進階組態選項
NetApp Solutions

NetApp
September 26, 2024

目錄

進階組態選項 1
探索負載平衡器選項 1
建立私有映像登錄 20

進階組態選項

探索負載平衡器選項

探索負載平衡器選項：採用**NetApp**的**Red Hat OpenShift**

在大多數情況下、Red Hat OpenShift會透過路由、讓外部使用者能夠使用應用程式。提供可從外部存取的主機名稱、即可公開服務。OpenShift路由器可以使用定義的路由及其服務所識別的端點、以提供與外部用戶端的命名連線。

不過在某些情況下、應用程式需要部署和設定自訂的負載平衡器、才能提供適當的服務。其中一個例子是NetApp Astra Control Center。為了滿足這項需求、我們評估了許多自訂負載平衡器選項。本節將說明其安裝與組態。

以下頁面提供有關Red Hat OpenShift with NetApp解決方案中驗證的負載平衡器選項的其他資訊：

- "MetalLB."
- "F5 BIG-IP"

安裝**MetalLB**負載平衡器：**Red Hat OpenShift with NetApp**

本頁列出MetalLB負載平衡器的安裝與組態指示。

MetalLB是安裝在OpenShift叢集上的自我代管網路負載平衡器、可在未在雲端供應商上執行的叢集中、建立類型負載平衡器的OpenShift服務。MetalLB的兩項主要功能是位址分配和外部宣告、這些功能可搭配運作以支援負載平衡器服務。

MetalLB組態選項

根據MetalLB如何宣告指派給OpenShift叢集外部負載平衡器服務的IP位址、它以兩種模式運作：

- *第2層模式。*在此模式下、OpenShift叢集中的一個節點會取得服務的所有權、並回應該IP的ARP要求、以便在OpenShift叢集外部存取。因為只有節點會通告IP、所以它會有頻寬瓶頸和緩慢的容錯移轉限制。如需詳細資訊、請參閱文件 "請按這裡"。
- * BGP模式。*在此模式下、OpenShift叢集中的所有節點都會與路由器建立BGP對等工作階段、並通告路由以將流量轉送到服務IP。這項作業的先決條件是將MetalLB與該網路中的路由器整合。由於BGP中的雜湊機制、因此在變更服務的IP對節點對應時、會有一定的限制。如需詳細資訊、請參閱文件 "請按這裡"。

針對本文件、我們將在第2層模式中設定MetalLB。

安裝**MetalLB**負載平衡器

1. 下載MetalLB資源。

```
[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. 編輯檔案「metallb.yaml」、並從「控制器部署」和「示範演講者」中移除「pec.template.spec.securityContext`」。

   要刪除的行數：

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

3. 建立「metallb-system」命名空間。

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

4. 建立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. 在設定MetalLB揚聲器之前、請先授予揚聲器示範設定提高權限、以便執行所需的網路組態、使負載平衡器正常運作。

```
[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. 在「metallb-system」命名空間中建立「ConfigMap」來設定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. 現在、當建立負載平衡器服務時、MetalLB會指派外部IP給服務、並回應ARP要求來通告IP位址。

如果您想要在BGP模式中設定MetalLB、請跳過上述步驟6、然後依照MetalLB文件中的程序進行 "請按這裡"。

安裝F5 BIG-IP負載平衡器

F5 BIG-IP是應用程式交付控制器（ADC）、提供一系列進階的正式作業級流量管理與安全服務、例如L4-L7負載平衡、SSL/TLS卸載、DNS、防火牆等。這些服務可大幅提升應用程式的可用度、安全性和效能。

您可以在專屬硬體、雲端或內部部署的虛擬應用裝置上、以各種方式部署和使用F5 BIG-IP。請參閱此處的文件、依照需求探索及部署F5 BIG-IP。

為有效整合使用Red Hat OpenShift的F5 BIG-IP服務、F5提供Big IP Container Ingress Service（CI）。CI是以控制器Pod的形式安裝、可針對特定的自訂資源定義（CRD）來觀看OpenShift API、並管理F5 BIG-IP系統組態。您可以在OpenShift中設定F5 BIG-IP CI、以控制服務類型負載平衡器和路由。

此外、若要自動分配IP位址以服務負載平衡器類型、您可以使用F5 IPAM控制器。將F5 IPAM控制器安裝為控制器Pod、會使用ipamLabel附註來監視負載平衡器服務的OpenShift API、以便從預先設定的集區分配IP位址。

本頁列出適用於F5 BIG-IP CI和IPAM控制器的安裝與組態指示。您必須部署並授權使用F5 BIG-IP系統、才能做為先決條件。也必須授權使用SDN服務、此服務預設隨附於Big IP VE基礎授權中。

可以在獨立或叢集模式中部署F5 BIG-IP。為了進行此驗證、在獨立模式下部署了F5 BIG-IP、但為了正式作業目的、最好使用一個BIG-IP叢集、以避免單點故障。

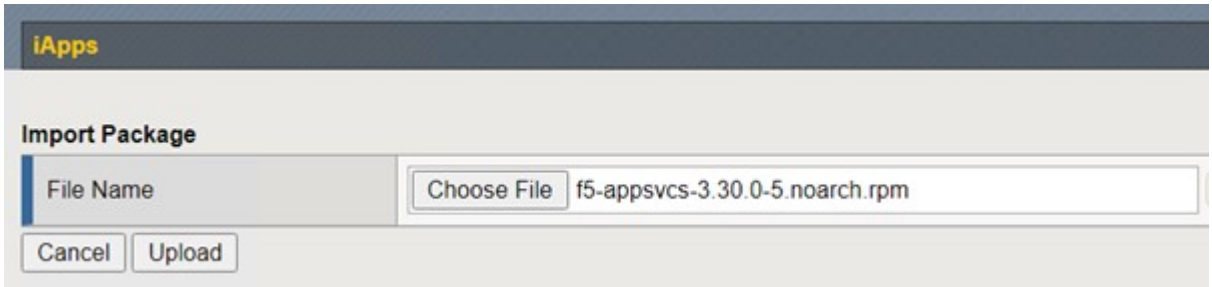
您可以在專屬硬體、雲端或內部部署的虛擬應用裝置上部署一個F5 BIG-IP系統、其版本超過12.x、以便與F5 CI整合。就本文件而言、以虛擬應用裝置（例如使用BIG-IP VE版本）的形式驗證的F5 BIG-IP系統。

已驗證的版本

技術	軟體版本
Red Hat OpenShift	4.6 EUS、4.7
F5 BIG-IP VE版本	16.1.0
F5 Container Ingress服務	2.5.1
F5 IPAM控制器	0.1.4
F5 AS3	3.30.0

安裝

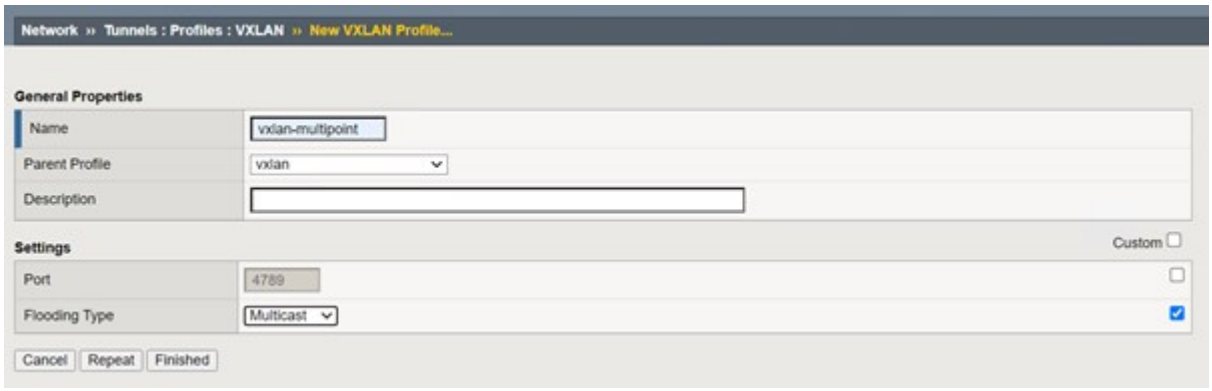
1. 安裝F5 Application Services 3擴充功能、讓BIG-IP系統接受Json中的組態、而非命令命令。前往 "F5 AS3 GitHub儲存庫"下載最新的RPM檔案。
2. 登入F5 BIG-IP系統、瀏覽至iApps >「套件管理Lx」、然後按一下「匯入」。
3. 按一下"選擇檔案"並選取下載的AS3 RPM檔案、按一下"確定"、然後按一下"上傳"。



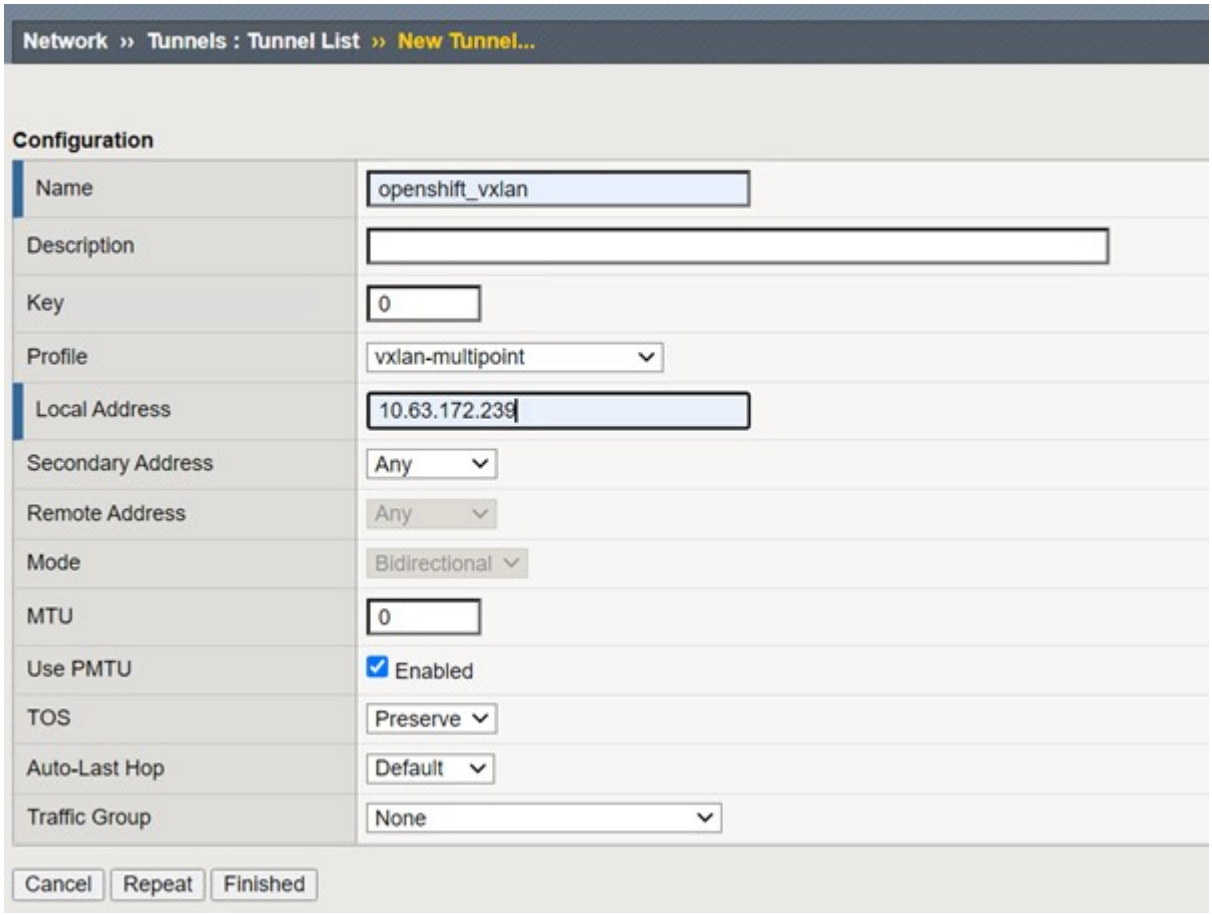
4. 確認已成功安裝AS3擴充功能。

5. 接下來、設定OpenShift與BIG-IP系統之間通訊所需的資源。首先在OpenShift和Big IP伺服器之間建立通

道、方法是在適用於OpenShift SDN的Big IP系統上建立VXLAN通道介面。瀏覽至「Network（網路）」>「Tunnels（通道）」>「Profiles（設定檔）」、按一下「Create（建立）」、然後將「Parent Profile（父設定檔）」設定為VXLAN、「輸入設定檔的名稱、然後按一下「完成」。



6. 瀏覽至「網路」>「通道」>「通道清單」、按一下「建立」、然後輸入通道的名稱和本機IP位址。選取在上一個步驟中建立的通道設定檔、然後按一下「完成」。



7. 以叢集管理權限登入Red Hat OpenShift叢集。
8. 在OpenShift上為F5 BIG-IP伺服器建立主機子網路、將子網路從OpenShift叢集延伸至F5 BIG-IP伺服器。下載主機子網路Yaml定義。

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

9. 編輯主機子網路檔案、並為OpenShift SDN新增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
```

變更適用於您環境的主機IP和其他詳細資料。

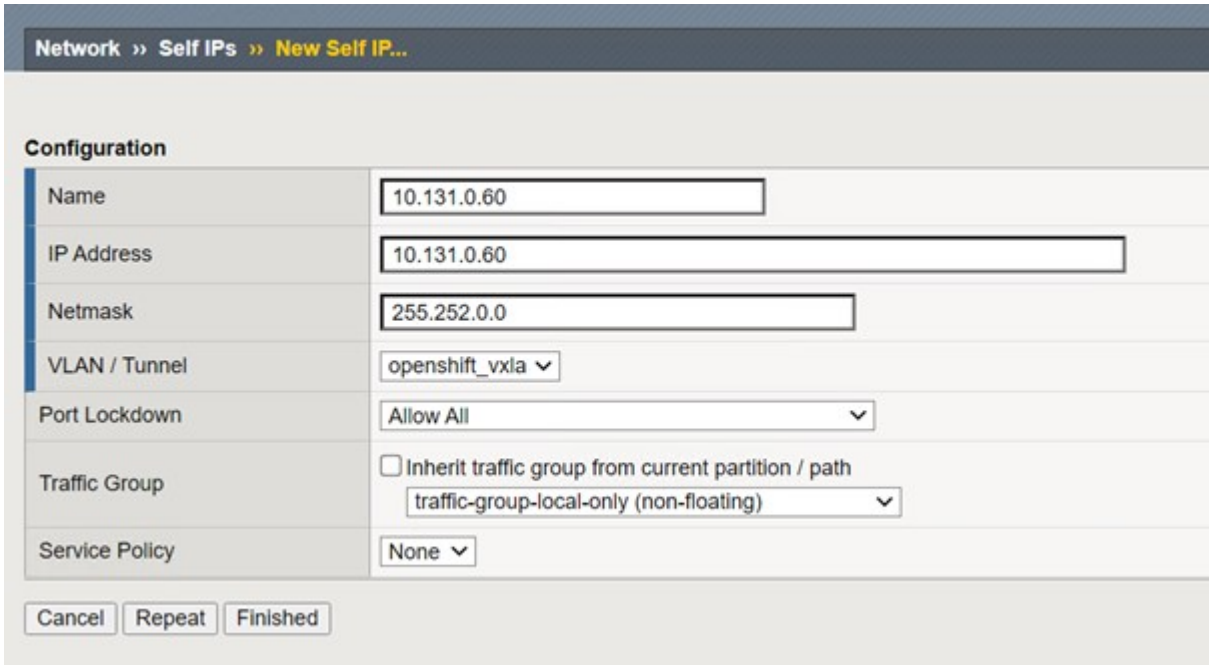
10. 建立主機子網路資源。

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

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

11. 取得為F5 BIG-IP伺服器所建立之主機子網路的叢集IP子網路範圍。

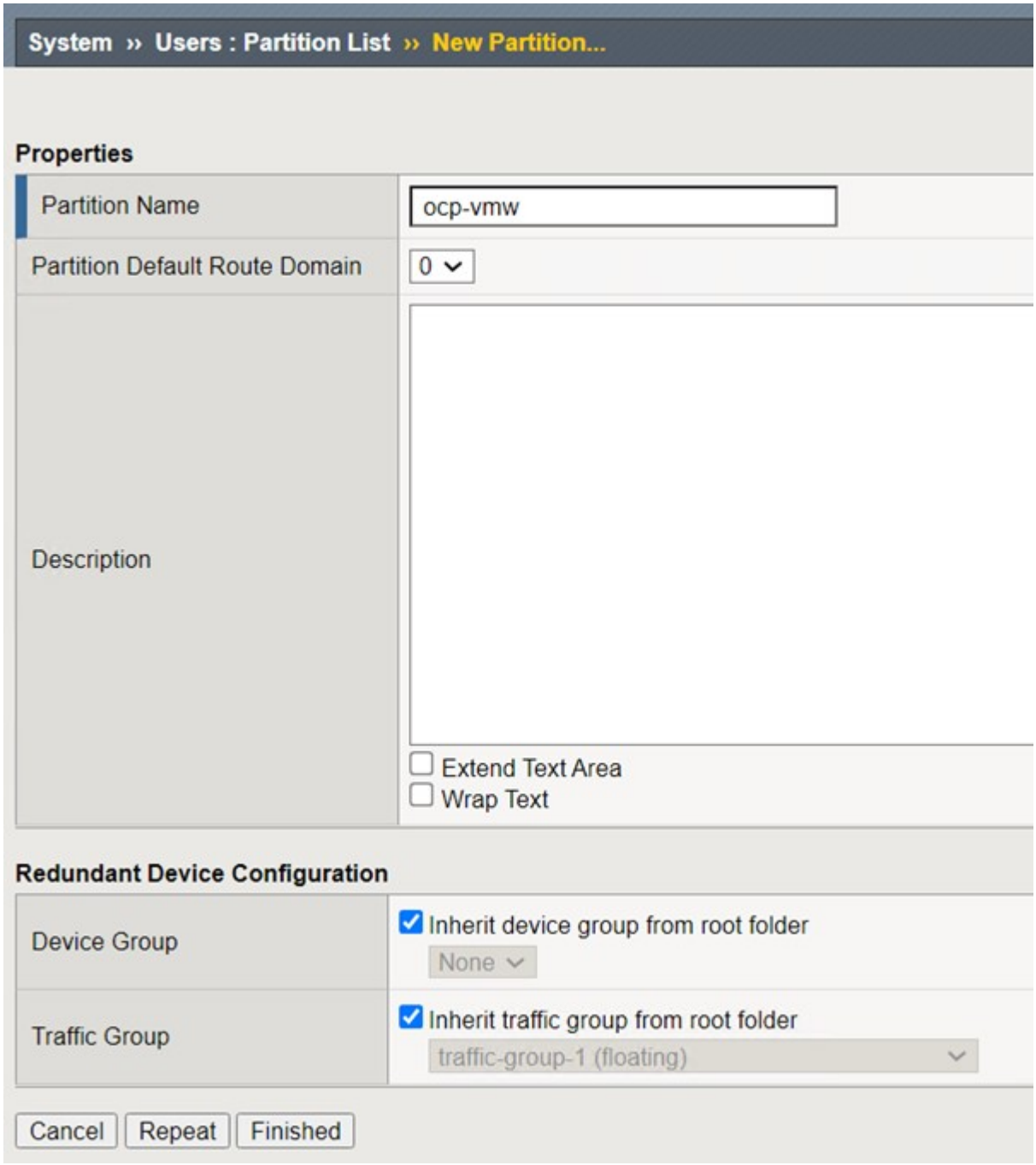
```
[admin@rhel-7 ~]$ oc get hostsubnet

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上建立一個自有IP、並在OpenShift的主機子網路範圍中建立對應於F5 BIG-IP伺服器的IP。登入F5 BIG-IP系統、瀏覽至「網路」>「自助IP」、然後按一下「建立」。從為F5 BIG-IP主機子網路建立的叢集IP子網路輸入IP、選取VXLAN通道、然後輸入其他詳細資料。然後按一下「完成」。



13. 在要設定並搭配CI使用的F5 BIG-IP系統中建立分割區。瀏覽至「系統」>「使用者」>「分割清單」、按一

下「建立」、然後輸入詳細資料。然後按一下「完成」。



F5建議您不要在由CI管理的分割區上進行手動設定。

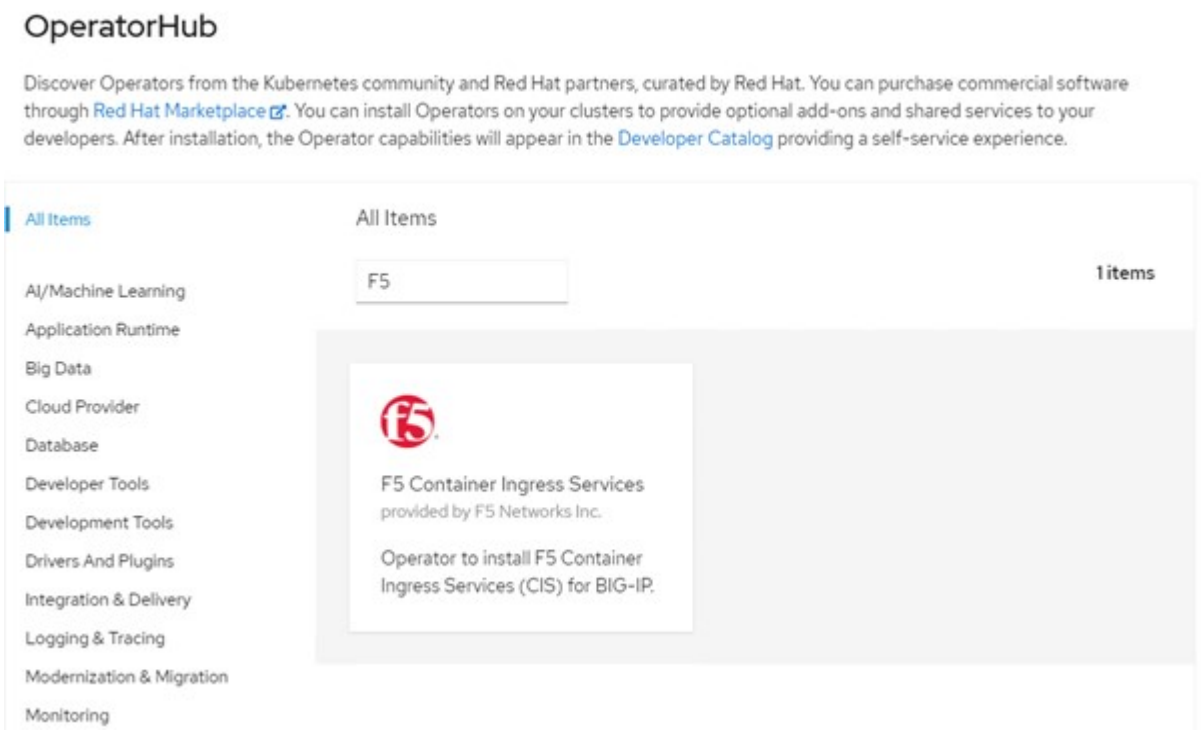
14. 使用來自作業系統集線器的操作員來安裝F5 BIG-IP CI。以叢集管理權限登入Red Hat OpenShift叢集、並使用F5 BIG-IP系統登入認證建立密碼、這是操作員的必要條件。

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

secret/bigip-login created
```

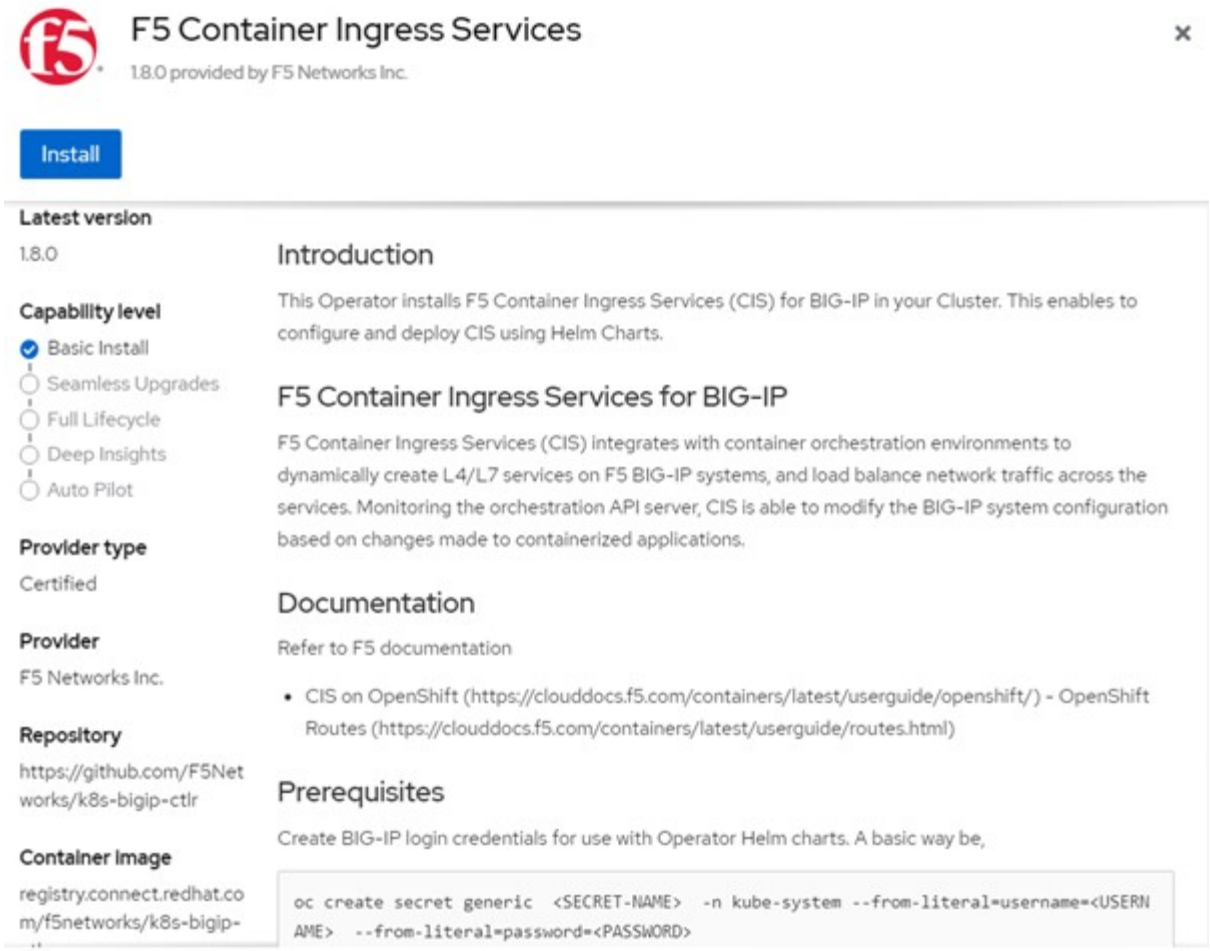
15. 安裝5個CI客戶需求日。

```
[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
```

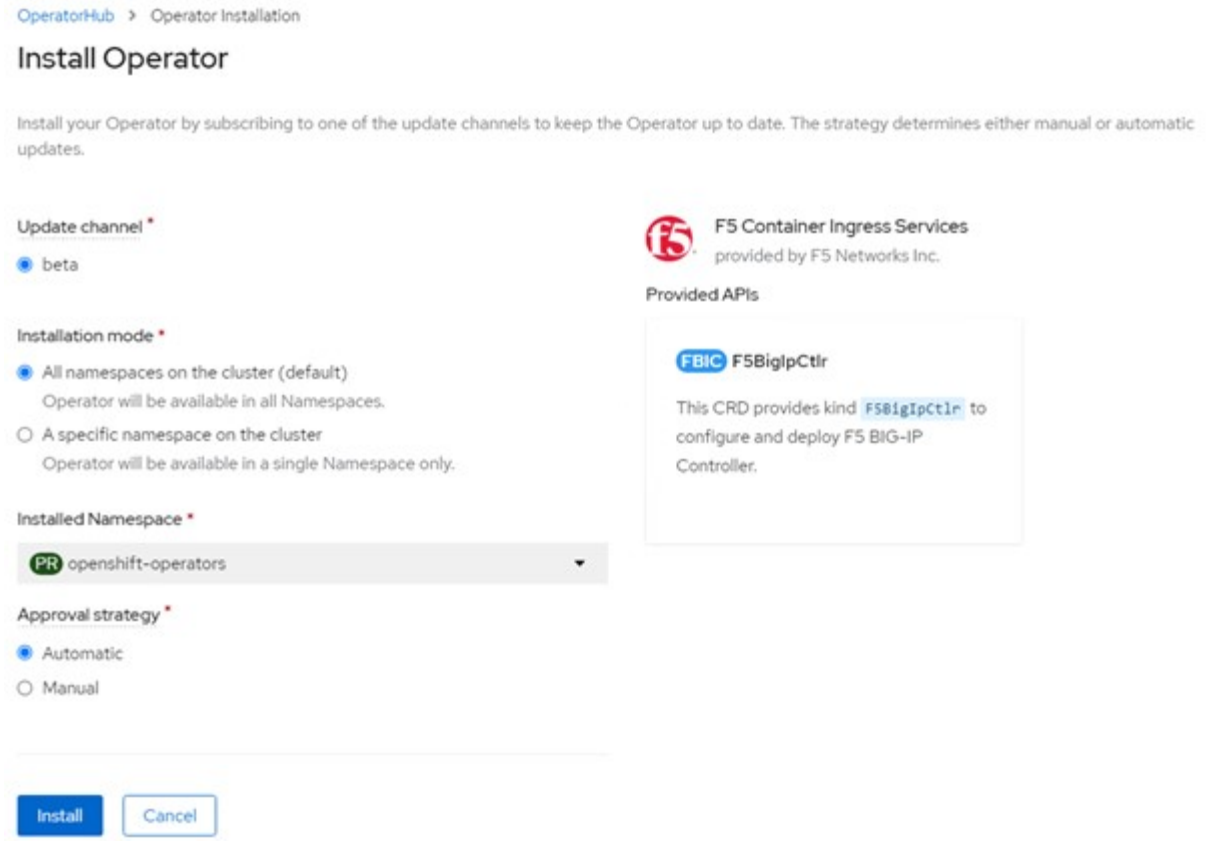
16. 瀏覽至「運算子」>「作業系統集線器」、搜尋關鍵字F5、然後按一下「F5 Container Ingress Service」方塊。



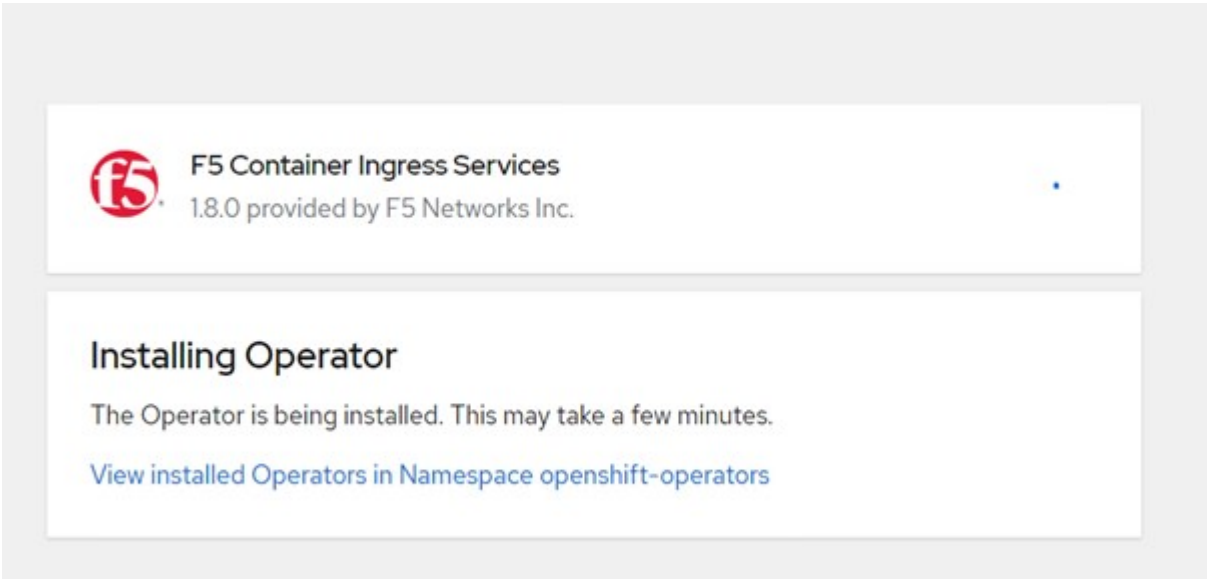
17. 閱讀操作員資訊、然後按一下「Install（安裝）」。



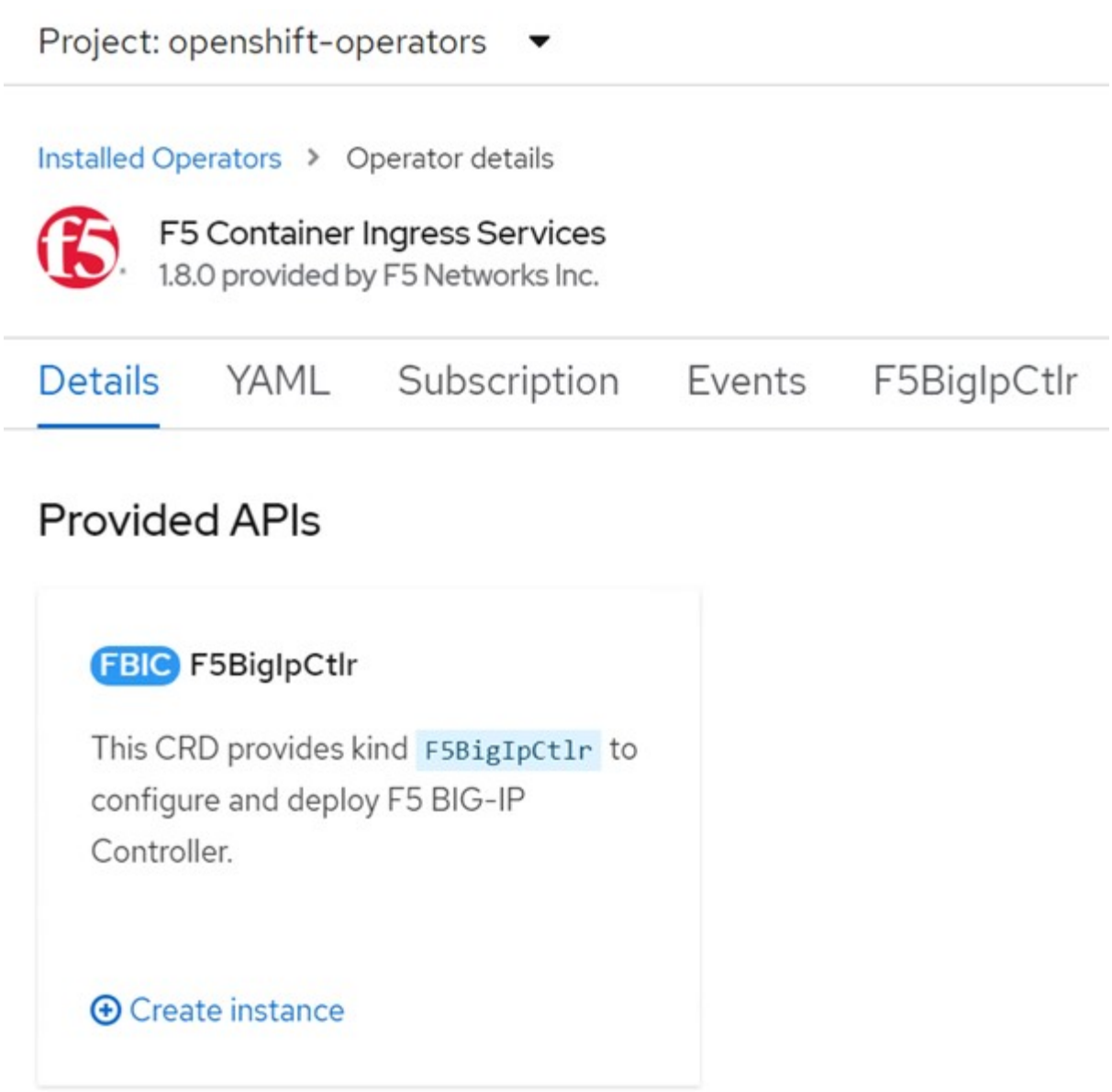
18. 在「Install（安裝）」操作員畫面上、保留所有預設參數、然後按一下「Install（安裝）」。



19. 安裝操作員需要一段時間。



20. 安裝操作員之後、會顯示安裝成功訊息。
21. 瀏覽至「運算子」>「安裝的運算子」、按一下「F5 Container Ingress Service」、然後按一下「F5Bigiprcr」方塊下方的「Create Instance（建立執行個體）」。

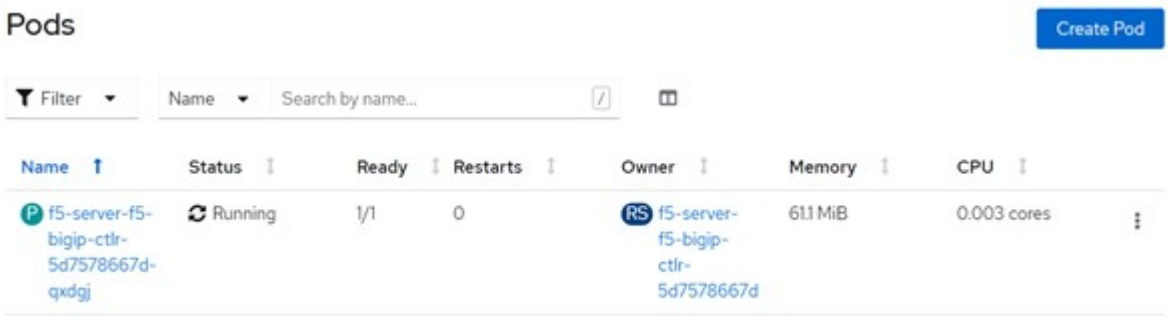


22. 按一下「Yaml View（Yaml檢視）」、然後在更新必要的參數後貼上下列內容。

請更新下列參數「bigip_partition'、「openshift_SDN_name'、「bigip_URL」和「bigip_login_secret」、以反映設定值、然後再複製內容。

```
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. 貼上此內容之後、按一下「建立」。這會在K資料庫 系統命名空間中安裝CI Pod。



Red Hat OpenShift依預設提供一種方法、可透過L7負載平衡的路由來公開服務。內建的OpenShift路由器負責廣告和處理這些路由的流量。不過、您也可以設定F5 CI來支援透過外部的F5 BIG-IP系統的路由、以便作為輔助路由器執行、或取代自行代管的OpenShift路由器。CI會在Big IP系統中建立虛擬伺服器、做為OpenShift路由的路由器、而Big IP則負責通告和流量路由。如需啟用此功能的參數資訊、請參閱此處的文件。請注意、這些參數是針對APS/v1 API中的OpenShift部署資源所定義。因此、將這些項目搭配F5BigIprvtrr資源cis.f5.com/v1 API使用時、請將參數名稱的連字號（-）取代為底線（_）。

24. 傳遞給CI資源建立的引數包括「ipam: true」和「custom_resource_mode：true」。這些參數是啟用與IPAM控制器的CI整合所需的參數。建立F5 IPAM資源、確認CI已啟用IPAM整合。

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

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

25. 建立F5 IPAM控制器所需的服務帳戶、角色和角色繫結。建立Yaml檔案並貼上下列內容。

```
[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. 建立資源。

```
[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. 建立Yaml檔案、然後貼上以下提供的F5 IPAM部署定義。

請更新下方spec.template.spec.contains[0].args中的IP範圍參數、以反映與您設定相對應的ipamLabel和IP位址範圍。

IPAM控制器的負載平衡器類型服務需要註釋ipamLabels ['range1'和'range2'、才能從定義的範圍偵測和指派IP位址。

```
[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. 建立F5 IPAM控制器部署。

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

deployment/f5-ipam-controller created
```

29. 確認F5 IPAM控制器Pod正在執行。

```
[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. 建立F5 IPAM架構。

```
[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
```

驗證

1. 建立負載平衡器類型的服務

```
[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. 檢查IPAM控制器是否指派外部IP給它。

```
[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. 建立部署並使用所建立的負載平衡器服務。

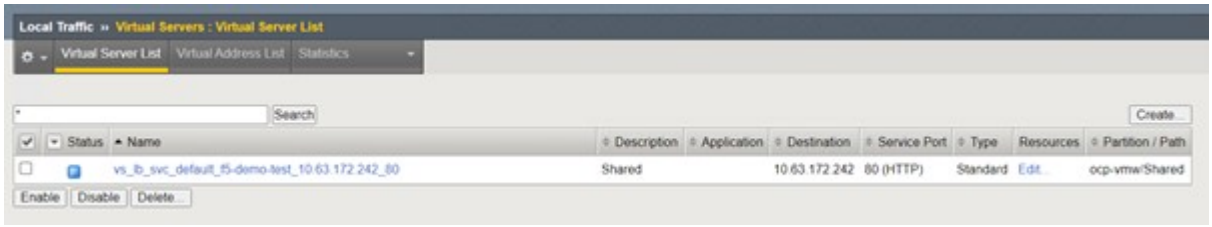
```
[admin@rhel-7 ~]$ vi example_deployment.yaml

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. 檢查Pod是否正在執行。

```
[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. 檢查OpenShift中是否針對負載平衡器類型的服務、在Big IP系統中建立對應的虛擬伺服器。瀏覽至本機流量>虛擬伺服器>虛擬伺服器清單。



建立私有映像登錄

對於大部分的Red Hat OpenShift部署、請使用類似的公用登錄 "Quay.IO" 或 "DockerHub" 滿足大多數客戶的需求。不過有時候客戶可能想要裝載自己的私有或自訂映像。

本程序說明如何建立私有映像登錄、並以Astra Trident和NetApp ONTAP 支援所提供的持續磁碟區作為後盾。

Astra Control Center需要登錄來裝載Astra容器所需的映像。下節說明在Red Hat OpenShift叢集上設定私有登錄的步驟、以及推送支援Astra Control Center安裝所需的映像。

建立私有映像登錄

1. 移除目前預設儲存類別的預設註釋、並在OpenShift叢集的Trident備份儲存類別中註記為預設值。

```
[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「s pec」區段中輸入下列儲存參數、以編輯影像登錄操作員。

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

storage:
  pvc:
    claim:
```

3. 在「最新」區段中輸入下列參數、以建立具有自訂主機名稱的OpenShift路由。儲存並結束。

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

當您想要為路由建立自訂主機名稱時、會使用上述路由組態。如果希望OpenShift使用預設主機名稱來建立路由、您可以將下列參數新增至「預設路由：true」區段。

自訂TLS憑證

當您使用路由的自訂主機名稱時、預設會使用OpenShift Ingress運算子的預設TLS組態。不過、您可以將自訂TLS組態新增至路由。若要這麼做、請完成下列步驟。

a. 使用路由的TLS憑證和金鑰建立秘密。

```
[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. 編輯影像註冊運算子、並將下列參數新增至「s pec」區段。

```
[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. 再次編輯影像註冊業者、並將營運者的管理狀態變更為「老舊」狀態。儲存並結束。

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. 如果滿足所有先決條件、就會為私有映像登錄建立PVCS、Pod和服務。幾分鐘後、登錄就會啟動。

```
[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. 如果您使用入口操作員OpenShift登錄路由的預設TLS憑證、則可以使用下列命令擷取TLS憑證。

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

7. 若要允許OpenShift節點存取及從登錄中提取影像、請將憑證新增至OpenShift節點上的Docker用戶端。使用TLS憑證在「openshift-config」命名空間中建立組態對應、並將其修補至叢集映像組態、使憑證成為信任的憑證。

```
[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內部登錄是由驗證控制。所有OpenShift使用者都能存取OpenShift登錄、但登入使用者可以執行的作業取決於使用者權限。

a. 若要允許使用者或使用者群組從登錄擷取映像、使用者必須指派登錄檢視器角色。

```
[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. 若要允許使用者或使用者群組寫入或推送映像、使用者必須指派登錄編輯器角色。

```
[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. 若要讓OpenShift節點存取登錄並推送或拉出映像、您需要設定拉出密碼。

```
[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. 這種拉出密碼可修補至服務帳戶、或在對應的Pod定義中參考。

a. 若要將IT修補為服務帳戶、請執行下列命令。

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

b. 若要參考Pod定義中的Pull機密、請將下列參數新增至「s pec」區段。

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

11. 若要從工作站推送或拉出OpenShift節點以外的映像、請完成下列步驟。

a. 將TLS憑證新增至Docker用戶端。

```
[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occ登入命令登入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使用者認證登入登錄。

podman

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

+附註：如果您使用「kubeadmin」使用者登入私有登錄、請使用權杖而非密碼。

Docker

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

+附註：如果您使用「kubeadmin」使用者登入私有登錄、請使用權杖而非密碼。

d. 推或拉映像。

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
```