高级配置选项

NetApp Solutions

NetApp
September 26, 2024

目录

高级配置选项

了解负载平衡器选项

了解负载平衡器选项： **Red Hat OpenShift** 与 **NetApp**

在大多数情况下， Red Hat OpenShift 会通过路由向外部世界提供应用程序。通过为服务提供一个可从外部访问的主机名来公开该服务。OpenShift 路由器可以使用定义的路由及其服务标识的端点，以便为外部客户端提供此命名连接。

但是，在某些情况下，应用程序需要部署和配置自定义负载平衡器才能公开相应的服务。其中一个示例是 NetApp Astra 控制中心。为了满足这一需求，我们评估了许多自定义负载平衡器选项。本节将介绍其安装和配置。

以下页面提供了有关负载平衡器选项的追加信息，这些选项已在 Red Hat OpenShift with NetApp 解决方案中进行验证：

- "元 LB"
- "F5 BIG-IP"

安装 **MetalLB** 负载平衡器： **Red Hat OpenShift** 与 **NetApp**

此页面列出了 MetalLB 负载平衡器的安装和配置说明。

MetalLB 是一种安装在 OpenShift 集群上的自托管网络负载平衡器，可用于在未在云提供程序上运行的集群中创建类型为负载平衡器的 OpenShift 服务。MetalLB 可协同工作以支持负载平衡器服务的两个主要功能是地址分配和外部公告。

MetalLB 配置选项

根据 MetalLB 如何公布分配给 OpenShift 集群以外的负载平衡器服务的 IP 地址，它可在两种模式下运行：

- * 第 2 层模式。 * 在此模式下， OpenShift 集群中的一个节点将接管此服务的所有权，并对该 IP 的 ARP 请求做出响应，使其可在 OpenShift 集群之外访问。由于只有节点才公布 IP ，因此存在带宽瓶颈和较慢的故障转移限制。有关详细信息，请参见文档 "此处"。
- * BGP 模式。 * 在此模式下， OpenShift 集群中的所有节点都与路由器建立 BGP 对等会话，并公布路由以将流量转发到服务 IP 。前提条件是将 MetalLB 与该网络中的路由器集成在一起。由于 BGP 中采用哈希机制，因此在服务的 IP 到节点映射发生更改时，它具有一定的限制。有关详细信息，请参见文档 "此处"。

> 在本文档中，我们将在第 2 层模式下配置 MetalLB 。

安装 **MetalLB** 负载平衡器

1. 下载 MetalLB 资源。

```
[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. 编辑文件 `metallb.yaml` 并从控制器部署和主讲人 DemonSet 中删除 `spec.template.spec.securityContext` 。

   ◦ 要删除的行： *

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

3. 创建 `metallb-system` 命名空间。

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

4. 创建 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. 在配置 MetalLB 扬声器之前，请授予扬声器 DemonSet 提升权限，使其能够执行使负载平衡器正常工作所需的网络配置。

```
[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. 通过在 `metallb-system` 命名空间中创建 `ConfigMap` 来配置 MetalLB 。

```
[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. 现在，在创建负载平衡器服务时， MetalLB 会为这些服务分配一个外部 IP ，并通过响应 ARP 请求来公布 IP 地址。

如果要在 BGP 模式下配置 MetalLB ，请跳过上述步骤 6 并按照 MetalLB 文档中的操作步骤进行操作 "此处"。

安装 **F5 BIG-IP** 负载平衡器

F5 BIG-IP 是一款应用程序交付控制器（ Application Delivery Controller ， AD ），可提供一系列高级生产级流量管理和安全服务，例如 L4-L7 负载平衡， SSL/TLS 卸载， DNS ，防火墙等。这些服务可显著提高应用程序的可用性，安全性和性能。

F5 BIG-IP 可以在专用硬件上，云中或内部虚拟设备上以各种方式进行部署和使用。请参见此处的文档，了解如何根据需要部署 F5 BIG-IP 。

为了将 F5 BIG-IP 服务与 Red Hat OpenShift 高效集成， F5 提供了 BIG-IP 容器传入服务（ BIG-IP Container Ingress Service ， CIS ）。CI 作为控制器 POD 进行安装，用于监控 OpenShift API 以获取某些自定义资源定

义（ Custom Resource Definitions ， CRD ），并管理 F5 BIG-IP 系统配置。可以配置 F5 BIG-IP CIS ，以控制 OpenShift 中的服务类型 LoadBalbalers" 和 " 路由 " 。

此外，要自动分配 IP 地址以服务类型负载平衡器，您可以使用 F5 IPAM 控制器。F5 IPAM 控制器作为控制器 POD 进行安装，该控制器 POD 会通过 ipamLabel 标注监视 OpenShift API 以获取负载平衡器服务，以便从预配置的池分配 IP 地址。

此页面列出了 F5 BIG-IP CIS 和 IPAM 控制器的安装和配置说明。作为前提条件，您必须已部署并获得 F5 BIG-IP 系统的许可。此外，它还必须获得 SDN 服务的许可，这些服务默认包含在 BIG-IP VE 基础许可证中。

F5 BIG-IP 可以在独立模式或集群模式下部署。出于此验证的目的， F5 BIG-IP 部署在独立模式下，但出于生产目的，最好使用由大型 IP 组成的集群来避免单点故障。

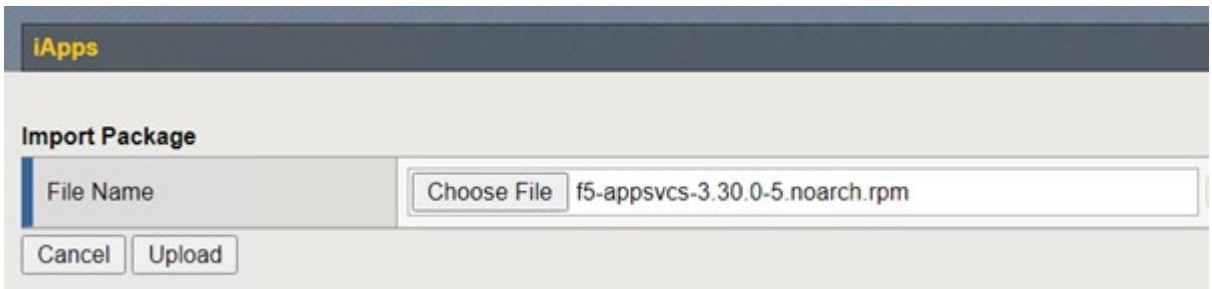
F5 BIG-IP 系统可以部署在专用硬件上，云中或内部部署的虚拟设备中，其版本高于 12.x ，以便与 F5 CIS 集成。在本文档中，我们已将 F5 BIG-IP 系统验证为虚拟设备，例如使用 BIG-IP VE 版本。

经过验证的版本

技术	软件版本
Red Hat OpenShift	4.6 EUS ， 4.7
F5 BIG-IP VE 版本	16.1.0
F5 容器传入服务	2.5.1
F5 IPAM 控制器	0.1.4
F5 AS3	3.30.0

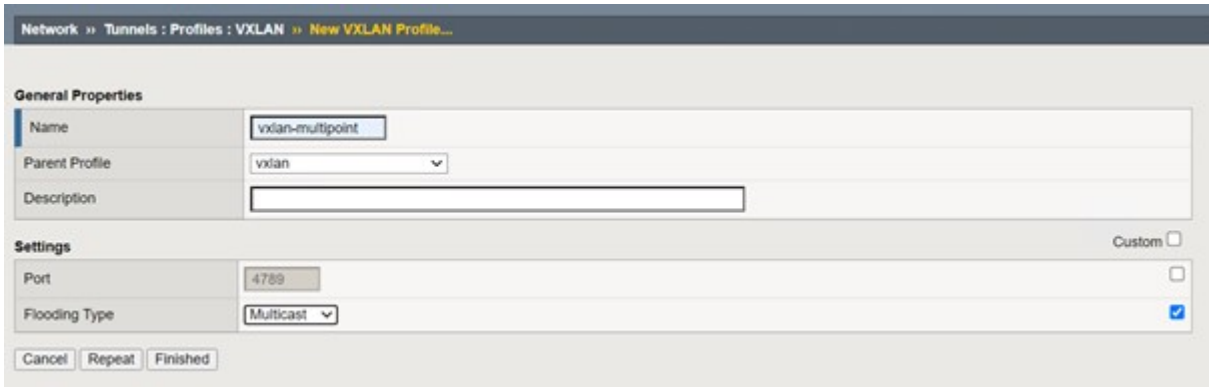
安装

1. 安装 F5 Application Services 3 扩展，以允许 BIG-IP 系统接受 JSON 中的配置，而不是强制命令。转至 "F5 AS3 GitHub 存储库"，并下载最新的 RPM 文件。
2. 登录到 F5 BIG-IP 系统，导航到 "iApps" > "Package Management LX" ，然后单击 "Import" 。
3. 单击选择文件并选择已下载的 AS3 RPM 文件，单击确定，然后单击上传。

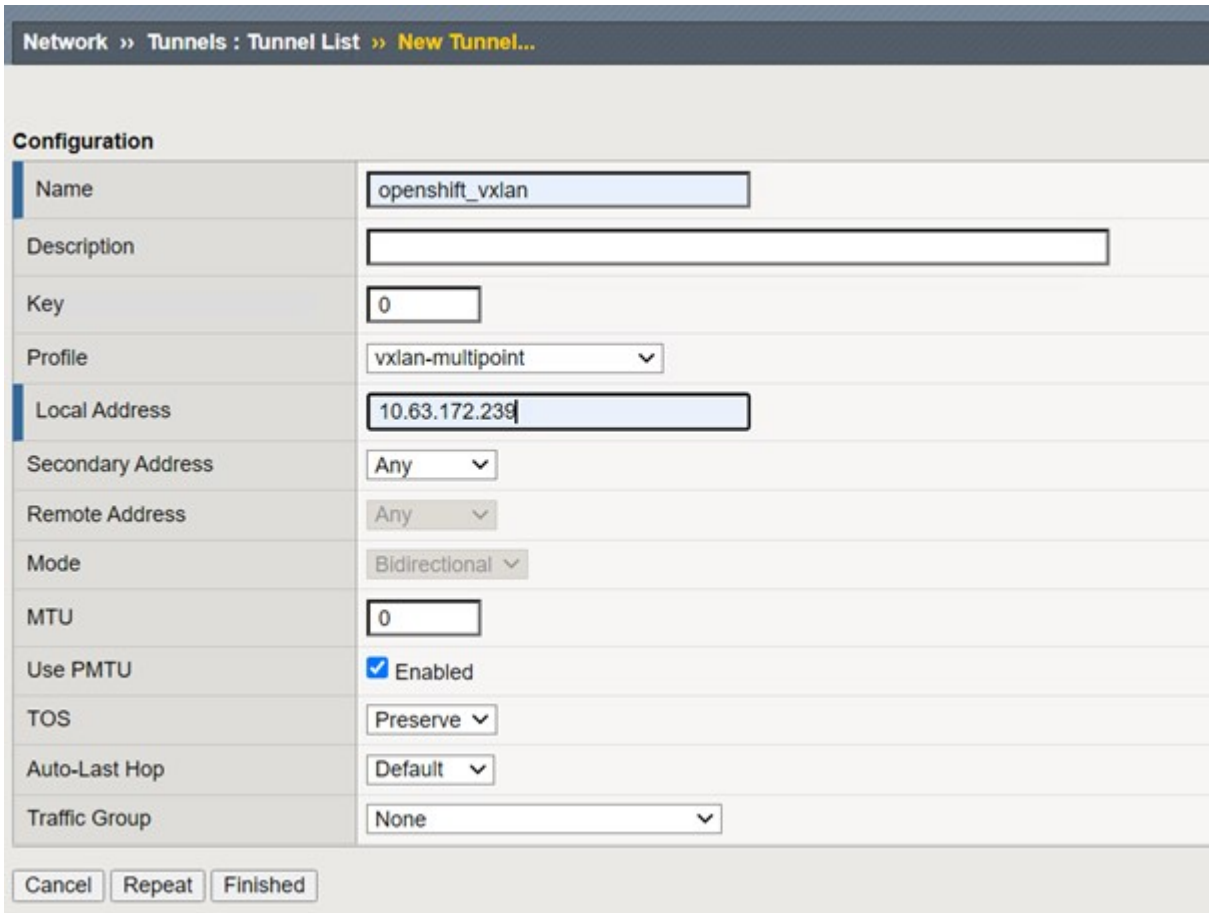


4. 确认 AS3 扩展已成功安装。

5. 接下来，配置 OpenShift 和 BIG-IP 系统之间通信所需的资源。首先，通过在 OpenShift SDN 的 BIG-IP 系统上创建 VXLAN 通道接口，在 OpenShift 和 BIG-IP 服务器之间创建通道。导航到 " 网络 ">" 通道 ">" 配置文件 " ，单击 " 创建 " ，然后将父配置文件设置为 VXLAN ，并将 " 洪水类型 " 设置为 " 多播 " 。输入配置文件的名称，然后单击完成。



6. 导航到 " 网络 ">" 通道 ">" 通道列表 " ，单击 " 创建 " ，然后输入通道的名称和本地 IP 地址。选择在上一步中创建的通道配置文件，然后单击完成。



7. 使用 cluster-admin 权限登录到 Red Hat OpenShift 集群。
8. 在 OpenShift 上为 F5 BIG-IP 服务器创建一个子网，从而将子网从 OpenShift 集群扩展到 F5 BIG-IP 服务器。下载主机子网 YAML 定义。

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

9. 编辑主机子网文件并为 OpenShift SDN 添加 BIG-IP VTEP （ VXLAN 通道） IP 。

```
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
```

根据您的环境情况更改主机提示和其他详细信息。

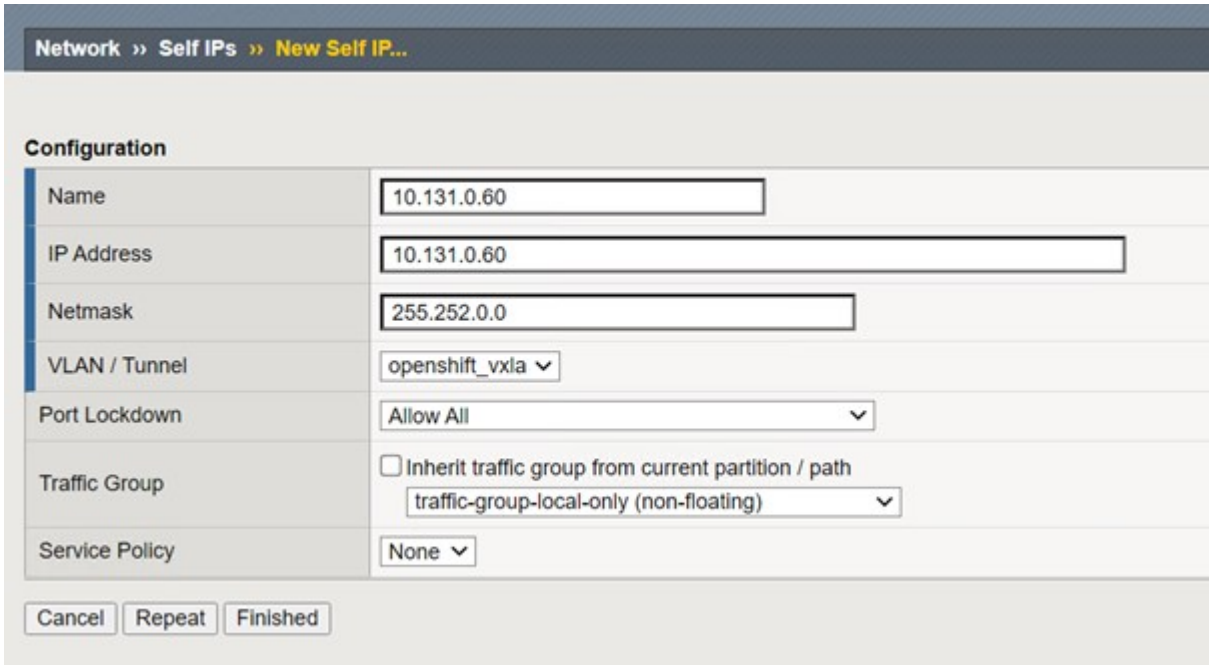
10. 创建 HostSubnet 资源。

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

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

11. 获取为 F5 BIG-IP 服务器创建的主机子网的集群 IP 子网范围。

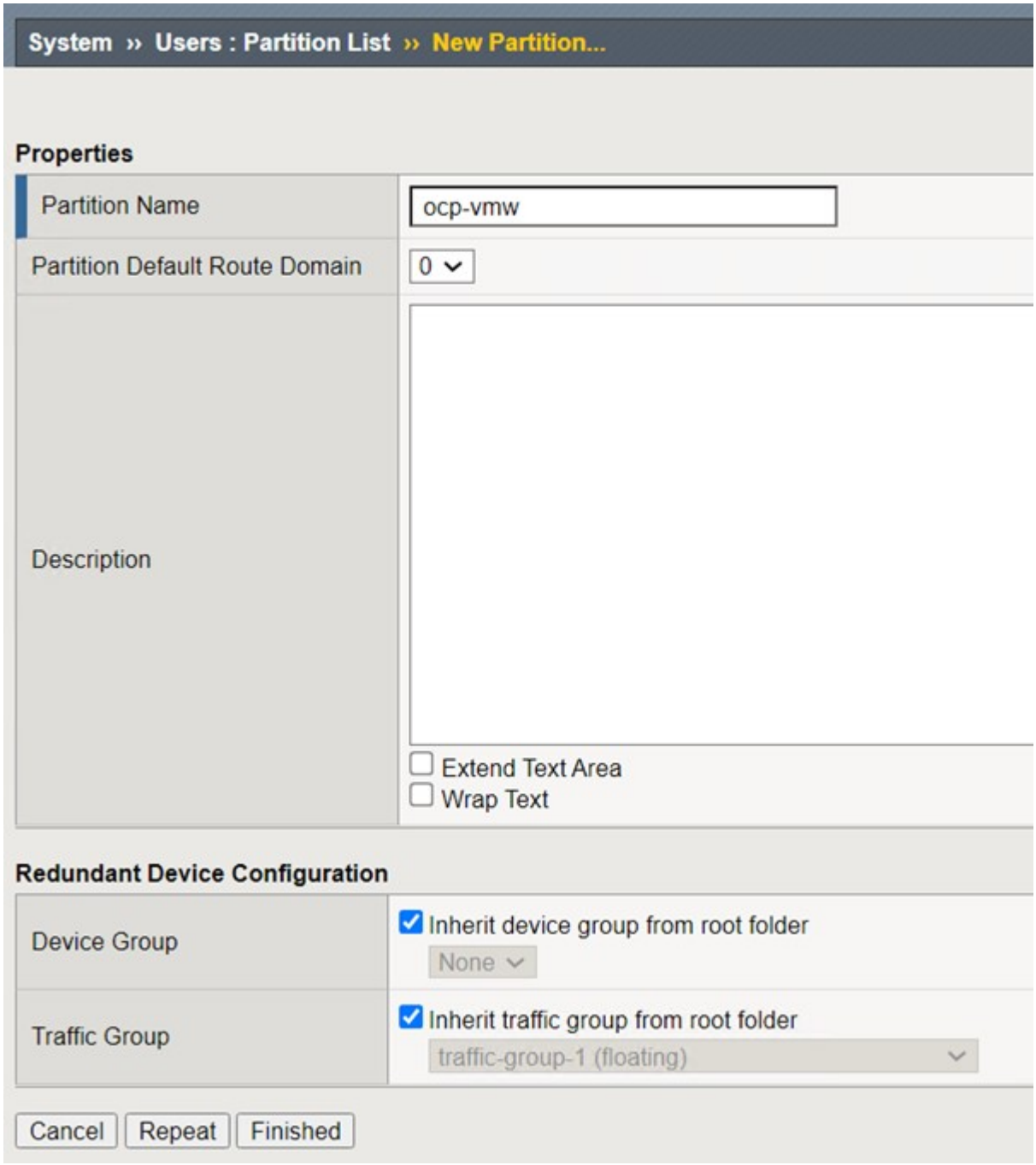
```
[admin@rhel-7 ~]$ oc get hostsubnet

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. 在 OpenShift VXLAN 上使用与 F5 BIG-IP 服务器对应的 OpenShift 主机子网范围中的 IP 创建自 IP 。登录到 F5 BIG-IP 系统，导航到 " 网络 ">" 自 IP " ，然后单击 " 创建 " 。输入为 F5 BIG-IP 主机子网创建的集群 IP 子网中的 IP ，选择 VXLAN 通道，然后输入其他详细信息。然后单击完成。



13. 在 F5 BIG-IP 系统中创建一个分区，以便在 CIS 中配置和使用。导航到系统 > 用户 > 分区列表，单击创建

，然后输入详细信息。然后单击完成。



F5 建议不要对由 CIS 管理的分区进行手动配置。

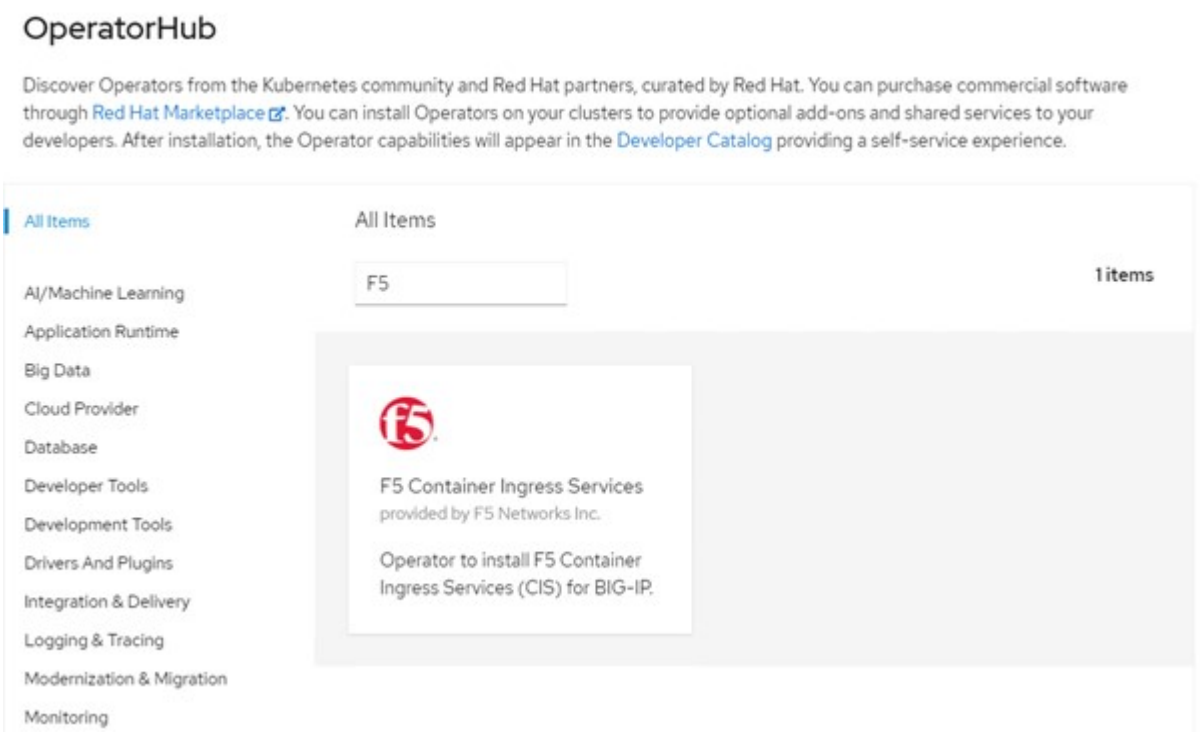
14. 使用 OperatorHub 中的运算符安装 F5 BIG-IP CIS 。使用集群管理员权限登录到 Red Hat OpenShift 集群，并使用 F5 BIG-IP 系统登录凭据创建一个密钥，这是操作员的前提条件。

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

secret/bigip-login created
```

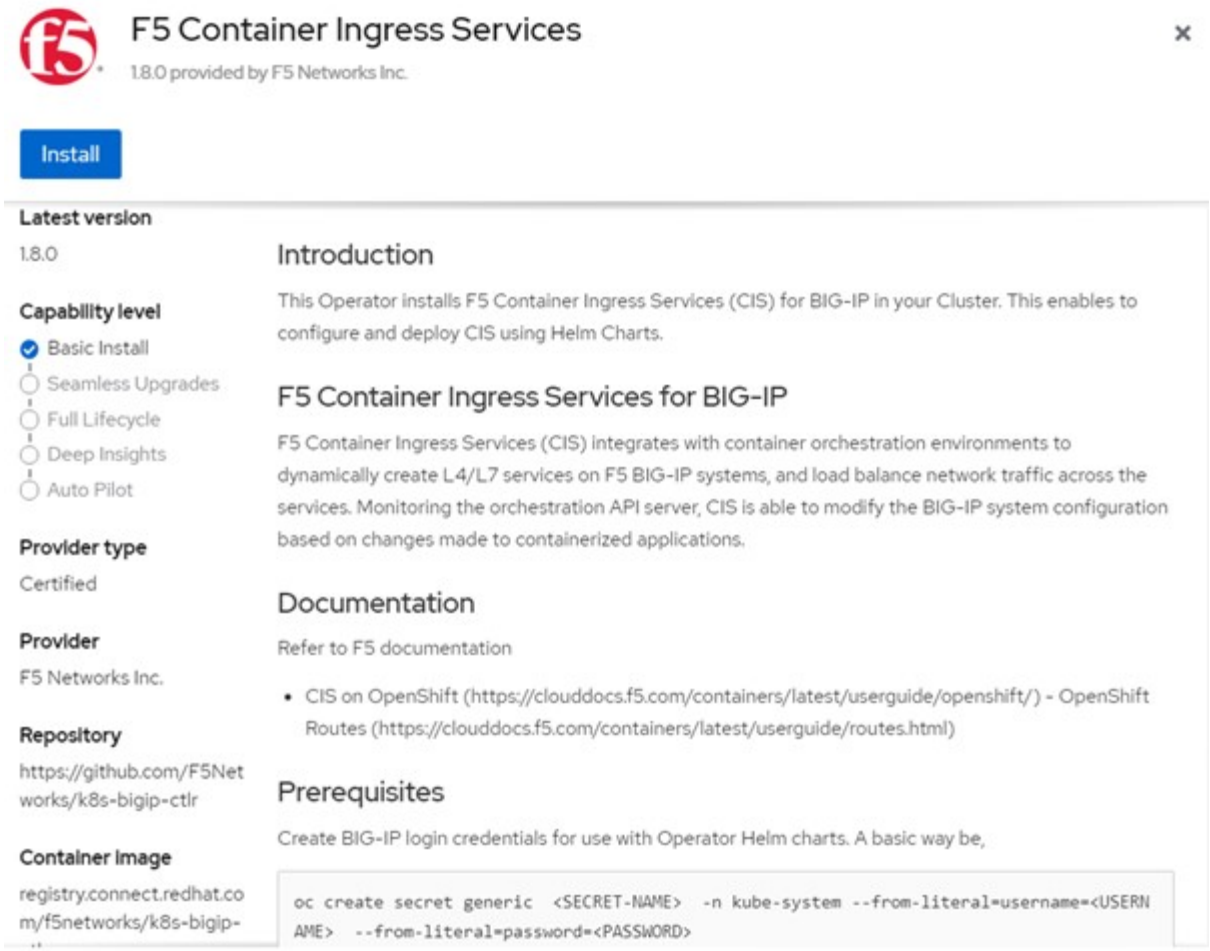
15. 安装 F5 CIS CRD 。

```
[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
```

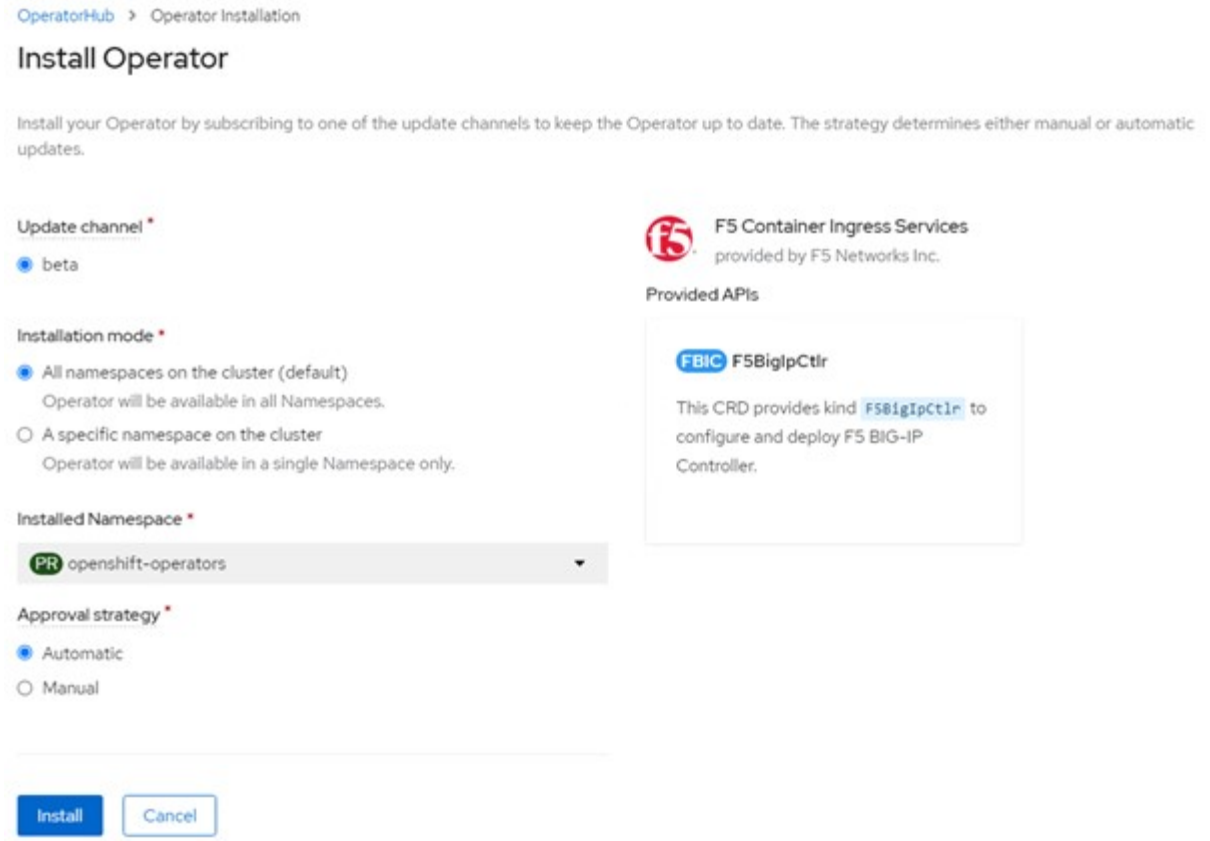
16. 导航到 Operators > OperatorHub ，搜索关键字 F5 ，然后单击 F5 Container In出口 服务磁贴。



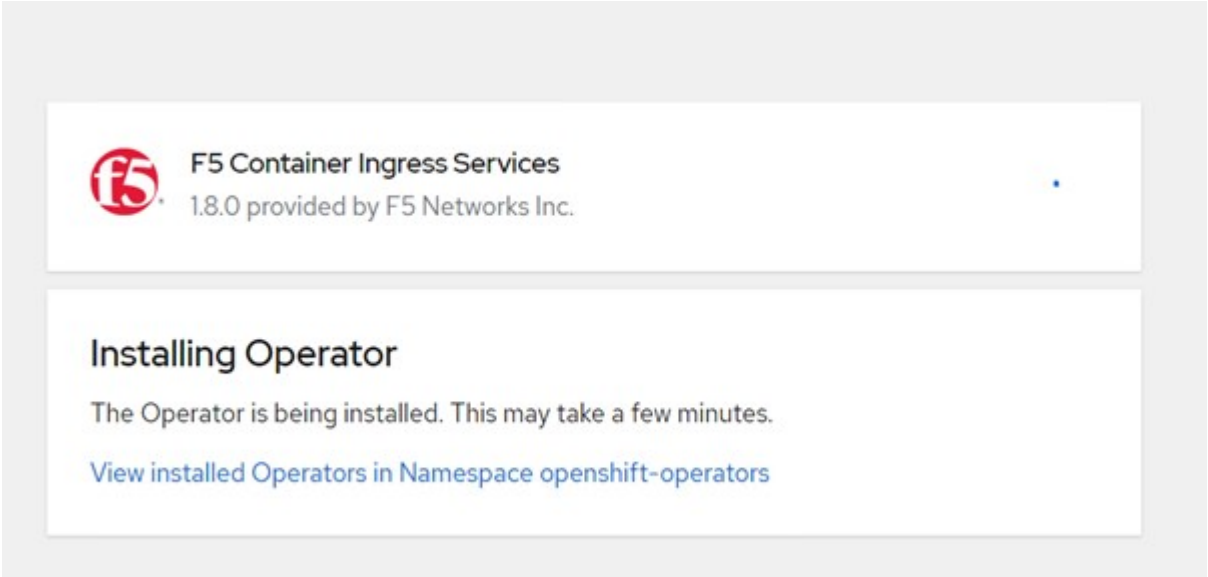
17. 阅读操作员信息，然后单击安装。



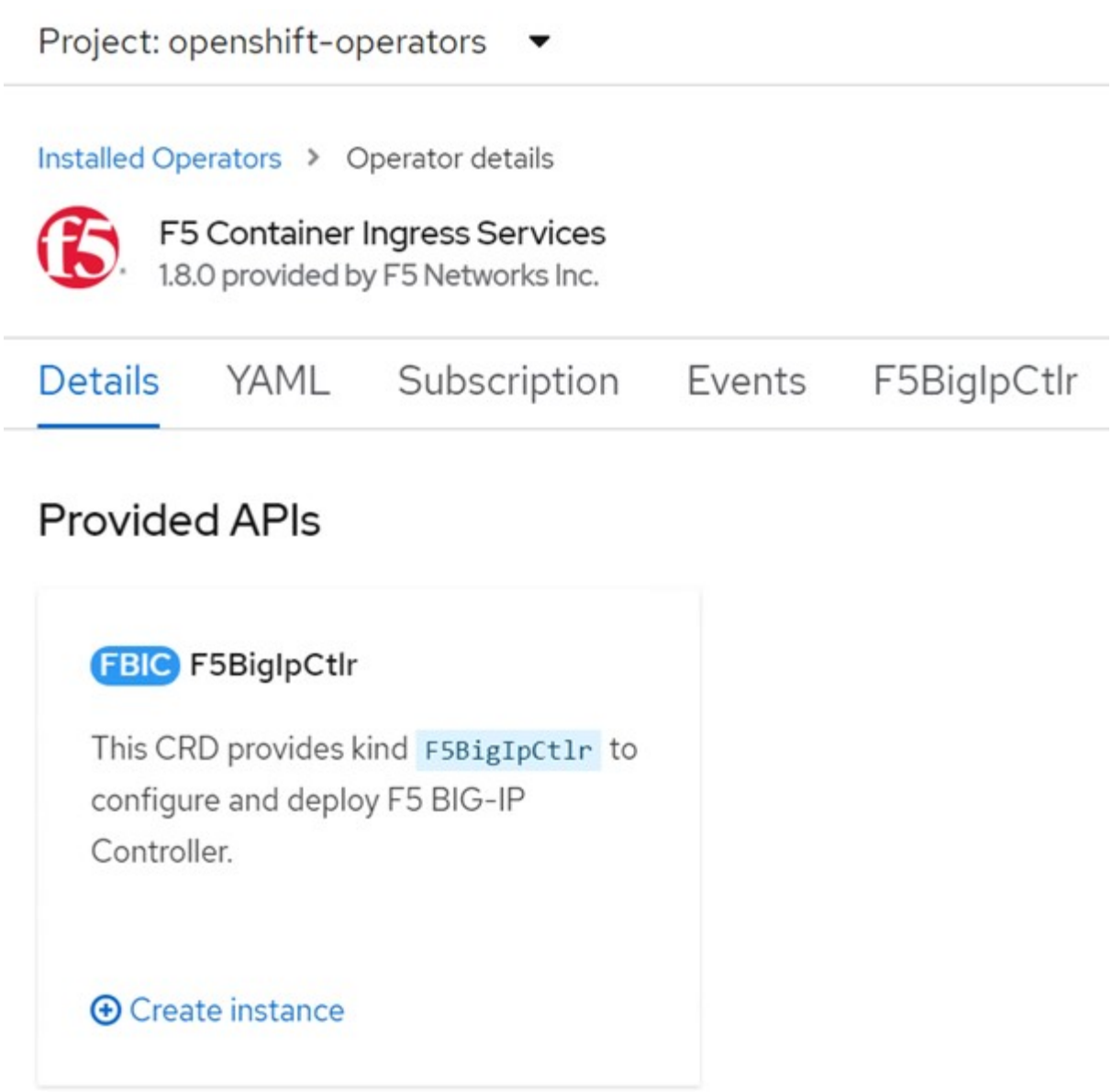
18. 在 Install Operator 屏幕上，保留所有默认参数，然后单击 Install 。



19. 安装操作员需要一段时间。



20. 安装操作员后，将显示安装成功消息。
21. 导航到 Operators > Installed Operators ，单击 F5 Container In出口 服务，然后单击 F5BigIpCtrl+Alt+Del 图块下的 Create Instance 。

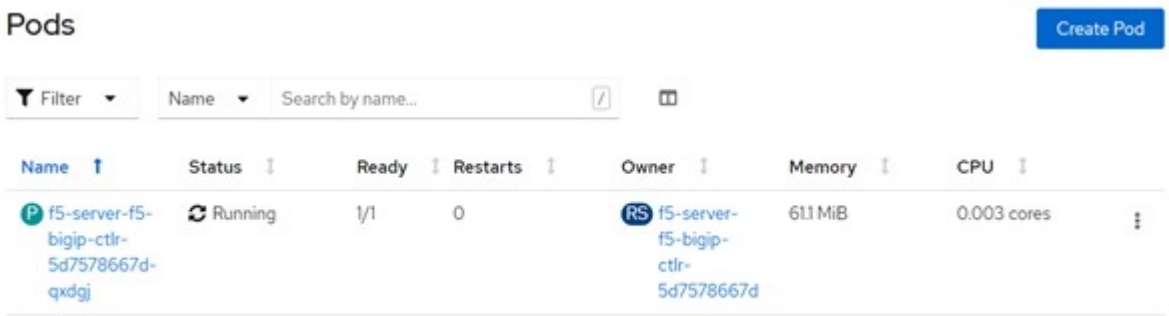


22. 单击 YAML View ，并在更新必要参数后粘贴以下内容。

> 在复制内容之前，更新以下参数 `bigip_partition` ， ` OpenShift_SDN_name` ， `bigip_url` 和 `bigip_login_secret` ，以反映您的设置值。

```
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. 粘贴此内容后，单击创建。此操作将在 Kube-system 命名空间中安装 CIS Pod 。



默认情况下， Red Hat OpenShift 提供了一种通过路由公开服务以实现 L7 负载平衡的方法。内置的 OpenShift 路由器负责公布和处理这些路由的流量。但是，您也可以将 F5 CIS 配置为支持通过外部 F5 BIG-IP 系统的路由，该系统可以作为辅助路由器运行，也可以替代自托管 OpenShift 路由器运行。CIS 在 BIG-IP 系统中创建一个虚拟服务器，充当 OpenShift 路由的路由器， BIG-IP 负责处理公告和流量路由。有关启用此功能的参数的信息，请参见此处的文档。请注意，这些参数是在 APPS/v1 API 中为 OpenShift 部署资源定义的。因此，在将这些参数与 F5BigIpartl 资源 cis.f5.com/v1 API 结合使用时，请将参数名称的连字符（ - ）替换为下划线（ _ ）。

24. 传递给创建 CIS 资源的参数包括 `ipam ： true` 和 `custom_resource_mode ： true` 。要启用与 IPAM 控制器的 CIS 集成，需要使用这些参数。通过创建 F5 IPAM 资源验证 CIS 是否已启用 IPAM 集成。

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

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

25. 创建 F5 IPAM 控制器所需的服务帐户，角色和角色绑定。创建 YAML 文件并粘贴以下内容。

```
[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. 创建资源。

```
[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. 创建一个 YAML 文件并粘贴下面提供的 F5 IPAM 部署定义。

更新以下 spec.template.spec.containers[0].args 中的 ip-range 参数，以反映与您的设置对应的 ipamLabel 和 IP 地址范围。

要使 IPAM 控制器能够从定义的范围检测和分配 IP 地址，需要为类型为 loadbalancer 的服务标注 ipamLabels （`range1` 和` range2 `in below example] ）。

```
[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. 创建 F5 IPAM 控制器部署。

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

deployment/f5-ipam-controller created
```

29. 验证 F5 IPAM 控制器 Pod 是否正在运行。

```
[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. 创建 F5 IPAM 模式。

```
[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
```

验证

1. 创建类型为 loadbalancer 的服务

```
[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. 检查 IPAM 控制器是否为其分配了外部 IP 。

```
[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. 创建部署并使用已创建的负载平衡器服务。

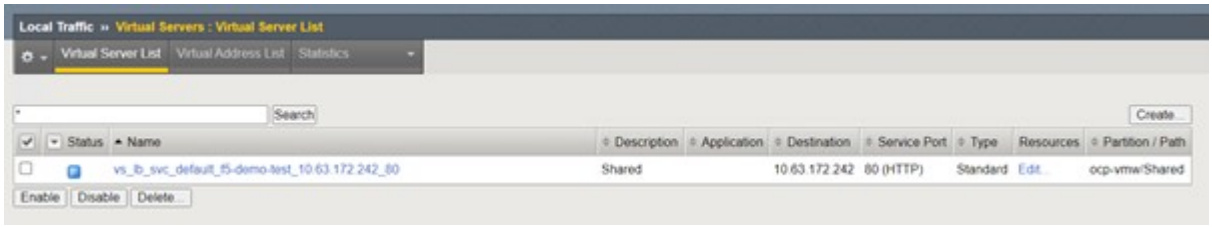
```
[admin@rhel-7 ~]$ vi example_deployment.yaml

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. 检查 Pod 是否正在运行。

```
[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. 检查是否在 OpenShift 中为 loadbalancing 类型的服务在 BIG-IP 系统中创建了相应的虚拟服务器。导航到 " 本地流量 ">" 虚拟服务器 ">" 虚拟服务器列表 " 。



创建私有映像注册表

对于大多数 Red Hat OpenShift 部署，请使用等公有注册表 "Quay.io" 或 "DockerHub" 满足大多数客户的需求。但是，有时客户可能希望托管自己的私有或自定义映像。

本操作步骤介绍了如何创建私有映像注册表，该注册表由 Astra Trident 和 NetApp ONTAP 提供的永久性卷提供支持。

Astra 控制中心需要注册表来托管 Astra 容器所需的映像。以下部分介绍了在 Red Hat OpenShift 集群上设置专用注册表以及推送支持安装 Astra 控制中心所需的映像的步骤。

创建私有映像注册表

1. 从当前默认存储类中删除默认标注，并将支持 Trident 的存储类标注为 OpenShift 集群的默认值。

```
[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. 在 `sPec` 部分中输入以下存储参数，以编辑 imageeregistry 运算符。

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

storage:
  pvc:
    claim:
```

3. 在 `sPec` 部分中输入以下参数，以便使用自定义主机名创建 OpenShift 路由。保存并退出。

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

如果要为路由设置自定义主机名，则会使用上述路由配置。如果您希望 OpenShift 使用默认主机名创建路由，可以将以下参数添加到 `sPec` 部分： `defaultRoute ： true` 。

自定义 TLS 证书

默认情况下，当您为路由使用自定义主机名时，它会使用 OpenShift 入口操作员的默认 TLS 配置。但是，您可以向路由添加自定义 TLS 配置。为此，请完成以下步骤：

a. 使用路由的 TLS 证书和密钥创建密钥。

```
[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. 编辑 imageeregistry 运算符，并将以下参数添加到 `sPec` 部分。

```
[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. 再次编辑 imageeregistry 运算符，并将该运算符的管理状态更改为 `Maned` state 。保存并退出。

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. 如果满足所有前提条件，则会为专用映像注册表创建 PVC ， Pod 和服务。几分钟后，注册表就会启动。

```
[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. 如果您对传入操作员 OpenShift 注册表路由使用默认 TLS 证书，则可以使用以下命令提取 TLS 证书。

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

7. 要允许 OpenShift 节点访问并从注册表中提取映像，请将证书添加到 OpenShift 节点上的 Docker 客户端。使用 TLS 证书在 `OpenShift-config` 命名空间中创建一个配置映射，并将其修补到集群映像配置中以使此证书可信。

```
[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. OpenShift 内部注册表由身份验证控制。所有 OpenShift 用户都可以访问 OpenShift 注册表，但登录用户可以执行的操作取决于用户权限。

a. 要允许用户或用户组从注册表中提取映像，必须为用户分配注册表查看器角色。

```
[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. 要允许用户或用户组写入或推送映像，必须为用户分配注册表编辑器角色。

```
[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. 要使 OpenShift 节点能够访问注册表并推送或拉取映像，您需要配置拉取密钥。

```
[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. 然后，可以将此提取密钥修补到服务帐户或在相应的 POD 定义中引用。

a. 要将其修补到服务帐户，请运行以下命令。

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

b. 要在 Pod 定义中引用 Pull secret ，请将以下参数添加到 `sPec` 部分。

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

11. 要从 OpenShift 节点以外的工作站推送或拉取映像，请完成以下步骤。

a. 将 TLS 证书添加到 Docker 客户端。

```
[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. 使用 oc login 命令登录到 OpenShift 。

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

c. 使用 podman/Docker 命令使用 OpenShift 用户凭据登录到注册表。

podman

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

+ 注意：如果您使用 `kubeadmin` user 登录到专用注册表，请使用 token 代替密码。

Docker

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

+ 注意：如果您使用 `kubeadmin` user 登录到专用注册表，请使用 token 代替密码。

d. 推送或拉图像。

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
```

版权信息

商标信息