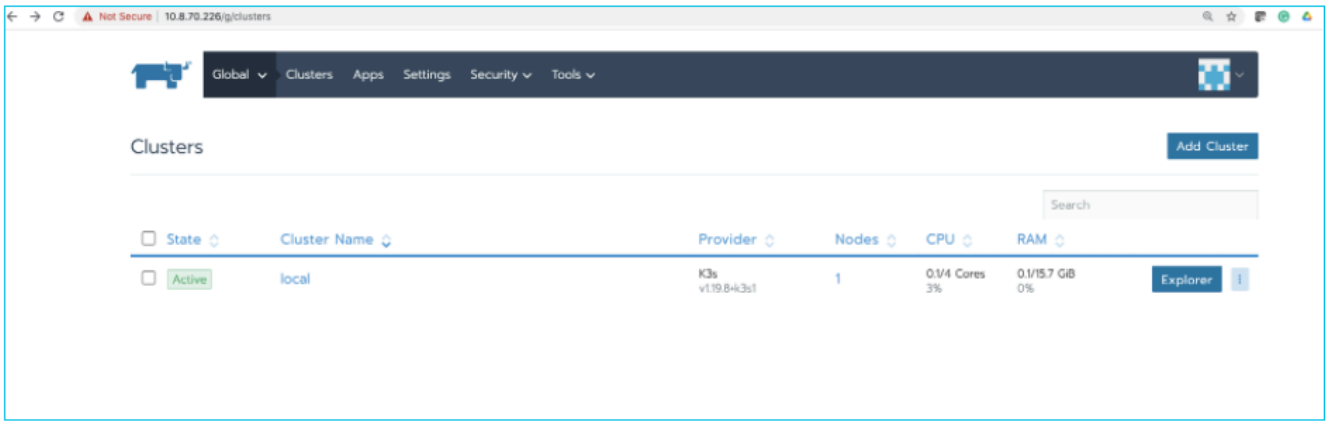
Suite à la création de l'utilisateur admin, un cluster local sera créé comme indiqué ci-dessous :



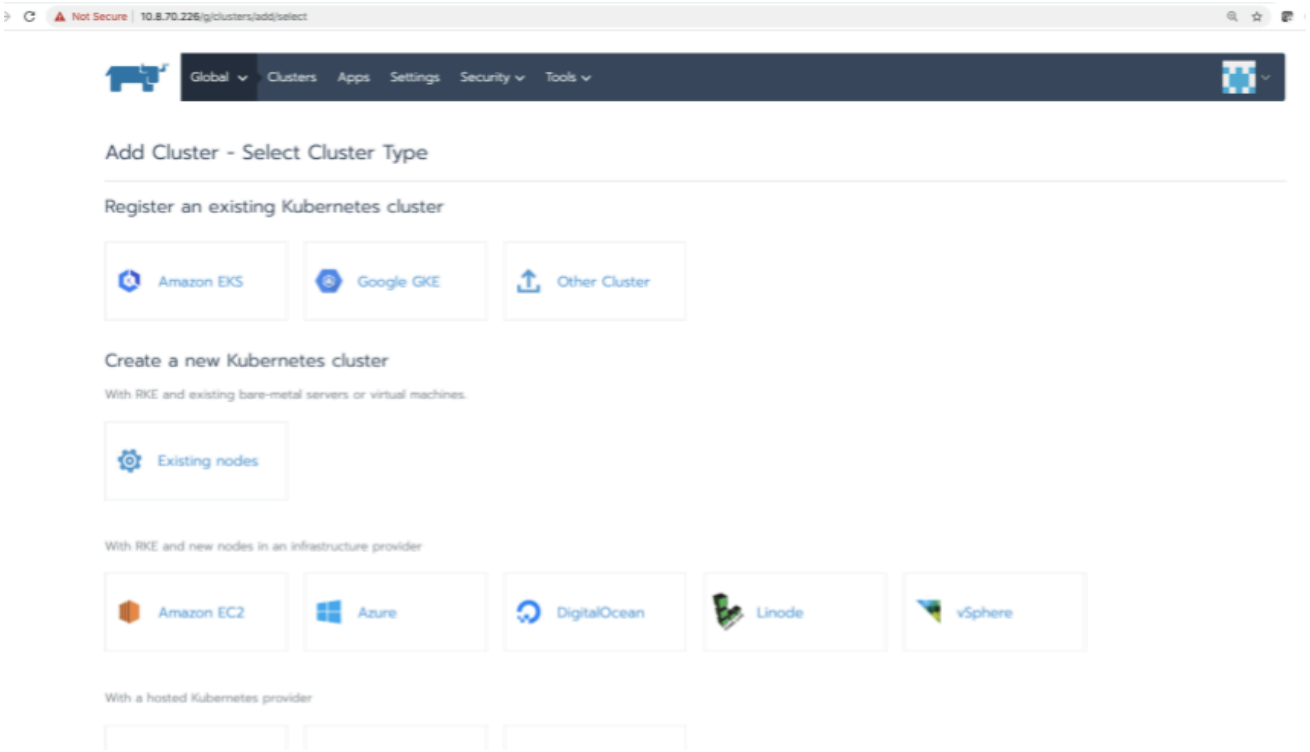
Configurer le nœud maître et le nœud esclave sur le cluster Rancher

Après avoir créé un cluster local sur l'interface utilisateur Rancher, configurez un nœud maître et esclave, puis procédez comme suit :

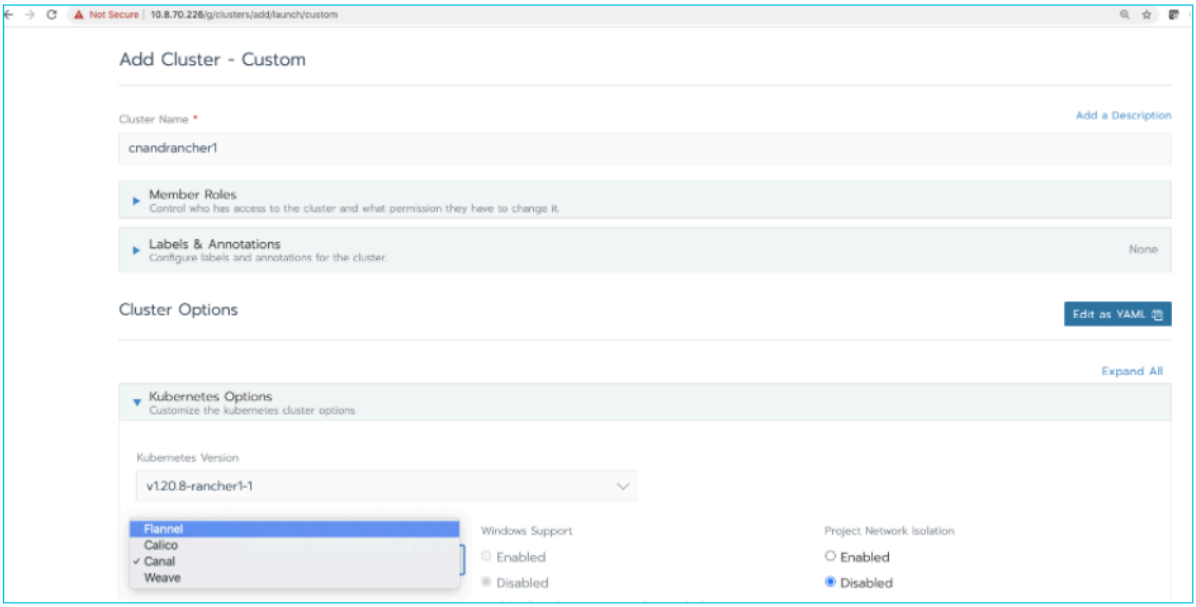
1. Accédez à l'interface utilisateur de Rancher et cliquez sur **Add Cluster** (Ajouter un cluster).



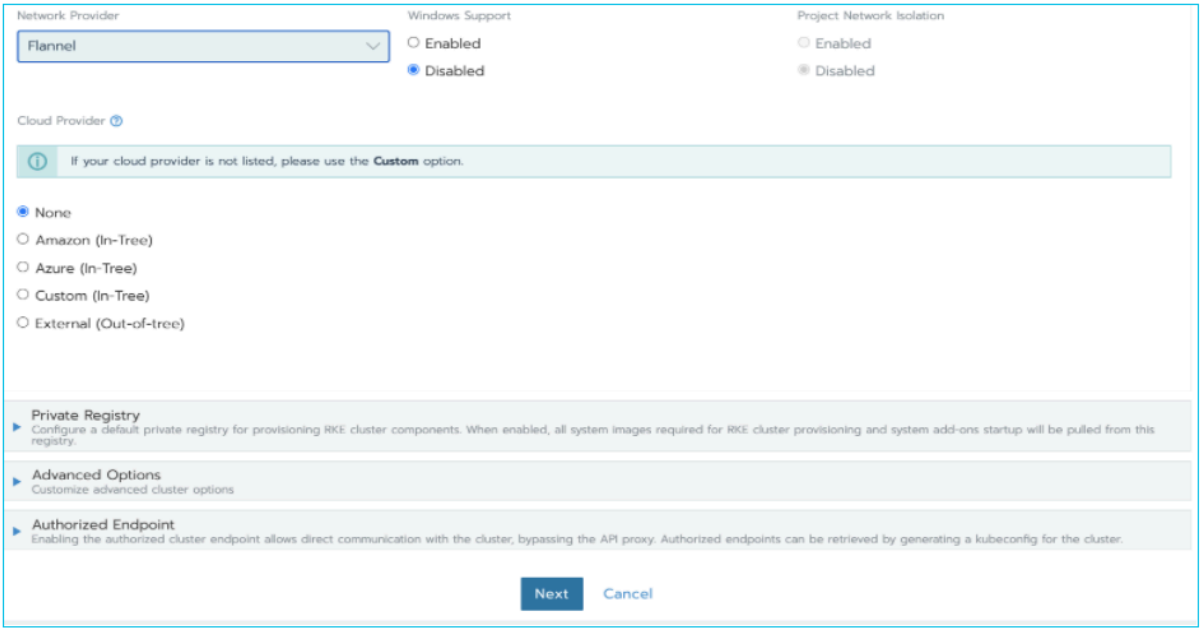
2. Cliquez sur **Existing nodes** (Nœuds existants).



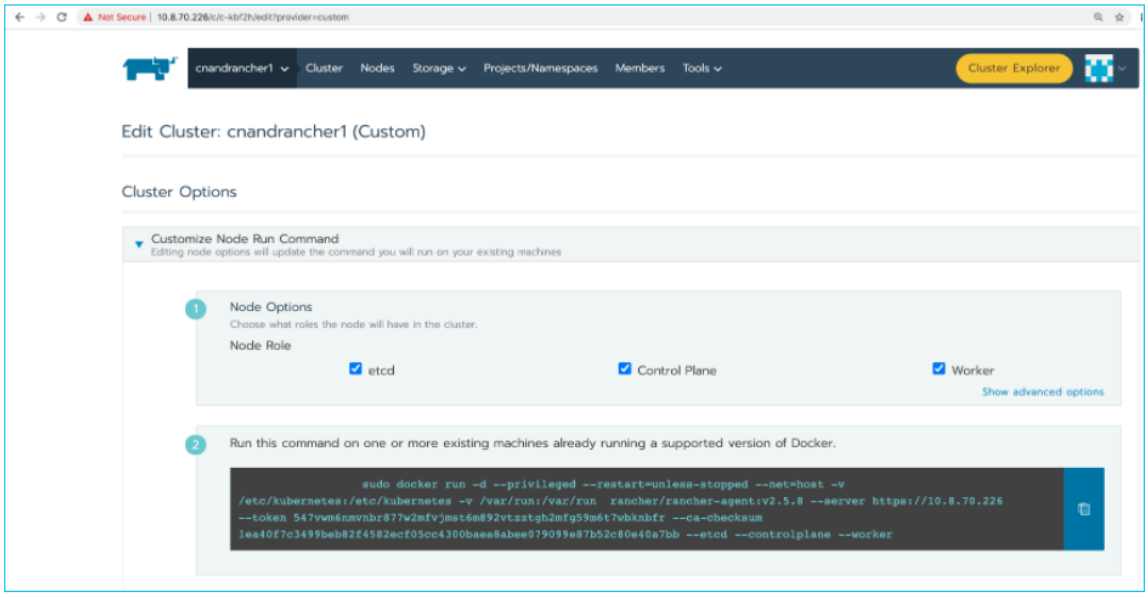
3. Entrez le **Cluster name** (Nom du cluster), puis sélectionnez Flannel dans la liste déroulante **Network provider** (Fournisseur de réseau).



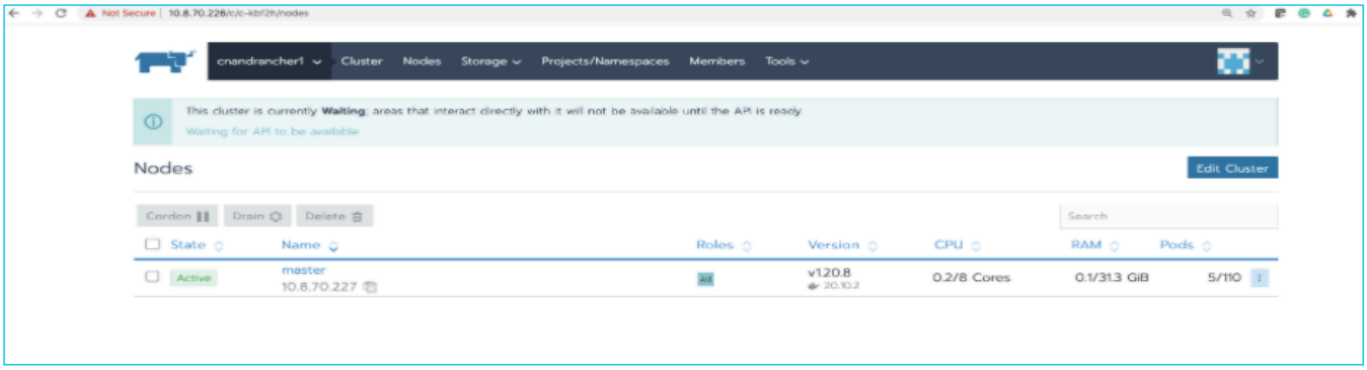
4. Conservez les valeurs par défaut de tous les autres champs, puis cliquez sur **Next** (Suivant).



5. Sous Node options (Options de nœud), sélectionnez les trois **Node Role** (Rôle de nœud), puis exécutez la commande donnée dans le nœud maître à l'aide de SSH.



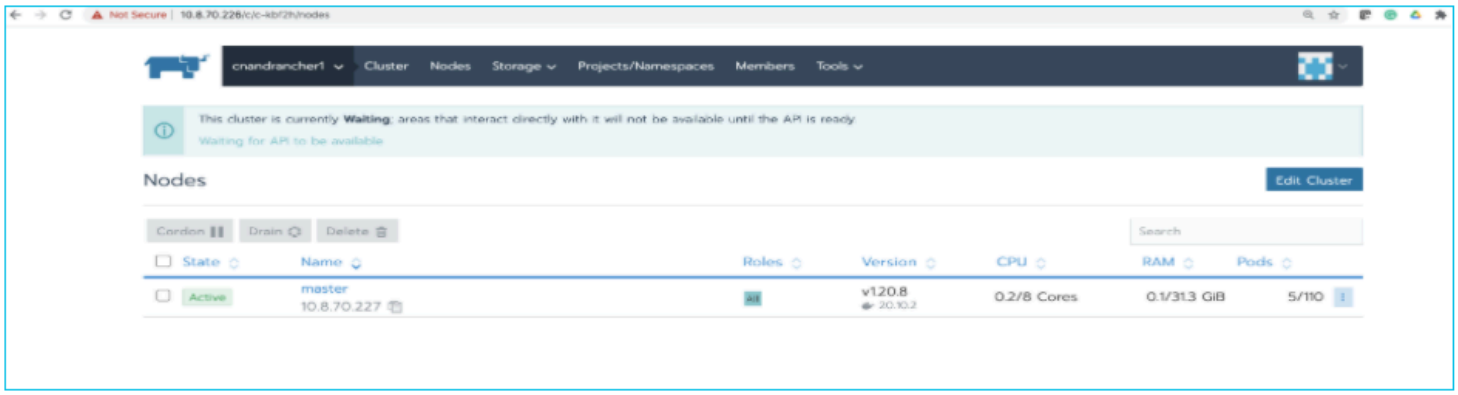
6. Vérifiez que le nœud maître est ajouté avec succès.



7. SSH into each Worker node and run the following command:

```
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
```

Lors de l'exécution réussie de la commande sur un nœud maître et deux nœuds esclaves, vous verrez que le cluster Rancher est prêt comme indiqué ci-dessous :



Modifier le fichier YAML des options du cluster Rancher

Avant de déployer le pare-feu CN-Series, vous devez modifier le fichier YAML des options de cluster comme décrit ci-dessous.

Le pare-feu CN-Series avec Rancher prend en charge Rancher 2.5 ou version ultérieure avec k8s 1.20.5.

STEP 1 | À l'aide des informations d'identification d'administrateur que vous avez créées précédemment, connectez-vous au portail Rancher.

STEP 2 | Cliquez sur le **Navigation Menu (Menu de navigation)** et sélectionnez **Cluster Management (Gestion du cluster)**.

STEP 3 | Recherchez le cluster à modifier, cliquez sur le menu ellipses vertical, puis sélectionnez **Edit Config (Modifier la configuration)**.

STEP 4 | Cliquez sur **Edit as YAML** (Modifier en tant que YAML).

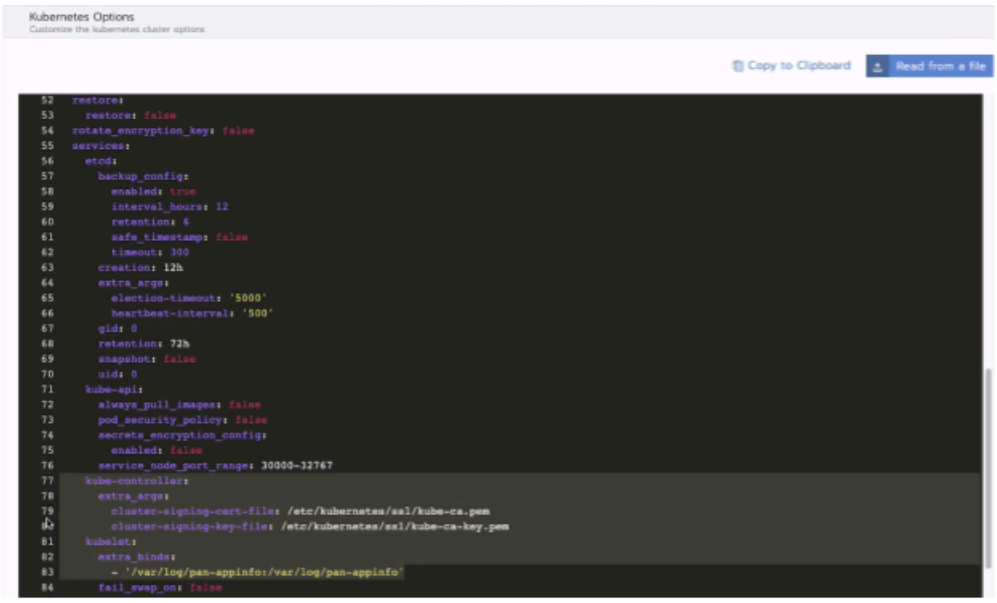
Pour connaître les différentes versions de Rancher, consultez la documentation Rancher.

STEP 5 | Ajoutez les lignes suivantes sous la section **Services** dans le fichier YAML existant.

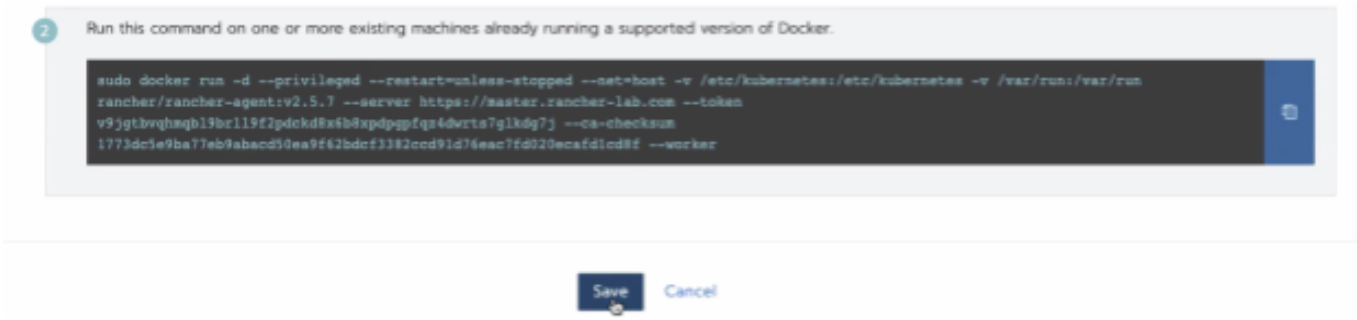
```
kube-controller: extra_args: cluster-signing-cert-file: "/etc/
kubernetes/ssl/kube-ca.pem" cluster-signing-key-file: "/etc/
kubernetes/ssl/kube-ca-key.pem"
```

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

Si vous utilisez un chemin de stockage autre que `'/mnt'`, vous devez vous assurer de modifier le chemin de stockage sous le `extra_binds`.



STEP 6 | Cliquez sur **Save (Enregistrer)** et attendez que la mise à niveau du cluster devienne active avant de déployer le pare-feu CN-Series.



Paramètres modifiables dans les fichiers YAML de déploiement CN-Series

Les fichiers YAML comprennent plusieurs paramètres modifiables. Les tableaux suivants énumèrent ceux que vous devez modifier pour Déployer les pare-feu CN-Series correctement.

- 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	
Routage avancé (requis pour les déploiements Kubernetes 3.0.0) `PAN_ADVANCED_ROUTING:”true”`	Si vous utilisez le routage avancé avec le plug-in Kubernetes 3.0.0, vous devez d’abord le configurer de manière à l’activer dans PAN-OS, puis manuellement sur la pile de modèles. Après l’avoir activé, validez et appliquez la configuration. Pour plus d’informations, reportez-vous à la section Routage avancé.
Adresse IP Panorama `PAN_PANORAMA_IP:`	Indiquez l’adresse IP de Panorama à laquelle le Pod CN-MGMT se connectera. Si vous avez configuré vos serveurs de gestion Panorama dans une configuration High Availability (haute disponibilité - HA), indiquez l’adresse IP du Panorama primaire actif. Vous pouvez localiser l’adresse IP Panorama dans **Dashboard** > **General Information**.
Nom du groupe d’appareils `PAN_DEVICE_GROUP:`	Indiquez le nom du groupe d’appareils auquel vous souhaitez attribuer les pods CN-NGFW. Depuis Panorama, vous appliquerez des politiques identiques à tous les Pods CN-NGFW qui sont gérés par une paire de Pods CN-MGMT (ou qui appartiennent à un PAN-SERVICE-NAME).

PAN-CN-MGMT-CONFIGMAP	
	Vous pouvez localiser le nom du groupe de périphériques dans **Panorama** > **Device Groups** (Groupe d'appareils).
Nom de pile de modèles `PAN_TEMPLATE_STACK:`	Vous permet de configurer les paramètres qui permettent aux pare-feu (Pods CN-NGFW) de fonctionner sur le réseau. Vous pouvez localiser le nom de la pile de modèles dans **Panorama** > **Templates**.
Nom du groupe de collecteurs de journaux `PAN_PANORAMA_CGNAME:`	Permet le stockage des journaux générés sur les pare-feu CN-NGFW. Sans groupe de collecteurs, les journaux du pare-feu ne sont pas sauvegardés. Vous pouvez localiser le nom du groupe de collecteurs dans **Panorama** > **Collector Groups**.
(Facultatif) `#CLUSTER_NAME:`	Précisez le nom du cluster. Le nom d'hôte du pod CN-MGMT combine le nom StatefulSet défini dans le `PAN-CN-MGMT.yaml` et ce CLUSTER_NAME facultatif. Ce nom d'hôte vous permet d'identifier les pods qui sont associés à différents clusters, si vous gérez plusieurs clusters sur le même appareil Panorama. La meilleure pratique consiste à utiliser le même nom ici et sur le plug-in Kubernetes sur Panorama.
(Facultatif) Adresse IP de l'homologue HA Panorama `#PAN_PANORAMA_IP2:`	Adresse IP de l'homologue Panorama (passif secondaire) qui est configuré dans une configuration High Availability (haute disponibilité - HA). Vérifiez que le PAN_PANORAMA_IP est celui du Panorama primaire actif. Vous pouvez localiser l'adresse IP du pair Panorama HA dans **Panorama** > **High Availability** > **Setup**.
(Requis pour GTP) Sécurité GTP `#PAN_GTP_ENABLED: "true"`	Activez ce paramètre pour la sécurité GTP sur le pare-feu CN-Series. Après avoir activé GTP, vous pouvez utiliser Panorama pour configurer la sécurité GTP et surveiller le trafic GTP sur le pare-feu.
(Requis pour le support de trame Jumbo, si le CNI primaire n'utilise pas de trame Jumbo) Mode de trame Jumbo	Le pod CN-MGMT pendant le démarrage utilise le MTU eth0 pour détecter automatiquement s'il faut activer le mode de

PAN-CN-MGMT-CONFIGMAP	
`#PAN_JUMBO_FRAME_ENABLED: "true"`	trame Jumbo. Ainsi, si votre CNI secondaire utilise des trames Jumbo, alors que le CNI primaire ne le fait pas, vous devez définir `PAN_JUMBO_FRAME_ENABLED: "True"` pour activer le mode de trames Jumbo sur le pare-feu VM-Series. Vous devez effectuer ce changement avant que le CN-MGMT StatefulSet ne soit déployé.

(Requis pour une allocation flexible des ressources du système)
• CN-Series en tant que DaemonSet
`#PAN_NGFW_MEMORY: "42Gi"`
• CN-Series en tant que service K8s
`#PAN_NGFW_MEMORY: "6.5Gi"`
`#PAN_NGFW_MEMORY: "42Gi"`
 	
Pour la sécurité 5G-Native, 48Gio est recommandé
	Si vous avez besoin d'une capacité plus élevée et souhaitez configurer plus de mémoire pour répondre à vos besoins de déploiement, définissez la valeur de la mémoire à l'aide de ce paramètre.
	• CN-Series en tant que DaemonSet
	La petite capacité est de 42 Gi ou moins et la grande capacité est supérieure à 42 Gi.
	• CN-Series en tant que service K8s
	La petite capacité est inférieure à 6,5 Gi, la capacité moyenne est comprise entre 6,5 Gi et 42 Gi et la grande capacité est supérieure à 42 Gi.
	
	Ce changement nécessite également une allocation de mémoire identique ou supérieure sur le fichier `pan-cn-ngfw.yaml`.

(Facultatif) AF-XDP `#PAN_DATA_MODE: "next-gen"`	Ce paramètre est requis pour activer le chemin de données eXpress de la famille d'adresses (AF-XDP). AF-XDP est un socket basé sur eBPF qui est optimisé pour le traitement de paquets hautes performances adapté aux services natifs du cloud, afin d'augmenter le débit effectif. Cela nécessite la version 5.4 ou ultérieure du noyau. De plus, le mode jumbo n'est pas pris en charge ; EKS ne peut pas utiliser ce paramètre car le mode jumbo est activé par défaut. De plus, le mode privilégié est requis dans PAN-CN-NGFW.

PAN-CN-MGMT-CONFIGMAP	
(Requis pour activer HPA) (AKS et GKE) `#HPA_NAME` (EKS uniquement) `#PAN_NAMESPACE_EKS` (AKS uniquement) `#PAN_INSTRUMENTATION_KEY`	Plusieurs paramètres sont requis pour activer la mise à l'échelle horizontale des pods (HPA) sur le pare-feu CN-Series en tant que service. • Pour chaque environnement, vous devez fournir un nom unique pour identifier la ressource HPA par espace de noms ou par locataire. • Pour le déploiement AKS, vous devez fournir une clé d'instrumentation Azure Application Insight.

Les valeurs par défaut suivantes sont définies dans le fichier `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
```

Ces valeurs par défaut vous permettent d'utiliser ces fichiers pour une preuve de concept rapide. Si vous souhaitez les modifier, par exemple pour déployer plus d'une paire de PAN-MGMT à tolérance de pannes qui gèrent jusqu'à 30 Pods PAN-NGFW, vous devez modifier `pan-mgmt-svc` pour utiliser un autre nom de service. Lorsque vous modifiez ces valeurs, vous devez mettre à jour les références correspondantes dans les autres fichiers YAML pour qu'elles correspondent aux valeurs que vous définissez dans ce fichier.

PAN-CN-MGMT-SECRET

PAN-CN-MGMT-SECRET	
Clé d'authentification VM `PAN_PANORAMA_AUTH_KEY:`	Permet à Panorama d'authentifier les pare-feu afin de pouvoir ajouter chaque pare-feu en tant qu'appareil géré. La clé d'authentification VM est requise pour la durée de vie du déploiement. Sans clé valide dans la requête de connexion, le pare-feu CN-Series ne pourra pas s'enregistrer auprès de Panorama. Voir Installer le plug-in Kubernetes pour le pare-feu CN-Series.

Certificat de périphérique pour CN-Series
`CN-SERIES-AUTO-REGISTRATION-PIN-ID`
`CN-SERIES-AUTO-REGISTRATION-PIN-VALUE`
	Le pare-feu a besoin du certificat de périphérique pour obtenir toute licence de site et accéder de manière sécurisée aux services fournis dans le cloud de Palo Alto. Générez l'ID et la valeur du PIN sur le CSP de Palo Alto Networks, et utilisez le PIN avant qu'il n'expire. Par exemple :

PAN-CN-MGMT-SECRET	

	`CN-SERIES-AUTO-REGISTRATION-PIN-ID:`
	`"01cc5-0431-4d72-bb84-something”`
	`CN-SERIES-AUTO-REGISTRATION-PIN-VALUE:`
	`"12.....13e"`
	Le champ supplémentaire suivant pour SÉRIE-CN - AUTO-REGISTRATION-API-CSP est commenté et n’est pas requis : `"certificate.paloaltonetworks.com"`
	Voir Installer un certificat de périphérique sur le pare-feu CN-Series.

PAN-CN-MGMT

PAN-CN-MGMT	
Chemin d’accès de l’image du conteneur Init pour le pare-feu CN-MGMT `initContainers: - name: pan-mgmt-init image: <your-private-registry-image-path>`	Le conteneur init génère des certificats qui sont utilisés pour sécuriser la communication entre les instances des pods CN-MGMT et entre les pods CN-MGMT et les pods CN-NGFW. Modifiez le chemin d’accès de l’image pour qu’elle pointe vers l’emplacement où vous avez téléchargé l’image du docker pour le conteneur CN-MGMT.
Chemin d’accès de l’image pour les conteneurs d’images CN-MGMT : `initContainers: - name: pan-mgmt image: <your-private-registry-image-path>`	Modifiez le chemin d’accès de l’image pour qu’elle pointe vers l’emplacement où vous avez téléchargé l’image du docker pour le conteneur CN-MGMT.
Nom d’hôte du pare-feu CN-MGMT `kind: StatefulSet metadata: name: pan-mgmt-sts`	Le nom d’hôte du pare-feu CN-MGMT est dérivé en combinant le nom StatefulSet et le nom de cluster optionnel que vous avez peut-être défini dans le fichier `pan-cn-mgmt-configmap.yaml`. Le nom d’hôte par défaut des pods CN-MGMT est pan-mgmt-sts-0 et pan-mgmt-

PAN-CN-MGMT	
	sts-1, car le nom StatefulSet est pan-mgmt-sts et le nom du cluster n'est pas défini. *Si le nom d'hôte comporte plus de 30 caractères, le nom sera tronqué à 30 caractères.*
(Requis si vous avez défini la mémoire pour une allocation flexible des ressources du système)	Si vous avez attribué une valeur de mémoire supérieure ou égale à 40Gio pour `#PAN_NGFW_MEMORY: «40Gi»`dans le fichier `pan-cn-mgmt-configmap.yaml`, assurez-vous que vous avez des valeurs identiques dans `request` et `limit` pour le processeur et la mémoire pour obtenir une utilisation plus élevée de la capacité sous `containers: resources: requests: # configurable based on desired logging, capacities cpu: "4" memory: "16.0Gi" limits: cpu: "4" memory: "16.0Gi"` Pour la sécurité 5G-Native, les valeurs recommandées sont processeur = 4, mémoire = 16 Gio
(Uniquement pour un déploiement sur site ou autogéré de Native Kubernetes) `storageClassName: local`	Pour un déploiement autogéré, la configuration par défaut est « storageClassName: local ». Si votre cluster a provisionné dynamiquement des volumes persistants (PV), vous devez modifier le « storageClassName: local » pour qu'il corresponde à cette storageClass ou supprimer ces lignes si DefaultStorageClass est utilisé. Si votre cluster n'a pas de PV provisionné dynamiquement, l'administrateur du cluster peut créer des PV statiques avec le fichier `pan_cn_pv_local.yaml` fourni qui comporte 2 ensembles de quelques PV, un pour chaque pod statefulSet PAN-CN-MGMT. Vous pouvez modifier `pan_cn_pv_local.yaml` pour qu'il corresponde aux volumes de votre configuration et le déployer avant de déployer le `PAN-CN-MGMT.yaml`.

PAN-CN-NGFW-CONFIGMAP

Vous n'avez pas besoin de modifier les valeurs PAN, à moins que vous ne deviez modifier les éléments suivants :

- `PAN_SERVICE_NAME: pan-mgmt-svc`

  Le nom du service doit correspondre à celui que vous avez défini sur PAN-CN-MGMT-CONFIGMAP.
- `FAILOVER_MODE: failopen`

  Vous pouvez changer ce mode en « failclose ». Il ne prend effet que lorsque CN-NGFW ne parvient pas à obtenir une licence.
  - En mode « fail-open », le pare-feu reçoit le paquet et l'envoie sans l'inspecter. Le passage en mode « fail-open » entraîne un redémarrage interne et une brève interruption du trafic.
  - En mode « fail-close », le pare-feu supprime tous les paquets qu'il reçoit. Le mode « fail-close » réduit le CN-NFGW et libère l'emplacement alloué pour permettre à d'autres CN-NFGW autorisés d'utiliser cet emplacement.
- Épinglage du processeur - Dans le `pan-cn-ngfw-configmap.yaml`, l'épinglage du processeur et l'hyperthreading sont désactivés. Ne changez pas ce paramètre pour activer l'épinglage du processeur pour des cœurs physiques dédiés au lieu de cœurs logiques avec hyperthreading, à moins d'être guidé par l'assistance de Palo Alto Networks.

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

PAN-CN-NGFW

PAN-CN-NGFW	
Chemin d'image pour l'image du conteneur CN-NGFW image `containers: - name: pan-ngfw-container image: <your-private-registry-image-path>`	Modifiez le chemin d'accès de l'image pour qu'elle pointe vers l'emplacement où vous avez téléchargé l'image du docker pour le conteneur CN-NGFW.
(Requis si vous avez défini la mémoire pour une allocation flexible des ressources du système)	Si vous avez attribué une valeur de mémoire supérieure ou égale à 40Gio pour `#PAN_NGFW_MEMORY: "40Gi"`dans le fichier `pan-cn-mgmt-configmap.yaml`, assurez-vous que vous avez des valeurs identiques dans `request` et `limit` pour le processeur et la mémoire pour obtenir une QoS garantie sous `containers: resources: requests: #configurable based`

PAN-CN-NGFW	
	`on desired throughput, number of running pods cpu: "1" memory: "40.0Gi" limits: cpu: "1" memory: "40.0Gi"` Pour la sécurité 5G-Native, les valeurs recommandées sont processeur = 12, mémoire = 48 Gio.

Note :
• L'annotation suivante identifie le daemonset du PAN-NGFW :
`paloaltonetworks.com/app: pan-ngfw-ds`
Ne modifiez pas cette valeur.
• L'annotation suivante identifie le nom du pare-feu (« pan-fw ») :
`paloaltonetworks.com/firewall: pan-fw`
Dans `pan-cni-configmap.yaml`, le nom du pare-feu doit correspondre exactement dans le `cni_network_config: "firewall"`
Et cette annotation doit correspondre exactement dans l'application yaml que vous utilisez pour déployer chaque pod d'application.
	Le Pod CN-NGFW sur chaque nœud sécurise les pods d'application et les espaces de noms qui portent l'annotation :
	`paloaltonetworks.com/firewall: pan-fw`
	Conservez cette annotation telle quelle.

(Facultatif) AF-XDP `imagePullPolicy: Always securityContext: capabilities: #add: ["NET_ADMIN","NET_RAW","NET_BROADCAST","NET_BIND_SERVICE"] add: ["ALL"] privileged: true resources:`	Vous devez ajouter `privileged: true` à la section affichée à gauche. Ce paramètre est requis pour activer le chemin de données eXpress de la famille d'adresses (AF-XDP). Vous devez également activer AF-XDP dans le fichier PAN-CN-MGMT-CONFIGMAP.

PAN-CNI-CONFIGMAP

Ces paramètres sont facultatifs.

PAN-CNI-CONFIGMAP	
Liste des noms de pare-feu auxquels le pod d'application peut appartenir : `"firewall": [ "pan-fw" ]`	Bien qu'aucune modification ne soit nécessaire, si vous changez l'annotation `paloaltonetworks.com/firewall: pan-fw` dans le `pan-cn-ngfw.yaml`, vous devez remplacer la valeur dans `"firewall": [ "pan-fw" ]` pour correspondre.
`"exclude_namespaces": []`	Bien qu'aucune modification ne soit nécessaire, si vous souhaitez exclure des espaces de noms spécifiques, ajoutez-le à **`"exclude_namespaces"`**, de sorte que l'annotation du pod d'application dans cet espace de noms soit ignorée et que le trafic ne soit pas redirigé vers le pod CN-NGFW pour inspection.
`"security_namespaces": [ "kube-system" ]`	Ajoutez les espaces de noms dans lesquels vous avez déployé le daemonset CN-NGFW dans security_namespaces. L'espace de noms par défaut est kube-system.

`“interfaces”`	Ajoutez les interfaces dans les pods d'application à partir desquels vous voulez rediriger le trafic vers le pod CN-NGFW pour inspection. Par défaut, seul le trafic eth0 est inspecté, et vous pouvez ajouter des interfaces supplémentaires sous la forme d'une liste de chaînes séparées par des virgules, par exemple [« eth0 », « net1 », « net 2 »].
	`cni_network_config:`
	`{ "cniVersion": "0.3.0", "name": "pan-cni", "type": "pan-cni", "log_level": "debug", "appinfo_dir": "/var/log/pan-appinfo", "mode": "daemonset", "firewall": [ "pan-fw" ], "interfaces": ["eth0", "net1", "net2", "net3"],`
	`}`

PAN-CNI-CONFIGMAP	
	En plus de cela, vous devez également ajouter `pan-cni` *à l'annotation* `k8s.v1.cni.cncf.io/networks` *dans le pod de l'application.* *Par exemple :* `metadata: name: testpod annotations: paloaltonetworks.com/firewall: pan-fw k8s.v1.cni.cncf.io/networks: sriov-net1, sriov-net2, macvlan-conf, pan-cni` *CN-Series ne prend pas en charge actuellement DPDK et ne permet pas au pod de l'application d'utiliser DPDK. Vous devrez peut-être modifier le pod de l'application si celle-ci ne s'adapte pas automatiquement au mode non DPDK.*

(CN-Series en tant que service Kubernetes uniquement)
`"dpservicename"`
`"dpservicenamespace"`
	Lorsque CN-Series est déployé en tant que service, un `dpservicename` et un `dpservicenamespace` sont requis. Par défaut, `dpservicename` est `"pan-ngfw-svc"` et `dpservicenamespace` est `"kube-system"`.

PAN-CNI

PAN-CNI	
Chemin d'accès de l'image du conteneur PAN-CNI qui contient les binaires CNI et le fichier	Modifiez le chemin d'accès de l'image pour qu'elle pointe vers l'emplacement où vous

PAN-CNI	
de configuration du réseau CNI sur chaque nœud. `containers: name: install-pan-cni image: <your-private-registry-image-path>`	avez téléchargé l'image du docker pour le conteneur AN-CNI.

PAN-CNI-MULTUS

Si vous utilisez Multus CNI sur une implémentation native ou autogérée de Kubernetes comme avec VMware TKG+, utilisez le fichier `pan-cni-multus.yaml` au lieu de `pan-cni.yaml`.

Sécuriser la 5G avec le pare-feu CN-Series

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Pour la visibilité et le contrôle du trafic 5G pour les entreprises privées et les déploiements 5G Mobile Packet Core dans un réseau d'opérateur mobile sur Kubernetes, consultez les sections suivantes pour en savoir plus sur les environnements pris en charge et sur la manière de modifier les fichiers pour déverrouiller la Sécurité GTP et la Sécurité native 5G sur le pare-feu CN-Series. En plus d'activer ces capacités lorsque vous déployez le pare-feu CN-Series, vous devez également activer la Sécurité GTP et/ou la Sécurité SCTP sur Panorama.

Runtime de conteneur	Docker CRI-O Containerd
Version de Kubernetes	1.17 à 1.27
Kubernetes gérés par le fournisseur cloud	• AWS EKS (de 1.17 à 1.27 pour le mode de déploiement CN-Series en tant que DaemonSet et CN-Series en tant que service.) • AWS EKS (1.17 à 1.22 pour CN-Series en tant que mode de déploiement CNF.) • AWS EKS (1.22 à 1.27 pour CN-Series en tant que déploiement CN-Cluster.) • EKS sur AWS Outpost (1.17 à 1.25) *CN-Series pour EKS sur AWS Outpost ne prend pas en charge SR-IOV ou Multus.*

	• Azure AKS (1.17 à 1.27)  *Dans Azure AKS, PAN-OS 11.0.2 est la version minimale requise pour prendre en charge Kubernetes 1.25 et versions ultérieures.* • GCP GKE (1.17 à 1.27)  *Comprend le plan de données GKE V2.* • OCI OKE (1.23)
Kubernetes gérés par le client	Sur le cloud public ou le centre de données sur site. Assurez-vous que la version de Kubernetes, les types de CNI et les versions du SE de la VM hôte sont répertoriés dans ce tableau. VMware TKG+ version 1.1.2 • Plateforme d'infrastructure – vSphere 7.0 • SE de la VM hôte Kubernetes – Photon OS
VM hôte Kubernetes	Système d'exploitation : • Ubuntu 16.04 • Ubuntu 18.04 • Ubuntu-22.04 • RHEL/Centos 7.3 et versions ultérieures • CoreOS 21XX, 22XX • SE optimisé pour les conteneurs
	Version du noyau Linux : • 4.18 ou version ultérieure (mode de service K8s uniquement) • 5.4 ou version ultérieure requise pour activer le mode AF_XDP. Pour plus d'informations, consultez Paramètres modifiables dans les fichiers YAML de déploiement CN-Series.
	Netfilter du noyau Linux : Iptables
Plug-ins CNI	Spécification CNI 0.3 et versions ultérieures : • AWS-VPC

	• Azure
	• Calico
	• Flannel
	• Weave
	• Pour Openshift, OpenshiftSDN
	• Les éléments suivants sont pris en charge sur le pare-feu CN-Series en tant que DaemonSet.
	  • Multus
	  • Pont
	  • SR-IOV
	  • Macvlan

OpenShift	• Versions 4.2, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 et 4.13.  *OpenShift 4.7 est qualifié sur CN-Series en tant que DaemonSet uniquement.* *PAN-OS 11.0.2 est la version minimale requise pour prendre en charge 4.12 et versions ultérieures.* • OpenShift sur AWS

Exécution du conteneur	Version
Pare-feu CN-Series	PAN-OS 10.0.3 ou version ultérieure
Plug-in Kubernetes	1.0.1 ou version ultérieure
Panorama	10.0.0 ou version ultérieure
La liste suivante comprend tous les paramètres modifiables du fichier YAML que vous utilisez pour déployer le pare-feu CN-Series : Pour plus de détails, voir Paramètres modifiables dans les fichiers yaml de déploiement CN-Seriesset Blocs de construction de base de CN-Series.	
Activer GTP	Dans le fichier pan-cn-mgmt-configmap.yaml, définissez : `PAN_GTP_ENABLED: "True"`, avant de déployer le StatefulSet CN-MGMT.

Exécution du conteneur	Version
Activer le mode de trame Jumbo	Dans le fichier pan-cn-mgmt-configmap.yaml, définissez : `PAN_JUMBO_FRAME_ENABLED: "True"`, avant de déployer le StatefulSet CN-MGMT. Le pod CN-MGMT pendant le démarrage utilise le MTU « eth0 » pour détecter automatiquement s'il faut activer le mode de trames Jumbo. Ainsi, si votre CNI secondaire utilise des trames Jumbo, alors que le CNI primaire ne le fait pas, vous devez définir `PAN_JUMBO_FRAME_ENABLED: "True"` pour activer le mode de trames jumbo sur le pare-feu CN-Series. *CN-Series ne prend pas en charge actuellement DPDK et ne permet pas au pod de l'application d'utiliser DPDK. Vous devrez peut-être modifier le pod de l'application si celle-ci ne s'adapte pas automatiquement au mode non DPDK.*
Activer la flexibilité des ressources du système	Si vous avez besoin d'un débit plus élevé et souhaitez configurer plus de mémoire pour répondre à vos besoins de déploiement dans le fichier pan-cn-mgmt-configmap.yaml, définissez : PAN_NGFW_MEMORY="48Gi" *Pour la création de modèles (Helm), la même variable peut être utilisée que celle qui est attribuée pour le pod CN-NGFW. Lors de l'activation d'une empreinte de mémoire plus importante, le StatefulSet CN-MGMT prend uniquement en charge un pod CN-NGFW.*
Configurer le vCPU et la mémoire pour la 5G	La configuration recommandée pour les pods CN-MGMT (dans pan-cn-mgmt.yaml) et les pods NGFW (dans pan-cn-ngfw.yaml) est d'avoir des valeurs identiques dans "request" et "limit" pour les processeurs et la mémoire afin de parvenir à une QoS garantie.

Exécution du conteneur	Version
	Pour les pods CN-MGMT, les valeurs recommandées sont cpu=4, memory=16Gi. Pour contrôler le placement sur les pods CN-MGMT, par exemple sur les mêmes nœuds ou sur des nœuds différents de ceux où les pods CN-NGFW sont déployés, utilisez la fonctionnalité de sélecteur de nœuds dans k8s.
	Pour les pods CN-NGFW , les valeurs recommandées sont cpu=12, memory=48Gi. Pour contrôler le placement sur les pods CN-NGFW, par exemple sur les mêmes nœuds ou sur des nœuds différents de ceux où les pods CN-NGFW sont déployés, utilisez la fonctionnalité de sélecteur de nœuds dans k8s.
Sélectionner le fichier CNI yaml	Le CNI Multus fonctionne comme un « méta-plug-in » qui appelle d'autres plug-ins CNI. Sur les environnements OpenShift; Multus est activé par défaut, vous pouvez donc utiliser le fichier `pan-cni.yaml`. Sur les autres environnements où Multus est pris en charge, mais facultatif, comme c'est le cas avec les environnements autogérés (natifs), utilisez le fichier `pan-cni-multus.yaml` au lieu du fichier `pan-cni.yaml`.

Consultez également la Configuration système requise pour le pare-feu CN-Series avant de poursuivre le déploiement du pare-feu CN-Series.

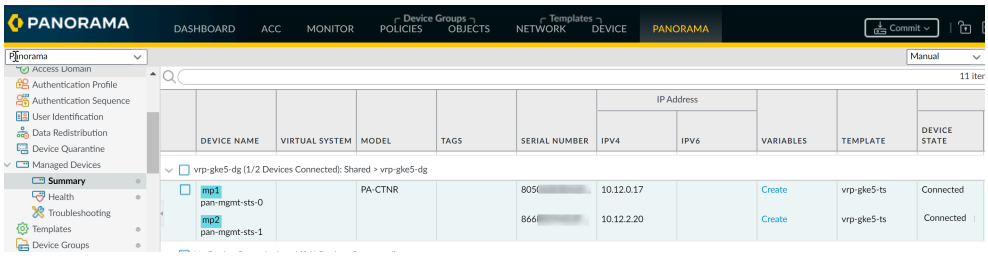
Configurer Panorama pour sécuriser un déploiement Kubernetes

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec graphique Helm

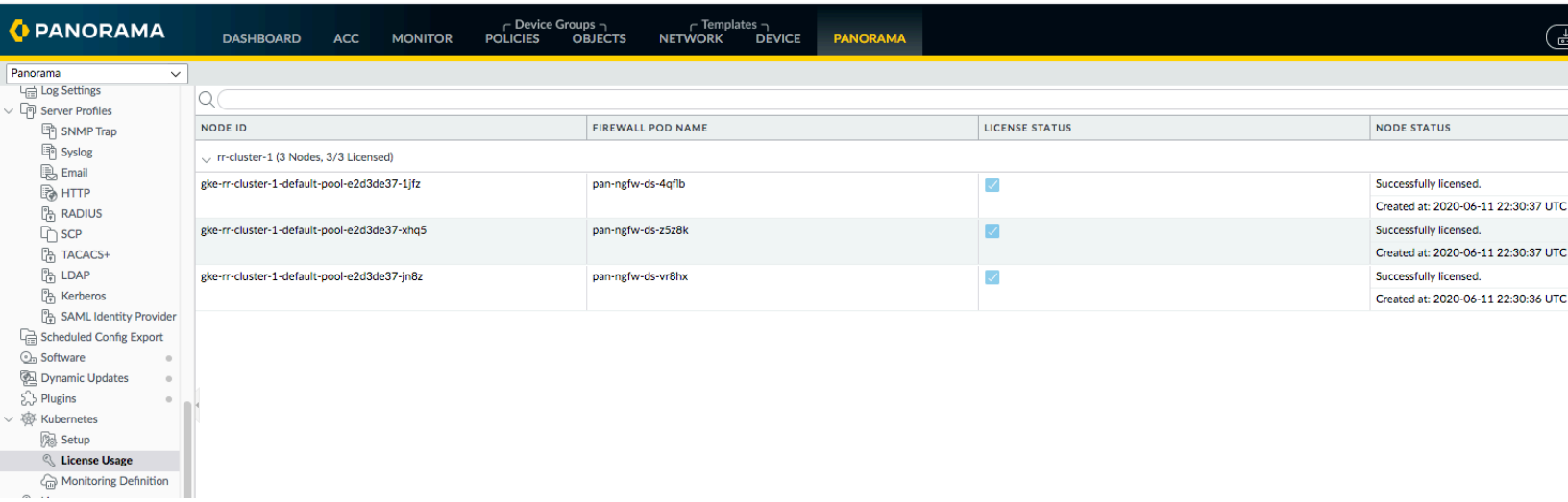
Après avoir installé le plug-in Kubernetes pour CN-Series et déployé le pare-feu CN-Series, pour surveiller le cluster Kubernetes et configurer les stratégies de sécurité qui permettent l'application du trafic, vous devez effectuer les tâches suivantes.

STEP 1 | Vérifiez que les pods CN-MGMT sont enregistrés sur Panorama et que les pods CN-NGFW sont sous licence.

1. Sélectionnez **Panorama** > **Managed Devices (Périphériques gérés)** > **Summary (Récapitulatif)**.



2. Sélectionnez **Panorama** > **Plugins (Plug-ins)** > **Kubernetes** > **License Usage (Utilisation de licence)** pour vérifier qu'un jeton de licence est attribué à chaque nœud du cluster.



STEP 2 | Créez un profil de transfert des journaux pour transférer les journaux à Panorama.

Le profil définit les destinations des différents journaux qui seront générés sur le pare-feu.

1. Sélectionnez le groupe d'appareils que vous avez créé pour le déploiement k8s dans la liste déroulante **Device Group (Groupe d'appareils)**.
2. Sélectionnez **Objects (Objets)** > **Log Forwarding (Transfert des journaux)** et cliquez sur **Add (Ajouter)**.
3. Saisissez un **Name (Nom)** pour identifier le profil. Si vous souhaitez associer automatiquement le profil aux nouvelles règles et zones de sécurité, saisissez **`default`**. Si vous ne souhaitez pas disposer d'un profil par défaut, ou si vous souhaitez remplacer un profil par défaut existant, saisissez un **Name (Nom)** qui vous permettra d'identifier le profil lors de son association à des règles et zones de sécurité.
4. Cliquez sur **Add** (Ajouter) pour ajouter les types de journaux à transférer.
5. Cliquez sur **OK**.

STEP 3 | Configurez le plug-in Kubernetes pour qu'il transmette les étiquettes aux groupes d'appareils spécifiés.

Vous devez ajouter une définition de surveillance comprenant le nom du cluster Kubernetes depuis lequel Panorama récupère les étiquettes prédéfinies et, éventuellement, un groupe de notification.

Un groupe de notification est requis si CN-Series est déployé dans un espace de noms autre que kube-system.

Un groupe de notification est une liste de groupes d'appareils recevant des mises à jour d'étiquettes. Pour le plug-in Kubernetes, le groupe de notification doit inclure des pare-feu externes au cluster (autrement dit, des pare-feu qui n'appartiennent pas au même groupe d'appareils que le cluster Kubernetes depuis lequel vous collectez des attributs).

Étant donné que le nom des groupes d'appareils est spécifié dans les fichiers YAML utilisés pour déployer les pare-feu CN-Series, le plug-in Kubernetes détecte automatiquement tous les groupes d'appareils internes au cluster et, par défaut, il transmet automatiquement toutes les étiquettes prédéfinies à ces groupes d'appareils.

Le plug-in Kubernetes utilise des secrets Kubernetes pour détecter de manière dynamique les groupes d'appareils de chaque cluster. À chaque fois que vous déployez un StatefulSet CN-MGMT, le secret est publié sur le serveur d'API Kubernetes et Panorama le détecte au cours du prochain intervalle de surveillance.

1. Configurez le plug-in Kubernetes pour la surveillance des clusters.
2. Ajoutez des groupes de notification. Ajoutez un groupe de notification et sélectionnez les groupes d'appareils qui reçoivent les étiquettes associées à votre cluster Kubernetes.
   1. Sélectionnez **Panorama** > **Plugins (Plug-ins)** > **Kubernetes** > **Setup (Configuration)** > **Notify Groups (Groupes de notification)**, puis cliquez sur **Add (Ajouter)**.
   2. Donnez un **Name (Nom)** au groupe de notification (31 caractères au maximum).
   3. Sélectionnez **Enable sharing internal tags with Device Groups (Activer le partage d'étiquettes internes avec les groupes d'appareils)** si vous souhaitez partager des étiquettes internes en plus des étiquettes externes (par défaut) créées pour le cluster.
   4. Sélectionnez les groupes d'appareils sur lesquels vous souhaitez enregistrer les étiquettes.

      Pour le groupe de notification sélectionné, Panorama ne transmet que les étiquettes externes.

      Une étiquette externe est une étiquette accessible depuis l'extérieur du cluster, comme les étiquettes générées pour l'adresse IP et le port d'un service externe pour

une adresse IP de cluster, l'adresse IP externe de tous les nœuds et ports de nœud, et l'adresse IP et le port ou port de nœud d'un équilibreur de charge externe.

Les étiquettes internes contiennent des informations sur les adresses IP, l'adresse IP du pod, les nœuds et les ports de nœud des clusters internes.

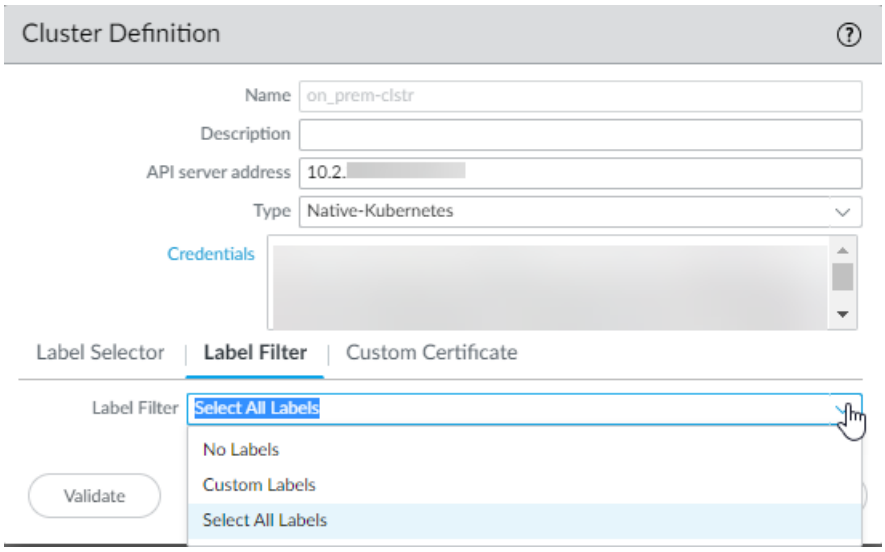
Par défaut, Panorama transmet toutes les étiquettes qu'il découvre (en fonction des filtres d'étiquette que vous sélectionnez) au groupe d'appareils associé au cluster, tel que défini dans le fichier YAML utilisé pour déployer les pods CN-MGMT.

3. Ajoutez une définition de surveillance pour chaque cluster.
   1. Sélectionnez **Panorama > Plugins (Plug-ins)** > **Kubernetes** > **Monitoring Definition (Définition de surveillance)**, puis cliquez sur **Add (Ajouter)**.
   2. Saisissez un **Name (Nom)** pour la définition de surveillance.
   3. Sélectionnez le **Cluster** que vous souhaitez surveiller.
   4. (Facultatif) Sélectionnez un **Notify Group (Groupe de notification)** auquel vous souhaitez envoyez les informations de mappage étiquette/adresse IP.

      Par défaut, les étiquettes sont partagées avec tous les pods CN-NFGW du cluster.
   5. Cliquez sur **OK** pour enregistrer vos modifications.
4. Cliquez sur **Commit (Valider)** pour valider sur Panorama.

STEP 4 | (Facultatif) Configurez le plug-in Kubernetes pour qu'il récupère les étiquettes définies par l'utilisateur des fichiers YAML de votre application.

1. Sélectionnez **Panorama > Plugins (Plug-ins)** > **Kubernetes** > **Setup (Configuration)** > **Cluster**, puis sélectionnez la définition du cluster dans la liste.
2. Sélectionnez le filtre d'étiquette parmi les options suivantes :



1. **No Labels (Aucune étiquette)** – Aucune étiquette n'est créée pour les étiquettes Kubernetes.
2. **Custom Labels (Étiquettes personnalisées)** – Crée uniquement des étiquettes pour les étiquettes que vous choisissez.

   Pour utiliser des étiquettes personnalisées, vous devez d'abord annoter les fichiers YAML dans votre déploiement Kubernetes, puis utiliser l'une des combinaisons suivantes pour générer des étiquettes personnalisées pour les adresses IP correspondantes :

   Spécifiez l'espace de noms, la clé et la valeur. Utilisez « * » pour tout sélectionner. Le plug-in crée des étiquettes lorsque les trois entrées sont valides.

   Spécifiez l'espace de noms et la clé pour créer des étiquettes pour toutes les clés correspondantes dans cet espace de noms.

   Spécifiez uniquement l'espace de noms pour créer une étiquette personnalisée pour chaque étiquette dans cet espace de noms.
3. **Select All Labels (Sélectionner toutes les étiquettes)** – Permet de créer des étiquettes pour toutes les étiquettes Kubernetes, y compris les étiquettes personnalisées.

3. Ajoutez une expression de sélecteur d'étiquettes.

   Le sélecteur d'étiquettes fait correspondre l'étiquette spécifiée dans le cluster Kubernetes et mappe les adresses IP associées à l'étiquette dans une étiquette unique.

Pour obtenir une liste des préfixes pris en charge, reportez-vous à la section Mappage étiquette/adresse IP des attributs Kubernetes.

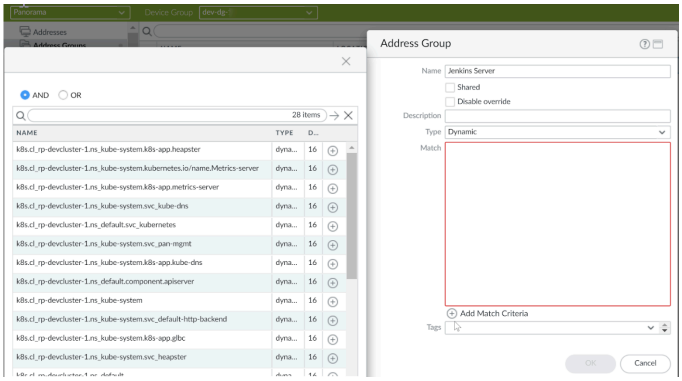
Pour chaque sélecteur d'étiquettes, Panorama génère une étiquette pouvant être utilisée comme critère de correspondance dans les groupes d'adresses dynamiques et vous permettant d'appliquer les politiques de sécurité :

1. **Tag Prefix (Préfixe de l'étiquette)** – Élément placé à la fin de chaque étiquette pour vous aider à les identifier. Par exemple, le sélecteur d'étiquettes k8s.cl_<clustername>.<selector-name> fait correspondre toutes les adresses IP de cluster et de pod correspondant au sélecteur. Ces adresses IP peuvent se trouver dans tous les espaces de noms ou dans des espaces spécifiques, selon votre configuration.
2. **Namespace (Espace de noms)** – Saisissez « * » pour sélectionner tous les espaces de noms ou indiquez une valeur pour l'espace de noms.
3. **Label Selector Filter (Filtre de sélecteur d'étiquettes)** – Le plug-in Kubernetes prend en charge les sélecteurs basés sur des ensembles ou sur l'égalité pour la clé et la valeur des étiquettes. Les sélecteurs suivants basés sur l'égalité sont pris en charge —key = value; key ==value; key != value, par exemple, app = redis. Vous pouvez également spécifier plusieurs sélecteurs dans une expression sous la forme d'une liste séparée par des virgules, comme app == web, tier != backend. Les sélecteurs basés sur les ensembles suivants sont pris en charge—key in (value1,value2), key notin (value1, value2), key, !key, par exemple, tier notin (frontend, backend).
4. **Apply On (Appliquer à)** – L'étiquette peut être appliquée aux types de ressources Service, Pod et All (Tout).

STEP 5 | Configurez les groupes d'adresses dynamiques.

1. Sélectionnez votre groupe d'appareils pour la gestion des pods CN-NGFW.
2. Sélectionnez **Objects (Objets)** > **Address Groups (Groupes d'adresses)**.
3. Cliquez sur **Add (Ajouter)** et saisissez un **Name (nom)** et une **Description (description)** pour identifier le groupe d'adresses.
4. Définissez le **Type (type)** sur **Dynamic (Dynamique)**.

STEP 6 | Cliquez sur **Add Match Criteria (Ajouter des critère de correspondance)** et sélectionnez l'opérateur **AND (ET)** ou **OR (OU)**, puis choisissez les attributs que vous souhaitez filtrer ou mettre en correspondance.

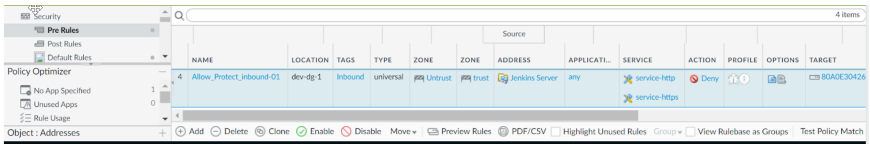


STEP 7 | Cliquez sur **OK** et sur **Commit on Panorama (Valider sur Panorama)**.

Utilisez le lien **more... (plus...)** pour afficher les adresses IP associées à l'objet, comme le serveur Jenkins dans cet exemple.

STEP 8 | Créez des règles de politique de sécurité pour l'application du trafic.

1. Sélectionnez **Policies (Politiques)** > **Security (Sécurité)**.
2. Cliquez sur **Add (Ajouter)** et saisissez un **Name (nom)** et une **Description (description)** pour identifier la politique.
3. Ajoutez la **Source Zone (Zone source)** pour indiquer la zone d'où provient le trafic.
4. Ajoutez la **Destination Zone (Zone de destination)** dans laquelle se termine le trafic.
5. Pour **Destination Address (Adresse de destination)**, sélectionnez le groupe d'adresses dynamiques que vous venez de créer.
6. Indiquez l'action (**Deny [Refuser]**) pour le trafic, puis associez éventuellement les profils de sécurité par défaut à la règle.
7. Sélectionnez **Actions** et sélectionnez le profil de **Log Forward (Transfert de journaux)** que vous avez créé.
8. Cliquez sur **Commit (Valider)**.



Vous pouvez également appliquer une politique de sécurité au trafic est-ouest dans les espaces de noms. Par exemple, admettons que vous disposez de deux espaces de noms (stage-NS et DB-NS) dans un cluster nommé « cluster intermédiaire », où les pods frontaux sont déployés dans stage-NS et les pods principaux Redis s'exécutent dans l'espace de noms DB-NS. Lorsque vous ajoutez ce cluster au plug-in Kubernetes sur Panorama pour le surveiller, il récupère les métadonnées des étiquettes pour créer des étiquettes. Vous pouvez ensuite utiliser ces nouvelles étiquettes pour appliquer des règles de politique de sécurité. Pour ce faire, vous devez répondre à quelques exigences :

- Assurez-vous que l'espace de noms ou les fichiers YAML que vous utilisez pour déployer les applications frontales et principales possèdent l'annotation paloaltonetworks.com/firewall: pan-fw.
- Créez le groupe d'adresses dynamiques pour les pods frontaux et principaux.

  Vous devez configurer les groupes d'adresses dynamiques dans le groupe de périphériques associé au cluster et sélectionner d'abord les balises des serveurs frontaux. Répétez ensuite le processus pour créer un autre groupe d'adresses dynamiques pour les serveurs principaux.

- Ajoutez la règle de politique de sécurité pour permettre le trafic des pods frontaux aux pods principaux pour l'application Redis.

  La source est le groupe d'adresses dynamiques des serveurs frontaux et la destination est le groupe d'adresses dynamiques pour les serveurs principaux, et l'action est autoriser.

Mappage étiquette/adresse IP des attributs Kubernetes

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Le plug-in Kubernetes sur Panorama crée des étiquettes pour les étiquettes prédéfinies de vos clusters Kubernetes, des étiquettes définies par l'utilisateur pour les pods et les services, et des objets de service.

Le plug-in crée des étiquettes pour les objets Kubernetes suivants :

- Classes de pods : ReplicaSets, DaemonSets, StatefulSets
- Types de services : ClusterIP, NodePort, LoadBalancer
- Objets de service : port, targetPort, nodePort et interfaces de pod

Par défaut, le plug-in Kubernetes sur Panorama récupère les étiquettes prédéfinies suivantes de chacun des clusters Kubernetes que vous surveillez sur Panorama et crée des étiquettes selon le format ci-dessous. Vous pouvez ensuite utiliser ces étiquettes comme critères de correspondance dans les groupes d'adresses dynamiques et appliquer la politique de sécurité aux adresses IP sous-jacentes associées à chaque étiquette.

La longueur maximale de chaque étiquette est de 127 caractères. Les étiquettes sont tronquées si elles dépassent le nombre maximal de caractères. Si deux étiquettes tronquées sont identiques, un hachage unique est ajouté aux balises pour les différencier l'une de l'autre.

Vous pouvez utiliser le plug-in Kubernetes pour distribuer le mappage étiquette/adresse IP pour les pods, les nœuds, les espaces de noms et les services déployés dans le cluster Kubernetes vers des pare-feu physiques ou VM-Series, et ce, même si vous n'avez pas déployé de pare-feu CN-Series dans ce cluster.

Étiquettes prédéfinies	Format des étiquettes sur Panorama	Adresse IP collectée
DaemonSet	k8s.cl_<cluster-name>.ns_<namespace>.ds_<pod-name>	Adresses IP des pods
ReplicaSet	k8s.cl_<cluster-name>.ns_<namespace>.rs_<pod-name>	Adresses IP des pods
StatefulSet	k8s.cl_<cluster-name>.ns_<namespace>.ss_<pod-name>	Adresses IP des pods
Service	k8s.cl_<cluster-name>.ns_<namespace>.svc_<svc-name>	Adresses IP des clusters Adresses IP des pods
Service externe	k8s.cl_<cluster-name>.ns_<namespace>.exsvc_<svc-name>	Adresses IP des services externes Adresses IP LoadBalancer
Nœuds	k8s.cl_<cluster-name>.nodes	Adresses IP privées de tous les nœuds
Nœuds externes	k8s.cl_<cluster-name>.ex_nodes	Adresses IP publiques de tous les nœuds
Espace de noms	k8s.cl_<cluster-name>.ns_<namespace>	Adresses IP de tous les clusters de l'espace de noms Adresses IP de tous les pods de l'espace de noms
Interface	• 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>	Toutes les adresses IP de toutes les interfaces de chaque pod de votre déploiement.

Si vous utilisez des étiquettes pour organiser les pods et les services dans le cluster Kubernetes, le plug-in Kubernetes sur Panorama peut interroger ces étiquettes et créer des étiquettes pour vous. Les étiquettes définies par l'utilisateur suivantes sont prises en charge :

Étiquettes définies par l'utilisateur	Format des étiquettes sur Panorama	Adresse IP collectée
Libellé	k8s.cl_<cluster-name>.ns_<namespace>.<label-key>.<label-value>	Les adresses IP de tous les clusters de cet espace de noms qui correspondent à l'étiquette spécifiée. Les adresses IP de tous les pods de cet espace de noms qui correspondent à l'étiquette spécifiée.
Sélecteur d'étiquettes	k8s.cl_<cluster-name>.<selector-name>	Les adresses IP de tous les clusters qui correspondent au sélecteur spécifié. Les adresses IP de tous les pods qui correspondent au sélecteur spécifié.

Le sélecteur d'étiquettes met en correspondance l'étiquette spécifiée avec les pods et les services du cluster Kubernetes et mappe les adresses IP associées à l'étiquette dans une étiquette unique. Le plug-in Kubernetes prend en charge les sélecteurs basés sur des ensembles ou sur l'égalité pour la clé et la valeur des étiquettes.

Les sélecteurs basés sur l'égalité suivants sont pris en charge :

- `clé = valeur ; clé ==`
- `valeur ; clé != valeur, par exemple, appli = redis`

Vous pouvez également spécifier plusieurs sélecteurs dans une expression sous la forme d'une liste séparée par des virgules. Par exemple :

```
appli == web, couche != principal
```

Les sélecteurs basés sur les ensembles suivants sont pris en charge :

- clé in (valeur1, valeur2)
- clé notin (valeur1, valeur2), par exemple, couche notin (frontal, principal)
- clé
- !clé

Pour les objets de service surveillés, le plug-in génère des ports pour les objets de service port, targetPort et nodePort en utilisant le schéma d'affectation de nom suivant :

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

Le hachage permet de s'assurer que les objets de service sont uniques, même si vous possédez des espaces de noms et des noms de services se chevauchant sur plusieurs clusters k8s.

Activer l’inspection du trafic VLAN étiqueté

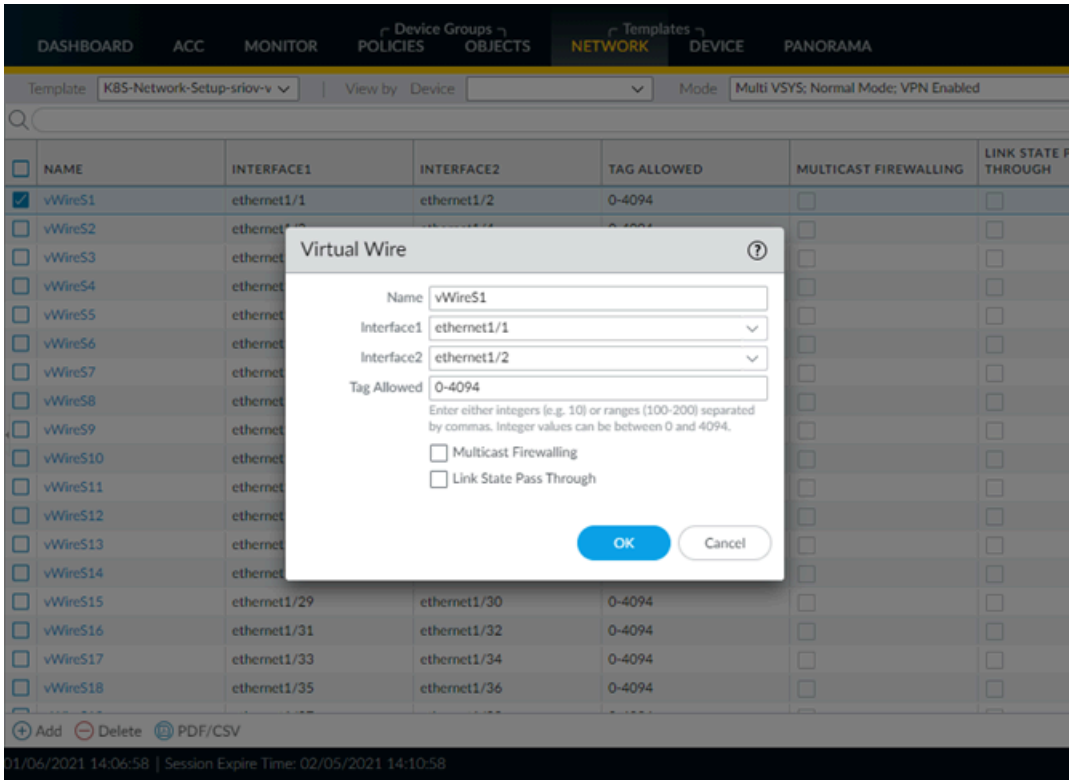
Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Effectuez la procédure suivante pour permettre au pare-feu CN-Series d’inspecter le trafic VLAN étiqueté. Pour inspecter le trafic VLAN étiqueté, vous devez mettre à jour la configuration de tous les câbles virtuels sur Panorama afin d’autoriser toutes les étiquettes VLAN. Vous devez ensuite annoter votre fichier YAML du pod d’application pour attribuer des étiquettes VLAN aux interfaces du pod d’application. Cette annotation indique au CN-NGFW quelles étiquettes sont appliquées aux paquets qui sont envoyés à travers le pare-feu.

Le double étiquetage VLAN n’est pas pris en charge.

STEP 1 | Activez tous les VLAN sur toutes les interfaces du CN-NGFW.

1. Connectez-vous à Panorama.
2. Sélectionnez **Network (Réseau)** > **Virtual Wires (Câbles virtuels)**.
3. Sélectionnez le modèle **K8S-Network-Setup** dans la liste déroulante du **Template (Modèle)**.
4. Sélectionnez le premier câble virtuel.
5. Définissez **Tag Allowed (Étiquette autorisée)** comme suit : 0-4094.
6. Répétez cette procédure pour chaque câble virtuel.
7. **Commit (Validez)** vos modifications.



STEP 2 | Ajoutez le fichier YAML du pod d'application avec les annotations suivantes pour appliquer un identifiant VLAN statique par interface.

Une seule étiquette VLAN est prise en charge par interface.

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 } ]'
```

Activer IPVLAN

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.1.x or above Container Images • Panorama sous PAN-OS 10.1.x ou une version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

IPVLAN est un pilote pour un périphérique réseau virtuel qui peut être utilisé dans un environnement conteneurisé pour accéder au réseau hôte. En mode L2, IPVLAN expose une seule adresse MAC au réseau externe, quel que soit le nombre de périphériques IPVLAN créés à l'intérieur du réseau hôte. Toutes les interfaces IP logiques utilisent la même adresse MAC. Cela vous permet d'éviter d'utiliser le mode promiscuité sur la carte réseau parente et d'éviter les limitations MAC potentielles sur la carte réseau ou le commutateur.

Vous pouvez désormais utiliser IPVLAN avec le pare-feu CN-Series avec les limitations suivantes.

- Nécessite PAN-OS 10.1.2 et versions ultérieures
- IPv4 uniquement
- Mode L2 uniquement
- Une adresse IP par interface
- Si vous utilisez Multus, déployez **`pan-cni-multus.yaml`** au lieu de **`pan-cni.yaml`**. En outre, vous devez déployer pan-cni-net-attach-def.yaml dans chaque espace de noms où les pods d'application Multus sont déployés.

La communication de l'interface enfant IPVLAN sur le même hôte (partage de la même interface parente) ne fonctionne pas.

Vous devez annoter vos fichiers yaml de pod d'application pour activer IPVLAN ; aucune modification n'est requise pour les fichiers yaml CN-Series avec IPVLAN activé. Voici un exemple de définition de pièce jointe réseau pour IPVLAN. Notez que le mode est réglé sur **`"l2"`**. Le pare-feu CN-Series ne prend en charge que le mode 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" } ] } }'
```

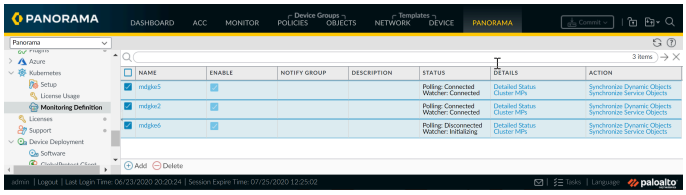
Désinstaller le plug-in Kubernetes sur Panorama

Utilisez le flux de travail suivant pour désinstaller le plug-in Kubernetes sur Panorama afin de pouvoir renvoyer tous les jetons vers les serveurs de mise sous licence de Palo Alto Networks, puis effacez le code d'autorisation. Ce flux de travail vous permet de vous assurer que les jetons sont disponibles pour être utilisés sur un autre Panorama. Si vous avez déployez votre serveur de gestion Panorama dans une configuration haute disponibilité, vous devez effectuer les étapes sur le Panorama actif-principal avant de passer à l'homologue Panorama passif-principal.

STEP 1 | Connectez-vous à l'homologue Panorama actif-principal, s'il est déployé dans une configuration HA.

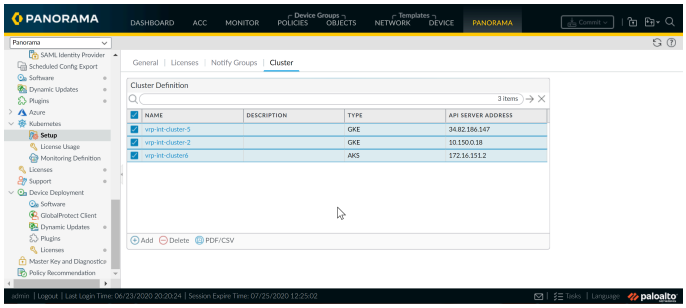
1. Supprimez la totalité de la configuration du cluster du plug-in.
   1. Supprimez les définitions de surveillance.

      Sélectionnez **Plugins** > **Kubernetes** > **Monitoring Definition**, sélectionnez les définitions de surveillance et **Delete** (Supprimer).

   2. Supprimez les définitions du cluster Kubernetes.

      Sélectionnez **Plugins** > **Kubernetes** > **Set up** > **Cluster**, sélectionnez les définitions de surveillance et **Delete (Supprimer)**.

2. Validez vos modifications sur Panorama.

   Commit (Valider) > **Commit to Panorama (Valider sur Panorama)**.
3. Vérifiez que le nombre de jetons utilisés est égal à zéro.

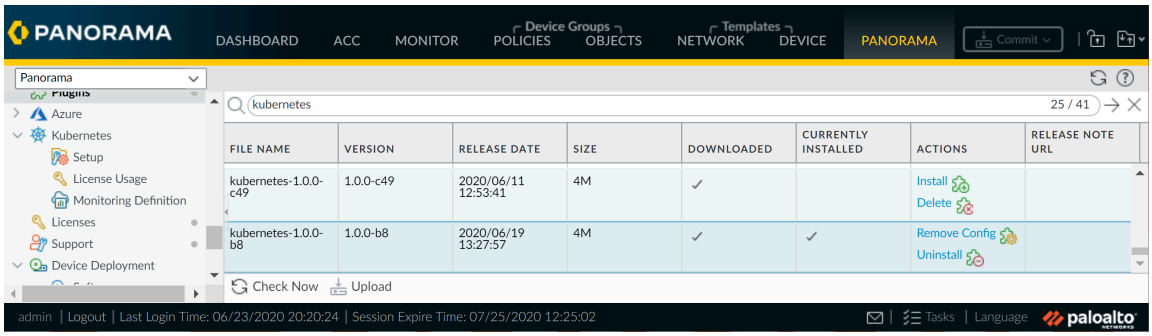
   Pour confirmer que tous les jetons sont renvoyés au serveur de mise sous licence.
4. Effacez le code d'autorisation et assurez-vous que le code d'autorisation de la colonne de licence est défini sur None (Aucun).
5. Supprimez la configuration et validez vos modifications.
   1. Sélectionnez **Plugins (Plug-ins)** et trouvez la version du plug-in Kubernetes que vous avez installée, et cliquez sur **Remove Config (Supprimer la configuration)**.
   2. **Commit (Valider)** > **Commit to Panorama (Valider sur Panorama)**.
6. Désinstallez le plug-in Kubernetes.
7. Suspendez l'homologue Panorama actif.

   Sélectionnez **Panorama** > **High Availability (Haute Disponibilité)** et cliquez sur le lien **Suspend local Panorama (Suspendre le Panorama local)** dans la section Operational Commands (Commandes opérationnelles).

STEP 2 | Connectez-vous à votre autre homologue Panorama.

Cet homologue est désormais l'homologue actif-secondaire.

1. Sélectionnez **Plugins (Plug-ins)** et trouvez la version du plug-in Kubernetes que vous avez installée, et cliquez sur **Remove Config (Supprimer la configuration)**.



2. Désinstallez le plug-in.
    1. Sélectionnez **Plugins** (Plug-ins) et trouvez la version du plug-in Kubernetes que vous avez installée, et cliquez sur **Uninstall** (Désinstaller).
    2. Vérifiez que la désinstallation est réussie.

Effacer le code d'autorisation pour les pare-feu CN-Series sur Panorama

N'utilisez la solution alternative indiquée ci-dessous que si vous avez supprimé la configuration du plug-in et validé vos modifications avant d'effacer le code d'autorisation. Cette solution alternative vous permet de libérer les jetons pour les renvoyer sur le serveur de mise sous licence, afin que vous puissiez l'utiliser sur un autre appareil Panorama.

STEP 1 | 1. Ajoutez un nouvel utilisateur de plug-in et validez vos modifications.

1. Sélectionnez **Panorama** > **(Administrateurs)**.
2. **Ajoutez** un nouvel utilisateur appelé **__kubernetes**.
3. **Commit (Valider)** > **Commit to Panorama (Valider sur Panorama)**.

STEP 2 | Effacez le code d'autorisation sur Panorama.

1. Sélectionnez **Panorama** > **Plugins (Plug-ins)** > **Kubernetes** > **Setup (Configuration)** > **Licenses (Licences)**.
2. Sélectionnez **Activate/update using authorization code (Activer / Mettre à jour à l'aide du code d'autorisation)** et **Clear Auth Code (Effacer le code d'autorisation)**.
3. Vérifiez que la colonne de licence affiche le code d'autorisation **None (Aucune)**.

STEP 3 | Supprimez l'utilisateur de plug-in **__kubernetes** que vous avez créé à l'étape 1.

STEP 4 | Validez vos modifications.

STEP 5 | Désinstallez le plug-in.

1. Sélectionnez **Plugins** (Plug-ins) et trouvez la version du plug-in Kubernetes que vous avez installée, et cliquez sur **Uninstall** (Désinstaller).
2. Vérifiez que la désinstallation est réussie.

Fonctionnalités non prises en charge par le pare-feu CN-Series

Les fonctionnalités suivantes prises en charge sur PAN-OS ne sont pas disponibles pour la série CN, sauf indication contraire ci-dessous :

Fonctionnalité	DaemonSet	Service K8s	Mode CNF	Mode HSF
Authentification	Non	Non	Non	Non
Journaux vers Cortex Data Lake	Non	Non	Non	Non
Enterprise DLP	Non	Non	Non	Non
Interfaces non-vWire	Non	Non	**Oui**	**Oui**
IoT Security	Non	Non	Non	Non
IPv6	**Oui**	Non	**Oui**	Non
NAT	Non	Non	**Oui**	Non
Transfert basé sur une politique	Non	Non	**Oui**	Non
QoS	Non	Non	Non	Non
SD-WAN	Non	Non	Non	Non
User-id	Non	Non	**Oui**	Non
WildFire Inline ML	Non	Non	Non	Non
SaaS Inline	Non	Non	Non	Non
IPSec	Non	Non	Non	Non
Inspection du contenu du tunnel	Non	Non	Non	Non

Haute disponibilité et support DPDK pour le pare-feu CN-Series

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

La High Availability (haute disponibilité ; HA) est une configuration dans laquelle deux pare-feu sont placés dans un groupe et où leur configuration est synchronisée afin d'éviter tout point de défaillance unique sur votre réseau. Une connexion de pulsation entre les pare-feu homologues garantit un basculement transparent en cas d'arrêt d'un homologue. Le paramétrage des pare-feu dans un cluster composé de deux périphériques fournit une redondance et vous permet d'assurer la continuité de l'activité.

Ce chapitre couvre les sections suivantes :

Prise en charge de la High Availability (haute disponibilité - HA) pour le pare-feu CN-Series en tant que Kubernetes CNF

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

La High Availability (haute disponibilité ; HA) est une configuration dans laquelle deux pare-feu sont placés dans un groupe et où leur configuration est synchronisée afin d'éviter tout point de défaillance unique sur votre réseau. Une connexion de pulsation entre les pare-feu homologues garantit un basculement transparent en cas d'arrêt d'un homologue. Le paramétrage des pare-feu dans un cluster composé de deux périphériques fournit une redondance et vous permet d'assurer la continuité de l'activité.

Vous pouvez maintenant déployer le CN-series-as-a-kubernetes-CNF dans HA. Ce mode de déploiement ne prend en charge que la HA active/passive avec synchronisation de session et de configuration.

Lorsque vous déployez le CN-Series-as-a-Kubernetes CNF dans HA, il y aura deux fichiers PAN-CN-MGMT-CONFIGMAP, PAN-CN-MGMT et PAN-CN-NGFW YAML, chacun pour les nœuds actifs et passifs.

Pour déployer avec succès le pare-feu CN-Series en tant que Kubernetes CNF dans HA avec prise en charge de la couche 3 :

- En HA, chaque nœud Kubernetes doit avoir au moins trois interfaces : Gestion (par défaut), interface HA2 et interface de données.
- Pour le pare-feu CN-Series en mode L3, il doit y avoir au moins deux interfaces : Gestion (par défaut) et interface de données.

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		

- Modifiez les nouveaux fichiers YAML de définition de pièce jointe réseau avec les modifications suivantes :
  - Assurez-vous que la valeur du paramètre PAN_HA_SUPPORT est **`true`** dans les fichiers YAML suivants :

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

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

  - Récupérez la valeur **`pciBusID`** à partir de l'interface de l'hyperviseur en exécutant la commande suivante :

    ```
    ethtool -i interface name
    ```

    Ajoutez la valeur **`pciBusID`** récupérée ci-dessus aux fichiers de définition de réseau suivants :

    ```
    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
    ```

  - Récupérez l'adresse IP statique de l'interface HA2 à partir de l'instance de nœud correspondante sur la console AWS et ajoutez-la au paramètre `d'adresse` du fichier `net-attach-def-ha2-0.yaml` et `net-attach-def-ha2-1.yaml`.

Si vous utilisez le **Advanced Routing (Routage avancé)**, n'oubliez pas que les pare-feu CN-Series déployés en mode CNF ne sont pris en charge que dans les environnements EKS et sur site. Si vous utilisez le **Advanced Routing (Routage avancé)** avec le plug-in Kubernetes 3.0.0, vous devez le configurer manuellement sur la pile de modèles ; dans le fichier `pan-cn-mgmt-console.yaml`, définissez l'indicateur `PAN_ADVANCED_ROUTING:"true"`.

High Availability (haute disponibilité - HA) du pare-feu CN-Series dans AWS EKS

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Vous pouvez maintenant déployer le CN-Series-as-a-Kubernetes-CNF dans HA. Ce mode de déploiement ne prend en charge que la HA active/passive avec synchronisation de session et de configuration.

Le déploiement CN-Series-as-a-Kubernetes CNF en HA avec IPV6 n'est pas pris en charge sur l'environnement AWS.

Pour assurer la redondance, vous pouvez déployer les pare-feu CN-Series sous forme de configuration High Availability (haute disponibilité - HA) active/passive dans AWS. L'homologue actif synchronise continuellement les informations de configuration et de session avec l'homologue passif ayant la même configuration. Une connexion de pulsation entre les deux périphériques garantit un basculement dans l'éventualité où le périphérique actif tombe en panne. Vous pouvez déployer le pare-feu CN-Series dans AWS EKS en HA via le déplacement IP secondaire.

Pour vous assurer que tout le trafic vers vos applications Internet passe par le pare-feu, vous pouvez configurer le routage d'entrée AWS. La fonction de routage d'entrée AWS vous permet d'associer des tables de routage à la passerelle Internet AWS et d'ajouter des règles de routage pour rediriger le trafic de l'application à travers le pare-feu CN-Series. Cette redirection garantit que tout le trafic Internet passe par le pare-feu sans avoir à reconfigurer les terminaisons des applications.

Mouvement secondaire

Lorsque l'homologue actif tombe en panne, l'homologue passif détecte cette défaillance et devient actif. Il déclenche également des appels d'API vers l'infrastructure AWS pour déplacer les adresses IP secondaires configurées depuis les interfaces de plan de données de l'homologue défaillant vers lui-même. En outre, AWS met à jour les tables de routage pour s'assurer que le trafic est dirigé vers l'instance de pare-feu active. Ces deux opérations permettent de s'assurer que les sessions de trafic entrant et sortant sont rétablies après le basculement. Cette option vous permet de tirer parti de DPDK pour améliorer les performances de vos instances de pare-feu CN-Series.

Rôles IAM pour la HA

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

AWS exige que toutes les requêtes de l'API reçoivent une signature cryptographique à l'aide d'authentifiants émis par ces derniers. Afin d'activer les permissions API pour les pare-feu CN-Series qui seront déployés en tant que paire HA, vous devez créer une politique et associer cette politique à un rôle dans le service de gestion de l'identité et des accès (Identity and Access Management ; IAM) d'AWS. Le rôle doit être associé aux pare-feu CN-Series lors du lancement. La politique donne à l'IAM les permissions de rôle pour initier les actions de l'API afin de déplacer des interfaces ou des adresses IP secondaires entre l'homologue actif et l'homologue passif lorsqu'un basculement est déclenché.

Pour les instructions détaillées sur la création d'une politique, reportez-vous à la documentation AWS sur la Création de politiques gérées par le client. Pour obtenir des instructions détaillées sur la création d'un rôle IAM, la détermination des comptes ou des services AWS qui peuvent assumer le rôle et la détermination des actions et des ressources API que l'application peut utiliser après avoir assumé le rôle, reportez-vous à la documentation d'AWS qui porte sur les rôles IAM dans Amazon EC2.

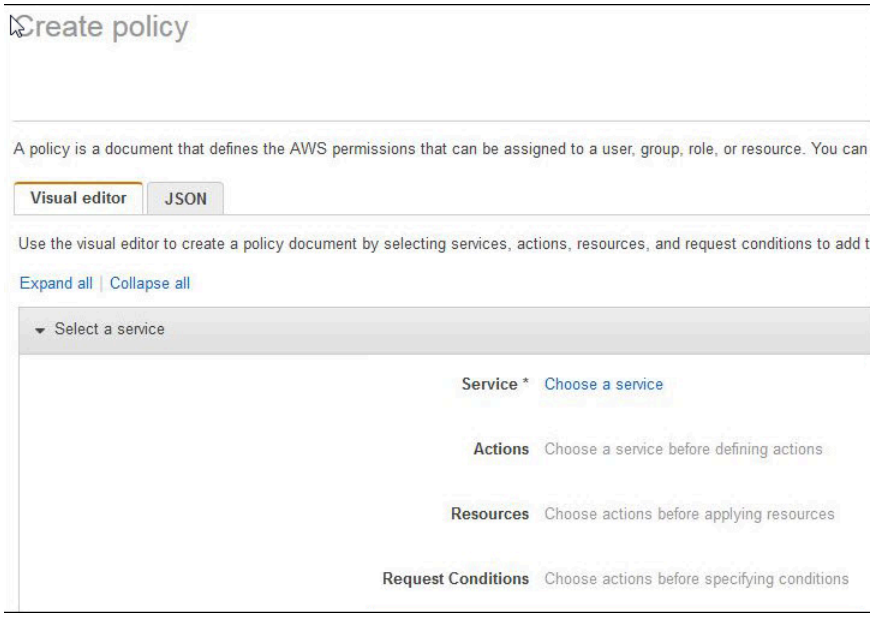
La politique IAM, qui est configurée dans la console AWS, doit disposer des permissions pour les actions et ressources suivantes (au minimum) :

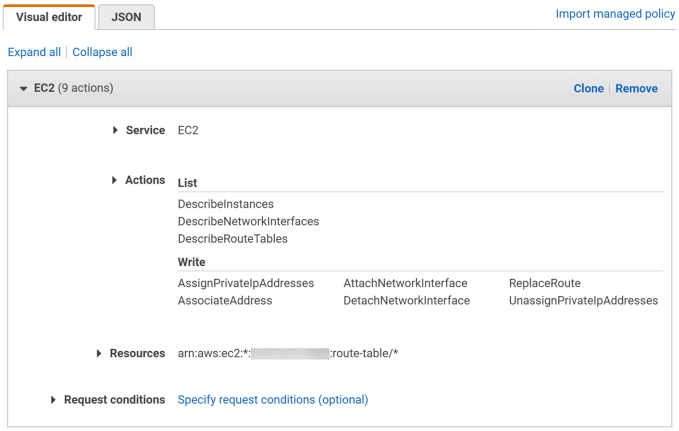
Les actions, autorisations et ressources IAM suivantes sont nécessaires pour autoriser le mode HA.

Action, autorisation ou ressource IAM	Description	Déplacer l'adresse IP secondaire
AttachNetworkInterface	Pour l'autorisation d'associer une ENI à une instance.	✓
DescribeNetworkInterfaces	Pour télécharger les paramètres de l'ENI afin d'associer une interface à une instance.	✓
DetachNetworkInterface	Pour l'autorisation de dissocier l'ENI de l'instance EC2.	✓
DescribeInstances	Pour l'autorisation d'obtenir des informations sur les instances EC2 sur le VPC.	✓

Action, autorisation ou ressource IAM	Description	Déplacer l'adresse IP secondaire
AssociateAddress	Pour les autorisations de déplacer des adresses IP publiques associées aux adresses IP primaires des interfaces passives vers les interfaces actives.	✓
AssignPrivateIpAddresses	Pour les autorisations d'attribuer des adresses IP secondaires et des adresses IP publiques associées aux interfaces sur l'homologue passif.	✓
DescribeRouteTables	Pour l'autorisation de récupérer toutes les tables de routage associées aux instances du pare-feu CN-Series.	✓
ReplaceRoute	Pour les autorisations de mettre à jour des entrées de la table de routage AWS.	✓
GetPolicyVersion	Pour l'autorisation de récupérer les informations sur la version de la politique AWS.	✓
GetPolicy	Pour l'autorisation de récupérer les informations sur la politique AWS.	✓
ListAttachedRolePolicies	Pour l'autorisation de récupérer la liste de toutes les politiques gérées associées à un rôle IAM spécifié.	✓
ListRolePolicies	Pour l'autorisation de récupérer une liste des noms des politiques en ligne intégrées dans un rôle IAM spécifié.	✓
GetRolePolicy	Pour l'autorisation de récupérer une politique en ligne spécifique intégrée dans un rôle IAM spécifié.	✓
policy	Pour l'autorisation d'accéder à la politique IAM Amazon Resource Name (ARN).	✓
role	Pour l'autorisation d'accéder aux rôles IAM ARN.	✓
route-table	Pour l'autorisation d'accéder à la table de routage Amazon Resource Name (ARN) afin de la mettre à jour lors du basculement.	✓
Caractère générique (*)	Dans le champ ARN, utilisez* comme caractère générique.	✓

Les captures d'écran suivantes présentent les paramètres de gestion d'accès pour le rôle IAM décrit ci-dessus pour la HA d'IP secondaire :





Les autorisations minimales dont vous avez besoin pour une HA de déplacement d'IP secondaire sont les suivantes : {"Version": "2012-10-17","Statement": [{"Sid": "VisualEditor0","Effect": "Allow","Action": ["ec2:AttachNetworkInterface","ec2:DetachNetworkInterface","ec2:DescribeInstances","ec2:DescribeNetwork "ec2:AssignPrivateIpAddresses","ec2:AssociateAddress","ec2:DescribeRouteTables"],"Resource": "*"}{"Sid": "VisualEditor1","Effect": "Allow","Action": "ec2:ReplaceRoute", "Resource": "arn:aws:ec2:*:*:route-table/*"}]}

Liaisons HA

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
	• Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Les périphériques d'une paire HA utilisent des liaisons HA pour synchroniser des données et garantir des informations d'état. Sur AWS, le pare-feu CN-Series utilise les ports suivants :

- **Liaison de contrôle**—La liaison HA1 permet d'échanger des salutations, des pulsations et des informations d'état HA, ainsi que des synchronisations de plans de gestion pour le routage. Cette liaison permet également de synchroniser des modifications de configuration apportées au périphérique actif ou passif avec son homologue.

  Le port de gestion est utilisé pour la liaison HA1. Les ports TCP 28769 et 28260 pour les communications en texte clair ; le port 28 pour les communications cryptées (SSH sur TCP).
- **Liaison de données**—La liaison HA2 permet de synchroniser des sessions, des tables de transfert, des associations de sécurité IPSec et des tables ARP entre des périphériques d'une paire HA. Le flux de données de la liaison HA2 est toujours unidirectionnel (sauf pour la persistance HA2), du périphérique actif vers le périphérique passif.

  Ethernet1/1 doit être attribué comme liaison HA2. Cette opération est nécessaire pour déployer le pare-feu CN-Series dans AWS en mode HA. La liaison de données HA peut être configurée pour utiliser IP (numéro de protocole 99) ou UDP (port 29281) comme protocole de transport.

Dans AWS, le pare-feu CN-Series ne prend pas en charge les liens de secours pour HA1 ou HA2.

Analyse des pulsations et messages Hello

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Les pare-feu utilisent les messages hello et les pulsations pour vérifier que le périphérique homologue est réactif et opérationnel. Les messages Hello sont envoyés par un homologue à un autre à un *intervalle Hello* configuré afin de vérifier l'état du périphérique. La pulsation est une requête ping ICMP envoyée à l'homologue HA sur la liaison de contrôle et l'homologue y répond pour indiquer que les périphériques sont connectés et réactifs. Pour plus d'informations sur les minuteurs HA qui déclenchent un basculement, reportez-vous à la section Minuteurs HA. (Les minuteurs HA du pare-feu CN-Series sont identiques à ceux des pare-feu PA-5200 Series.)

Priorité et préemption des périphériques

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Les périphériques d'une paire HA peuvent être assignés à une valeur *Priorité du périphérique* afin d'indiquer une préférence pour laquelle un périphérique doit assumer un rôle actif et gérer le trafic lors d'un basculement. Si vous devez utiliser un périphérique spécifique dans la paire HA pour la sécurisation active du trafic, vous devez activer le comportement préemptif sur les deux pare-feu et assigner une valeur de priorité pour chaque périphérique. Le périphérique affichant la valeur numérique la plus basse et, par conséquent, la *priorité la plus élevée*, est désigné comme étant actif et gère l'ensemble du trafic sur le réseau. L'autre périphérique est dans un état passif et synchronise les informations de configuration et d'état avec le périphérique actif afin d'être prêt à passer en état actif en cas d'échec.

La valeur numérique inférieure devient active lors du premier déploiement. Si la valeur numérique la plus élevée est déployée en premier et que la préemption est désactivée, la valeur numérique la plus élevée deviendra active.

La préemption n'est pas recommandée pour la HA dans le pare-feu CN-Series sur AWS.

Par défaut, la préemption est désactivée sur les pare-feu. Lorsqu'il est activé, le comportement préemptif autorise le pare-feu affichant la *priorité la plus élevée* (valeur numérique la plus basse) à reprendre en état actif après avoir récupéré d'un échec. En cas de préemption, l'événement est consigné dans les journaux système.

Pour ajouter une priorité, vous devez vous assurer que la valeur du paramètre `PAN_HA_PRIORITY` est définie sur une valeur numérique dans les `pan-cn-mgmt-configmap-0.yaml` et `pan-cn-mgmt-configmap-1.yaml`.

Par exemple :

```
PAN_HA_PRIORITY : “10”
```

Minuteurs HA

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
	• Helm 3.6 or above version client pour le déploiement CN-Series avec Helm

Les minuteurs haute disponibilité (HA) permettent de détecter une défaillance du pare-feu et de déclencher un basculement. Pour réduire la complexité de configuration des minuteurs HA, vous pouvez sélectionner l'un des trois profils suivants : **Recommended (Recommandé)**, **Aggressive (Agressif)** et **Advanced (Avancé)**. Ces profils renseignent automatiquement les valeurs optimales des minuteurs HA pour une plate-forme de pare-feu spécifique afin de permettre un déploiement HA accéléré.

Utilisez le profil **Recommended (Recommandé)** si vous souhaitez des paramètres de minuteur de basculement types ou le profil **Aggressive (Agressif)** si vous préférez des paramètres de minuteur de basculement plus rapides. Le profil **Advanced (Avancé)** vous permet de personnaliser les valeurs des minuteurs pour répondre à vos besoins en matière de réseau.

Minuteur HA sur la série CN dans AWS	Valeurs par défaut pour les profils recommandé ou agressif
Délai de maintien de promotion	2000/500 ms
Intervalle Hello	8000/8000 ms
Intervalle de pulsation	2000/1000 ms
Nombre maximal de battements	3/3
Délai de maintien de préemption	1/1 min
Temps d'attente actif après l'échec de la surveillance	0/0 ms
Temps d'attente actif principal supplémentaire	500/500 ms

Configurer la HA active/passive dans AWS EKS à l'aide d'un IP secondaire

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure

Effectuez la procédure suivante pour déployer les nouveaux pare-feu CN-Series en tant que paire HA avec des adresses IP secondaires.

STEP 1 | Avant de déployer les pare-feu CN-Series pour votre paire HA, assurez-vous de ce qui suit :

- Déployez les deux homologues HA dans la même zone de disponibilité AWS. Reportez-vous à la section Rôles IAM pour la HA.
- Créez un rôle IAM et attribuez-le aux nœuds de travail exécutant les pare-feu CN-Series lorsque vous déployez les instances.
- Les pare-feu actifs et passifs doivent avoir au moins trois interfaces chacun : interface de gestion, interface HA2 et interface de données.

  Par défaut, l'interface de gestion sera utilisée comme interface HA1.
- Créez des interfaces réseau sur AWS dans la même zone de disponibilité que le cluster. Ajoutez une étiquette sur l'`eni` afin qu'il ne soit pas géré par AWS et puisse être utilisé par multus :

  ```
  node.k8s.amazonaws.com/no_manage: True
  ```
- Vérifiez que les composants réseau et de sécurité sont définis de manière appropriée.
  - Activez la communication avec Internet. Le VPC par défaut inclut une passerelle Internet et si vous installez le pare-feu CN-Series dans le sous-réseau par défaut, il a accès à Internet.
  - Créez des sous-réseaux. Les sous-réseaux sont des segments de la plage d'adresses IP affectée au VPC dans lequel vous lancez les instances EC2. Le pare-feu CN-Series doit appartenir au sous-réseau public pour pouvoir être configuré pour l'accès à Internet.
  - Créez un groupe de sécurité des données qui comprend les interfaces de données du pare-feu. De plus, configurez la sécurité pour autoriser tout le trafic, afin que la sécurité soit renforcée par les pare-feu. Ceci est nécessaire pour maintenir les sessions existantes pendant le basculement.
  - Ajoutez des itinéraires à la table de routage pour un sous-réseau privé afin de vous assurer que le trafic peut être acheminé entre des sous-réseaux et des groupes de sécurité du VPC, le cas échéant.

  Lors du déploiement du pare-feu CN-Series dans EKS, la récupération du jeton IMDSv2 échoue si la valeur `http-put-response-hop-limit` est définie sur la valeur par défaut 1. Vous devez vous assurer que la valeur limite de saut est définie sur 3 ou plus lorsque IMDSv2 est activé.

  Par exemple :

  Exécutez la commande suivante :

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


STEP 2 | Déployez le pare-feu CN-Series dans EKS.

1. Configurez ethernet 1/1 comme interface HA2 sur chaque homologue HA.
   1. Ouvrez la console EC2 Amazon.
   2. Sélectionnez Network Interface (Interface réseau), puis choisissez et sélectionnez votre interface réseau.
   3. Sélectionnez **Actions** > **Manage IP Addresses (Gérer les adresses IP).**
   4. Laissez le champ vide pour permettre à AWS d'attribuer une adresse IP de manière dynamique ou saisissez une adresse IP dans la plage du sous-réseau pour le pare-feu CN-Series. Cela attribuera une adresse IP secondaire à l'interface HA2.
   5. Cliquez sur **Yes (Oui)** et sur **Update (Mettre à jour)**.
   6. Sélectionnez **Actions** > **Change Source/Dest. Check (Modifier la vérification de la source/destination)** et sélectionnez **Disable (Désactiver)**.
   7. Répétez ce processus sur le deuxième (à être passif) homologue HA.
2. Ajoutez une adresse IP secondaire à vos interfaces de plan de données sur le premier (à être actif) homologue HA.
   1. Sélectionnez **Network Interface (Interface réseau)**, puis choisissez et sélectionnez votre interface réseau.
   2. Sélectionnez **Actions** > **Manage IP Addresses (Gérer les adresses IP)** > **IPv4 Addresses (Adresses IPv4)** > **Assign new IP (Attribuer une nouvelle IP)**.
   3. Laissez le champ vide pour permettre à AWS d'attribuer une adresse IP de manière dynamique ou saisissez une adresse IP dans la plage du sous-réseau pour le pare-feu CN-Series.
   4. Cliquez sur **Yes (Oui)** et sur **Update (Mettre à jour)**.
3. Associez une adresse IP élastique (publique) secondaire à l'interface non approuvée de l'homologue actif.
   1. Sélectionnez **Elastic IPs (IP élastiques)** puis choisissez et sélectionnez l'adresse IP élastique à associer.
   2. Sélectionnez **Actions** > **Associate Elastic IP (Associer une IP élastique)**.
   3. Sous **Resource Type (Type de ressource)**, sélectionnez **Network Interface (Interface réseau)**.
   4. Choisissez l'interface réseau à laquelle associer l'adresse IP élastique.
   5. Cliquez sur **Associate (Associer)**.
4. Pour l'inspection du trafic sortant, ajoutez une entrée à la table de routage du sous-réseau qui définit le prochain saut comme l'interface approuvée du pare-feu.
   1. Sélectionnez **VPC** > **Route Tables (Tables de routage)**.
   2. Choisissez votre table de routage de sous-réseau.
   3. Sélectionnez **Actions** > **Edit routes (Modifier les routes)** > **Add route (Ajouter une route)**.
   4. Saisissez l'adresse IP ou le bloc CIDR de **Destination**.
   5. Pour **Target (Cibler)**, entrez dans l'interface réseau de l'interface approuvée du pare-feu.

6. Cliquez sur **Save routes (Enregistrer routes)**.

5. Pour utiliser le routage des entrées AWS, créez une table de routage et associez-y la passerelle Internet. Ensuite, ajoutez une entrée avec le prochain saut défini comme l'interface non approuvée du pare-feu actif.
   1. Sélectionnez **Route Tables (Tables de routage)** > **Create route table (Créer une table de routage)**.
   2. (En option) Saisissez une **Name tag (Étiquette de nom)** descriptive pour votre table de routage.
   3. Cliquez sur **Create (Créer)**.
   4. Cliquez sur votre table de routage et sélectionnez **Actions** > **Edit edge associations (Modifier les associations périphériques)**.
   5. Sélectionnez **Internet gateways (Passerelles Internet)** et choisissez votre passerelle Internet VPC.
   6. Cliquez sur **Save (Enregistrer)**.
   7. Cliquez sur votre table de routage et sélectionnez **Actions** > **Edit routes (Modifier les routes)**.
   8. Pour **Target (Cible)**, sélectionnez **Network Interface (Interface réseau)** et choisissez l'interface non approuvée du pare-feu actif.
   9. Cliquez sur **Save routes (Enregistrer routes)**.

STEP 3 | Activez HA.

Pour activer la prise en charge HA, vous devez vous assurer que la valeur du paramètre PAN_HA_SUPPORT est « true » dans les fichiers YAML suivants :

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

L'adresse IP HA1 de l'homologue est configurée automatiquement.

STEP 4 | Récupérez l'adresse IP statique de l'interface HA2 à partir de l'instance de nœud correspondante sur la console AWS et ajoutez-la au paramètre d'adresse du fichier `net-attach def-ha2-0.yaml` et `net-attach-def-ha2-1.yaml`.

(Facultatif) Modifiez le **Threshold** (Seuil) des paquets de **HA2 Keep-alive** (Persistance HA2). Par défaut, la **HA2 Keep-alive (Persistance HA2)** est activée à des fins de surveillance de la liaison de données HA2 entre les homologues. En cas d'échec, si ce seuil (établi par défaut à 10000 ms) est dépassé, l'action définie se produira. Un message critique du journal système est généré en cas d'échec de la persistance HA2.

*Vous pouvez configurer l'option **HA2 Keep-alive (Persistance HA2)** sur les deux périphériques ou sur un seul périphérique de la paire HA. Si vous activez cette option sur un seul périphérique, il sera le seul à envoyer des messages de persistance.*

STEP 5 | Vérifiez que les pare-feu sont appariés en HA actif/passif.

1. Accédez au **Dashboard (Tableau de bord)** sur les deux pare-feu et affichez le widget High Availability (Haute disponibilité).
2. Sur l'homologue HA actif, cliquez sur **Sync to peer (Synchroniser à l'homologue)**.
3. Vérifiez que les pare-feu sont appariés et synchronisés.
   - Sur le pare-feu passif : l'état du pare-feu local doit afficher **Passive (Passif)** et la **Running Config (Configuration en cours)** doit être Synchronized (Synchronisée).
   - Sur le pare-feu actif : l'état du pare-feu local doit afficher **Active (Actif)** et la **Running Config (Configuration en cours)** doit être Synchronized (Synchronisée).
4. Depuis l'interface de ligne de commande (CLI) du pare-feu, exécutez les commandes suivante :
   - Pour vérifier l'état de préparation du basculement :

     ```
     show plugins vm_series aws ha state
     ```
   - Pour afficher le mappage d'IP secondaire :

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


Configurer DPDK sur le pare-feu CN-Series

Où puis-je utiliser ceci ?	De quoi ai-je besoin ?
• Déploiement CN-Series	• CN-Series 10.2.x or above Container Images • Panorama sous PAN-OS 10.2.x ou version supérieure • Helm 3.6 or above version client

Le kit de développement de plan de données (DPDK) fournit un cadre simple pour le traitement rapide des paquets dans les applications de plan de données.

Le mode DPDK est pris en charge sur le pare-feu CN-Series en tant que fonction de réseau de conteneur (CNF) Kubernetes uniquement.

DHCP IPAM n'est pas pris en charge en mode DPDK.

Configuration système requise

Pour exécuter une application DPDK, vous devez effectuer les personnalisations suivantes sur votre machine cible.

- **Configuration du noyau**—Activez l'option HUGETLBFS dans le noyau du système d'exploitation hôte.
- **KNI et UIO/VFIO**—Insérez KNI et UIO/VFIO dans le noyau du système d'exploitation hôte.
- **HugePages**

  1. **Réserver HugePages**

     - Réservez HugePages pendant l'exécution, avant le démarrage des pods. Ajoutez le nombre de HugePages requises au fichier `nr_hugepages` dans le répertoire `/sys/kernel/` correspondant à une taille de page spécifique (en Ko). Par exemple, utilisez la commande suivante pour un système à nœud unique si 1 024 pages sur 2 millions sont requises.

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

     - Réserver HugePages au démarrage. Par exemple, pour réserver un HugePage de mémoire 4G sous la forme de quatre pages 1G, les options suivantes doivent être transmises au noyau.

       ```
       default_hugepagesz=1G énormespagesz=1G énormespages=4
       ```

2. **Utiliser HugePages avec DPDK**—Créez des points de montage pour HugePages, car PanOS 10.2 utilise des processus secondaires DPDK.

   Voici un exemple de commande pour créer un HugePage e de taille 1 Go pour une utilisation DPDK.

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

3. Redémarrez le service kubelet sur l'hôte après avoir activé HugePages à l'aide de la commande suivante.

   ```
   sudo systemctl restart kubelet
   ```

4. Vérifiez `/sys/fs/cgroup/hugetlb/kubepods.slice/hugetlb.2MB.limit_in_bytes` pour vous assurer que la taille correspond à la taille

du HugePage. Si la taille ne correspond pas à la taille du HugePage, mettez à jour la taille à l'aide de la commande suivante.

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

Dans un pod, les applications peuvent allouer et consommer des HugePages pré-alloués de plusieurs tailles. Les applications consomment des HugePages via les besoins en ressources au niveau du conteneur en utilisant le nom de la ressource<size>. Par exemple, hugepages-2Mi ou hugepages-1Gi.

Contrairement au processeur ou à la mémoire, les HugePages ne prennent pas en charge la survalidation.

Le mode privilégié est activé pour accéder à l'espace du périphérique hôte. Pour répertorier et lier les périphériques réseau à un conteneur, montez `/sys` sur le conteneur afin que DPDK puisse accéder aux fichiers du répertoire.

Voici un extrait de code pour activer HugePages sur 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_NAME
 valueFrom: fieldRef: fieldPath: metadata.name - 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
```

- **Épinglage NUMA et du processeur**—Plusieurs processus DPDK ne peuvent pas s'exécuter sur le même cœur, car ils provoquent, entre autres, une corruption du cache du pool de mémoire.

Les processus secondaires sont épinglés à un noyau différent. Utilisez les options d'épinglage du processeur dans *configmap* pour contrôler les processus secondaires.

- **Modifications de configuration et de pod**
  - Activez `PAN_DATA_MODE: "dpdk"` dans `pan-cn-ngfw-configmap-0.yaml` et `pan-cn-ngfw-configmap-1.yaml`.

    DPDK n'est pas le mode par défaut pour CN-Series-as-a-kubernetes-CNF.

  - Faites correspondre le paramètre `#HUGEPAGE_MEMORY_REQUEST` avec la demande de mémoire HugePage dans `pan-cn-ngfw-configmap-0.yaml` et `pan-cn-ngfw-configmap-1.yaml`.

    Si la mémoire Hugepage n'est pas disponible, elle est par défaut sur MMAP.

Pour plus d'informations, consultez la configuration système requise pour DPDK.

Vous pouvez configurer DPDK sur des nœuds de travail sur site et un cluster AWS EKS

- Configurer DPDK sur des nœuds de travail sur site
- Configurer DPDK dans AWS EKS

Configurer DPDK sur des nœuds de travail sur site

STEP 1 | Installez les dépendances suivantes :

Exécutez toutes les commandes sur le noeud esclave sur lequel vous souhaitez configurer DPDK.

- Pour CentOS :

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

- Pour Ubuntu OS :

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


STEP 2 | Après avoir installé les dépendances :

- Téléchargez le fichier DPDK tar sur https://fast.dpdk.org/rel/. Voir la documentation DPDK pour les étapes de compilation.

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

- Décompressez le fichier.

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

- Compilez le fichier. Le fichier compilé sera dans le sous-dossier `x86_64-native-linuxapp-gcc`

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

STEP 3 | Insérez le module noyau compilé statistiquement ou dynamiquement pendant l'exécution (modprobe/insmod). Pour plus d'informations, voir module noyau.

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

Sur Ubuntu, si vous voyez l'erreur—`inmod: ERROR: could not insert module igb_uio.ko (ERREUR : impossible d'insérer le module igb_uio.ko)`, insérez le module `uio` en premier.

```
modprobe uio
```

STEP 4 | Utilisez des moyens spécifiques à la distribution pour insérer des modules pendant le démarrage. Sinon, vous pouvez créer un service qui exécute les commandes modprobe/insmod à chaque démarrage du système.

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

STEP 5 | Activez et montez 2M hugepages de taille 2048K.

Vous pouvez également activer hugepages en utilisant le script de service à l'étape 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 | Créez un instantané de la VM pour une utilisation future.

Configurer DPDK dans AWS EKS

Sur AWS EKS, chaque pod possède une interface réseau affectée par le plug-in CNI Amazon VPC. En utilisant Multus, vous pouvez créer des pods avec plusieurs interfaces.

STEP 1 | Créez un compte AWS si vous n'en avez pas déjà un.

STEP 2 | Créez un cluster EKS à l'aide de l'AMI personnalisée. Pour plus d'informations, consultez Création d'un cluster Amazon EKS.

STEP 3 | Modifiez les paramètres du VPC et du nœud. Pour plus d'informations, consultez la documentation AWS EKS.

STEP 4 | (Multus)Ajoutez plusieurs ENI au nœud EKS et chargez les pilotes KNI et UIO.

- Utilisez l'étiquette suivante pour ajouter plusieurs ENI au nœud EKS.

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

  Lorsque l'étiquette est détectée, l'interface peut alors être utilisée par Multus CNI. Pour plus d'informations, consultez la documentation AWS.

- Exécutez les commandes suivantes dans le CLI d'AWS.

  ```
  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 vous n'utilisez pas d'AMI personnalisée) Activez les pages énormes sur les nœuds de travail.

  ```
  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
  ```

