OpenShift Dedicated 4

教程

OpenShift Dedicated 教程

Last Updated: 2025-12-09

OpenShift Dedicated 4 教程

OpenShift Dedicated 教程

Legal Notice

Abstract

管理 OpenShift Dedicated 集群的教程。

Table of Contents

第 1 章 教程概述

使用红帽专家的逐步教程，充分利用您的受管 OpenShift 集群。

重要

此内容由红帽专家编写，但尚未测试每个支持的配置。

第 2 章 教程：使用自定义域和 TLS 证书更新组件路由

本指南演示了如何修改 Google Cloud 版本 4.14 及更高版本上的 OpenShift Dedicated 中的 Web 控制台、OAuth 服务器和下载组件路由的主机名和 TLS 证书。[1]

对组件路由所做的更改[2] 在本指南中，请参阅 自定义内部 OAuth 服务器 URL、自定义控制台路由以及自定义 下载路由 OpenShift Dedicated 文档。

2.1. 先决条件

- OCM CLI (**ocm**)版本 1.0.5 或更高版本
- gcloud CLI (**gcloud**)
- Google Cloud 集群版本 4.14 或更高版本上的 OpenShift Dedicated
- OpenShift CLI (**oc**)
- **jq** CLI
- 使用具有 **cluster-admin** 角色的用户访问集群。
- openssl （用于生成演示 SSL/TLS 证书）

2.2. 设置您的环境

1. 使用具有 **cluster-admin** 权限的账户登录集群。
2. 为集群名称配置环境变量：

```
$ export CLUSTER_NAME=$(oc get infrastructure cluster -o=jsonpath="
{.status.infrastructureName}"  | sed 's/-[a-z0-9]\{5\}$//')
```

3. 在移至下一部分前，确保所有字段都正确输出：

```
$ echo "Cluster: ${CLUSTER_NAME}"
```

输出示例

```
Cluster: my-osd-cluster
```

2.3. 查找当前路由

1. 验证您是否可以访问其默认主机名上的组件路由。
   您可以通过查询 **openshift-console** 和 **openshift-authentication** 项目中的路由列表来查找主机名。

```
$ oc get routes -n openshift-console
$ oc get routes -n openshift-authentication
```

输出示例

```
NAME        HOST/PORT                                                    PATH  SERVICES
PORT    TERMINATION          WILDCARD
console     console-openshift-console.apps.my-example-cluster-
gcp.z9a9.p2.openshiftapps.com    ... 1 more  console    https   reencrypt/Redirect   None
downloads   downloads-openshift-console.apps.my-example-cluster-
gcp.z9a9.p2.openshiftapps.com  ... 1 more  downloads  http    edge/Redirect        None
NAME            HOST/PORT                                                  PATH  SERVICES
PORT    TERMINATION          WILDCARD
oauth-openshift   oauth-openshift.apps.my-example-cluster-gcp.z9a9.p2.openshiftapps.com
... 1 more  oauth-openshift   6443   passthrough/Redirect   None
```

在这个输出中，您可以看到我们的基本主机名为 **z9a9.p2.openshiftapps.com**。

2. 运行以下命令，获取默认入口的 ID：

```
$ export INGRESS_ID=$(ocm list ingress -c ${CLUSTER_NAME} -o json | jq -r '.[] |
select(.default == true) | .id')
```

3. 在移至下一部分前，确保所有字段都正确输出：

```
$ echo "Ingress ID: ${INGRESS_ID}"
```

输出示例

```
Ingress ID: r3l6
```

通过运行这些命令，您可以看到集群的默认组件路由是：

- **console-openshift-console.apps.my-example-cluster-gcp.z9a9.p2.openshiftapps.com** for Console
- **downloads-openshift-console.apps.my-example-cluster-gcp.z9a9.p2.openshiftapps.com** for Downloads
- **oauth-openshift.apps.my-example-cluster-gcp.z9a9.p2.openshiftapps.com** for OAuth

我们可以使用 **ocm edit ingress** 命令更改每个服务的主机名，并为我们的所有组件路由添加一个 TLS 证书。**ocm edit ingress** 命令的命令行帮助摘录中显示了相关的参数：

```
$ ocm edit ingress -h
Edit a cluster ingress for a cluster. Usage:
  ocm edit ingress ID [flags]
  [...]
  --component-routes string                Component routes settings. Available keys [oauth, console,
downloads]. For each key a pair of hostname and tlsSecretRef is expected to be supplied. Format
should be a comma separate list 'oauth: hostname=example-hostname;tlsSecretRef=example-secret-
ref,downloads:...'
```

在本例中，我们将使用以下自定义组件路由：

- Console 的 **console.my-new-domain.dev**
- **download.my-new-domain.dev** for Downloads

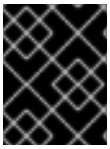
- **oauth.my-new-domain.dev** for OAuth

2.4. 为每个组件路由创建有效的 TLS 证书

在本节中，我们创建三个单独的自签名证书密钥对，然后信任它们，以使用真实的 Web 浏览器访问我们的新组件路由。

警告

这仅用于演示目的，不建议将其作为生产工作负载的解决方案。请参考您的证书颁发机构了解如何为生产工作负载创建具有类似属性的证书。



重要

要防止 HTTP/2 连接合并出现问题，您必须为每个端点使用单独的证书。不支持使用通配符或 SAN 证书。

- 为每个组件路由生成一个证书，注意将证书的主题(**-subj**)设置为您要使用的组件路由的自定义域：

  Example

  ```
  $ openssl req -newkey rsa:2048 -new -nodes -x509 -days 365 -keyout key-console.pem -out cert-console.pem -subj "/CN=console.my-new-domain.dev"
  $ openssl req -newkey rsa:2048 -new -nodes -x509 -days 365 -keyout key-downloads.pem -out cert-downloads.pem -subj "/CN=downloads.my-new-domain.dev"
  $ openssl req -newkey rsa:2048 -new -nodes -x509 -days 365 -keyout key-oauth.pem -out cert-oauth.pem -subj "/CN=oauth.my-new-domain.dev"
  ```

  这会生成三对 **.pem** 文件、**key-<component>.pem** 和 **cert-<component>.pem**。

2.5. 将证书作为 SECRET 添加到集群中

- 在 **openshift-config** 命名空间中创建三个 TLS secret。
  在本指南的稍后更新组件路由时，这些 secret 将成为您的 secret 引用。

  ```
  $ oc create secret tls console-tls --cert=cert-console.pem --key=key-console.pem -n openshift-config
  $ oc create secret tls downloads-tls --cert=cert-downloads.pem --key=key-downloads.pem -n openshift-config
  $ oc create secret tls oauth-tls --cert=cert-oauth.pem --key=key-oauth.pem -n openshift-config
  ```

2.6. 在集群中查找负载均衡器的负载均衡器 IP

当您创建集群时，该服务会创建一个负载均衡器，并为该负载均衡器生成负载均衡器 IP。为了为集群创建 DNS 记录，我们需要知道负载均衡器 IP。

您可以通过针对 **openshift-ingress** 命名空间运行 **oc get svc** 命令来查找负载均衡器 IP。负载均衡器的负载均衡器 IP 是与 **openshift-ingress** 命名空间中的 **router-default** 服务关联的 **EXTERNAL-IP**。

```
$ oc get svc -n openshift-ingress
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)                      AGE
router-default LoadBalancer  172.30.237.88  34.85.169.230     80:31175/TCP,443:31554/TCP  76d
```

在我们的情形中，负载均衡器 IP 为 **34.85.169.230**。

稍后保存这个值，因为我们需要为新组件路由主机名配置 DNS 记录。

2.7. 在托管供应商中添加组件路由 DNS 记录

在 DNS 设置中创建 A 记录，将域指向 router-default 的负载均衡器的 IP 地址。

2.8. 使用 OCM CLI 更新组件路由和 TLS SECRET

更新 DNS 记录后，您可以使用 OCM CLI 更改组件路由。

1. 使用 **ocm edit ingress** 命令，使用新的基域和与其关联的 secret 引用更新您的默认入口路由，需要小心更新每个组件路由的主机名。

```
$ ocm edit ingress -c ${CLUSTER_NAME} ${INGRESS_ID} --component-routes 'console:
hostname=console.my-new-domain.dev;tlsSecretRef=console-tls,downloads:
hostname=downloads.my-new-domain.dev;tlsSecretRef=downloads-tls,oauth:
hostname=oauth.my-new-domain.dev;tlsSecretRef=oauth-tls'
```

注意

您还可以通过保留您不想更改设置为空字符串的组件路由来只编辑组件路由的子集。例如，如果您只想更改 Console 和 OAuth 服务器主机名和 TLS 证书，您将运行以下命令：

```
$ ocm edit ingress -c ${CLUSTER_NAME} ${INGRESS_ID} --component-
routes 'console: hostname=console.my-new-
domain.dev;tlsSecretRef=console-tls,downloads:
hostname="";tlsSecretRef="", oauth: hostname=oauth.my-new-
domain.dev;tlsSecretRef=oauth-tls'
```

2. 运行 **ocm list ingress** 命令来验证您的更改是否成功：

```
$ ocm list ingress -c ${CLUSTER_NAME} -ojson | jq ".[] | select(.id == \"${INGRESS_ID}\") |
.component_routes"
```

输出示例

```
{
  "console": {
    "kind": "ComponentRoute",
    "hostname": "console.my-new-domain.dev",
    "tls_secret_ref": "console-tls"
  },
```

```
  "downloads": {
    "kind": "ComponentRoute",
    "hostname": "downloads.my-new-domain.dev",
    "tls_secret_ref": "downloads-tls"
  },
  "oauth": {
    "kind": "ComponentRoute",
    "hostname": "oauth.my-new-domain.dev",
    "tls_secret_ref": "oauth-tls"
  }
}
```

3. 将您的证书添加到本地系统上的信任存储中，然后确认您可以使用本地 Web 浏览器通过新路由访问组件。

2.9. 使用 OCM CLI 将组件路由重置为默认值

如果要将组件路由重置为默认配置，请运行以下 **ocm edit ingress** 命令：

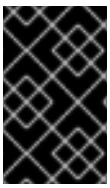
```
$ ocm edit ingress -c ${CLUSTER_NAME} ${INGRESS_ID} --component-routes 'console:
hostname="";tlsSecretRef="",downloads: hostname="";tlsSecretRef="", oauth:
hostname="";tlsSecretRef=""'
```

[1] 通常不支持在 4.14 之前的 OpenShift Dedicated OCM 版本上修改这些路由。但是，如果您有一个使用版本 4.13 的集群，可以通过 提交支持问题单，在版本 4.13 集群上启用对此功能的支持。

[2] 我们使用术语"组件路由"来指代首次安装 OCM 时提供的 OAuth、控制台和下载路由。

第 3 章 教程：使用 GOOGLE CLOUD NEXT 代防火墙限制出口

使用本指南，使用 Google Cloud 的 Next Generation Firewall (NGFW)在 Google Cloud 上为 OpenShift Dedicated 实现出口限制。NGFW 是一个完全分布式的防火墙服务，它允许防火墙规则中完全限定域名(FQDN)对象。对于 OpenShift Dedicated 依赖的许多外部端点，这是必需的。



重要

使用防火墙或其他网络设备限制出口流量的功能仅支持使用私有 Service Connect (PSC)部署的 OpenShift Dedicated 集群。不使用 PSC 的集群需要支持例外才能使用此功能。如需其他帮助，请创建一个支持问题单。

3.1. 查看您的先决条件

- 已安装 Google Cloud 命令行界面(**gcloud**)。
- 您已登录到 Google Cloud CLI，并选择您要部署 OpenShift Dedicated 的 Google Cloud 项目。
- 在 Google Cloud 中具有最低所需的权限，包括：
  - **Compute Network Admin**
  - **DNS Administrator**
- 您已通过在终端中运行以下命令来启用某些服务：

```
$ gcloud services enable networksecurity.googleapis.com
```

```
$ gcloud services enable networkservices.googleapis.com
```

```
$ gcloud services enable servicenetworking.googleapis.com
```

3.2. 设置您的环境

在终端中，配置以下环境变量：

```
export project_id=$(gcloud config list --format="value(core.project)")
export region=us-east1
export prefix=osd-ngfw
export service_cidr="172.30.0.0/16"
export machine_cidr="10.0.0.0/22"
export pod_cidr="10.128.0.0/14"
```

本例使用 **us-east1** 作为区域，为集群资源部署 **osd-ngfw** 前缀。为服务和 pod 网络分配默认 CIDR 范围。机器 CIDR 基于子网范围，将在本教程中稍后设置。修改参数以满足您的需要。

3.3. 创建 VPC 和子网

在部署 Google Cloud NGFW 之前，您必须首先创建用于 OpenShift Dedicated 的 Virtual Private Cloud (VPC)和子网：

1. 运行以下命令来创建 VPC：

```
$ gcloud compute networks create ${prefix}-vpc --subnet-mode=custom
```

2. 运行以下命令来创建 worker 子网：

```
$ gcloud compute networks subnets create ${prefix}-worker \
    --range=10.0.2.0/23 \
    --network=${prefix}-vpc \
    --region=${region} \
    --enable-private-ip-google-access
```

3. 运行以下命令来创建 control plane 子网：

```
$ gcloud compute networks subnets create ${prefix}-control-plane \
    --range=10.0.0.0/25 \
    --network=${prefix}-vpc \
    --region=${region} \
    --enable-private-ip-google-access
```

4. 运行以下命令来创建 PSC 子网：

```
$ gcloud compute networks subnets create ${prefix}-psc \
    --network=${prefix}-vpc \
    --region=${region} \
    --stack-type=IPV4_ONLY \
    --range=10.0.0.128/29 \
    --purpose=PRIVATE_SERVICE_CONNECT
```

这些示例使用 worker 子网为 10.0.2.0/23 的子网范围，control plane 子网为 10.0.0.0/25，为 PSC 子网使用 10.0.0.128/29。修改参数以满足您的需要。确保参数值包含在本教程前面设置的机器 CIDR 中。

3.4. 部署全局网络防火墙策略

1. 运行以下命令来创建全局网络防火墙策略：

```
$ gcloud compute network-firewall-policies create \
    ${prefix} \
    --description "OpenShift Dedicated Egress Firewall" \
    --global
```

2. 运行以下命令，将新创建的全局网络防火墙策略与上面创建的 VPC 关联：

```
$ gcloud compute network-firewall-policies associations create \
    --firewall-policy ${prefix} \
    --network ${prefix}-vpc \
    --global-firewall-policy
```

3.5. 创建云路由器和云网络地址转换网关

网络地址转换(NAT)网关通过伪装单个公共 IP 地址下的所有流量，为您的私有虚拟机启用互联网连接。在指定的退出点中，它为任何出站请求（如获取更新）转换其内部 IP。这个过程有效地允许他们访问互联网，而无需公开其专用地址。

1. 运行以下命令，为 Cloud NAT 保留 IP 地址：

```
$ gcloud compute addresses create ${prefix}-${region}-cloudnatip \
    --region=${region}
```

2. 运行以下命令来创建云路由器：

```
$ gcloud compute routers create ${prefix}-router \
    --region=${region} \
    --network=${prefix}-vpc
```

3. 运行以下命令来创建 Cloud NAT：

```
$ gcloud compute routers nats create ${prefix}-cloudnat-${region} \
    --router=${prefix}-router --router-region ${region} \
    --nat-all-subnet-ip-ranges \
    --nat-external-ip-pool=${prefix}-${region}-cloudnatip
```

3.6. 为私有 GOOGLE 访问创建私有域名系统记录

私有域名系统(DNS)区通过确保流量永远不会通过公共互联网来优化您的资源连接到 Google API 的方式。它的功能通过截获 Google 服务的 DNS 请求，并将它们解析为私有 IP 地址，强制连接到 Google 的内部网络，以便更快地进行数据交换。

1. 运行以下命令，为 googleapis.com 域创建私有 DNS 区域：

```
$ gcloud dns managed-zones create ${prefix}-googleapis \
    --visibility=private \
    --
networks=https://www.googleapis.com/compute/v1/projects/${project_id}/global/networks/${prefix}-vpc \
    --description="Private Google Access" \
    --dns-name=googleapis.com
```

2. 运行以下命令开始记录设置事务：

```
$ gcloud dns record-sets transaction start \
    --zone=${prefix}-googleapis
```

3. 运行以下命令，将 Google API 的 DNS 记录暂存在 googleapis.com 域下：

```
$ gcloud dns record-sets transaction add --name="*.googleapis.com." \
    --type=CNAME restricted.googleapis.com. \
    --zone=${prefix}-googleapis \
    --ttl=300
```

```
$ gcloud dns record-sets transaction add 199.36.153.4 199.36.153.5 199.36.153.6
199.36.153.7 \
    --name=restricted.googleapis.com. \
    --type=A \
    --zone=${prefix}-googleapis \
    --ttl=300
```

4. 运行以下命令应用上面启动的暂存记录设置事务：

```
$ gcloud dns record-sets transaction execute \
    --zone=$prefix-googleapis
```

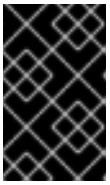
3.7. 创建防火墙规则

1. 运行以下命令，为私有 IP (RFC 1918)地址空间创建一个空白允许规则：

```
$ gcloud compute network-firewall-policies rules create 500 \
    --description "Allow egress to private IP ranges" \
    --action=allow \
    --firewall-policy=${prefix} \
    --global-firewall-policy \
    --direction=EGRESS \
    --layer4-configs all \
    --dest-ip-ranges=10.0.0.0/8,172.16.0.0/12,192.168.0.0/16
```

2. 运行以下命令，为 OpenShift Dedicated 所需的 HTTPS (tcp/443)域创建一个允许规则：

```
$ gcloud compute network-firewall-policies rules create 600 \
    --description "Allow egress to OpenShift Dedicated required domains (tcp/443)" \
    --action=allow \
    --firewall-policy=${prefix} \
    --global-firewall-policy \
    --direction=EGRESS \
    --layer4-configs tcp:443 \
    --dest-fqdns accounts.google.com,pull.q1w2.quay.rhcloud.com,http-inputs-
osdsecuritylogs.splunkcloud.com,nosnch.in,api.deadmanssnitch.com,events.pagerduty.com,api
pagerduty.com,api.openshift.com,mirror.openshift.com,observatorium.api.openshift.com,observ
atorium-
mst.api.openshift.com,console.redhat.com,infogw.api.openshift.com,api.access.redhat.com,cert
api.access.redhat.com,catalog.redhat.com,sso.redhat.com,registry.connect.redhat.com,registry.
access.redhat.com,cdn01.quay.io,cdn02.quay.io,cdn03.quay.io,cdn04.quay.io,cdn05.quay.io,cd
n06.quay.io,cdn.quay.io,quay.io,registry.redhat.io,quayio-production-s3.s3.amazonaws.com
```



重要

如果没有允许流量的匹配规则，则防火墙会阻断它。要允许访问其他资源，如内部网络或其他外部端点，请创建优先级小于 1000 的额外规则。有关如何创建防火墙规则的更多信息，请参阅 使用全局网络防火墙策略和规则。

3.8. 创建集群

现在，您可以在 Google Cloud 集群上创建 OpenShift Dedicated。如需更多信息，请参阅使用 Workload Identity Federation 身份验证在 Google Cloud 上创建集群。

3.9. 删除集群

要删除集群，请参阅 删除 Google Cloud 上的 OpenShift Dedicated 集群 。

3.10. 清理资源

为了防止持续收费，在删除集群后，您必须手动删除您创建的 Google Cloud 网络基础架构。删除集群不会自动删除这些底层资源。您可以使用 gcloud CLI 命令和 Google Cloud 控制台中的操作的组合来清理这些资源。

在开始清理本教程创建的资源的过程前，运行以下命令并完成任何提示。

1. 要验证您的身份，请运行以下命令：

```
$ gcloud init
```

2. 要登录到 Google Cloud 帐户，请运行以下命令：

```
$ gcloud auth application-default login
```

3. 要登录到 OpenShift Cluster manager CLI 工具，请运行以下命令：

```
$ ocm login --use-auth-code
```

现在，您可以清理在本教程中创建的资源。要遵守资源依赖项，以相反的顺序删除它们。

1. 运行以下命令来删除防火墙策略与 VPC 的关联：

```
$ gcloud compute network-firewall-policies associations delete \
    --firewall-policy=${prefix} \
    --network=${prefix}-vpc \
    --global-firewall-policy
```

2. 运行以下命令来删除全局网络防火墙策略：

```
$ gcloud compute network-firewall-policies delete ${prefix} --global
```

3. 在删除所有用户定义的记录集前，Google Cloud 中的受管 DNS 区域无法被删除。运行以下命令，定义用于目标特定 Google Cloud 项目和清理受管 DNS 区域的变量：

```
$ cat /tmp/delete_records.sh
PROJECT_ID=<your-project-id>
ZONE_NAME=<your-managed-zone-name>
```

4. 运行以下命令，列出 Private DNS 区域中包含的记录集：

```
$ gcloud \
    dns record-sets list \
    --project=$PROJECT_ID \
    --zone=$ZONE_NAME \
    --filter="type!=NS AND type!=SOA" \
    --format="value(name,type)" | while read name type;
```

5. 运行以下命令，删除该私有 DNS 区域中包含的记录集：

```
$ gcloud --project=$PROJECT_ID dns record-sets delete "$name" --zone=$ZONE_NAME --
type="$type"
```

6. 运行以下命令来删除 Private DNS 区域：

```
$ gcloud dns managed-zones delete ${prefix}-googleapis
```

7. 删除 Cloud NAT 网关：

```
$ gcloud compute routers nats delete ${prefix}-cloudnat-${region} \
    --router=${prefix}-router \
    --router-region=${region}
```

8. 运行以下命令来删除云路由器：

```
$ gcloud compute routers delete ${prefix}-router --region=${region}
```

9. 运行以下命令来删除保留的 IP 地址：

```
$ gcloud compute addresses delete ${prefix}-${region}-cloudnatip --region=${region}
```

10. 运行以下命令来删除 worker 子网：

```
$ gcloud compute networks subnets delete ${prefix}-worker --region=${region}
```

11. 运行以下命令来删除 control plane 子网：

```
$ gcloud compute networks subnets delete ${prefix}-control-plane --region=${region}
```

12. 运行以下命令来删除 PSC 子网：

```
$ gcloud compute networks subnets delete ${prefix}-psc --region=${region}
```

13. 运行以下命令来删除 VPC：

```
$ gcloud compute networks delete ${prefix}-vpc
```