Erweiterte Konfigurationsoptionen

NetApp Solutions

NetApp
December 19, 2024

Inhalt

Erweiterte Konfigurationsoptionen

Informationen Zu Load Balancer Options

Load Balancer-Optionen: Red hat OpenShift mit NetApp

In den meisten Fällen stellt Red hat OpenShift Anwendungen für die Außenwelt über Routen zur Verfügung. Ein Service wird durch die Angabe eines extern zugänglichen Host-Namens zugänglich gemacht. Die definierte Route und die vom Dienst identifizierten Endpunkte können von einem OpenShift-Router genutzt werden, um diese benannte Verbindung externen Clients bereitzustellen.

In einigen Fällen müssen jedoch Applikationen zum Einsatz kommen und eine Konfiguration von angepassten Lastausgleich erforderlich sein, um die entsprechenden Services bereitzustellen. Ein Beispiel hierfür ist das NetApp Astra Control Center. Um diesem Bedarf gerecht zu werden, haben wir eine Reihe von benutzerdefinierten Load Balancer-Optionen evaluiert. Die Installation und Konfiguration der Lösung wird in diesem Abschnitt beschrieben.

Auf den folgenden Seiten sind zusätzliche Informationen zu den in Red hat OpenShift mit NetApp validierten Load Balancer-Optionen verfügbar:

- "MetalLB"
- "F5 BIG-IP"

Installation von MetalLB Load Balancer: Red hat OpenShift mit NetApp

Auf dieser Seite werden die Installations- und Konfigurationsanweisungen für den MetalLB Load Balancer aufgeführt.

MetalLB ist ein selbst gehosteter Network Load Balancer, der auf Ihrem OpenShift-Cluster installiert ist und das die Erstellung von OpenShift-Diensten vom Typ Load Balancer in Clustern ermöglicht, die nicht auf einem Cloud-Provider ausgeführt werden. Die zwei Hauptmerkmale der MetalLB, die zur Unterstützung der Load Balancer-Dienste zusammenarbeiten, sind Adresszuweisung und externe Ankündigung.

MetalLB-Konfigurationsoptionen

Basierend darauf, wie MetalLB die IP-Adresse ankündigt, die den Load Balancer-Diensten außerhalb des OpenShift-Clusters zugewiesen ist, arbeitet es in zwei Modi:

- **Layer 2-Modus.** in diesem Modus übernimmt ein Knoten im OpenShift-Cluster die Verantwortung für den Service und antwortet auf ARP-Anfragen, damit er außerhalb des OpenShift-Clusters erreichbar ist. Da nur der Node die IP bereitstellt, kommt es zu einem Bandbreitenengpass und zu langsamen Failover-Einschränkungen. Weitere Informationen finden Sie in der Dokumentation "Hier".
- **BGP-Modus.** in diesem Modus etablieren alle Knoten im OpenShift-Cluster BGP-Peering-Sitzungen mit einem Router und werben die Routen an die Service-IPs weiter. Voraussetzung dafür ist die Integration von MetalLB in einen Router in dieses Netzwerk. Aufgrund des Hashing-Mechanismus in BGP hat es bestimmte Einschränkungen, wenn IP-zu-Node-Zuordnung für einen Service geändert wird. Weitere Informationen finden Sie in der Dokumentation "Hier".

Im Rahmen dieses Dokuments konfigurieren wir MetalLB im Layer-2-Modus.

Installieren des MetalLB Load Balancer

1. Laden Sie die MetalLB-Ressourcen herunter.

```
[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. Datei bearbeiten `metallb.yaml` Und entfernen `spec.template.spec.securityContext` Von der Controller-Bereitstellung und dem Lautsprecher DemonSet.

   Zu löschende Zeilen:

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

3. Erstellen Sie die `metallb-system` Namespace.

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

4. Erstellen Sie den MetalLB CR.

```
[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. Bevor Sie den MetalLB-Lautsprecher konfigurieren, gewähren Sie dem Lautsprecher DemonSet erhöhte Berechtigungen, damit er die für den Lastausgleich erforderliche Netzwerkkonfiguration ausführen kann.

```
[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. Konfigurieren Sie MetalLB, indem Sie eine erstellen `ConfigMap` Im `metallb-system` Namespace.

```
[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. Wenn nun Load Balancer Dienste erstellt werden, weist MetalLB den Diensten eine externe IP zu und gibt die IP-Adresse durch Antwort auf ARP-Anfragen an.

> Wenn Sie MetalLB im BGP-Modus konfigurieren möchten, überspringen Sie Schritt 6 oben und befolgen Sie das Verfahren in der MetalLB-Dokumentation "Hier".

Installation von F5 BIG-IP Load Balancer

F5 BIG-IP ist ein Application Delivery Controller (ADC), der ein breites Spektrum erweiterter Traffic Management und Sicherheitsservices wie L4-L7 Load Balancing, SSL/TLS-Entlastung, DNS, Firewall und vieles mehr in Produktionsqualität bietet. Diese Services sorgen für eine signifikante Steigerung der Verfügbarkeit, Sicherheit und Performance der Applikationen.

F5 BIG-IP kann auf verschiedene Arten implementiert und genutzt werden: Auf dedizierter Hardware, in der Cloud oder als virtuelle Appliance vor Ort. In der Dokumentation finden Sie Informationen zur Nutzung und Implementierung von F5 BIG-IP nach Bedarf.

Für eine effiziente Integration von F5 BIG-IP-Diensten in Red hat OpenShift bietet F5 den BIG-IP Container Ingress Service (CIS) an. CIS wird als Controller-Pod installiert, der die OpenShift API für bestimmte Custom Resource Definitions (CRDs) überwacht und die F5 BIG-IP-Systemkonfiguration verwaltet. F5 BIG-IP CIS kann für die Steuerung von Servicetypen für Loadbalancer und Routen in OpenShift konfiguriert werden.

Für die automatische IP-Adresszuweisung zur Wartung des Typs loadbalancer können Sie außerdem den F5 IPAM-Controller nutzen. Der F5 IPAM-Controller wird als Controller-Pod installiert, der OpenShift API für Load Balancer-Dienste mit einer ipamLabel-Anmerkung überwacht, um die IP-Adresse aus einem vorkonfigurierten Pool zuzuweisen.

Auf dieser Seite sind die Installations- und Konfigurationsanweisungen für die F5 BIG-IP CIS- und IPAM-Controller aufgeführt. Voraussetzung ist, dass ein F5 BIG-IP-System implementiert und lizenziert werden

muss. Es muss auch für SDN-Dienste lizenziert sein, die standardmäßig in DER BIG-IP VE-Basislizenz enthalten sind.

F5 BIG-IP kann im Standalone- oder Cluster-Modus implementiert werden. Im Rahmen dieser Validierung wurde F5 BIG-IP im Standalone-Modus implementiert, jedoch wird es für Produktionszwecke bevorzugt, ein Cluster VON BIG-IPs zu verwenden, um Single Point of Failure zu vermeiden.

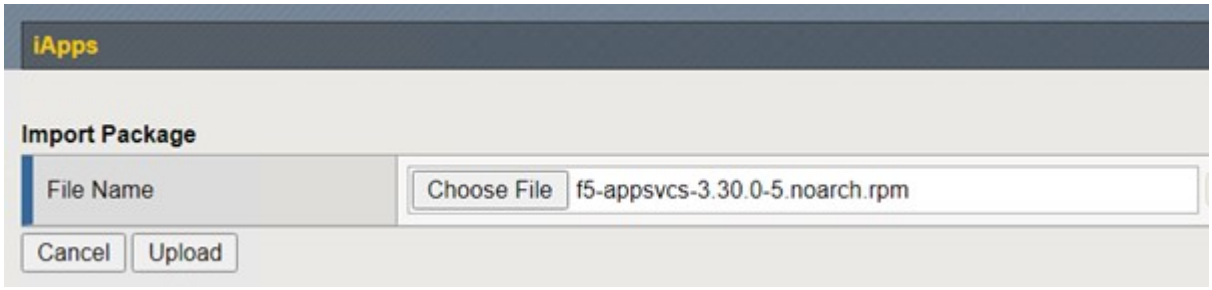
Ein F5 BIG-IP System kann auf dedizierter Hardware, in der Cloud oder als virtuelle Appliance on-Premises mit Versionen über 12.x bereitgestellt werden, damit es mit F5 CIS integriert werden kann. Für dieses Dokument wurde das F5 BIG-IP System als virtuelle Appliance validiert, beispielsweise mit DER BIG-IP VE Edition.

Validierte Versionen

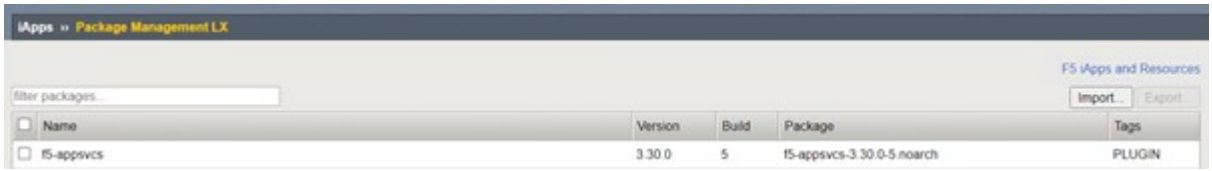
Technologie	Softwareversion
Red hat OpenShift	4.6 EUS, 4.7
F5 BIG-IP VE EDITION	16.1.0
F5 Container Ingress Service	2.5.1
F5 IPAM Controller	0.1.4
F5 AS3	3.30.0

Installation

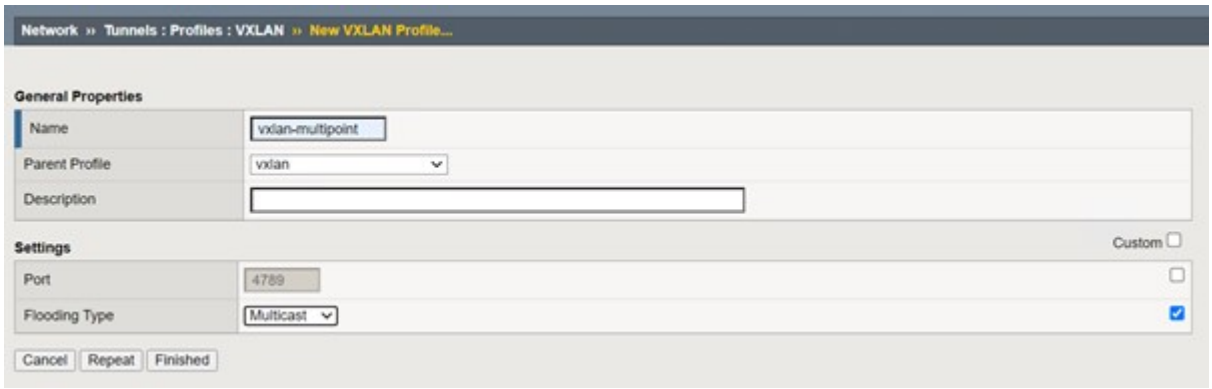
1. Installieren Sie die Erweiterung F5 Application Services 3, damit BIG-IP-Systeme Konfigurationen in JSON anstelle von Imperativ-Befehlen akzeptieren können. Gehen Sie zu "F5 AS3 GitHub-Repository", Und laden Sie die neueste RPM-Datei.
2. Melden Sie sich bei F5 BIG-IP-System an, navigieren Sie zu iApps > Package Management LX, und klicken Sie auf Importieren.
3. Klicken Sie auf Datei auswählen und wählen Sie die heruntergeladene AS3 RPM-Datei aus, klicken Sie auf OK und anschließend auf Hochladen.



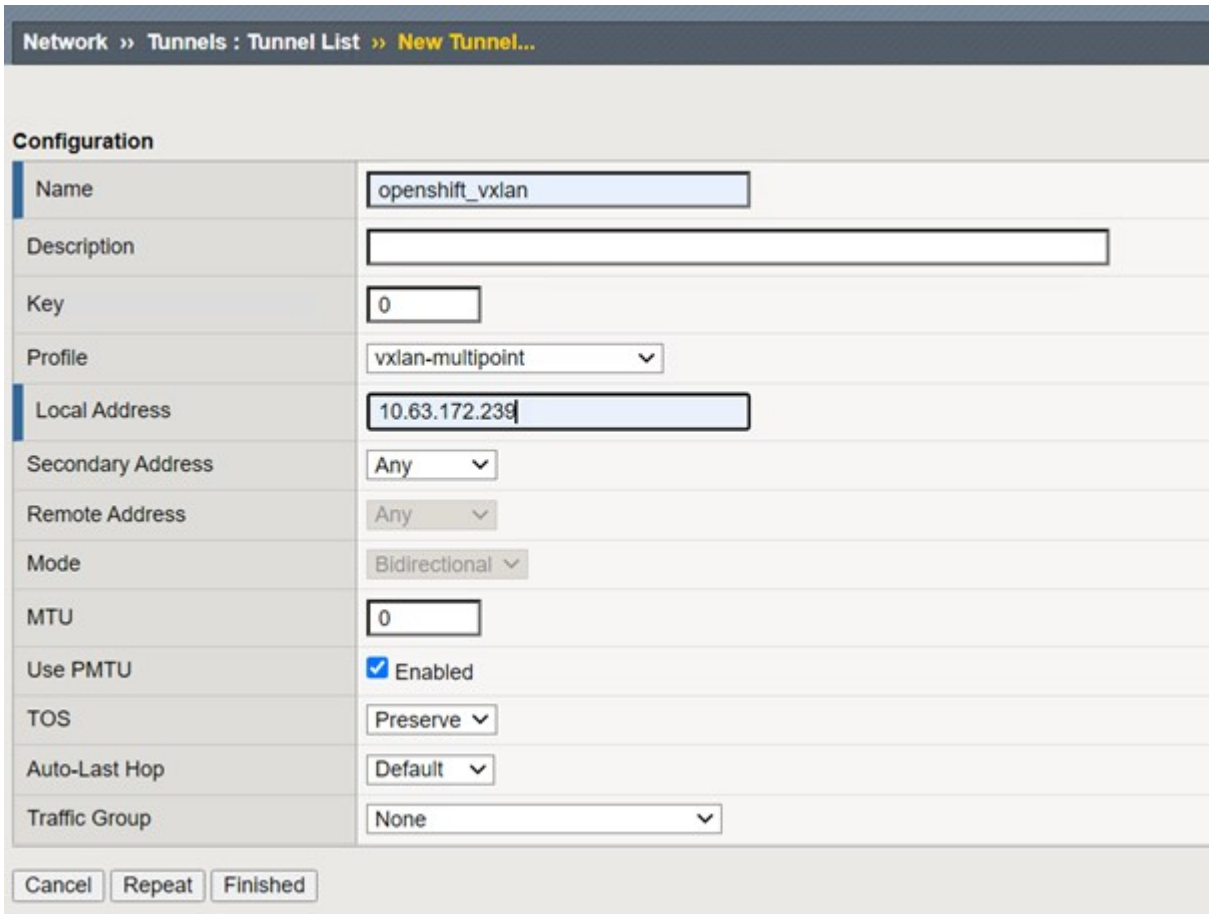
4. Vergewissern Sie sich, dass die AS3-Erweiterung erfolgreich installiert wurde.



5. Konfigurieren Sie dann die Ressourcen, die für die Kommunikation zwischen OpenShift- und BIG-IP-Systemen benötigt werden. Erstellen Sie zunächst einen Tunnel zwischen OpenShift und DEM BIG-IP-Server, indem Sie eine VXLAN-Tunnelschnittstelle auf dem BIG-IP-System für OpenShift SDN erstellen. Navigieren Sie zu Netzwerk > Tunnel > Profile, klicken Sie auf Erstellen, und legen Sie das übergeordnete Profil auf vxlan und den Hochwassertyp auf Multicast fest. Geben Sie einen Namen für das Profil ein und klicken Sie auf Fertig.



6. Navigieren Sie zu Netzwerk > Tunnel > Tunnelliste, klicken Sie auf Erstellen, und geben Sie den Namen und die lokale IP-Adresse für den Tunnel ein. Wählen Sie das Tunnelprofil aus, das im vorherigen Schritt erstellt wurde, und klicken Sie auf Fertig.



7. Melden Sie sich beim Red hat OpenShift-Cluster mit Clusteradministratorrechten an.
8. Erstellen Sie auf OpenShift ein Hostsubnetz für den F5 BIG-IP-Server, der das Subnetz vom OpenShift-Cluster auf den F5 BIG-IP-Server erweitert. Laden Sie die YAML-Hostsubnet-Definition herunter.

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

9. Bearbeiten Sie die Hostsubnet-Datei und fügen Sie die BIG-IP-VTEP-IP (VXLAN-Tunnel)-IP für das SDN OpenShift hinzu.

```
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
```

Ändern Sie HostIP und andere Details, je nach Ihrer Umgebung.

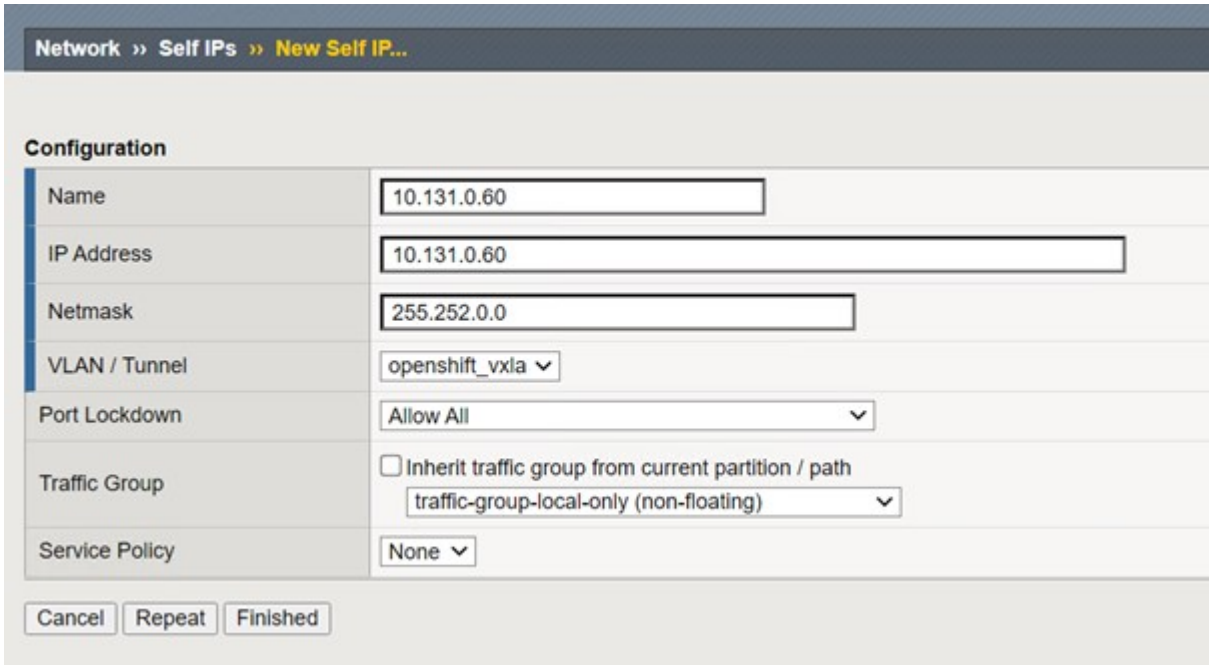
10. Hostsubnet-Ressource erstellen.

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

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

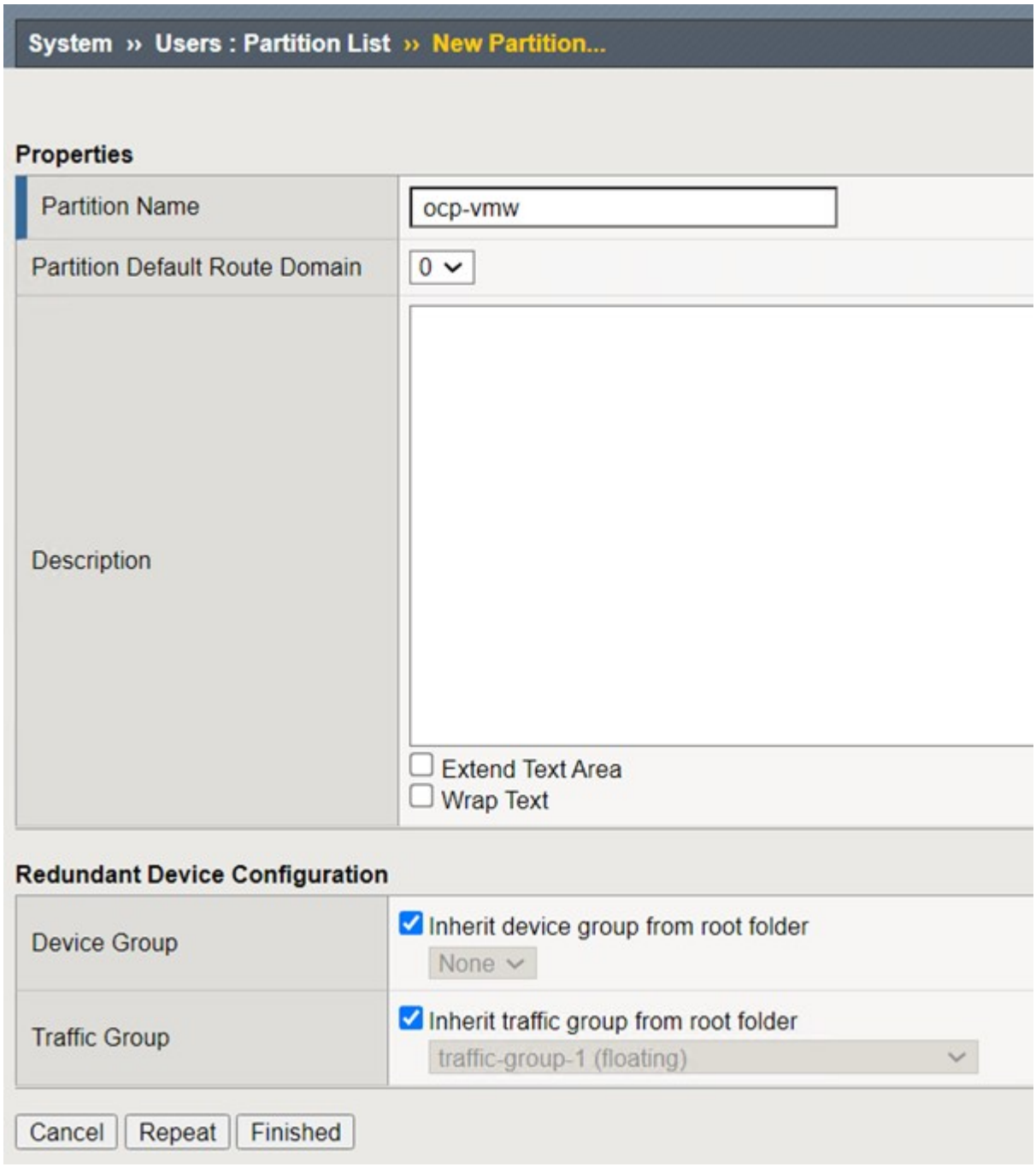
11. Abrufen des Cluster-IP-Subnetzes für das für den F5 BIG-IP-Server erstellte Host-Subnetz.

```
[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. Erstellen Sie auf OpenShift VXLAN mit einer IP im Host-Subnetz-Bereich von OpenShift eine eigene IP-Adresse, die dem F5 BIG-IP-Server entspricht. Melden Sie sich beim F5 BIG-IP-System an, navigieren Sie zu Netzwerk > selbst-IPs, und klicken Sie auf Erstellen. Geben Sie eine IP aus dem Cluster-IP-Subnetz ein, das für das F5 BIG-IP Host-Subnetz erstellt wurde, wählen Sie den VXLAN-Tunnel aus, und geben Sie die anderen Details ein. Klicken Sie anschließend auf Fertig.



13. Erstellen Sie eine Partition im F5 BIG-IP-System, die mit CIS konfiguriert und verwendet werden soll. Navigieren Sie zu System > Users > Partitionsliste, klicken Sie auf Create, und geben Sie die Details ein. Klicken Sie anschließend auf Fertig.



F5 empfiehlt, auf der von CIS verwalteten Partition keine manuelle Konfiguration durchzuführen.

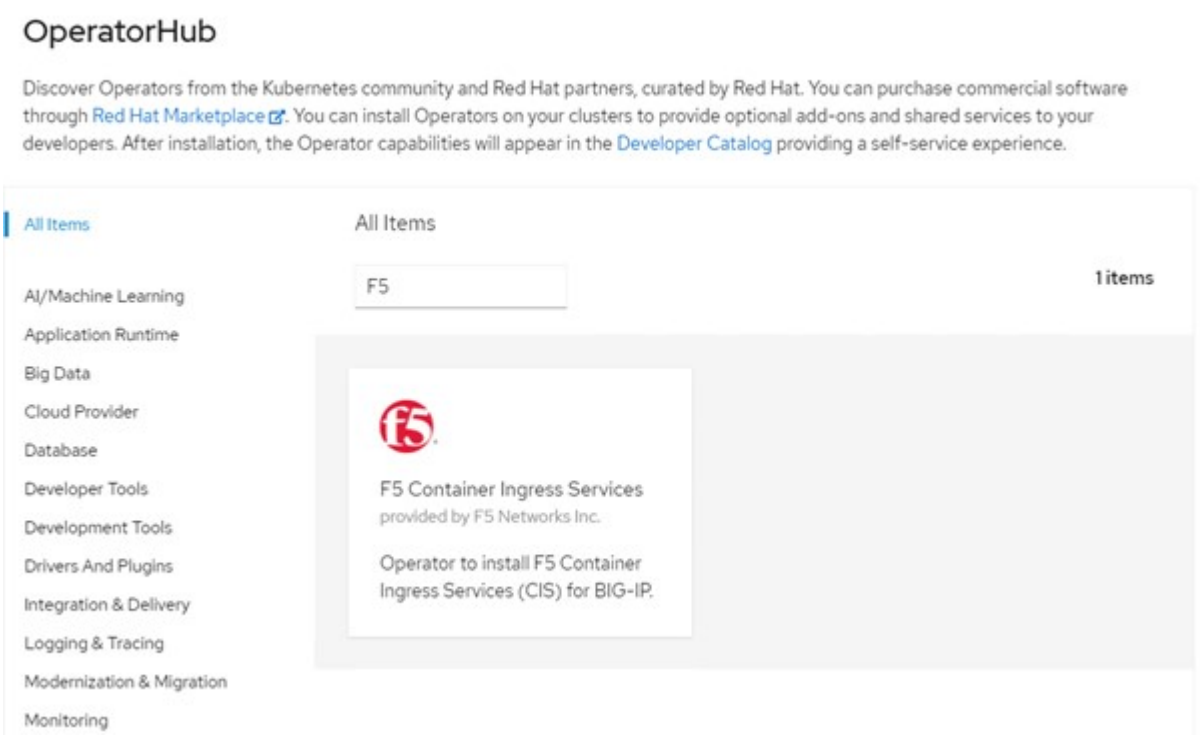
14. Installieren Sie die F5 BIG-IP CIS mit dem Operator von OperatorHub. Melden Sie sich mit Clusteradministratorrechten beim Red hat OpenShift-Cluster an und erstellen Sie mit den Anmeldedaten des F5 BIG-IP-Systems ein Geheimnis, das Voraussetzung für den Operator ist.

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

secret/bigip-login created
```

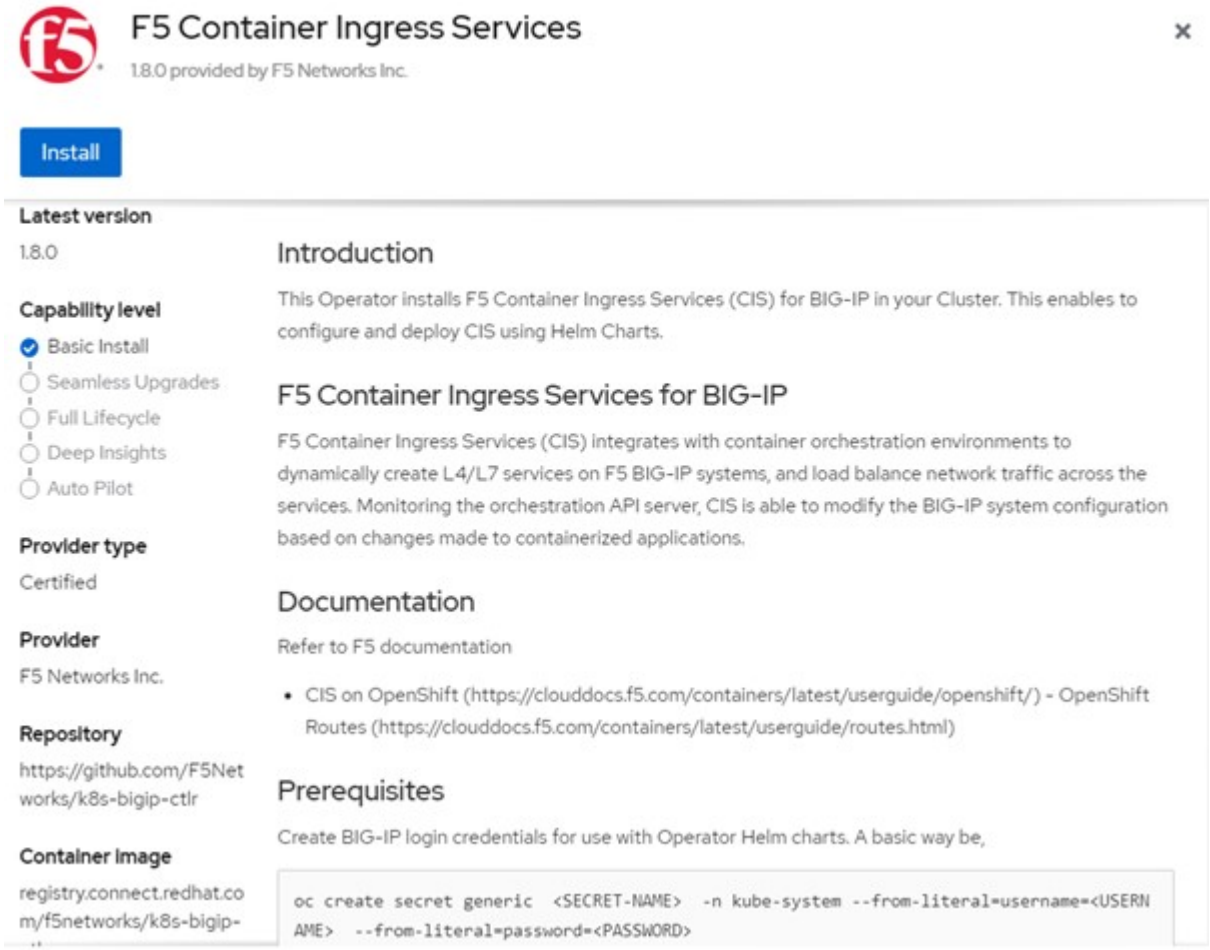
15. Installieren Sie die F5 CIS CRDs.

```
[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
```

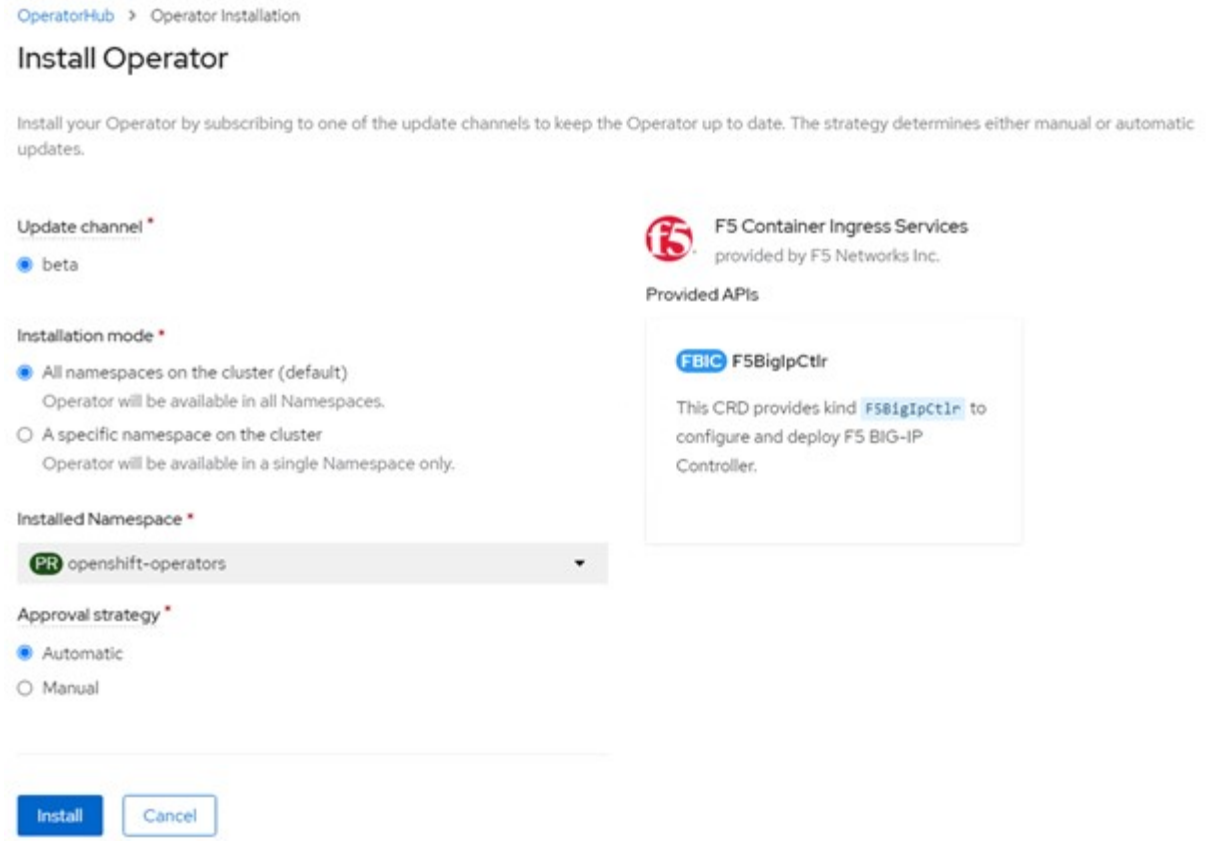
16. Navigieren Sie zu Operators > OperatorHub, suchen Sie nach dem Schlüsselwort F5 und klicken Sie auf die Kachel F5 Container Ingress Service.



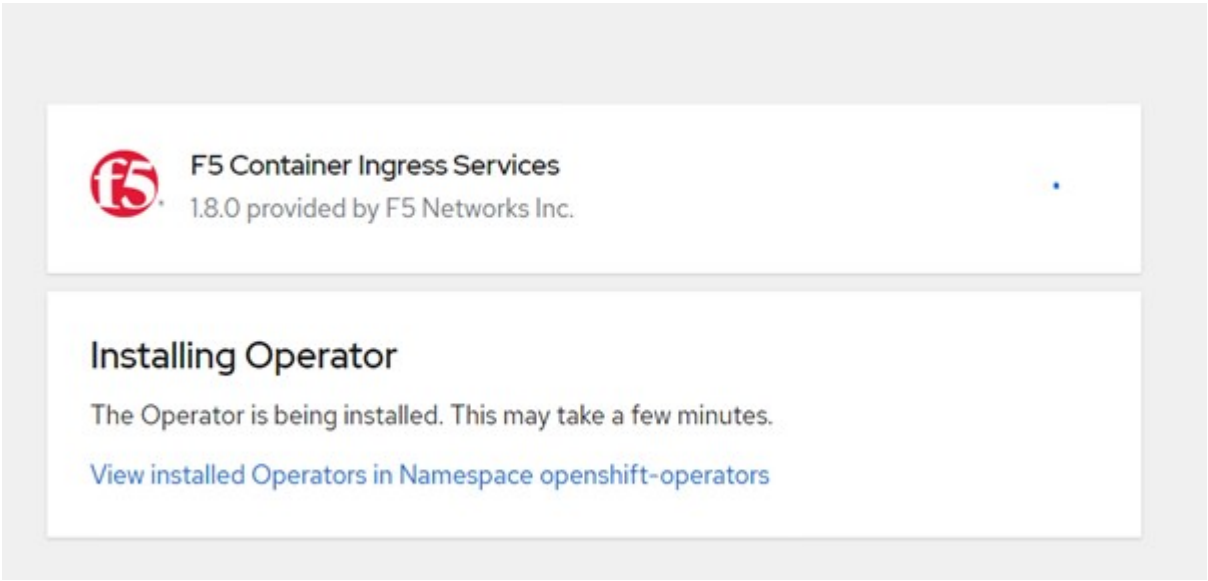
17. Lesen Sie die Bedienerinformationen, und klicken Sie auf Installieren.



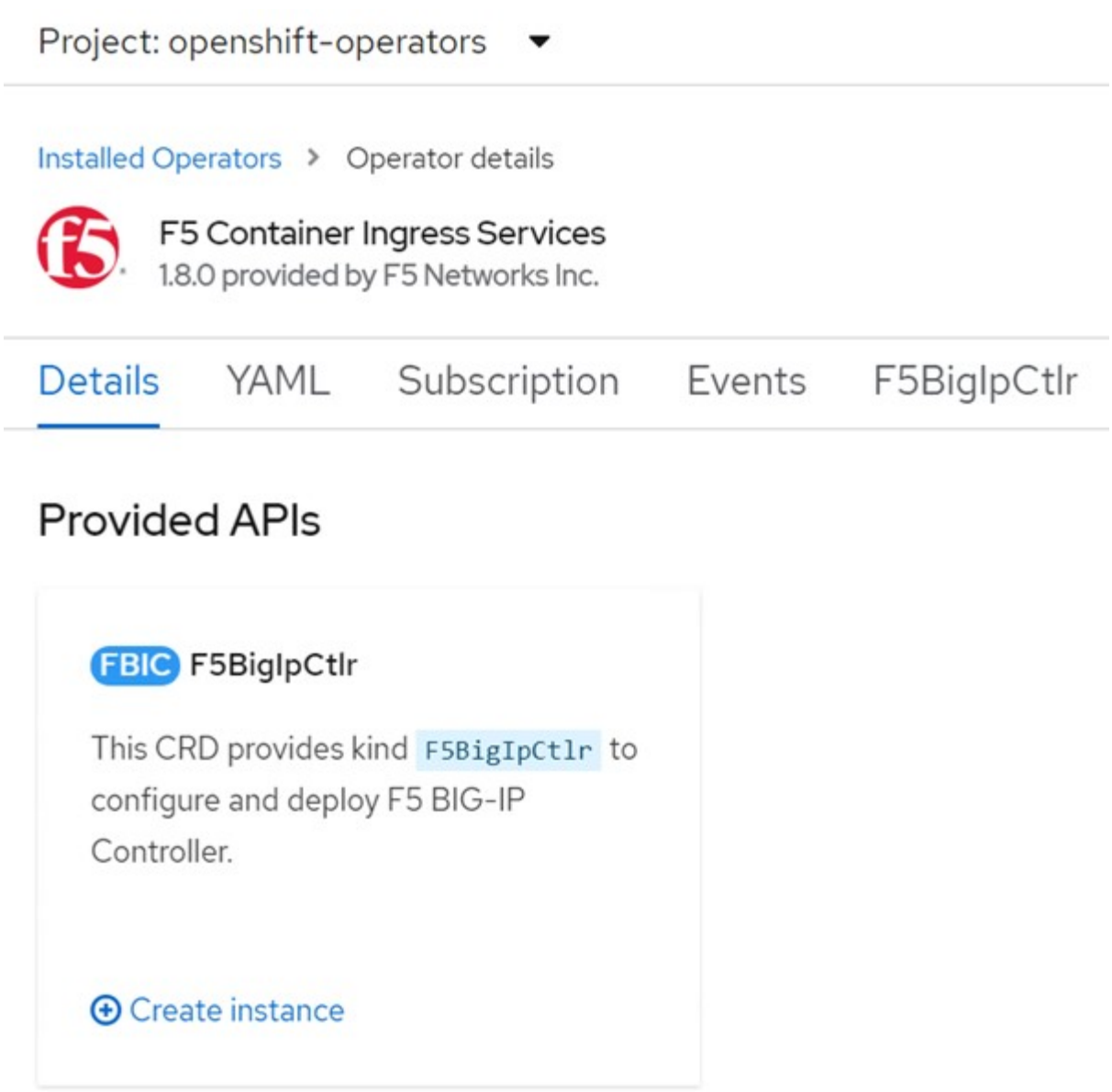
18. Lassen Sie auf dem Bildschirm Install Operator alle Standardparameter stehen, und klicken Sie auf Install.



19. Es dauert eine Weile, bis der Bediener installiert wird.



20. Nach der Installation des Bedieners wird die Meldung Installation erfolgreich angezeigt.
21. Navigieren Sie zu Operatoren > Installed Operators, klicken Sie auf F5 Container Ingress Service und klicken Sie dann unter der Kachel F5BigIpCtlr auf Create Instance.

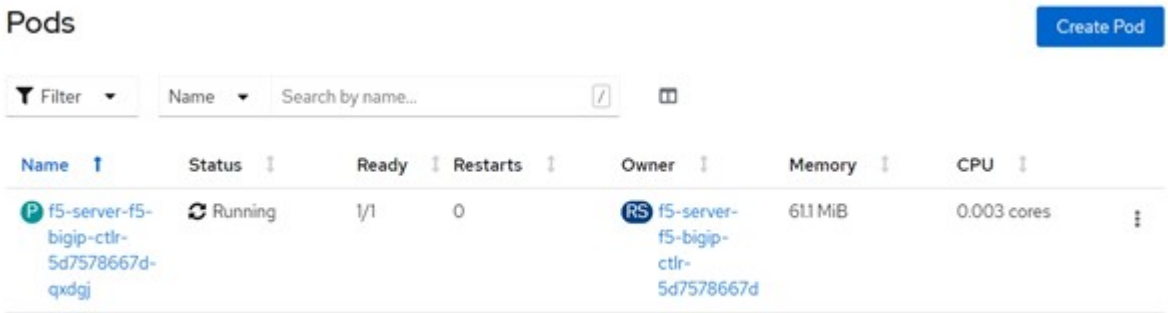


22. Klicken Sie auf YAML View und fügen Sie den folgenden Inhalt ein, nachdem Sie die erforderlichen Parameter aktualisiert haben.

Aktualisieren Sie die Parameter `bigip_partition`, ` openshift_sdn_Name`, `bigip_url` Und `bigip_login_secret` Geben Sie unten die Werte für Ihr Setup vor dem Kopieren des Inhalts an.

```
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. Klicken Sie nach dem Einfügen dieses Inhalts auf Erstellen. Damit werden die CIS-Pods im Namespace des kube-Systems installiert.



Red hat OpenShift bietet standardmäßig eine Möglichkeit, die Dienste über Routen für L7-Lastenausgleich zur Verfügung zu stellen. Ein eingebauter OpenShift-Router ist für die Werbung und den Umgang mit dem Verkehr auf diesen Routen verantwortlich. Sie können die F5 CIS jedoch auch so konfigurieren, dass sie die Routen über ein externes F5 BIG-IP-System unterstützen, das entweder als Hilfrouter oder als Ersatz für den selbst gehosteten OpenShift-Router ausgeführt werden kann. CIS erstellt einen virtuellen Server im BIG-IP-System, der als Router für die OpenShift-Routen fungiert, und BIG-IP übernimmt das Werbe- und Traffic-Routing. Informationen zu Parametern, die diese Funktion aktivieren, finden Sie in der Dokumentation hier. Beachten Sie, dass diese Parameter für die OpenShift Deployment-Ressource in der Apps/v1-API definiert sind. Wenn Sie diese also mit der F5BigIpCtlr Resource cis.f5.com/v1 API verwenden, ersetzen Sie die Bindestriche (-) durch Unterstriche (_) für die Parameternamen.

24. Die Argumente, die an die Erstellung von CIS-Ressourcen übergeben werden, umfassen `ipam: true` Und `custom_resource_mode: true`. Diese Parameter sind für die CIS-Integration mit einem IPAM-Controller erforderlich. Überprüfen Sie, ob die CIS die IPAM-Integration aktiviert hat, indem Sie die F5 IPAM-Ressource erstellen.

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

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

25. Erstellen Sie das Servicekonto, die Rolle und die Einbindung, die für den F5 IPAM-Controller erforderlich sind. Erstellen Sie eine YAML-Datei, und fügen Sie den folgenden Inhalt ein.

```
[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. Erstellen Sie die Ressourcen.

```
[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. Erstellen Sie eine YAML-Datei, und fügen Sie die nachfolgend angegebene F5 IPAM-Bereitstellungsdefinition ein.

Aktualisieren Sie den ip-Bereich-Parameter in spec.template.spec.Containers[0].args unten, um die ipamLabels und IP-Adressbereiche zu berücksichtigen, die Ihrem Setup entsprechen.

IpamLabels [`range1` Und `range2` Im folgenden Beispiel] müssen für die Dienste des Typs loadbalancer Anmerkungen gemacht werden, damit der IPAM-Controller eine IP-Adresse aus dem definierten Bereich erkennt und zuweist.

```
[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. Erstellen Sie die F5 IPAM Controller-Implementierung.

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

deployment/f5-ipam-controller created
```

29. Überprüfen Sie, ob die F5 IPAM-Controller-Pods ausgeführt werden.

```
[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. Erstellen Sie das F5 IPAM-Schema.

```
[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
```

Verifizierung

1. Erstellen Sie einen Service vom Typ Load Balancer

```
[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. Überprüfen Sie, ob der IPAM-Controller ihm eine externe IP zuweist.

```
[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. Erstellen Sie eine Implementierung, und verwenden Sie den erstellten Load Balancer Service.

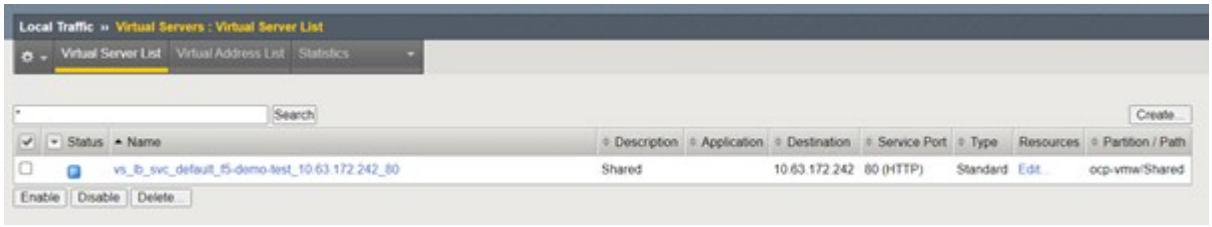
```
[admin@rhel-7 ~]$ vi example_deployment.yaml

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. Prüfen Sie, ob die Pods ausgeführt werden.

```
[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. Prüfen Sie, ob der entsprechende virtuelle Server im BIG-IP-System für den Dienst vom Typ loadbalancer in OpenShift erstellt wird. Navigieren Sie zu lokalem Verkehr > Virtuelle Server > Liste virtueller Server.



Private Bildregistrien Werden Erstellt

Für die meisten Bereitstellungen von Red hat OpenShift verwenden Sie eine öffentliche Registrierung wie "Quay.io" Oder "DockerHub" Erfüllt die meisten Kundenanforderungen. Es gibt jedoch Zeiten, in denen ein Kunde seine eigenen privaten oder angepassten Bilder hosten möchte.

Dieses Verfahren dokumentiert die Erstellung einer privaten Image-Registrierung, die durch ein persistentes Volume von Trident und NetApp ONTAP gesichert wird.

> Astra Control Center erfordert eine Registrierung, um die Bilder zu hosten, die die Astra-Container benötigen. Im folgenden Abschnitt werden die Schritte zum Einrichten einer privaten Registrierung auf dem Red hat OpenShift-Cluster sowie das Drücken der Bilder beschrieben, die für die Installation des Astra Control Center erforderlich sind.

Erstellen einer privaten Bildregistrierung

1. Entfernen Sie die Standardanmerkung aus der aktuellen Standard-Storage-Klasse und versehen Sie die Trident-gestützte Storage-Klasse als Standard für das OpenShift-Cluster.

```
[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. Bearbeiten Sie den Operator imageregistry, indem Sie die folgenden Speicherparameter in eingeben `spec` Abschnitt.

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

storage:
  pvc:
    claim:
```

3. Geben Sie die folgenden Parameter in das ein `spec` Abschnitt zum Erstellen einer OpenShift-Route mit einem benutzerdefinierten Hostnamen. Speichern und beenden.

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

Die obige Routenkonfiguration wird verwendet, wenn Sie einen benutzerdefinierten Hostnamen für Ihre Route wünschen. Wenn OpenShift eine Route mit einem Standard-Hostnamen erstellen soll, können Sie dem die folgenden Parameter hinzufügen `spec` Abschnitt: `defaultRoute: true`.

Benutzerdefinierte TLS-Zertifikate

Wenn Sie einen benutzerdefinierten Hostnamen für die Route verwenden, wird standardmäßig die TLS-Standardkonfiguration des OpenShift Ingress-Operators verwendet. Sie können der Route jedoch eine benutzerdefinierte TLS-Konfiguration hinzufügen. Um das zu tun, führen Sie folgende Schritte durch.

a. Erstellen Sie ein Geheimnis mit den TLS-Zertifikaten und dem Schlüssel der 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. Bearbeiten Sie den Operator Imageregistry, und fügen Sie dem die folgenden Parameter hinzu `spec` Abschnitt.

```
[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. Bearbeiten Sie den Operator Imageregistry erneut, und ändern Sie den Verwaltungsstatus des Bedieners in das `Managed` Bundesland. Speichern und beenden.

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. Wenn alle Voraussetzungen erfüllt sind, werden PVCs, Pods und Services für die private Image Registry erstellt. In wenigen Minuten sollte die Registrierung betriebsbereit sein.

```
[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. Wenn Sie die standardmäßigen TLS-Zertifikate für die Registrierungsroute Ingress Operator OpenShift verwenden, können Sie die TLS-Zertifikate mit dem folgenden Befehl abrufen.

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

7. Damit OpenShift-Knoten auf die Bilder zugreifen und sie aus der Registrierung ziehen können, fügen Sie die Zertifikate dem Docker-Client auf den OpenShift-Knoten hinzu. Erstellen Sie im eine Konfigurationskarte `openshift-config` Namespace mit den TLS-Zertifikaten und patchen Sie ihn mit der Konfiguration des Cluster-Images, um das Zertifikat vertrauenswürdig zu machen.

```
[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. Die interne OpenShift-Registrierung wird durch Authentifizierung gesteuert. Alle OpenShift-Benutzer können auf die OpenShift-Registrierung zugreifen, aber die Vorgänge, die der angemeldete Benutzer durchführen kann, sind von den Benutzerberechtigungen abhängig.
   a. Damit ein Benutzer oder eine Gruppe von Benutzern Bilder aus der Registrierung ziehen kann, müssen den Benutzern die Rolle Registry-Viewer zugewiesen sein.

```
[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. Damit ein Benutzer oder eine Benutzergruppe Bilder schreiben oder übertragen kann, muss dem/den Benutzer die Rolle des Registrierungs-Editors zugewiesen sein.

```
[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. Damit OpenShift-Knoten auf die Registrierung zugreifen und die Bilder per Push oder Pull übertragen können, müssen Sie einen Pull Secret konfigurieren.

```
[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. Dieses Pull-Secret kann dann auf Dienstkonten gepatcht oder in der entsprechenden Pod-Definition referenziert werden.
    a. Führen Sie den folgenden Befehl aus, um ihn auf Dienstkonten zu patchen.

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

    b. Um den Pull-Secret in der Pod-Definition zu referenzieren, fügen Sie dem den folgenden Parameter hinzu `spec` Abschnitt.

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

11. Führen Sie die folgenden Schritte aus, um ein Bild von den Arbeitsstationen getrennt vom OpenShift-Knoten zu schieben oder zu ziehen.
    a. Fügen Sie die TLS-Zertifikate zum Docker-Client hinzu.

```
[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. Melden Sie sich über den oc-Anmeldebefehl bei OpenShift an.

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

    c. Melden Sie sich mit den OpenShift-Benutzeranmeldeinformationen über den Befehl podman/Docker bei der Registrierung an.

Podman

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

+ HINWEIS: Wenn Sie verwenden `kubeadmin` Benutzer, um sich bei der privaten Registrierung anzumelden, dann Token statt Passwort verwenden.

docker

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

+ HINWEIS: Wenn Sie verwenden `kubeadmin` Benutzer, um sich bei der privaten Registrierung anzumelden, dann Token statt Passwort verwenden.

d. Drücken oder ziehen Sie die Bilder.

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
```