Red Hat Advanced Cluster Management for Kubernetes 2.10

网络

网络

Last Updated: 2024-07-01

Red Hat Advanced Cluster Management for Kubernetes 2.10 网络

网络

法律通告

摘要

了解更多有关 hub 集群和受管集群的网络要求。

目录

第 1 章 网络

了解 hub 集群和受管集群的网络要求。

- hub 集群网络配置
- 受管集群网络配置
- 高级网络配置
- Submariner multicluster networking 和 service discovery

1.1. HUB 集群网络配置

重要：可信 CA 捆绑包在 Red Hat Advanced Cluster Management 命名空间中可用，但该增强需要更改您的网络。可信 CA 捆绑包 ConfigMap 使用 **trusted-ca-bundle** 的默认名称。您可以通过在名为 **TRUSTED_CA_BUNDLE** 的环境变量中提供 Operator 来更改此名称。如需更多信息，请参阅 Red Hat OpenShift Container Platform 的网络部分中的配置集群范围代理。

您可以引用 hub 集群网络的配置。

1.1.1. hub 集群网络配置表

请参阅下表中的 hub 集群网络要求：

方向	协议	连接	端口（如果指定）	源地址	目标地址
出站到受管集群	HTTPS	从搜索控制台为受管集群的 pod 动态检索日志，使用受管集群中运行的 **klusterlet-addon-workmgr** 服务	443	无	用于访问受管集群路由的 IP 地址
出站到受管集群	HTTPS	安装过程中置备的受管集群的 Kubernetes API 服务器来安装 klusterlet	6443	无	Kubernetes 受管集群 API 服务器的 IP
到频道源的外向流量	HTTPS	频道源，包括 GitHub、Object Store 和 Helm 仓库，只有在您使用应用程序生命周期、OpenShift GitOps 或 Argo CD 时才需要它	443	无	频道源的 IP

方向	协议	连接	端口（如果指定）	源地址	目标地址
来自受管集群的内向流量	HTTPS	用于推送只为运行 OpenShift Container Platform 版本 4.13 或更高版本的受管集群收集的指标和警报的受管集群	443	无	hub 集群访问路由的 IP 地址
来自受管集群的内向流量	HTTPS	监视受管集群的 Kubernetes API 服务器，用于监视受管集群的更改	6443	无	hub 集群 Kubernetes API 服务器的 IP 地址
出站到 ObjectStore	HTTPS	当 Cluster Backup Operator 运行时，为长期存储发送 Observability 指标数据	443	无	ObjectStore 的 IP 地址
出站到镜像存储库	HTTPS	访问 OpenShift Container Platform 和 Red Hat Advanced Cluster Management 的镜像	443	无	镜像存储库的 IP 地址

1.2. 受管集群网络配置

您可以引用受管集群网络的配置。

1.2.1. 受管集群网络配置表

下表中查看受管集群网络要求：

方向	协议	连接	端口（如果指定）	源地址	目标地址
来自 hub 集群的内向流量	HTTPS	从搜索控制台为受管集群的 pod 动态发送日志，使用受管集群中运行的 **klusterlet-addon-workmgr** 服务	443	无	用于访问受管集群路由的 IP 地址
来自 hub 集群的内向流量	HTTPS	安装过程中置备的受管集群的 Kubernetes API 服务器来安装 klusterlet	6443	无	Kubernetes 受管集群 API 服务器的 IP
出站到镜像存储库	HTTPS	访问 OpenShift Container Platform 和 Red Hat Advanced Cluster Management 的镜像	443	无	镜像存储库的 IP 地址
到 hub 集群的外向流量	HTTPS	用于推送只为运行 OpenShift Container Platform 版本 4.13 或更高版本的受管集群收集的指标和警报的受管集群	443	无	hub 集群访问路由的 IP 地址
到 hub 集群的外向流量	HTTPS	监视 hub 集群的 Kubernetes API 服务器的变化	6443	无	hub 集群 Kubernetes API 服务器的 IP 地址

方向	协议	连接	端口（如果指定）	源地址	目标地址
到频道源的外向流量	HTTPS	频道源，包括 GitHub、Object Store 和 Helm 仓库，只有在您使用应用程序生命周期、OpenShift GitOps 或 Argo CD 时才需要它	443	无	频道源的 IP

1.3. 高级网络配置

- *基础架构 operator 表的额外网络要求*
- *Submariner 网络要求表*
- *Hive 表的额外网络要求*
- *托管 control planes 网络要求表（技术预览）*
- *应用程序部署网络要求表*
- *命名空间连接网络要求表*

1.3.1. 基础架构 operator 表的额外网络要求

当使用 Infrastructure Operator 安装裸机受管集群时，请参阅 multicluster engine for Kubernetes operator 文档中的 网络配置 以了解额外网络要求。

1.3.2. Submariner 网络要求表

使用 Submariner 的集群需要三个打开的端口。下表显示了您可以使用的端口：

方向	协议	连接	端口（如果指定）
出站和入站	UDP	每个受管集群	4800
出站和入站	UDP	每个受管集群	4500、500 以及网关节点上 IPSec 流量的任何其他端口
入站	TCP	每个受管集群	8080

1.3.3. Hive 表的额外网络要求

*当使用 Hive Operator 安装裸机受管集群（包括使用中央基础架构管理）时，您必须在 hub 集群和 **libvirt** 置备主机间配置第 2 层或第 3 层端口连接。在使用 Hive 创建基本集群的过程中，需要它们来与置备主机进行连接。如需更多信息，请参阅下表：*

方向	协议	连接	端口（如果指定）
到 **libvirt** 置备主机的 hub 集群的内向和外向流量	IP	将 hub 集群（Hive operator 安装的位置）连接到 **libvirt** 置备主机（在创建裸机集群时作为一个 bootstrap）	

***注：** 这些要求只适用于安装时，在升级使用 Infrastructure Operator 安装的集群时不需要。*

1.3.4. 托管 control planes 网络要求表（技术预览）

*使用托管的 control plane 时，**HypershiftDeployment** 资源必须具有与下表中列出的端点的连接：*

方向	连接	端口（如果指定）
出站	OpenShift Container Platform control-plane 和 worker 节点	
出站	仅限 Amazon Web Services 上的托管集群：到 AWS API 和 S3 API 的出站连接	
出站	对于 Microsoft Azure 云服务上的托管集群：到 Azure API 的出站连接	
出站	OpenShift Container Platform 镜像存储库，用于存储 coreOS 的 ISO 镜像和 OpenShift Container Platform pod 的镜像 registry	
出站	托管集群中 klusterlet 的本地 API 客户端与 HyperShift 托管集群的 API 通信	

1.3.5. 应用程序部署网络要求表

*通常，应用程序部署通信是从受管集群到 hub 集群的一种方法。连接使用 **kubeconfig**，后者由受管集群上的代理配置。受管集群中的应用程序部署需要访问 hub 集群中的以下命名空间：*

- *频道资源的命名空间*
- *受管集群的命名空间*

1.3.6. 命名空间连接网络要求表

- 应用程序生命周期连接：
  - 命名空间 **open-cluster-management** 需要访问端口 4000 上的控制台 API。
  - 命名空间 **open-cluster-management** 需要在端口 3001 上公开 Application UI。
- 应用程序生命周期后端组件(pod)：
  在 hub 集群中，所有应用程序生命周期 pod 都安装在 **open-cluster-management** 命名空间中，包括以下 pod：
  - multicluster-operators-hub-subscription
  - multicluster-operators-standalone-subscription
  - multicluster-operators-channel
  - multicluster-operators-application
  - multicluster-integrations
    由于这些 pod 位于 **open-cluster-management** 命名空间中：
  - 命名空间 **open-cluster-management** 需要通过端口 6443 访问 Kube API。

  在受管集群中，只有 **klusterlet-addon-appmgr** 应用程序生命周期 pod 安装在 **open-cluster-management-agent-addon** 命名空间中：
  - 命名空间 **open-cluster-management-agent-addon** 需要通过端口 6443 访问 Kube API。
- 监管和风险：
  在 hub 集群中，需要以下访问权限：
  - 命名空间 **open-cluster-management** 需要通过端口 6443 访问 Kube API。
  - 命名空间 **open-cluster-management** 需要访问端口 5353 上的 OpenShift DNS。

  在受管集群中，需要以下访问权限：
  - 命名空间 **open-cluster-management-addon** 需要通过端口 6443 访问 Kube API。

1.4. SUBMARINER MULTICLUSTER NETWORKING 和 SERVICE DISCOVERY

Submariner 是一个开源工具，可与 Red Hat Advanced Cluster Management for Kubernetes 一起使用，用来在您的环境中（内部环境或云中）在两个或多个受管集群之间提供直接联网和服务发现。Submariner 与 Multi-Cluster Services API (Kubernetes 增强 Proposal #1645) 兼容。有关 Submariner 的更多信息，请参阅 Submariner 站点。

如需了解有关基础架构供应商支持级别的更多详细信息，请参阅 Red Hat Advanced Cluster Management 支持列表，包括哪些供应商支持 自动控制台部署 或需要 手动部署。

请参阅以下主题以了解有关如何使用 Submariner 的更多信息：

- 在断开连接的集群中部署 Submariner
- 配置 Submariner
- 安装 subctl 命令工具

- 使用控制台部署 Submariner
- 手动部署 Submariner
- 自定义 Submariner 部署
- 为 Submariner 管理服务发现
- 卸载 Submariner

1.4.1. 在断开连接的集群中部署 Submariner

在断开连接的集群中部署 Submariner 可以帮助出现安全问题，从而降低集群的外部攻击风险。要在断开连接的集群中使用 Red Hat Advanced Cluster Management for Kubernetes 部署 Submariner，您必须首先完成在断开连接的环境中安装中所述的步骤。

1.4.1.1. 在断开连接的集群中配置 Submariner

按照在断开连接的环境中安装中所述的步骤后，您必须在安装过程中配置 Submariner，以支持在断开连接的集群上部署。请参见以下主题：

1.4.1.1.1. 在本地 registry 中 mirror 镜像

在断开连接的集群上部署 Submariner 前，请确保在本地 registry 中部署 **Submariner Operator** 捆绑包镜像。

1.4.1.1.2. 自定义 catalogSource 名称

默认情况下，**submariner-addon** 会搜索名为 **redhat-operators** 的 **catalogSource**。当使用具有不同名称的 **catalogSource** 时，您必须使用 **catalogSource** 自定义名称更新与您的受管集群关联的 **SubmarinerConfig** 中的 **SubmarinerConfig.Spec.subscriptionConfig.Source** 参数的值。

1.4.1.1.3. 在 SubmarinerConfig 中启用 airGappedDeployment

当从 Red Hat Advanced Cluster Management for Kubernetes 控制台在受管集群中安装 **submariner-addon** 时，您可以选择 **Disconnected cluster** 选项，以便 Submariner 不向外部服务器发出 API 查询。

如果要使用 API 安装 Submariner，您必须在与受管集群关联的 **SubmarinerConfig** 中将 **airGappedDeployment** 参数设置为 **true**。

1.4.2. 配置 Submariner

Red Hat Advanced Cluster Management for Kubernetes 提供 Submariner 作为 hub 集群的附加组件。要了解如何配置 Submariner，请阅读以下主题：

- 先决条件
- Submariner 端口表
- Globalnet

1.4.2.1. 先决条件

在使用 Submariner 前，请确保已满足以下先决条件：

- 使用 **cluster-admin** 权限访问 hub 集群的凭证。
- 在网关节点之间必须配置 IP 连接。连接两个集群时，网关节点必须利用分配给网关节点的公共或私有 IP 地址访问至少一个集群。如需更多信息，请参阅 Submariner NAT Traversal。
- 如果使用 OVN Kubernetes，集群必须位于 Red Hat OpenShift Container Platform 版本 4.13 或更高版本。
- 如果您的 Red Hat OpenShift Container Platform 集群使用 OpenShift SDN CNI，则每个受管集群中所有节点的防火墙配置必须同时允许 4800/UDP。
- 防火墙配置必须允许网关节点上的 4500/UDP 和 4490/UDP 在受管集群之间建立隧道。
- 如果网关节点可以通过其私有 IP 直接访问，且没有其中任何 NAT，请确保防火墙配置允许网关节点上的 ESP 协议。
  注：当您的集群部署在 Amazon Web Services、Google Cloud Platform、Microsoft Azure 或 Red Hat OpenStack 环境中时，这会自动配置，但必须为其他环境中的集群和保护私有云的防火墙手动配置。
- **managedcluster** 名称必须遵循 RFC 1123 中定义的 DNS 标签标准，并满足以下要求：
  - 包含 63 个字符或更少
  - 仅包含小写字母数字字符或 '-'
  - 以字母数字字符开头
  - 以字母数字字符结尾

1.4.2.2. Submariner 端口表

查看下表以查看您必须启用的 Submariner 端口：

Name	默认值	Customizable	可选或必需的
IPsec NATT	4500/UDP	是	必填
VXLAN	4800/UDP	否	必填
NAT 发现端口	4490/UDP	否	必填

1.4.2.3. Globalnet

Globalnet 是一个 Submariner 附加组件功能，它允许您连接带有重叠无类别域间路由(CIDR)的集群，而无需更改现有集群上的 CIDR。Globalnet 是一个集群范围的配置，您可以在将第一个受管集群添加到集群集时选择。

如果启用 Globalnet，则每个受管集群都会从虚拟全局专用网络接收全局 CIDR，用于促进集群间通信。

重要： 当集群集合中的集群有重叠的 CIDR 时，您必须启用 Globalnet。

在为集群集中启用 Submariner add-on 时，选择 **Enable Globalnet** 选项在控制台中启用 Globalnet。如果要在启用后禁用 Globalnet，您必须首先从集群集中删除所有受管集群。

1.4.2.3.1. 通过创建 submariner-broker 对象来启用 Globalnet

使用 Red Hat Advanced Cluster Management API 时，**ClusterAdmin** 可以通过在 **<ManagedClusterSet>-broker** 命名空间中创建 **submariner-broker** 对象来启用 Globalnet。

ClusterAdmin 角色具有在代理命名空间中创建 **submariner-broker** 对象所需的权限。**ManagedClusterSetAdmin** 角色 (有时被创建为作为集群集的代理管理员) 没有所需的权限。

要提供所需的权限，**ClusterAdmin** 必须将 **access-to-brokers-submariner-crd** 的角色权限与 **ManagedClusterSetAdmin** 用户关联。

通过创建 **submariner-broker** 对象来启用 Globalnet ：

1. 运行以下命令来检索 **<broker-namespace>** ：

```
oc get ManagedClusterSet <cluster-set-name> -o jsonpath="
{.metadata.annotations['cluster\.open-cluster-management\.io/submariner-broker-ns']}"
```

2. 创建一个 **submariner-broker** 对象，通过创建名为 **submariner-broker** 的 YAML 文件来指定 Globalnet 配置。在 YAML 文件中添加类似以下行的内容：

```
apiVersion: submariner.io/v1alpha1
kind: Broker
metadata:
  name: submariner-broker 1
  namespace: broker-namespace 2
spec:
  globalnetEnabled: true-or-false 3
```

1. 名称必须是 **submariner-broker**。
2. 将 **broker-namespace** 替换为代理命名空间的名称。
3. 将 **true-or-false** 替换为 **true** 来启用 Globalnet。

3. 运行以下命令来应用该文件：

```
oc apply -f submariner-broker.yaml
```

1.4.2.3.2. 配置全局 IP 数量

您可以通过更改 **ClusterGlobalEgressIP** 资源中的 **numberOfIPs** 字段的值来分配可配置的全局 IP 数。默认值为 8。请参见以下示例：

```
apiVersion: submariner.io/v1
kind: ClusterGlobalEgressIP
metadata:
  name: cluster-egress.submariner.io
spec:
  numberOfIPs: 8
```

1.4.2.3.3. 其他资源

- 请参阅 Submariner 文档 来了解更多有关 Submariner 的信息
- 如需有关网关节点之间的 IP 连接的更多信息，请参阅 Submariner NAT Traversal。
- 有关先决条件的详情，请参阅 Submariner 先决条件 文档。
- 有关 Globalnet 的更多信息，请参阅 Submariner 文档中的 Globalnet 控制器 。

1.4.3. 安装 subctl 命令工具

subctl 工具包含在容器镜像中。要在本地安装 **subctl** 工具，请完成以下步骤：

1. 运行以下命令登录到 registry，并在提示时输入您的凭证：

```
oc registry login --registry registry.redhat.io
```

2. 输入以下命令下载 **subctl** 容器，并将 **subctl** 二进制文件的压缩版本提取到 **/tmp** 中：

```
oc image extract registry.redhat.io/rhacm2/subctl-rhel8:v0.16 --path="/dist/subctl-*-linux-amd64.tar.xz":/tmp/ --confirm
```

3. 输入以下命令解压缩 **subctl** 工具：

```
tar -C /tmp/ -xf /tmp/subctl-v0.16*-linux-amd64.tar.xz
```

4. 输入以下命令安装 **subctl** 工具：

```
install -m744 /tmp/subctl-v0.16*/subctl-v0.16*-linux-amd64 /$HOME/.local/bin/subctl
```

备**注**：

- 确保 **subctl** 和 Submariner 版本匹配。
- 对于断开连接的环境，请确保镜像 **submariner-nettest** 镜像。

1.4.3.1. 使用 subctl 命令

在路径中添加工具后，请查看下表以了解可用命令的简短描述：

export service	为指定服务创建一个 **ServiceExport** 资源，它允许 Submariner 部署中的其他集群发现对应的服务。
unexport service	删除指定服务的 **ServiceExport** 资源，这可防止 Submariner 部署中的其他集群发现对应的服务。
show	提供有关 Submariner 资源的信息。
verify	当 Submariner 在一组集群中配置时，验证连接、服务发现和其他 Submariner 功能。
benchmark	使用 Submariner 或单个集群中启用的一对集群的基准测试吞吐量和延迟。

diagnose	运行检查以识别防止 Submariner 部署正常工作的问题。
gather	从集群收集信息，以帮助对 Submariner 部署进行故障排除。
version	显示 **subctl** 二进制工具的版本详情。

*注 ：红帽构建的 **subctl** 只包含与 Red Hat Advanced Cluster Management for Kubernetes 相关的命令。有关 **subctl** 工具及其命令的更多信息，请参阅 Submariner 文档中的**subctl** 部分。*

1.4.4. 使用控制台部署 Submariner

*在使用 Red Hat Advanced Cluster Management for Kubernetes 部署 Submariner 前，您必须在托管环境中准备集群。您可以使用 **SubmarinerConfig** API 或 Red Hat Advanced Cluster Management for Kubernetes 控制台在以下供应商上自动准备 Red Hat OpenShift Container Platform 集群：*

- *Amazon Web Services*
- *Google Cloud Platform*
- *IBM Power 系统虚拟服务器*
- *Red Hat OpenShift on IBM Cloud （技术预览）*
- *Red Hat OpenStack Platform*
- *Microsoft Azure*
- *VMware vSphere*

*备**注：***

- *VMware vSphere 仅支持非NSX 部署。*
- *如果要在 IBM Cloud 上使用 Red Hat OpenShift，则必须在集群中安装 Calico API 服务器。另外，您可以按照 Submariner 上游文档中的 CALICO CNI 主题手动创建跨集群通信所需的 IP 池。*

要在其他供应商上部署 Submariner，请按照手动部署 Submariner 中的说明进行操作。

完成以下步骤，使用 Red Hat Advanced Cluster Management for Kubernetes 控制台部署 Submariner：

需要的**访问权**限：**集群管理员

1. *在控制台中，选择 **Infrastructure** > **Clusters**。*
2. *在 Clusters 页面上，选择 Cluster sets 选项卡。要使用 Submariner 启用的集群必须位于同一集群集合中。*
3. *如果要在其上部署 Submariner 的集群已位于同一集群集中，请跳至第 5 步。*

4. *如果要在其上部署 Submariner 的集群不在同一个集群集中，请完成以下步骤为它们创建一个集群集：*

   a. *选择 **Create cluster set**。*

   b. *对集群集进行命名，然后选择 **Create**。*

   c. *选择 **Manage resource assignments** 以将集群分配到集群集。*

   d. *选择您要与 Submariner 连接的受管集群，将它们添加到集群集合中。*

   e. *选择 **Review** 来查看并确认您选择的集群。*

   f. *选择 **Save** 保存集群集，并查看生成的集群设置页面。*

5. *在集群集页面中，选择 Submariner add-ons 选项卡。*

6. *选择 **Install Submariner add-ons**。*

7. *选择您要在其上部署 Submariner 的集群。*

8. *请参阅下表中的字段，并在 Install Submariner add-ons 编辑器中输入所需的信息：*

字段	备注
AWS 访问密钥 ID	仅在导入 AWS 集群时可见。
AWS Secret 访问密钥	仅在导入 AWS 集群时可见。
基域资源组名称	仅在导入 Azure 集群时可见。
客户端 ID	仅在导入 Azure 集群时可见。
客户端 secret	仅在导入 Azure 集群时可见。
订阅 ID	仅在导入 Azure 集群时可见。
租户 ID	仅在导入 Azure 集群时可见。
Google Cloud Platform 服务帐户 JSON 密钥	仅在导入 Google Cloud Platform 集群时可见。
实例类型	在受管集群中创建的网关节点的实例类型。
IPsec NAT-T 端口	IPsec NAT 遍历端口的默认值为 **4500**。如果您的受管集群环境是 VMware vSphere，请确保在防火墙中打开此端口。

字段	备注
网关计数	要在受管集群中部署的网关节点数量。对于 AWS、GCP、Azure 和 OpenStack 集群，部署了专用网关节点。对于 VWware 集群，现有的 worker 节点被标记为网关节点。默认值为 **1**。如果值大于 1，则会自动启用 Submariner 网关高可用性(HA)。
电缆驱动程序	维护跨集群隧道的 Submariner 网关电缆引擎组件。默认值为 **Libreswan IPsec**。
断开连接的集群	如果启用，请告知 Submariner 无法访问公共 IP 解析的任何外部服务器。
Globalnet CIDR	仅在集群集合中选择了 Globalnet 配置时才可见。用于受管集群的 Globalnet CIDR。如果留空，则会从集群设置池中分配一个 CIDR。

9. *在编辑器末尾选择 **Next** 以移动到下一个集群的编辑器，并为您选择的每个剩余的集群完成这个步骤。*
10. *验证每个受管集群的配置。*
11. *点 **Install** 在所选受管集群上部署 Submariner。*
    安装和配置完成可能需要几分钟时间。您可以在 Submariner add-ons 选项卡中的列表中检查 Submariner 状态：
    - *连**接状**态指示在受管集群中建立多少个 Submariner 连接。*
    - ***代理状**态代表 Submariner 是否成功部署到受管集群中。控制台可能会报告 **Degraded** 状态，直到安装和配置为止。*
    - *标签**的网关**节**点**表示受管集群中的网关节点数量。*

Submariner 现在部署在所选集群中。

1.4.5. 手动部署 Submariner

在使用 Red Hat Advanced Cluster Management for Kubernetes 部署 Submariner 前，您必须在托管环境中为连接准备集群。请参阅使用控制台部署 Submariner 以了解如何使用控制台在支持的集群中自动部署 Submariner。

如果您的集群托管在不支持自动 Submariner 部署的供应商上，请参阅以下部分来手动准备基础架构。每个提供程序都有唯一的准备步骤，因此请确保选择正确的提供程序。

1.4.5.1. 为 Submariner 准备裸机

要准备裸机集群来部署 Submariner，请完成以下步骤：

1. 确保防火墙允许 4500/UDP 和 4490/UDP 端口上的外部客户端入站/出站流量。另外，如果集群使用 OpenShiftSDN CNI 部署，允许本地集群节点中的入站/出站 UDP/4800 流量。

2. 自定义并应用类似以下示例的 YAML 内容：

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  gatewayConfig:
    gateways: 1
```

将 **managed-cluster-namespace** 替换为受管集群的名称。**SubmarinerConfig** 的名称必须是 **submariner**，如示例所示。

此配置将其中一个 worker 节点标记为裸机集群上的 Submariner 网关。

默认情况下，Submariner 使用 IP 安全(IPsec) 在网关节点上的集群之间建立安全隧道。您可以使用默认 IPsec NATT 端口，或者指定您配置的不同端口。当您运行这个步骤时，没有指定 IPsec NATT 端口，4500/UDP 用于连接。

3. 识别 Submariner 配置的网关节点，并启用防火墙配置，以允许用于外部流量的 IPsec NATT (UDP/4500) 和 NatDiscovery (UDP/4490) 端口。

如需有关自定义选项的信息，请参阅自定义 Submariner 部署。

1.4.5.2. 使用命令行界面为 Submariner 准备 Microsoft Azure Red Hat OpenShift

Microsoft Azure Red Hat OpenShift 服务组合了各种工具和资源，可用于简化构建基于容器的应用程序的过程。要准备 Azure Red Hat OpenShift 集群以使用命令行界面部署 Submariner，请完成以下步骤：

1. 安装 Azure CLI。

2. 在 Azure CLI 中运行以下命令安装扩展：

```
az extension add --upgrade -s <path-to-extension>
```

将 **path-to-extension** 替换为您下载 **.whl** 扩展文件的路径。

3. 运行以下命令验证是否使用了 CLI 扩展：

```
az extension list
```

如果使用扩展，输出可能类似以下示例：

```
"experimental": false,
"extensionType": "whl",
"name": "aro",
"path": "<path-to-extension>",
"preview": true,
"version": "1.0.x"
```

4. 在 Azure CLI 中，运行以下命令来注册 preview 功能：

```
az feature registration create --namespace Microsoft.RedHatOpenShift --name
AdminKubeconfig
```

5. 运行以下命令来检索管理员 **kubeconfig** ：

```
az aro get-admin-kubeconfig -g <resource group> -n <cluster resource name>
```

注： az aro 命令将 **kubeconfig** 保存到本地目录，并使用名称 **kubeconfig**。要使用它，设置环境变量 **KUBECONFIG** 以匹配文件的路径。请参见以下示例：

```
export KUBECONFIG=<path-to-kubeconfig>
oc get nodes
```

6. 导入 Azure Red Hat OpenShift 集群。请参阅集群导入简介 以了解有关如何导入集群的更多信息。

1.4.5.2.1. 使用 API 为 Submariner 准备 Microsoft Azure Red Hat OpenShift

要准备 Azure Red Hat OpenShift 集群以使用 API 部署 Submariner，请自定义并应用类似以下示例的 YAML 内容：

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  loadBalancerEnable: true
```

将 **managed-cluster-namespace** 替换为受管集群的名称。

SubmarinerConfig 的名称必须是 **submariner**，如示例所示。

此配置将其中一个 worker 节点标记为 Azure Red Hat OpenShift 集群上的 Submariner 网关。

默认情况下，Submariner 使用 IP 安全(IPsec) 在网关节点上的集群之间建立安全隧道。您可以使用默认 IPsec NATT 端口，或者指定您配置的不同端口。当您运行这个步骤时，如果没有指定 IPsec NATT 端口，会使用 4500/UDP 用于连接。

如需有关自定义选项的信息，请参阅自定义 Submariner 部署。

1.4.5.3. 使用命令行界面为 Submariner 准备 Red Hat OpenShift Service on AWS

Red Hat OpenShift Service on AWS 为应用程序开发和现代化提供了一个稳定而灵活的平台。要准备 OpenShift Service on AWS 集群来部署 Submariner，请完成以下步骤：

1. 运行以下命令，登录到 OpenShift Service on AWS ：

```
rosa login
oc login <rosa-cluster-url>:6443 --username cluster-admin --password <password>
```

2. 运行以下命令，在 AWS 集群上为 OpenShift Service 创建 **kubeconfig** ：

```
oc config view --flatten=true > rosa_kube/kubeconfig
```

3. 在 AWS 集群上导入 OpenShift Service。请参阅集群导入简介 以了解有关如何导入集群的更多信息。

1.4.5.3.1. 使用 API 为 Submariner 准备 Red Hat OpenShift Service on AWS

要准备 OpenShift Service on AWS 来使用 API 部署 Submariner，请自定义并应用类似以下示例的 YAML 内容：

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  loadBalancerEnable: true
```

将 **managed-cluster-namespace** 替换为受管集群的名称。

SubmarinerConfig 的名称必须是 **submariner**，如示例所示。

默认情况下，Submariner 使用 IP 安全(IPsec) 在网关节点上的集群之间建立安全隧道。您可以使用默认 IPsec NATT 端口，或者指定您配置的不同端口。当您运行这个步骤时，如果没有指定 IPsec NATT 端口，会使用 4500/UDP 用于连接。

如需有关自定义选项的信息，请参阅自定义 Submariner 部署。

1.4.5.4. 使用 ManagedClusterAddOn API 部署 Submariner

手动准备所选托管环境后，您可以通过完成以下步骤来使用 **ManagedClusterAddOn** API 部署 Submariner：

1. 按照 Creating a ManagedClusterSet 在 hub 集群中创建一个 **ManagedClusterSet** 资源。确保您的 **ManagedClusterSet** 条目类似以下内容：

   ```
   apiVersion: cluster.open-cluster-management.io/v1beta2
   kind: ManagedClusterSet
   metadata:
     name: <managed-cluster-set-name>
   ```

   将 **managed-cluster-set-name** 替换为您要创建的 **ManagedClusterSet** 的名称。

   重要：Kubernetes 命名空间的最大字符长度为 63 个字符。可用于 **<managed-cluster-set-name>** 的最大字符长度为 56 个字符。如果 **<managed-cluster-set-name>** 的字符长度超过 56 个字符，则 **<managed-cluster-set-name>** 会从头开始切断。

   创建 **ManagedClusterSet** 后，**submariner-addon** 会创建一个名为 **<managed-cluster-set-name>-broker** 的命名空间，并将 Submariner 代理部署到其中。

2. 通过自定义并应用类似以下示例的 YAML 内容，在 **<managed-cluster-set-name>-broker** 命名空间中的 hub 集群上创建 **Broker** 配置：

   ```
   apiVersion: submariner.io/v1alpha1
   kind: Broker
   metadata:
   ```

```
    name: submariner-broker
    namespace: <managed-cluster-set-name>-broker
    labels:
      cluster.open-cluster-management.io/backup: submariner
spec:
    globalnetEnabled: <true-or-false>
```

将 **managed-cluster-set-name** 替换为受管集群的名称。

如果要在 **ManagedClusterSet** 中启用 Submariner Globalnet，请将 **globalnetEnabled** 的值设置为 **true**。

3. 运行以下命令，将一个受管集群添加到 **ManagedClusterSet** 中：

```
oc label managedclusters <managed-cluster-name> "cluster.open-cluster-
management.io/clusterset=<managed-cluster-set-name>" --overwrite
```

将 **<managed-cluster-name>** 替换为您要添加到 **ManagedClusterSet** 的受管集群的名称。

将 **<managed-cluster-set-name>** 替换为您要添加受管集群的 **ManagedClusterSet** 的名称。

4. 自定义并应用类似以下示例的 YAML 内容：

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:{}
```

将 **managed-cluster-namespace** 替换为受管集群的命名空间。

注： **SubmarinerConfig** 的名称必须是 **submariner**，如示例中所示。

5. 通过自定义并应用类似以下示例的 YAML 内容，在受管集群中部署 Submariner：

```
apiVersion: addon.open-cluster-management.io/v1alpha1
kind: ManagedClusterAddOn
metadata:
    name: submariner
    namespace: <managed-cluster-name>
spec:
    installNamespace: submariner-operator
```

将 **managed-cluster-name** 替换为您要使用 Submariner 的受管集群的名称。

ManagedClusterAddOn 的 spec 中的 **installNamespace** 字段是在受管集群上安装 Submariner 的命名空间。目前，Submariner 必须安装到 **submariner-operator** 命名空间中。

创建 **ManagedClusterAddOn** 后，**submariner-addon** 将 Submariner 部署到受管集群上的 **submariner-operator** 命名空间。您可以从这个 **ManagedClusterAddOn** 的状态查看 Submariner 的部署状态。

注： **ManagedClusterAddOn** 的名称必须是 **submariner**。

6. 对您要启用 Submariner 的所有受管集群重复第三、第四和第五步骤。

7. 在受管集群中部署了 Submariner 后，您可以通过输入以下命令检查 submarinerr **ManagedClusterAddOn** 的状态来验证 Submariner 部署状态：

```
oc -n <managed-cluster-name> get managedclusteraddons submariner -oyaml
```

将 **managed-cluster-name** 替换为受管集群的名称。

在 Submariner **ManagedClusterAddOn** 的状态中，三个条件代表 Submariner 的部署状态：

- **SubmarinerGatewayNodesLabeled** 条件代表受管集群中是否存在标记为 Submariner 网关节点。
- **SubmarinerAgentDegraded** 条件指示 Submariner 是否成功部署到受管集群中。
- **SubmarinerConnectionDegraded** 条件指示受管集群上使用 Submariner 建立多少连接。

1.4.6. 自定义 Submariner 部署

您可以自定义 Submariner 部署的一些设置，包括网络地址转换(NATT)端口、网关节点数量和网关节点的实例类型。这些自定义在所有供应商间都是一致的。

1.4.6.1. NATT 端口

如果要自定义 NATT 端口，请自定义并应用您的供应商环境以下 YAML 内容：

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  credentialsSecret:
    name: <managed-cluster-name>-<provider>-creds
  IPSecNATTPort: <NATTPort>
```

- 将 **managed-cluster-namespace** 替换为受管集群的命名空间。
- 将 **managed-cluster-name** 替换为受管集群的名称
  - AWS：使用 **aws** 替换 **provider**。**<managed-cluster-name>-aws-creds** 的值是 AWS 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
  - GCP: 使用 **gcp** 替换 **provider**。**<managed-cluster-name>-gcp-creds** 的值是 Google Cloud Platform 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
  - 替换openstack: 使用 **osp** 替换 **provider**。**<managed-cluster-name>-osp-creds** 的值是 Red Hat OpenStack Platform 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
  - Azure ：使用 **azure** 替换 **provider**。**<managed-cluster-name>-azure-creds** 的值是 Microsoft Azure 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
- 将 **managed-cluster-namespace** 替换为受管集群的命名空间。

- 将 **managed-cluster-name** 替换为受管集群的名称。**managed-cluster-name-gcp-creds** 的值是 Google Cloud Platform 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到该 secret。
- 将 **NATTPort** 替换为您要使用的 NATT 端口。

注： **SubmarinerConfig** 的名称必须是 **submariner**，如示例中所示。

1.4.6.2. 网关节点数量

如果要自定义网关节点的数量，请自定义并应用类似以下示例的 YAML 内容：

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  credentialsSecret:
    name: <managed-cluster-name>-<provider>-creds
 gatewayConfig:
    gateways: <gateways>
```

- 将 **managed-cluster-namespace** 替换为受管集群的命名空间。
- 将 **managed-cluster-name** 替换为受管集群的名称。
  - AWS：使用 **aws** 替换 **provider**。**<managed-cluster-name>-aws-creds** 的值是 AWS 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
  - GCP: 使用 **gcp** 替换 **provider**。**<managed-cluster-name>-gcp-creds** 的值是 Google Cloud Platform 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
  - 替换openstack: 使用 **osp** 替换 **provider**。**<managed-cluster-name>-osp-creds** 的值是 Red Hat OpenStack Platform 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
  - Azure ：使用 **azure** 替换 **provider**。**<managed-cluster-name>-azure-creds** 的值是 Microsoft Azure 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
- 使用您要使用的网关数量替换 **gateway**。如果值大于 1，则 Submariner 网关会自动启用高可用性。

注： **SubmarinerConfig** 的名称必须是 **submariner**，如示例中所示。

1.4.6.3. 网关节点的实例类型

如果要自定义网关节点的实例类型，请自定义并应用类似以下示例的 YAML 内容：

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  credentialsSecret:
```

```
    name: <managed-cluster-name>-<provider>-creds
  gatewayConfig:
    instanceType: <instance-type>
```

- 将 **managed-cluster-namespace** 替换为受管集群的命名空间。
- 将 **managed-cluster-name** 替换为受管集群的名称。
  - AWS：使用 **aws** 替换 **provider**。**<managed-cluster-name>-aws-creds** 的值是 AWS 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
  - GCP: 使用 **gcp** 替换 **provider**。**<managed-cluster-name>-gcp-creds** 的值是 Google Cloud Platform 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
  - 替换openstack: 使用 **osp** 替换 **provider**。**<managed-cluster-name>-osp-creds** 的值是 Red Hat OpenStack Platform 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
  - Azure ：使用 **azure** 替换 **provider**。**<managed-cluster-name>-azure-creds** 的值是 Microsoft Azure 凭证 secret 名称，您可以在 hub 集群的集群命名空间中找到它。
- 将 **instance-type** 替换为您要使用的 AWS 实例类型。

注： SubmarinerConfig 的名称必须是 **submariner**，如示例中所示。

1.4.6.4. 电缆驱动程序

Submariner Gateway Engine 组件创建到其他集群的安全隧道。电缆驱动程序组件通过使用网关引擎组件中的可插拔架构来维护隧道。您可以使用 Libreswan 或 VXLAN 实现用于电缆引擎组件的 **cableDriver** 配置。请参见以下示例：

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  cableDriver: vxlan
  credentialsSecret:
    name: <managed-cluster-name>-<provider>-creds
```

最佳实践：不要在公共网络上使用 VXLAN 电缆驱动程序。VXLAN 电缆驱动程序是未加密的。仅在为了避免在私有网络中进行不必要的双加密的情况下才使用 VXLAN。例如，一些内部环境可能会使用专用的线一级的硬件设备处理隧道的加密。

1.4.6.5. 使用自定义 Submariner 订阅

Submariner 附加组件会自动为 Submariner 配置订阅；这样可确保安装 Red Hat Advanced Cluster Management 的 Submariner 版本并保持最新状态。如果要更改此行为，或者要手动控制 Submariner 升级，您可以自定义 Submariner 订阅。

使用自定义 Submariner 订阅时，您必须完成以下字段：

- **source ：** 用于 Submariner 订阅的目录源。例如： **redhat-operators**。
- **源命名空间：** 目录源的命名空间。例如，**openshift-marketplace**。

- **Channel：** 要订阅的频道。例如，对于 Red Hat Advanced Cluster Management 2.9， **stable-0.16**。
- 启动 CSV （可选）： 初始 **ClusterServiceVersion**。
- **安装计划批准** ： 手动或自动批准安装计划的决定。

注： 如果要手动批准安装计划，则必须使用自定义 Submariner 订阅。

1.4.7. 为 Submariner 管理服务发现

在 Submariner 部署到与受管集群相同的环境中后，会将路由配置为受管集群集中的 pod 和服务间的安全 IP 路由。

1.4.7.1. 为 Submariner 启用服务发现

要从集群可见服务并可以被受管集群集中的其他集群发现，您必须创建一个 **ServiceExport** 对象。使用 **ServiceExport** 对象导出服务后，您可以使用以下格式访问该服务：**<service>.<namespace>.svc.clusterset.local**。如果多个集群导出具有相同名称的服务，并且来自同一命名空间中，则其他集群会把这个服务看作为一个单一的逻辑服务。

在本例在，在 **default** 命名空间中使用 **nginx** 服务，但您可以发现任何 Kubernetes **ClusterIP** 服务或无头服务：

1. 使用以下命令，在 **ManagedClusterSet** 中的受管集群中应用 **nginx** 服务实例：

```
oc -n default create deployment nginx --image=nginxinc/nginx-unprivileged:stable-alpine
oc -n default expose deployment nginx --port=8080
```

2. 通过输入带有类似以下示例的 **subctl** 工具的命令创建一个 **ServiceExport** 条目来导出服务：

```
subctl export service --namespace <service-namespace> <service-name>
```

   将 **service-namespace** 替换为服务所在的命名空间的名称。在本例中，是 **default**。

   使用您要导出的服务的名称替换 **service-name**。在本例中是 **nginx**。

   如需有关其他可用标记的更多信息，请参阅 Submariner 文档中的 导出。

3. 在不同的受管集群中运行以下命令，确认它可以访问 **nginx** 服务：

```
oc -n default run --generator=run-pod/v1 tmp-shell --rm -i --tty --image
quay.io/submariner/nettest -- /bin/bash curl nginx.default.svc.clusterset.local:8080
```

nginx 服务发现现在已为 Submariner 配置。

1.4.7.2. 为 Submariner 禁用服务发现

要禁用将服务导出到其他集群，请为 **nginx** 输入一个类似以下示例的命令：

```
subctl unexport service --namespace <service-namespace> <service-name>
```

将 **service-namespace** 替换为服务所在的命名空间的名称。

使用您要导出的服务的名称替换 **service-name**。

有关其他可用标记的更多信息，请参阅 Submariner 文档中的**取消导出**。

集群不再可用于发现该服务。

1.4.8. 卸载 Submariner

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台或命令行从集群中删除 Submariner 组件。对于早于 0.12 的 Submariner 版本，需要额外的步骤来完全删除所有数据平面组件。Submariner uninstall 是幂等的，因此您可以在没有任何问题的情况下重复步骤。

1.4.8.1. 使用控制台卸载 Submariner

要使用控制台从集群卸载 Submariner，请完成以下步骤：

1. 在控制台导航中选择 **Infrastructure** > **Clusters**，然后选择 Cluster sets 选项卡。
2. 选择包含您要从中删除 Submariner 组件的集群集合。
3. 选择 **Submariner Add-ons** 选项卡来查看部署了 Submariner 的集群集合中的集群。
4. 在您要卸载 Submariner 的集群的 Actions 菜单中，选择 **Uninstall Add-on**。
5. 在您要卸载 Submariner 的集群的 Actions 菜单中，选择 **Delete cluster set**。
6. 对您要从中删除 Submariner 的其他集群重复这些步骤。
   提示： 您可以通过选择多个集群并点 **Actions**，从同一集群集中的多个集群中删除 Submariner 附加组件。选择 **Uninstall Submariner add-ons**。

如果您要删除的 Submariner 版本早于 0.12 版本，请手动使用 Uninstalling Submariner。如果 Submariner 版本为 0.12 或更高版本，则 Submariner 会被删除。

重要：验证所有云资源是否已从云供应商中删除，以避免您的云供应商额外的费用。如需更多信息，请参阅验证 Submariner 资源删除。

1.4.8.2. 使用 CLI 卸载 Submariner

要使用命令行卸载 Submariner，请完成以下步骤：

1. 运行以下命令来删除集群的 Submariner 部署：

   ```
   oc -n <managed-cluster-namespace> delete managedclusteraddon submariner
   ```

   将 **managed-cluster-namespace** 替换为受管集群的命名空间。

2. 运行以下命令删除集群的云资源：

   ```
   oc -n <managed-cluster-namespace> delete submarinerconfig submariner
   ```

   将 **managed-cluster-namespace** 替换为受管集群的命名空间。

3. 运行以下命令删除集群集以删除代理详情：

   ```
   oc delete managedclusterset <managedclusterset>
   ```

将 **managedclusterset** 替换为受管集群集的名称。

如果您要删除的 Submariner 版本早于 0.12 版本，请手动使用 Uninstalling Submariner。如果 Submariner 版本为 0.12 或更高版本，则 Submariner 会被删除。

重要：验证所有云资源是否已从云供应商中删除，以避免您的云供应商额外的费用。如需更多信息，请参阅验证 Submariner 资源删除。

1.4.8.3. 手动卸载 Submariner

卸载早于 0.12 版本的 Submariner 版本时，在 Submariner 文档中的 Manual Uninstall 部分中完成步骤 5-8。

完成这些步骤后，Submariner 组件将从集群中移除。

重要：验证所有云资源是否已从云供应商中删除，以避免您的云供应商额外的费用。如需更多信息，请参阅验证 Submariner 资源删除。

1.4.8.4. 验证 Submariner 资源删除

卸载 Submariner 后，验证所有 Submariner 资源是否已从集群中移除。如果它们保留在集群中，某些资源将继续获得基础架构供应商的费用。通过完成以下步骤，确保集群中没有额外的 Submariner 资源：

1. 运行以下命令列出集群中保留的所有 Submariner 资源：

   ```
   oc get cluster <CLUSTER_NAME> grep submariner
   ```

   将 **CLUSTER_NAME** 替换为集群的名称。

2. 输入以下命令删除列表中的任何资源：

   ```
   oc delete resource <RESOURCE_NAME> cluster <CLUSTER_NAME>
   ```

   将 **RESOURCE_NAME** 替换为您要删除的 Submariner 资源的名称。

3. 对每个集群重复步骤 1-2，直到搜索无法识别任何资源。

Submariner 资源从集群中移除。