Options de configuration avancées

NetApp Solutions

NetApp
September 26, 2024

Sommaire

Options de configuration avancées 1
Exploration des options de Load Balancer 1
Création de registres d’images privées 22

Options de configuration avancées

Exploration des options de Load Balancer

Exploration des options d'équilibreur de charge avec Red Hat OpenShift avec NetApp

Dans la plupart des cas, Red Hat OpenShift met les applications à la disposition du monde extérieur via des routes. Un service est exposé en lui donnant un nom d'hôte accessible en externe. La route définie et les points de terminaison identifiés par son service peuvent être utilisés par un routeur OpenShift pour fournir cette connectivité nommée aux clients externes.

Cependant, dans certains cas, les applications nécessitent le déploiement et la configuration d'équilibreurs de charge personnalisés pour exposer les services appropriés. Il s'agit notamment du centre de contrôle NetApp Astra, Pour répondre à ce besoin, nous avons évalué un certain nombre d'options d'équilibrage de charge personnalisé. Leur installation et leur configuration sont décrites dans cette section.

Les pages suivantes présentent des informations supplémentaires sur les options de équilibreur de charge validées dans la solution Red Hat OpenShift avec NetApp :

- "MetalLB"
- "F5 BIG-IP"

Installation d'équilibreurs de charge MetalLB : Red Hat OpenShift avec NetApp

Cette page répertorie les instructions d'installation et de configuration de l'équilibreur de charge MetalLB.

MetalLB est un équilibreur de charge réseau hébergé automatiquement sur votre cluster OpenShift qui permet la création de services OpenShift d'équilibreur de charge dans les clusters qui ne s'exécutent pas sur un fournisseur cloud. Les deux principales caractéristiques de MetalLB qui fonctionnent ensemble pour prendre en charge les services LoadBalancer sont l'allocation d'adresses et l'annonce externe.

Options de configuration MetalLB

D'après la façon dont MetalLB annonce l'adresse IP attribuée aux services LoadBalancer en dehors du cluster OpenShift, elle fonctionne selon deux modes :

- **Mode de couche 2.** dans ce mode, un nœud du cluster OpenShift est propriétaire du service et répond aux demandes ARP pour cette IP pour la rendre accessible en dehors du cluster OpenShift. Seul le nœud annonce l'IP, il présente un goulot d'étranglement au niveau de la bande passante et des limitations de basculement lentes. Pour plus d'informations, reportez-vous à la documentation "ici".
- **Mode BGP.** dans ce mode, tous les nœuds du cluster OpenShift établissent des sessions de peering BGP avec un routeur et annoncent les routes pour transférer le trafic vers les adresses IP du service. La condition préalable est d'intégrer MetalLB à un routeur de ce réseau. En raison du mécanisme de hachage dans BGP, il possède une certaine limite lors du mappage d'IP à nœud pour les modifications de service. Pour plus d'informations, reportez-vous à la documentation "ici".

Pour les besoins de ce document, nous allons configurer MetalLB en mode couche 2.

Installation de l'équilibreur de charge MetalLB

1. Téléchargez les ressources MetalLB.

```
[netapp-user@rhel7 ~]$ wget
https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/name
space.yaml
[netapp-user@rhel7 ~]$ wget
https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/meta
llb.yaml
```

2. Modifier le fichier `metallb.yaml` et déposer `spec.template.spec.securityContext` À partir du déploiement du contrôleur et de l'ensemble des haut-parleurs.

   Lignes à supprimer :

```
securityContext:
  runAsNonRoot: true
  runAsUser: 65534
```

3. Créer le `metallb-system` espace de noms.

```
[netapp-user@rhel7 ~]$ oc create -f namespace.yaml
namespace/metallb-system created
```

4. Créer la CR du MetalLB.

```
[netapp-user@rhel7 ~]$ oc create -f metallb.yaml
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
role.rbac.authorization.k8s.io/controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller
created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker
created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
rolebinding.rbac.authorization.k8s.io/controller created
daemonset.apps/speaker created
deployment.apps/controller created
```

5. Avant de configurer le haut-parleur MetalLB, accordez à l'intervenant DemonSet des privilèges élevés afin qu'il puisse effectuer la configuration réseau requise pour que les équilibreurs de charge fonctionnent.

```
[netapp-user@rhel7 ~]$ oc adm policy add-scc-to-user privileged -n
metallb-system -z speaker
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:privileged
added: "speaker"
```

6. Configurez MetalLB en créant un `ConfigMap` dans le `metallb-system` espace de noms.

```
[netapp-user@rhel7 ~]$ vim metallb-config.yaml

apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
      addresses:
      - 10.63.17.10-10.63.17.200

[netapp-user@rhel7 ~]$ oc create -f metallb-config.yaml
configmap/config created
```

7. Maintenant que des services loadBALB sont créés, MetalLB attribue un IP externe aux services et annonce l'adresse IP en répondant aux demandes ARP.

> Si vous souhaitez configurer MetalLB en mode BGP, ignorez l'étape 6 ci-dessus et suivez la procédure décrite dans la documentation MetalLB "ici".

Installation des presses à balles F5 BIG-IP Load Balancers

F5 BIG-IP est un contrôleur de distribution d'applications (ADC) qui offre un large éventail de services avancés de gestion du trafic et de sécurité de niveau production, tels que L4-L7 d'équilibrage de charge, d'allègement de la charge SSL/TLS, de DNS, de pare-feu et bien d'autres. Ces services augmentent considérablement la disponibilité, la sécurité et les performances de vos applications.

F5 BIG-IP peut être déployé et utilisé de différentes façons, sur un matériel dédié, dans le cloud ou comme appliance virtuelle sur site. Reportez-vous à la documentation ici pour explorer et déployer F5 BIG-IP selon les besoins.

Pour une intégration efficace des services F5 BIG-IP avec Red Hat OpenShift, F5 propose UN service CIS (BIG-IP Container Ingress Service). CIS est installé en tant que pod de contrôleur qui surveille l'API OpenShift pour certaines définitions de ressources personnalisées (CRD) et gère la configuration système F5 BIG-IP. F5 BIG-IP CIS peut être configuré pour contrôler les types de service LoadBalancers et les routes dans OpenShift.

En outre, pour l'allocation automatique d'adresses IP pour le traitement du type LoadBalancer, vous pouvez utiliser le contrôleur F5 IPAM. Le contrôleur F5 IPAM est installé comme un pod de contrôleur qui surveille l'API OpenShift pour les services LoadBalancer avec une annotation ipamLabel afin d'allouer l'adresse IP à partir d'un pool préconfiguré.

Cette page répertorie les instructions d'installation et de configuration pour F5 BIG-IP CIS et contrôleur IPAM.

Un système F5 BIG-IP doit être déployé et sous licence. Il doit également être concédé sous licence pour les services SDN, qui sont inclus par défaut avec la licence de base BIG-IP VE.

F5 BIG-IP peut être déployé en mode autonome ou cluster. Aux fins de cette validation, F5 BIG-IP a été déployé en mode autonome, mais pour la production, il est préférable d'avoir un cluster de BIG-IP pour éviter un seul point de défaillance.

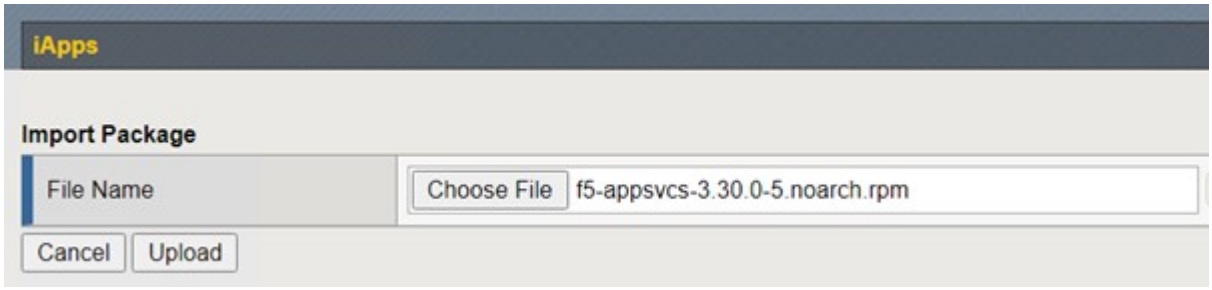
Un système F5 BIG-IP peut être déployé sur un matériel dédié, dans le cloud ou en tant qu'appliance virtuelle sur site avec des versions supérieures à 12.x pour une intégration avec F5 CIS. Dans le cadre de ce document, le système F5 BIG-IP a été validé en tant qu'appliance virtuelle, par exemple en utilisant L'édition BIG-IP VE.

Versions validées

De déduplication	Version logicielle
Red Hat OpenShift	4.6 EUS, 4.7
F5 BIG-IP VE EDITION	16.1.0
Service F5 d'entrée de conteneur	2.5.1
Contrôleur F5 IPAM	0.1.4
AS3 F5	3.30.0

Installation

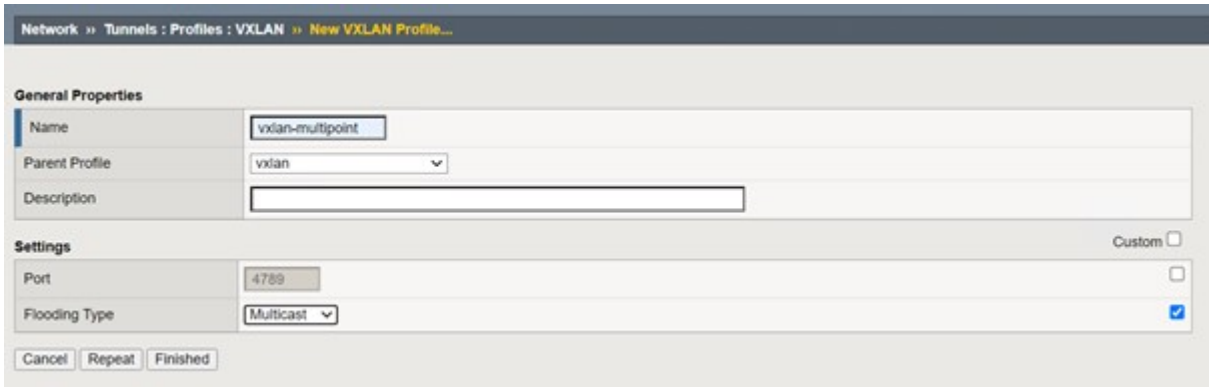
1. Installez l'extension F5 application Services 3 pour permettre aux systèmes BIG-IP d'accepter les configurations au format JSON au lieu de commandes impérative. Accédez à "Référentiel GitHub F5 AS3"Et téléchargez le dernier fichier RPM.
2. Connectez-vous au système F5 BIG-IP, accédez à iApps > Package Management LX et cliquez sur Importer.
3. Cliquez sur choisir un fichier et sélectionnez le fichier RPM AS3 téléchargé, cliquez sur OK, puis cliquez sur Télécharger.



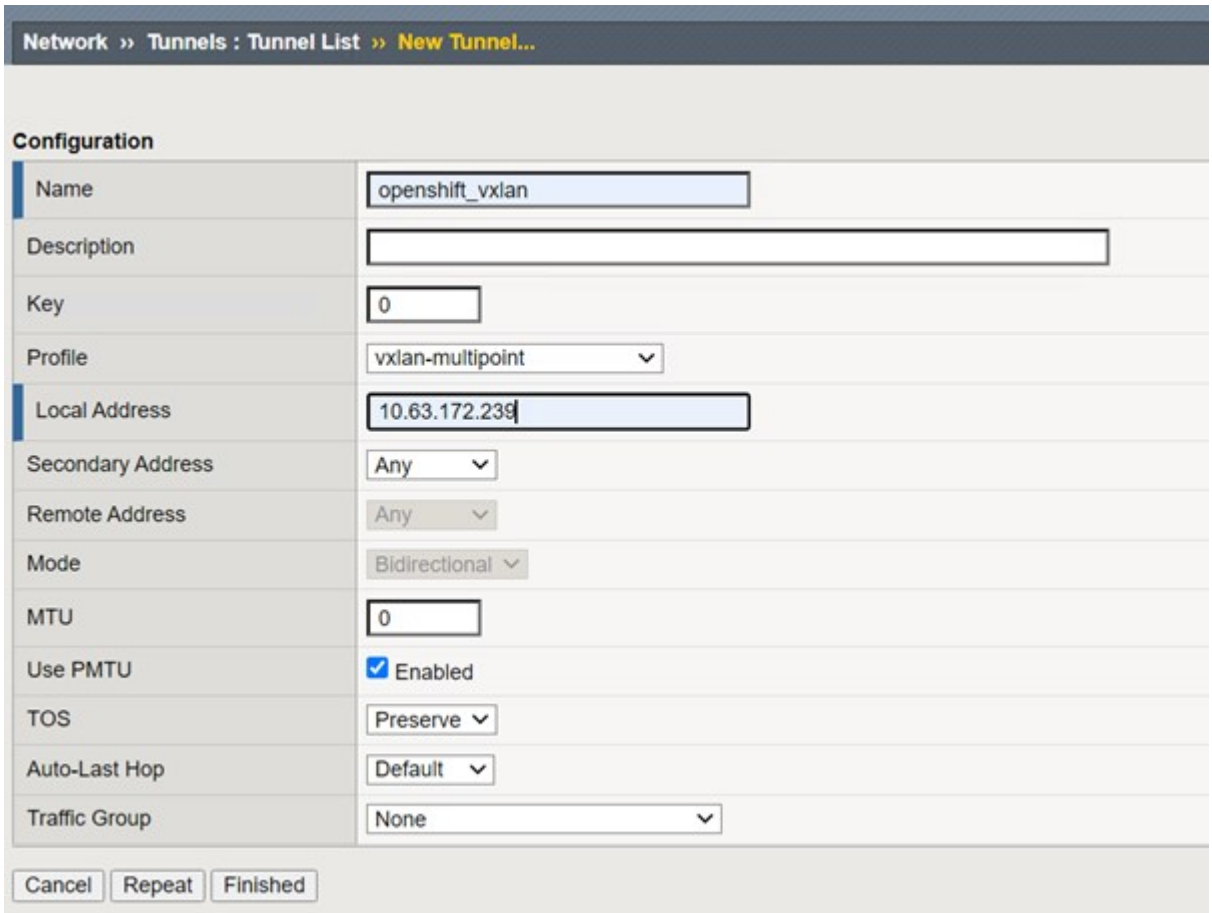
4. Vérifiez que l'extension AS3 est correctement installée.

5. Configurez ensuite les ressources requises pour la communication entre les systèmes OpenShift et BIG-IP.

Commencez par créer un tunnel entre OpenShift et LE serveur BIG-IP en créant une interface de tunnel VXLAN sur le système BIG-IP pour OpenShift SDN. Naviguez jusqu'à réseau > tunnels > profils, cliquez sur Créer, puis définissez le profil parent sur vxlan et le type d'inondation sur Multicast. Entrez un nom pour le profil et cliquez sur terminé.



6. Naviguez jusqu'à réseau > tunnels > liste de tunnels, cliquez sur Créer, puis entrez le nom et l'adresse IP locale du tunnel. Sélectionnez le profil de tunnel créé à l'étape précédente et cliquez sur terminé.



7. Connectez-vous au cluster Red Hat OpenShift avec les privilèges cluster-admin.
8. Créez un sous-réseau d'hôtes sur OpenShift pour le serveur F5 BIG-IP, qui étend le sous-réseau du cluster OpenShift au serveur F5 BIG-IP. Téléchargez la définition YAML du sous-réseau hôte.

```
wget https://github.com/F5Networks/k8s-bigip-
ctlr/blob/master/docs/config_examples/openshift/f5-kctlr-openshift-
hostsubnet.yaml
```

9. Modifiez le fichier de sous-réseau de l'hôte et ajoutez l'IP VTEP (VXLAN tunnel) BIG-IP pour le SDN OpenShift.

```
apiVersion: v1
kind: HostSubnet
metadata:
  name: f5-server
  annotations:
    pod.network.openshift.io/fixed-vnid-host: "0"
    pod.network.openshift.io/assign-subnet: "true"
# provide a name for the node that will serve as BIG-IP's entry into the
cluster
host: f5-server
# The hostIP address will be the BIG-IP interface address routable to
the
# OpenShift Origin nodes.
# This address is the BIG-IP VTEP in the SDN's VXLAN.
hostIP: 10.63.172.239
```

Modifiez l'adresse IP de l'hôte et d'autres détails applicables à votre environnement.

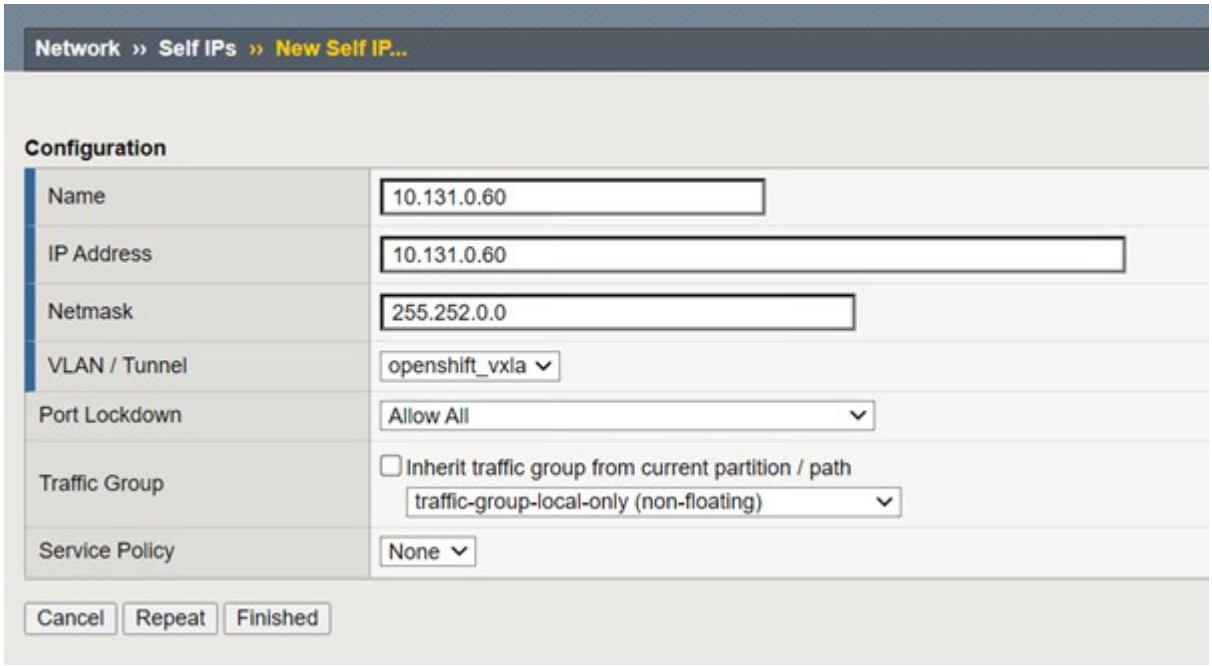
10. Créez la ressource HostSubnet.

```
[admin@rhel-7 ~]$ oc create -f f5-kctlr-openshift-hostsubnet.yaml

hostsubnet.network.openshift.io/f5-server created
```

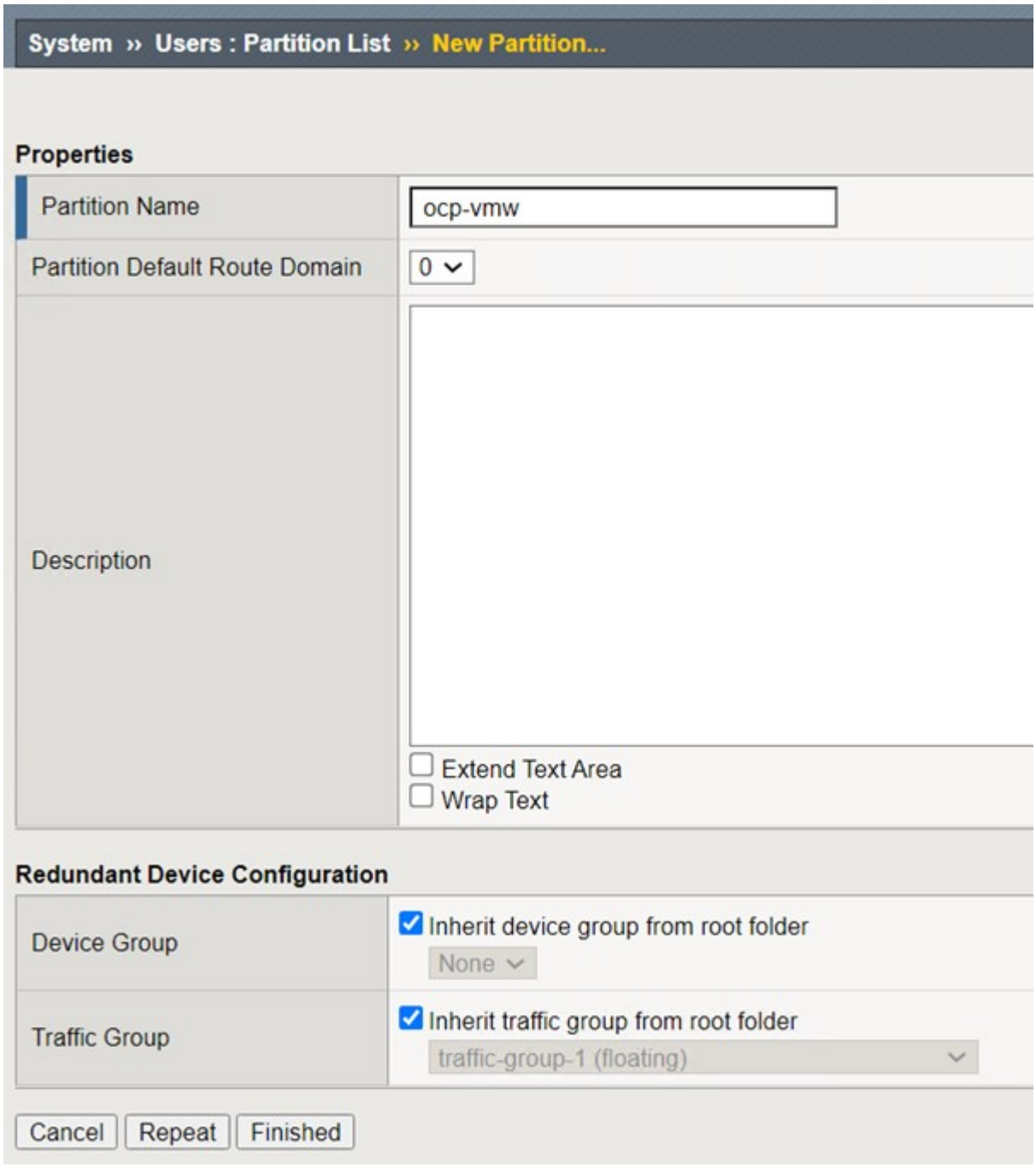
11. Obtenez la plage de sous-réseau IP du cluster pour le sous-réseau hôte créé pour le serveur F5 BIG-IP.

```
[admin@rhel-7 ~]$ oc get hostsubnet

NAME                           HOST                           HOST IP
SUBNET          EGRESS CIDRS   EGRESS IPS
f5-server                      f5-server                      10.63.172.239
10.131.0.0/23
ocp-vmw-nszws-master-0         ocp-vmw-nszws-master-0         10.63.172.44
10.128.0.0/23
ocp-vmw-nszws-master-1         ocp-vmw-nszws-master-1         10.63.172.47
10.130.0.0/23
ocp-vmw-nszws-master-2         ocp-vmw-nszws-master-2         10.63.172.48
10.129.0.0/23
ocp-vmw-nszws-worker-r8fh4     ocp-vmw-nszws-worker-r8fh4     10.63.172.7
10.130.2.0/23
ocp-vmw-nszws-worker-tvr46     ocp-vmw-nszws-worker-tvr46     10.63.172.11
10.129.2.0/23
ocp-vmw-nszws-worker-wdxhg     ocp-vmw-nszws-worker-wdxhg     10.63.172.24
10.128.2.0/23
ocp-vmw-nszws-worker-wg8r4     ocp-vmw-nszws-worker-wg8r4     10.63.172.15
10.131.2.0/23
ocp-vmw-nszws-worker-wtgfw     ocp-vmw-nszws-worker-wtgfw     10.63.172.17
10.128.4.0/23
```

12. Créez un auto-IP sur OpenShift VXLAN avec un IP dans la plage de sous-réseau hôte d'OpenShift correspondant au serveur F5 BIG-IP. Connectez-vous au système F5 BIG-IP, accédez à réseau > Auto-IP et cliquez sur Créer. Entrez une adresse IP à partir du sous-réseau IP du cluster créé pour le sous-réseau hôte F5 BIG-IP, sélectionnez le tunnel VXLAN et entrez les autres détails. Cliquez ensuite sur terminé.



13. Créez une partition dans le système F5 BIG-IP à configurer et à utiliser avec CIS. Accédez à système > utilisateurs > liste de partitions, cliquez sur Créer et entrez les détails. Cliquez ensuite sur terminé.



F5 recommande de ne pas effectuer de configuration manuelle sur la partition gérée par CIS.

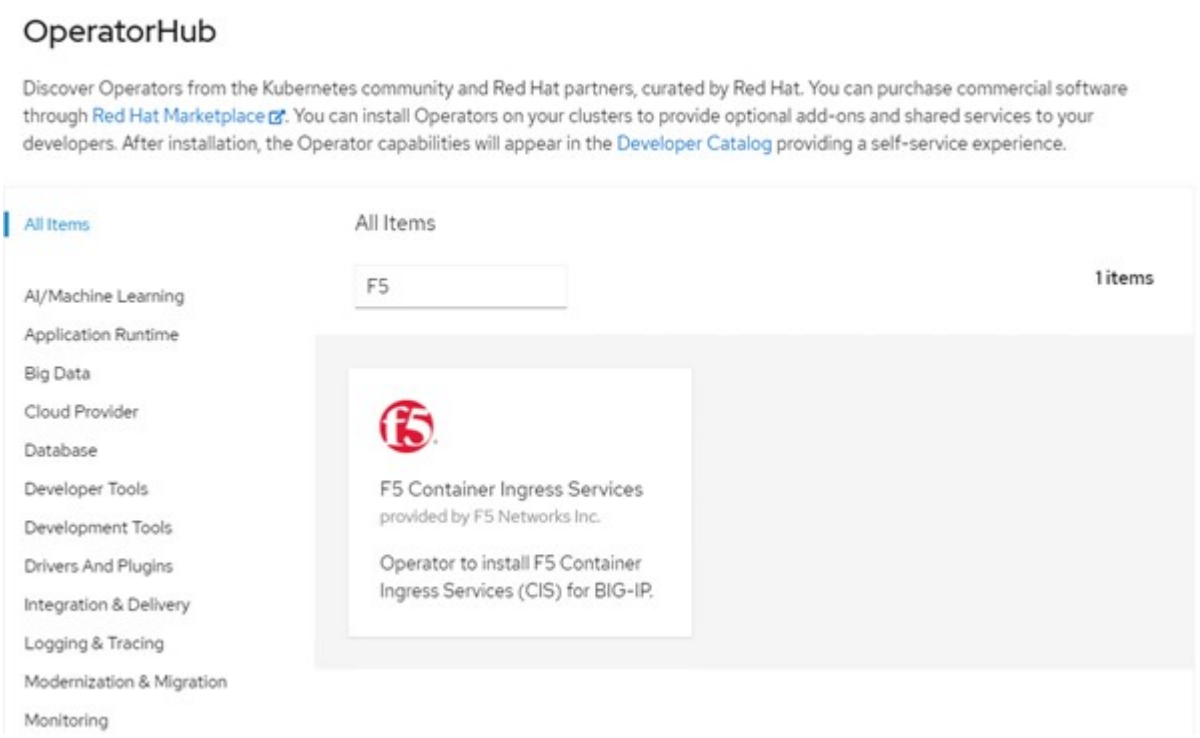
14. Installez F5 BIG-IP CIS à l'aide de l'opérateur depuis OperatorHub. Connectez-vous au cluster Red Hat OpenShift avec des privilèges cluster-admin et créez un secret avec les identifiants de connexion du système F5 BIG-IP. Il est indispensable pour l'opérateur.

```
[admin@rhel-7 ~]$ oc create secret generic bigip-login -n kube-system
--from-literal=username=admin --from-literal=password=admin

secret/bigip-login created
```

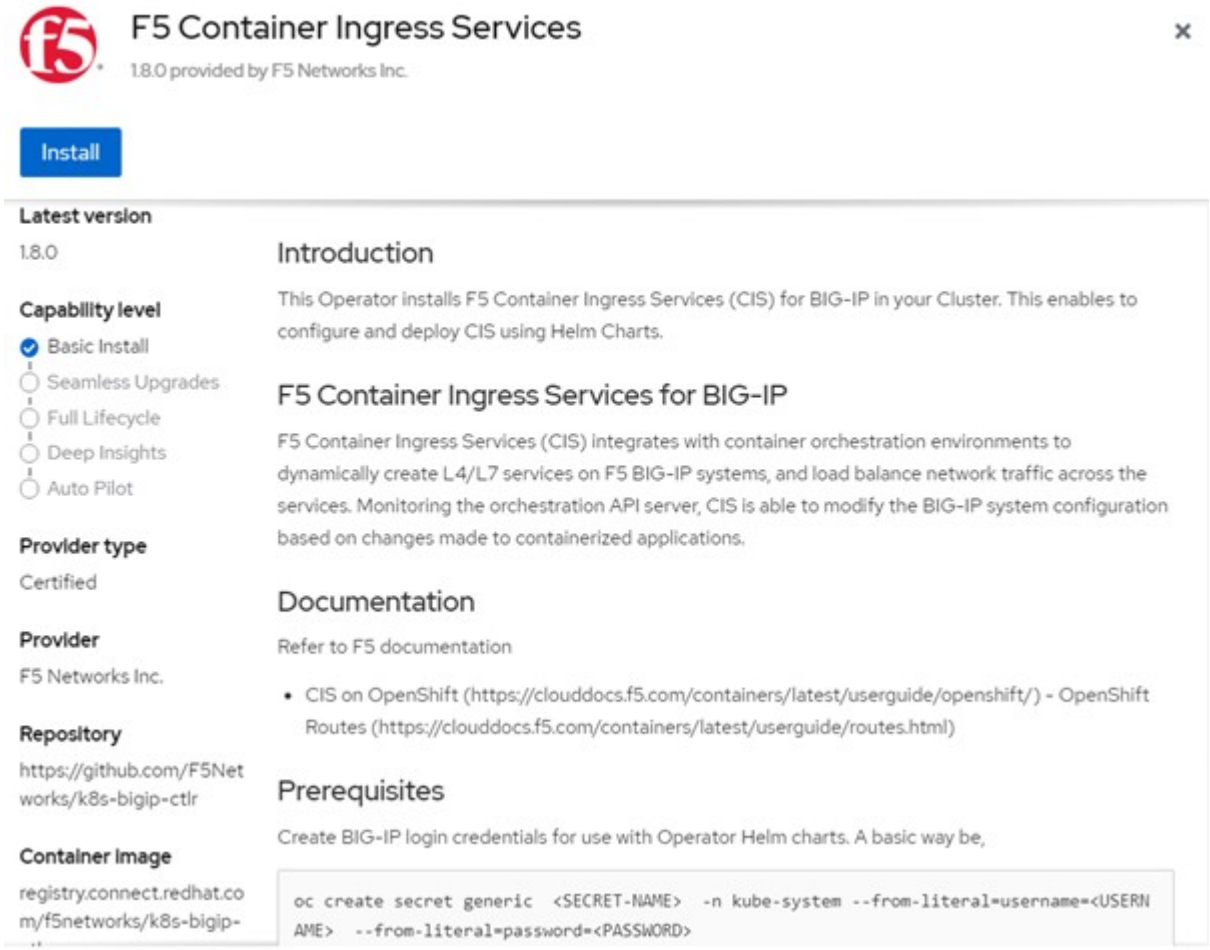
15. Installez les CRD F5 CIS.

```
[admin@rhel-7 ~]$ oc apply -f
https://raw.githubusercontent.com/F5Networks/k8s-bigip-
ctlr/master/docs/config_examples/crd/Install/customresourcedefinitions.y
ml

customresourcedefinition.apiextensions.k8s.io/virtualservers.cis.f5.com
created
customresourcedefinition.apiextensions.k8s.io/tlsprofiles.cis.f5.com
created
customresourcedefinition.apiextensions.k8s.io/transportservers.cis.f5.co
m created
customresourcedefinition.apiextensions.k8s.io/externaldnss.cis.f5.com
created
customresourcedefinition.apiextensions.k8s.io/ingresslinks.cis.f5.com
created
```

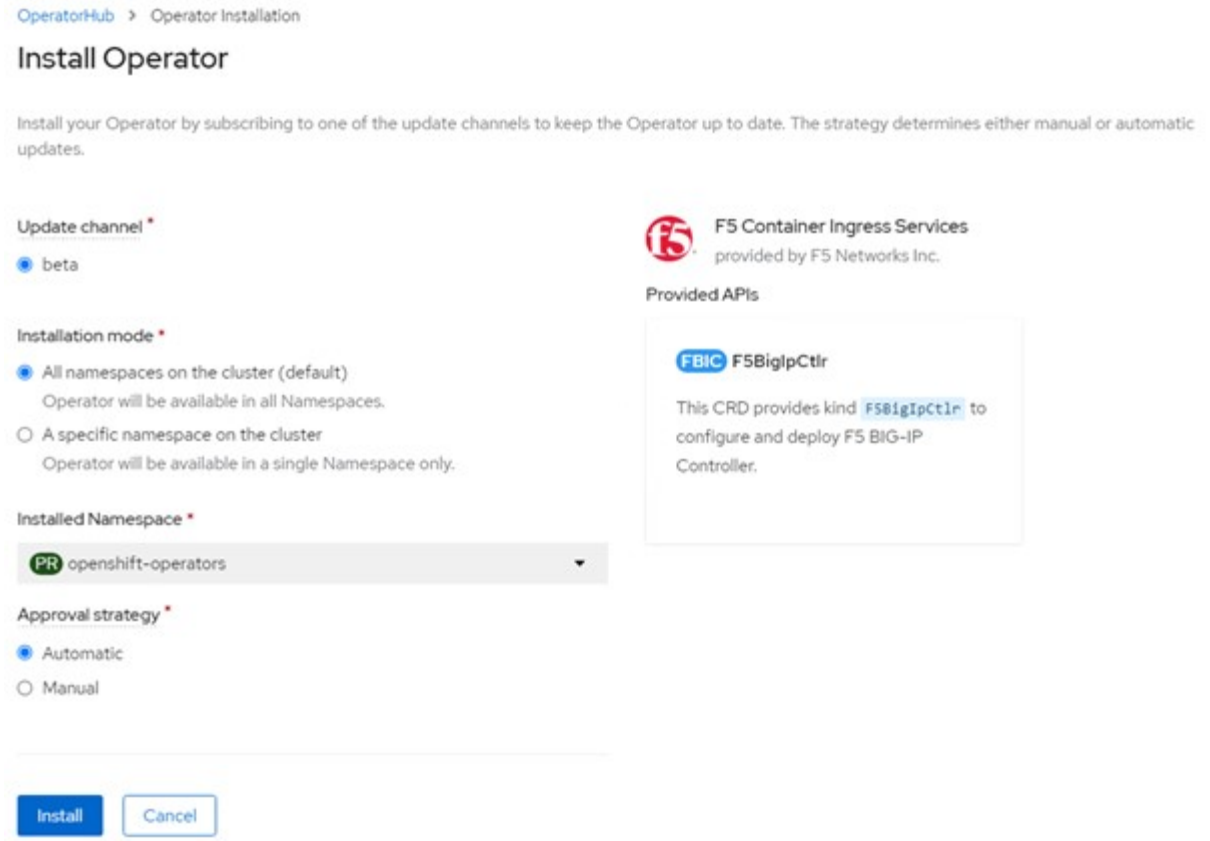
16. Accédez à Operators > OperatorHub, recherchez le mot-clé F5, puis cliquez sur la mosaïque F5 Container Ingress Service.



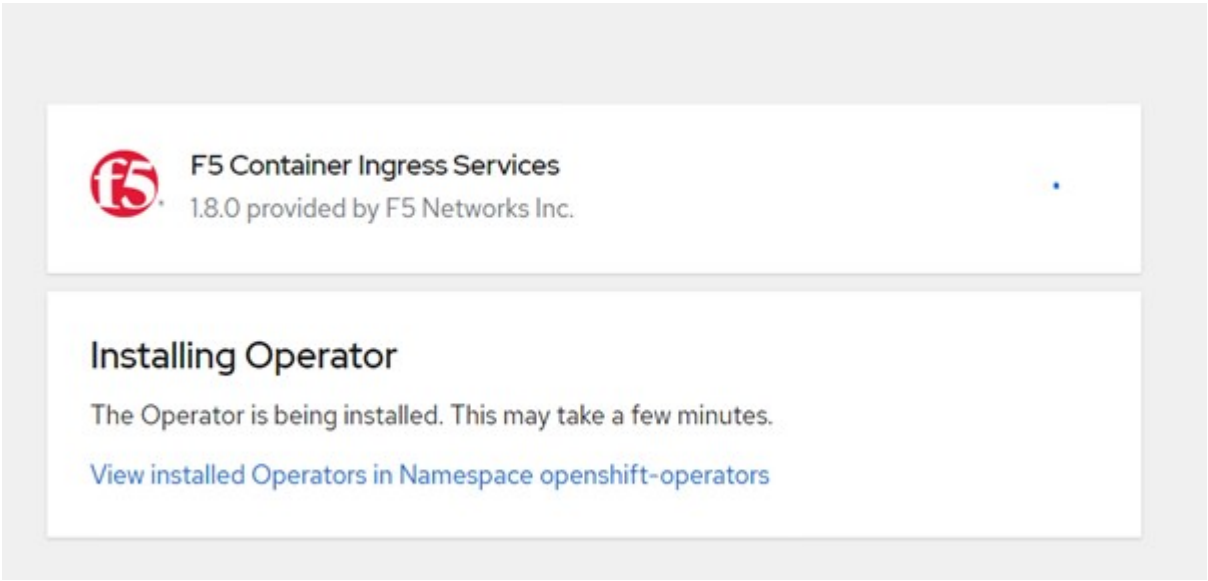
17. Lisez les informations de l'opérateur et cliquez sur installer.



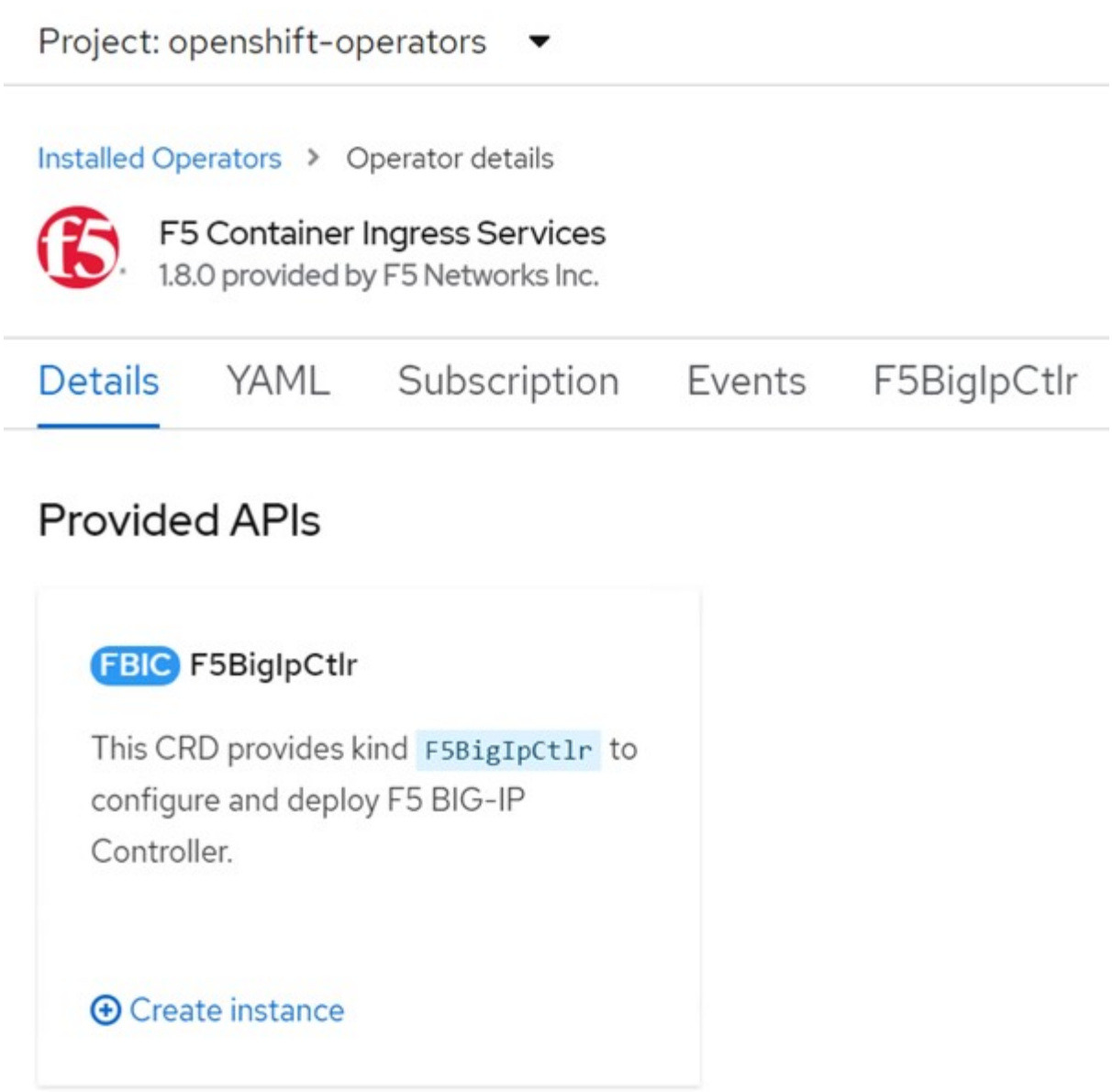
18. Sur l'écran de l'opérateur d'installation, conservez tous les paramètres par défaut, puis cliquez sur installer.



19. L'installation de l'opérateur prend un certain temps.



20. Une fois l'opérateur installé, le message installation réussie s'affiche.
21. Accédez à opérateurs > opérateurs installés, cliquez sur F5 Container Ingress Service, puis cliquez sur Créer une instance sous la mosaïque F5BigIpCtlr.

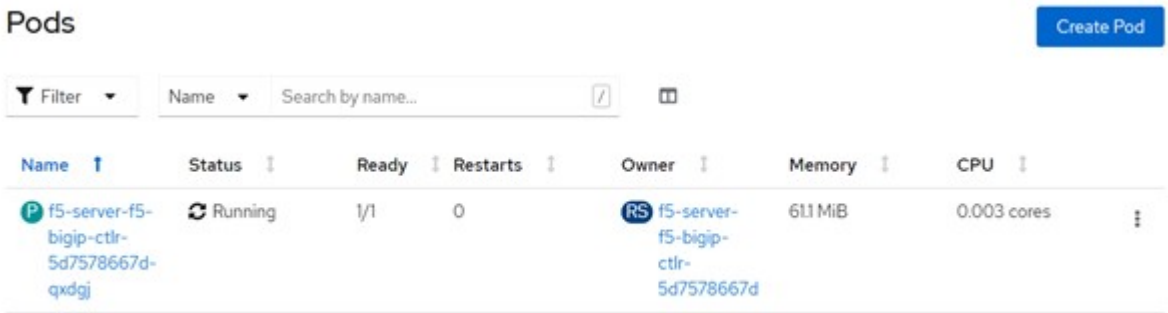


22. Cliquez sur vue YAML et collez le contenu suivant après la mise à jour des paramètres nécessaires.

> Mettre à jour les paramètres `bigip_partition`, ` openshift_sdn_name`, `bigip_url` et `bigip_login_secret` ci-dessous pour refléter les valeurs de votre configuration avant de copier le contenu.

```
apiVersion: cis.f5.com/v1
kind: F5BigIpCtlr
metadata:
  name: f5-server
  namespace: openshift-operators
spec:
  args:
    log_as3_response: true
    agent: as3
    log_level: DEBUG
    bigip_partition: ocp-vmw
    openshift_sdn_name: /Common/openshift_vxlan
    bigip_url: 10.61.181.19
    insecure: true
    pool-member-type: cluster
    custom_resource_mode: true
    as3_validation: true
    ipam: true
    manage_configmaps: true
  bigip_login_secret: bigip-login
  image:
    pullPolicy: Always
    repo: f5networks/cntr-ingress-svcs
    user: registry.connect.redhat.com
  namespace: kube-system
  rbac:
    create: true
  resources: {}
  serviceAccount:
    create: true
  version: latest
```

23. Après avoir collé ce contenu, cliquez sur Créer. Cela installe les modules CIS dans l'espace de noms du système kube.



Par défaut, Red Hat OpenShift permet d'exposer les services via des routes pour l'équilibrage de charge L7. Un routeur OpenShift intégré est chargé de la publicité et du traitement du trafic pour ces routes. Cependant, vous pouvez également configurer F5 CIS pour prendre en charge les routes via un système F5 BIG-IP externe, qui peut s'exécuter soit en tant que routeur auxiliaire, soit en remplacement du routeur OpenShift auto-hébergé. CIS crée un serveur virtuel dans le système BIG-IP qui sert de routeur pour les routes OpenShift, et BIG-IP gère la publicité et le routage du trafic. Pour plus d'informations sur les paramètres permettant d'activer cette fonctionnalité, reportez-vous à la documentation ci-dessous. Notez que ces paramètres sont définis pour la ressource OpenShift Deployment dans l'API apps/v1. Par conséquent, lors de l'utilisation de ces traits avec l'API F5BigIpCtlr ressource cis.f5.com/v1, remplacer les traits d'Union (-) par des traits de soulignement (_) pour les noms de paramètres.

24. Les arguments qui sont transmis à la création de ressources CIS sont notamment `ipam: true` et `custom_resource_mode: true`. Ces paramètres sont nécessaires pour activer l'intégration CIS avec un contrôleur IPAM. Vérifiez que le CIS a activé l'intégration IPAM en créant la ressource IP F5.

```
[admin@rhel-7 ~]$ oc get f5ipam -n kube-system

NAMESPACE   NAME                              AGE
kube-system   ipam.10.61.181.19.ocp-vmw       43s
```

25. Créez le compte de service, le rôle et la liaison en liaison rolerequises pour le contrôleur F5 IPAM. Créez un fichier YAML et collez le contenu suivant.

```
[admin@rhel-7 ~]$ vi f5-ipam-rbac.yaml

kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ipam-ctlr-clusterrole
rules:
  - apiGroups: ["fic.f5.com"]
    resources: ["ipams","ipams/status"]
    verbs: ["get", "list", "watch", "update", "patch"]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ipam-ctlr-clusterrole-binding
  namespace: kube-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: ipam-ctlr-clusterrole
subjects:
  - apiGroup: ""
    kind: ServiceAccount
    name: ipam-ctlr
    namespace: kube-system
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ipam-ctlr
  namespace: kube-system
```

26. Créez les ressources.

```
[admin@rhel-7 ~]$ oc create -f f5-ipam-rbac.yaml

clusterrole.rbac.authorization.k8s.io/ipam-ctlr-clusterrole created
clusterrolebinding.rbac.authorization.k8s.io/ipam-ctlr-clusterrole-
binding created
serviceaccount/ipam-ctlr created
```

27. Créez un fichier YAML et collez la définition de déploiement IPAM F5 indiquée ci-dessous.

Mettez à jour le paramètre de plage ip dans spec.template.spec.containers[0].args ci-dessous pour refléter les plages d'adresses IP et ipamLabels correspondant à votre configuration.

IpamLabels [`range1` et `range2` Dans l'exemple ci-dessous] sont nécessaires pour être annotés pour les services de type LoadBalancer pour le contrôleur IPAM afin de détecter et d'affecter une adresse IP à partir de la plage définie.

```
[admin@rhel-7 ~]$ vi f5-ipam-deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    name: f5-ipam-controller
  name: f5-ipam-controller
  namespace: kube-system
spec:
  replicas: 1
  selector:
    matchLabels:
      app: f5-ipam-controller
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: f5-ipam-controller
    spec:
      containers:
      - args:
        - --orchestration=openshift
        - --ip-range='{"range1":"10.63.172.242-10.63.172.249",
"range2":"10.63.170.111-10.63.170.129"}'
        - --log-level=DEBUG
        command:
        - /app/bin/f5-ipam-controller
        image: registry.connect.redhat.com/f5networks/f5-ipam-
controller:latest
        imagePullPolicy: IfNotPresent
        name: f5-ipam-controller
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      serviceAccount: ipam-ctlr
      serviceAccountName: ipam-ctlr
```

28. Créer le déploiement du contrôleur F5 IPAM.

```
[admin@rhel-7 ~]$ oc create -f f5-ipam-deployment.yaml

deployment/f5-ipam-controller created
```

29. Vérifiez que les modules de contrôleur F5 IPAM sont en cours d'exécution.

```
[admin@rhel-7 ~]$ oc get pods -n kube-system

NAME                                      READY   STATUS    RESTARTS
AGE
f5-ipam-controller-5986cff5bd-2bvn6       1/1     Running   0
30s
f5-server-f5-bigip-ctlr-5d7578667d-qxdgj  1/1     Running   0
14m
```

30. Créez le schéma F5 IPAM.

```
[admin@rhel-7 ~]$ oc create -f
https://raw.githubusercontent.com/F5Networks/f5-ipam-
controller/main/docs/_static/schemas/ipam_schema.yaml

customresourcedefinition.apiextensions.k8s.io/ipams.fic.f5.com
```

Vérification

1. Créez un service de type LoadBalancer

```
[admin@rhel-7 ~]$ vi example_svc.yaml

apiVersion: v1
kind: Service
metadata:
  annotations:
    cis.f5.com/ipamLabel: range1
  labels:
    app: f5-demo-test
  name: f5-demo-test
  namespace: default
spec:
  ports:
  - name: f5-demo-test
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: f5-demo-test
  sessionAffinity: None
  type: LoadBalancer
```

```
[admin@rhel-7 ~]$ oc create -f example_svc.yaml

service/f5-demo-test created
```

2. Vérifiez si le contrôleur IPAM lui attribue une adresse IP externe.

```
[admin@rhel-7 ~]$ oc get svc

NAME           TYPE           CLUSTER-IP       EXTERNAL-IP
PORT(S)        AGE
f5-demo-test   LoadBalancer   172.30.210.108   10.63.172.242
80:32605/TCP   27s
```

3. Créez un déploiement et utilisez le service LoadBalancer qui a été créé.

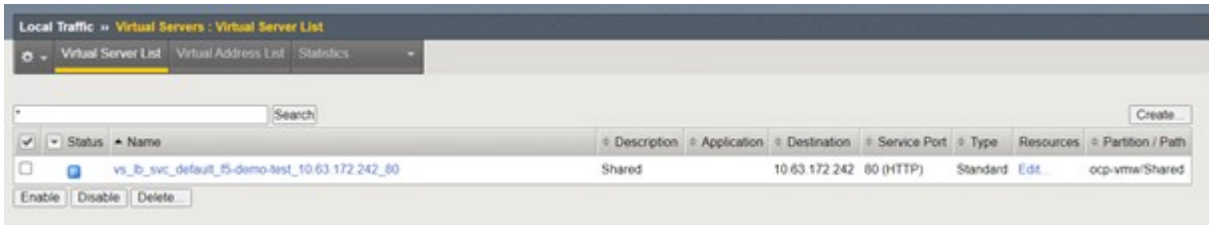
```
[admin@rhel-7 ~]$ vi example_deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: f5-demo-test
  name: f5-demo-test
spec:
  replicas: 2
  selector:
    matchLabels:
      app: f5-demo-test
  template:
    metadata:
      labels:
        app: f5-demo-test
    spec:
      containers:
      - env:
        - name: service_name
          value: f5-demo-test
        image: nginx
        imagePullPolicy: Always
        name: f5-demo-test
        ports:
        - containerPort: 80
          protocol: TCP
```

```
[admin@rhel-7 ~]$ oc create -f example_deployment.yaml

deployment/f5-demo-test created
```

4. Vérifiez que les modules sont en cours d'exécution.

```
[admin@rhel-7 ~]$ oc get pods

NAME                           READY   STATUS    RESTARTS   AGE
f5-demo-test-57c46f6f98-47wwp   1/1     Running   0          27s
f5-demo-test-57c46f6f98-cl2m8   1/1     Running   0          27s
```

5. Vérifiez si le serveur virtuel correspondant est créé dans LE système BIG-IP pour le service de type LoadBalancer dans OpenShift. Accédez à trafic local > serveurs virtuels > liste de serveurs virtuels.



Création de registres d'images privées

Pour la plupart des déploiements de Red Hat OpenShift, à l'aide d'un registre public comme "Quay.io" ou "DockerHub" répond à la plupart des besoins des clients. Cependant, il se peut qu'un client souhaite héberger ses propres images privées ou personnalisées.

Cette procédure décrit la création d'un registre d'images privées, sauvegardé par un volume persistant fourni par Astra Trident et NetApp ONTAP.

> Astra Control Center requiert un registre pour héberger les images dont les conteneurs Astra ont besoin. La section suivante décrit les étapes de configuration d'un registre privé sur un cluster Red Hat OpenShift et l'envoi des images requises pour prendre en charge l'installation d'Astra Control Center.

Création d'un registre d'images privé

1. Supprimez l'annotation par défaut de la classe de stockage par défaut actuelle et annoter la classe de stockage sauvegardée par Trident par défaut pour le cluster OpenShift.

```
[netapp-user@rhel7 ~]$ oc patch storageclass thin -p '{"metadata":
{"annotations": {"storageclass.kubernetes.io/is-default-class":
"false"}}}'
storageclass.storage.k8s.io/thin patched

[netapp-user@rhel7 ~]$ oc patch storageclass ocp-trident -p
'{"metadata": {"annotations": {"storageclass.kubernetes.io/is-default-
class": "true"}}}'
storageclass.storage.k8s.io/ocp-trident patched
```

2. Modifiez l'opérateur imageistry en saisissant les paramètres de stockage suivants dans le `spec` section.

```
[netapp-user@rhel7 ~]$ oc edit
configs.imageregistry.operator.openshift.io

storage:
  pvc:
    claim:
```

3. Entrez les paramètres suivants dans le `spec` Section permettant de créer une route OpenShift avec un nom d'hôte personnalisé. Enregistrer et quitter.

```
routes:
  - hostname: astra-registry.apps.ocp-vmw.cie.netapp.com
    name: netapp-astra-route
```

> La configuration de route ci-dessus est utilisée lorsque vous voulez un nom d'hôte personnalisé pour votre itinéraire. Si vous souhaitez qu'OpenShift crée une route avec un nom d'hôte par défaut, vous pouvez ajouter les paramètres suivants à l' `spec` section : `defaultRoute: true`.

Certificats TLS personnalisés

Lorsque vous utilisez un nom d'hôte personnalisé pour la route, il utilise par défaut la configuration TLS par défaut de l'opérateur OpenShift Ingress. Cependant, vous pouvez ajouter une configuration TLS personnalisée à la route. Pour ce faire, procédez comme suit.

a. Créez un secret avec les certificats TLS et la clé de la route.

```
[netapp-user@rhel7 ~]$ oc create secret tls astra-route-tls -n
openshift-image-registry –cert/home/admin/netapp-astra/tls.crt
--key=/home/admin/netapp-astra/tls.key
```

b. Modifiez l'opérateur imageistry et ajoutez les paramètres suivants à la `spec` section.

```
[netapp-user@rhel7 ~]$ oc edit
configs.imageregistry.operator.openshift.io

routes:
  - hostname: astra-registry.apps.ocp-vmw.cie.netapp.com
    name: netapp-astra-route
    secretName: astra-route-tls
```

4. Modifiez à nouveau l'opérateur imageistry et modifiez l'état de gestion de l'opérateur sur `Managed` état. Enregistrer et quitter.

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. Si toutes les conditions préalables sont remplies, les ESV, les pods et les services sont créés pour le

registre d’images privées. Dans quelques minutes, le registre devrait être mis en service.

```
[netapp-user@rhel7 ~]$oc get all -n openshift-image-registry

NAME                                                READY   STATUS
RESTARTS   AGE
pod/cluster-image-registry-operator-74f6d954b6-rb7zr   1/1     Running
3          90d
pod/image-pruner-1627257600-f5cpj                   0/1     Completed
0          2d9h
pod/image-pruner-1627344000-swqx9                   0/1     Completed
0          33h
pod/image-pruner-1627430400-rv5nt                   0/1     Completed
0          9h
pod/image-registry-6758b547f-6pnj8                  1/1     Running
0          76m
pod/node-ca-bwb5r                                   1/1     Running
0          90d
pod/node-ca-f8w54                                   1/1     Running
0          90d
pod/node-ca-gjx7h                                   1/1     Running
0          90d
pod/node-ca-lcx4k                                   1/1     Running
0          33d
pod/node-ca-v7zmx                                   1/1     Running
0          7d21h
pod/node-ca-xpppp                                   1/1     Running
0          89d

NAME                              TYPE        CLUSTER-IP       EXTERNAL-
IP   PORT(S)     AGE
service/image-registry            ClusterIP   172.30.196.167   <none>
5000/TCP    15h
service/image-registry-operator   ClusterIP   None             <none>
60000/TCP   90d

NAME                     DESIRED   CURRENT   READY   UP-TO-DATE
AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/node-ca   6         6         6       6            6
kubernetes.io/os=linux   90d

NAME                                           READY   UP-TO-DATE
AVAILABLE   AGE
deployment.apps/cluster-image-registry-operator   1/1     1            1
90d
deployment.apps/image-registry                    1/1     1            1
```

```
15h

NAME                                                              DESIRED
CURRENT   READY   AGE
replicaset.apps/cluster-image-registry-operator-74f6d954b6   1         1
1       90d
replicaset.apps/image-registry-6758b547f                     1         1
1       76m
replicaset.apps/image-registry-78bfbd7f59                    0         0
0       15h
replicaset.apps/image-registry-7fcc8d6cc8                    0         0
0       80m
replicaset.apps/image-registry-864f88f5b                     0         0
0       15h
replicaset.apps/image-registry-cb47fffb                      0         0
0       10h

NAME                                  COMPLETIONS   DURATION   AGE
job.batch/image-pruner-1627257600   1/1           10s        2d9h
job.batch/image-pruner-1627344000   1/1           6s         33h
job.batch/image-pruner-1627430400   1/1           5s         9h

NAME                          SCHEDULE    SUSPEND   ACTIVE   LAST
SCHEDULE   AGE
cronjob.batch/image-pruner   0 0 * * *   False     0        9h
90d

NAME                                       HOST/PORT
PATH   SERVICES         PORT    TERMINATION   WILDCARD
route.route.openshift.io/public-routes   astra-registry.apps.ocp-
vmw.cie.netapp.com          image-registry   <all>   reencrypt     None
```

6. Si vous utilisez les certificats TLS par défaut pour la route de registre OpenShift de l'opérateur d'entrée, vous pouvez récupérer les certificats TLS à l'aide de la commande suivante.

```
[netapp-user@rhel7 ~]$ oc extract secret/router-ca --keys=tls.crt -n
openshift-ingress-operator
```

7. Pour permettre aux nœuds OpenShift d'accéder aux images et de les extraire du registre, ajoutez les certificats au client docker sur les nœuds OpenShift. Créez une configuration dans le `openshift-config` Espace de noms à l'aide des certificats TLS et le patch dans la configuration d'image du cluster pour que le certificat soit fiable.

```
[netapp-user@rhel7 ~]$ oc create configmap astra-ca -n openshift-config
--from-file=astra-registry.apps.ocp-vmw.cie.netapp.com=tls.crt

[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster
--patch '{"spec":{"additionalTrustedCA":{"name":"astra-ca"}}}'
--type=merge
```

8. Le registre interne OpenShift est contrôlé par une authentification. Tous les utilisateurs OpenShift peuvent accéder au registre OpenShift, mais les opérations que l'utilisateur connecté peut exécuter dépendent des autorisations des utilisateurs.
   a. Pour permettre à un utilisateur ou à un groupe d'utilisateurs d'extraire des images du registre, le rôle du visualiseur de registre doit être affecté à l'utilisateur.

```
[netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-viewer
ocp-user

[netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-viewer
ocp-user-group
```

   b. Pour permettre à un utilisateur ou à un groupe d'utilisateurs d'écrire ou de diffuser des images, le rôle de l'éditeur de registre doit être affecté.

```
[netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-editor
ocp-user

[netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-editor
ocp-user-group
```

9. Pour que les nœuds OpenShift puissent accéder au registre et envoyer ou extraire les images, vous devez configurer un secret Pull.

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-registry-
credentials --docker-server=astra-registry.apps.ocp-vmw.cie.netapp.com
--docker-username=ocp-user --docker-password=password
```

10. Ce secret Pull peut ensuite être corrigé aux comptes de service ou être référencé dans la définition de pod correspondante.
    a. Pour le corriger aux comptes de service, exécutez la commande suivante.

```
[netapp-user@rhel7 ~]$ oc secrets link <service_account_name> astra-
registry-credentials --for=pull
```

b. Pour référencer le secret Pull dans la définition du pod, ajoutez le paramètre suivant à l' `spec` section.

```
imagePullSecrets:
  - name: astra-registry-credentials
```

11. Pour pousser ou extraire une image des postes de travail en dehors du nœud OpenShift, procédez comme suit.

a. Ajoutez les certificats TLS au client docker.

```
[netapp-user@rhel7 ~]$ sudo mkdir /etc/docker/certs.d/astra-
registry.apps.ocp-vmw.cie.netapp.com

[netapp-user@rhel7 ~]$ sudo cp /path/to/tls.crt
/etc/docker/certs.d/astra-registry.apps.ocp-vmw.cie.netapp.com
```

b. Connectez-vous à OpenShift à l'aide de la commande oc login.

```
[netapp-user@rhel7 ~]$ oc login --token=sha256~D49SpB_lesSrJYwrM0LIO
-VRcjWHu0a27vKa0 --server=https://api.ocp-vmw.cie.netapp.com:6443
```

c. Connectez-vous au registre à l'aide des informations d'identification de l'utilisateur OpenShift avec la commande podman/docker.

podman

```
[netapp-user@rhel7 ~]$ podman login astra-registry.apps.ocp-
vmw.cie.netapp.com -u kubeadmin -p $(oc whoami -t) --tls
-verify=false
```

+ REMARQUE : si vous utilisez `kubeadmin` l'utilisateur doit se connecter au registre privé, puis utiliser un jeton au lieu du mot de passe.

docker

```
[netapp-user@rhel7 ~]$ docker login astra-registry.apps.ocp-
vmw.cie.netapp.com -u kubeadmin -p $(oc whoami -t)
```

+ REMARQUE : si vous utilisez `kubeadmin` l'utilisateur doit se connecter au registre privé, puis utiliser un jeton au lieu du mot de passe.

d. Pousser ou extraire les images.

podman

```
[netapp-user@rhel7 ~]$ podman push astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
[netapp-user@rhel7 ~]$ podman pull astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
```

docker

```
[netapp-user@rhel7 ~]$ docker push astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
[netapp-user@rhel7 ~]$ docker pull astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
```