Red Hat OpenShift Data Foundation 4.18

规划部署

部署 Red Hat OpenShift Data Foundation 4.18 时的重要注意事项

Last Updated: 2025-12-13

Red Hat OpenShift Data Foundation 4.18 规划部署

部署 Red Hat OpenShift Data Foundation 4.18 时的重要注意事项

Legal Notice

Abstract

有关规划 Red Hat OpenShift Data Foundation 部署时的重要注意事项，请参阅本文档。

Table of Contents

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 CTO Chris Wright 的信息。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请告诉我们如何让它更好。

要提供反馈，请创建一个 JIRA ticket：

1. 登录到 JIRA。
2. 在顶部导航栏中点 **Create**
3. 在 Summary 字段中输入描述性标题。
4. 在 Description 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 在 Components 字段中选择 **Documentation**。
6. 点对话框底部的 Create。

第 1 章 OPENSHIFT DATA FOUNDATION 介绍

Red Hat OpenShift Data Foundation 是 Red Hat OpenShift Container Platform 的云存储和数据服务的高度集成集合。它作为 Red Hat OpenShift Container Platform Service Catalog 的一部分提供，它作为一个 operator 提供，以便于简单部署和管理。

Red Hat OpenShift Data Foundation 服务主要通过代表以下组件的存储类提供给应用程序：

- 块存储设备，主要服务于数据库工作负载。示例包括 Red Hat OpenShift Container Platform 日志记录和监控，以及 PostgreSQL。

重要

只有在不需要在多个容器间共享数据时，才会将块存储用于任何工作。

- 共享和分布式文件系统，主要服务于软件开发、消息传递和数据聚合工作负载。示例包括 Jenkins 构建源和工件、Wordpress 上传的内容、Red Hat OpenShift Container Platform registry，以及使用 JBoss AMQ 的消息传递。
- 多云对象存储，具有一个轻量级 S3 API 端点，可以从多个云对象存储中提取存储和检索数据。
- 在内部对象存储中，具有一个稳定的 S3 API 端点，可扩展到数十拍字节（PB）和数十亿个对象的环境，主要面向数据密集型应用。例如，使用 Spark、Pacesto、Red Hat AMQ Streams (Kafka) 等应用程序，以及 TensorFlow 和 Pytorch 等机器学习框架。

注意

不支持在 CephFS 持久性卷上运行 PostgresSQL 工作负载，建议使用 RADOS 块设备 (RBD) 卷。如需更多信息，请参阅 知识库解决方案 ODF 数据库工作负载不能使用 CephFS PV/PVC。

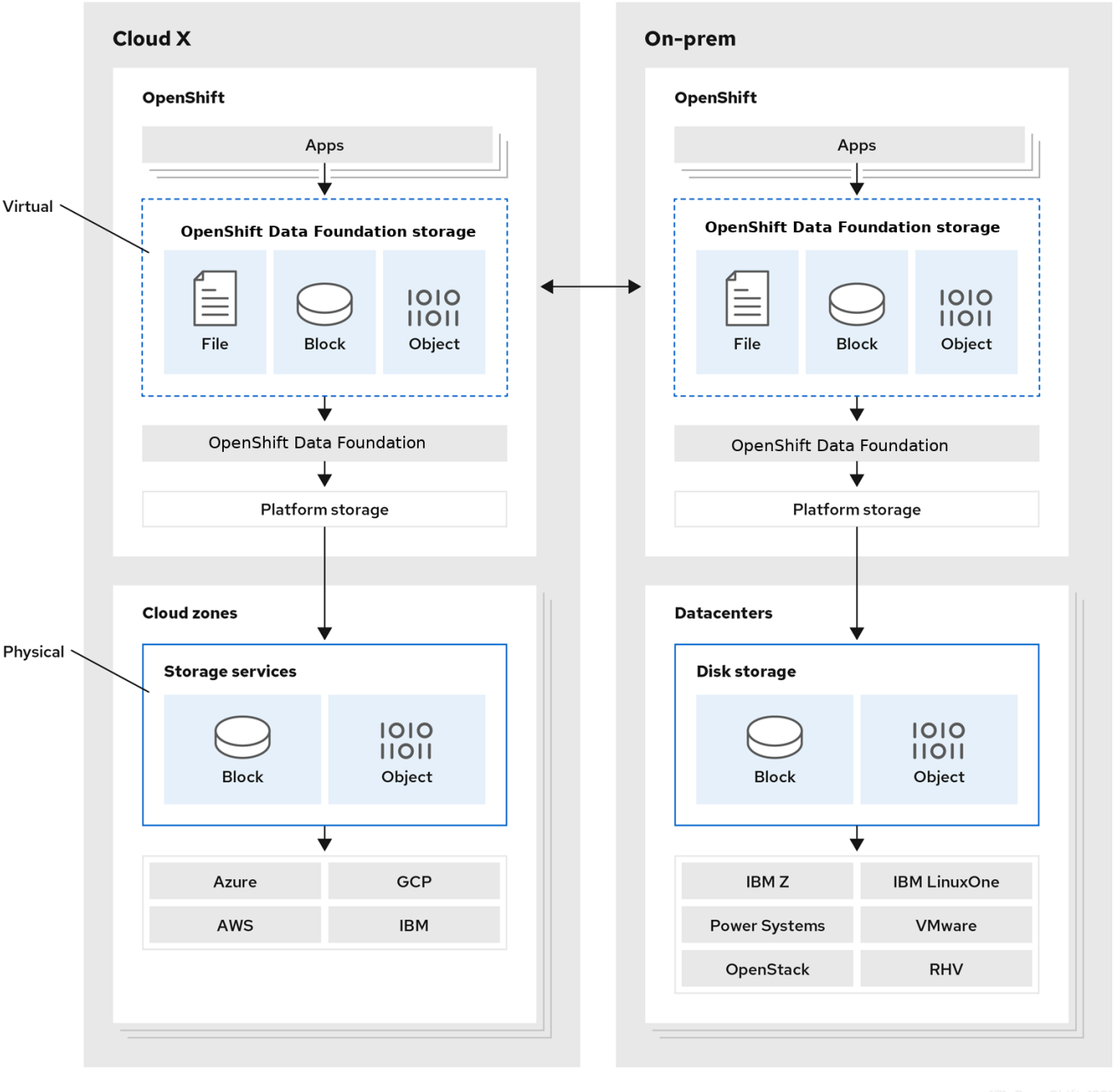
Red Hat OpenShift Data Foundation 版本 4.x 由一组软件项目组成，包括：

- Ceph，提供块存储、共享分布式文件系统以及内部对象存储
- Ceph CSI，用于管理持久性卷和声明的调配和生命周期
- NooBaa 提供多云对象网关
- OpenShift Data Foundation、Rook-Ceph 和 NooBaa 操作器，用于初始化和管理 OpenShift Data Foundation 服务。

第 2 章 OPENSHIFT DATA FOUNDATION 基础架构

Red Hat OpenShift Data Foundation 为 Red Hat OpenShift Container Platform 提供服务，也可以从 Red Hat OpenShift Container Platform 内部运行。

图 2.1. Red Hat OpenShift Data Foundation 架构



Red Hat OpenShift Data Foundation 支持部署到在安装程序置备的基础架构或用户置备的基础架构上部署的 Red Hat OpenShift Container Platform 集群中。

有关这两种方法的详情，请参阅 OpenShift Container Platform - 安装过程。

如需了解更多有关 Red Hat OpenShift Data Foundation 和 Red Hat OpenShift Container Platform 组件互操作性的信息，请参阅 Red Hat OpenShift Data Foundation 支持性和互操作性检查器 。

如需有关 OpenShift Container Platform 架构和生命周期的信息，请参阅 OpenShift Container Platform 架构。

提示

对于 IBM Power，请参阅在 IBM Power 上安装。

2.1. 关于 OPERATOR

Red Hat OpenShift Data Foundation 由三个主要操作器（operator）组成，它们协调管理任务和自定义资源，以便您可以轻松自动化任务和资源特征。管理员定义集群的所需最终状态，OpenShift Data Foundation 通过最少的管理员干预来确保集群处于该状态，或接近该状态。

OpenShift Data Foundation operator（操作器）

使用特定测试的方法在其他 Operator 上绘制并强制实施受支持的 Red Hat OpenShift Data Foundation 部署的建议和要求。rook-ceph 和 noobaa operator 提供了打包了这些资源的存储集群资源。

Rook-ceph operator

此 operator 自动打包、部署、管理、升级和扩展持久存储和文件、块和对象服务。它为所有环境创建块和文件存储类，并在内部环境中创建针对它的对象存储类和服务对象存储桶声明（Object Bucket Claims (OBCs)）。

此外，对于内部模式集群，它提供 Ceph 集群资源，它管理部署和服务，如下所示：

- 对象存储守护进程 (OSD)
- 监视器 (MON)
- 经理 (MGR)
- 元数据服务器 (MDS)
- 仅限内部 RADOS 对象网关 (RGW)

多云对象网关 operator

此 operator 自动打包、部署、管理、升级和扩展多云对象网关对象（MCG）服务。它创建一个对象存储类，以及 OBCs 提出的服务。

另外，它还提供 NooBaa 集群资源，用于管理 NooBaa core、数据库和端点的部署和服务。

注意

OpenShift Data Foundation 用于 MCG 的默认配置针对低资源消耗和不性能进行了优化。如果您计划经常使用 MCG，请参阅 Multicloud Object Gateway 的知识库文章性能调优指南中的 有关增加资源限值的信息。

2.2. 存储集群部署方法

日益增加的运营模式列表表明，灵活性是 Red Hat OpenShift Data Foundation 的核心原则。本节将为您提供信息，帮助您为您的环境选择最合适的方法。

Red Hat OpenShift Data Foundation 可以完全在 OpenShift Container Platform 中部署（内部方法），或者从 OpenShift Container Platform 外部运行的集群（外部方法）提供服务。

2.2.1. 内部方法

在 Red Hat OpenShift Container Platform 中完全部署 Red Hat OpenShift Data Foundation 具有基于 Operator 的部署和管理优势。图形用户界面（GUI）中的内部附加设备方法可用于使用本地存储 operator 和本地存储设备以内部模式部署 Red Hat OpenShift Data Foundation。

简化部署和管理是在 OpenShift Container Platform 内部运行 OpenShift Data Foundation 服务的关键。当 Red Hat OpenShift Data Foundation 完全在 Red Hat OpenShift Container Platform 中运行时，可以使用两种不同的部署模式：

- Simple（简单）
- Optimized（优化）

简单部署

Red Hat OpenShift Data Foundation 服务与应用程序共同运行。Red Hat OpenShift Container Platform 中的 Operator 管理这些应用程序。

简单的部署最适用于以下情况。

- 存储要求不明确。
- Red Hat OpenShift Data Foundation 服务与应用程序共同运行。
- 创建一个有特定大小的端点实例（例如在裸机上）比较困难。

要使 Red Hat OpenShift Data Foundation 与应用程序共同运行，节点必须具有动态附加本地存储设备或可移植存储设备，如 EC2 上的 EBS 卷或 VMware 上的 vSphere 虚拟磁盘或 SAN 卷。

注意

PowerVC 动态置备 SAN 卷。

优化的部署

Red Hat OpenShift Data Foundation 服务在专用的基础架构节点上运行。Red Hat OpenShift Container Platform 管理这些基础架构节点。

优化的方法最适合以下情况，

- 存储要求很明确。
- Red Hat OpenShift Data Foundation 服务在专用的基础架构节点上运行。
- 创建特定大小的节点实例很容易，例如在云、虚拟化环境中。

2.2.2. 外部方法

Red Hat OpenShift Data Foundation 将 OpenShift Container Platform 集群外运行的 Red Hat Ceph Storage 服务作为存储类公开。

在以下情况下最适合使用外部方法，

- 存储要求非常显著（超过 600 个存储设备）
- 多个 OpenShift Container Platform 集群需要消耗来自通用外部集群的存储服务。

- 其他团队，如站点可靠性工程 (SRE) 团队、存储团队等，需要管理提供存储服务的外部集群。可能已存在一个。

2.3. 节点类型

节点运行容器运行时和服务，以确保容器正在运行，并且维护容器集之间的网络通信和隔离。在 OpenShift Data Foundation 中，有三种类型的节点。

表 2.1. 节点类型

节点类型	描述
Master	这些节点运行公开 Kubernetes API、观察和调度新创建的 pod、维护节点健康和数量，以及控制与底层云供应商的交互的进程。
Infrastructure (Infra)	Infra 节点运行集群级别的基础架构服务，如日志记录、指标、registry 和路由。这些在 OpenShift Container Platform 集群中是可选的。为了将 OpenShift Data Foundation 工作负载与应用程序分离，请确保在虚拟化和云环境中使用 infra 节点作为 OpenShift Data Foundation。 要创建 Infra 节点，您可以置备标记为 **infra** 的新节点。如需更多信息，请参阅如何为 Red Hat OpenShift Data Foundation 使用专用 worker 节点
Worker	Worker 节点也称为应用节点，因为它们运行应用。 当 OpenShift Data Foundation 以内部模式部署时，您需要包含 3 个 worker 节点的最小集群。确保节点分散在 3 种不同的机架或可用性区域，以确保可用性。为了使 OpenShift Data Foundation 在 worker 节点上运行，您需要将本地存储设备或可移植的存储设备动态附加到 worker 节点。 当 OpenShift Data Foundation 以外部模式部署时，它会在多个节点上运行。这允许 Kubernetes 在故障时重新调度到可用节点上。

注意

OpenShift Data Foundation 需要与 OpenShift Container Platform 相同的订阅数。但是，如果 OpenShift Data Foundation 在 infra 节点上运行，OpenShift 不需要 OpenShift Container Platform 订阅用于这些节点。因此，OpenShift Data Foundation control plane 不需要额外的 OpenShift Container Platform 和 OpenShift Data Foundation 订阅。如需更多信息，请参阅 第 6 章 *订阅*。

第 3 章 内部存储服务

Red Hat OpenShift Data Foundation 服务可在内部被在以下基础架构上运行的 Red Hat OpenShift Container Platform 使用：

- Amazon Web Services (AWS)
- 裸机
- VMware vSphere
- Microsoft Azure
- Google Cloud
- Red Hat OpenStack 13 或更高版本 （安装程序置备的基础架构）[技术预览]
- IBM Power
- IBM Z 和 IBM® LinuxONE
- 带有托管的 control plane (HCP)的 ROSA

创建内部集群资源将导致内部置备 OpenShift Data Foundation 基础服务，并为应用程序提供额外的存储类。

第 4 章 外部存储服务

Red Hat OpenShift Data Foundation 可以从外部 Red Hat Ceph Storage 集群提供服务，以便通过在以下平台上运行的 OpenShift Container Platform 集群使用：

- VMware vSphere
- 裸机
- Red Hat OpenStack Platform（技术预览）
- IBM Power
- IBM Z

OpenShift Data Foundation operator 会创建和管理服务，以满足对外部服务的持久性卷声明(PV)和对象存储桶声明(OBC)。外部集群可以为 OpenShift Container Platform 上运行的应用程序提供块、文件和对象存储类。Operator 不会部署或管理外部集群。

第 5 章 安全考虑

5.1. FIPS CRYPTOGRAPHY

Federal Information Processing Standard Publication 140-2/140-3 (FIPS 140-2/140-3)是定义使用加密模块的一组安全要求的标准。这个标准受到美国政府机构和承包商的强制要求，在其他国际和行业特定的标准中也会引用该标准。

Red Hat OpenShift Data Foundation 为 FIPS 设计。当以 FIPS 模式运行 Red Hat Enterprise Linux (RHEL)或 Red Hat Enterprise Linux CoreOS (RHCOS)引导时，OpenShift Container Platform 核心组件（包括 OpenShift Data Foundation）使用已提交到 NIST for FIPS 140-2/140-3 Validation 的 RHEL 加密库，仅在 x86_64、ppc64le 和 s390x 架构上提交。如需更多信息，请参阅 OpenShift 文档中的对 FIPS 加密的支持 部分。

目前，Cryptographic Module Validation Program (CMVP) 用于处理加密模块。您可以在modules in Process List 中查看这些模块的状态。有关最新信息，请参阅红帽知识库解决方案 RHEL 内核加密组件。

5.2. 代理环境

代理环境是一个生产环境，它拒绝直接访问互联网并提供可用的 HTTP 或 HTTPS 代理。Red Hat Openshift Container Platform 被配置为使用代理，方法是修改现有集群的代理对象，或在新集群的 install-config.yaml 文件中配置代理设置。

当已根据配置集群范围的代理的内容配置了 OpenShift Container Platform，则红帽支持在代理环境中部署 OpenShift Data Foundation。

5.3. 数据加密选项

加密可让您对数据进行编码，使其在没有所需的加密密钥的情况下无法读取。通过这种机制，当物理性安全被破坏的情况下，您的数据所在的物理介质丢失时，可以保护您的数据的安全性。每个PV 加密也提供同一 OpenShift Container Platform 集群内其他命名空间的访问保护。当数据写入到磁盘时，数据会被加密，并在从磁盘读取数据时对其进行解密。使用加密的数据可能会对性能产生较小的影响。

只有使用 Red Hat OpenShift Data Foundation 4.6 或更高版本部署的新集群才支持加密。没有使用外部密钥管理系统 (KMS) 的现有加密集群无法迁移为使用外部 KMS。

以前，HashiCorp Vault 是唯一支持集群范围的 KMS 和持久性卷加密的 KMS。在 OpenShift Data Foundation 4.7.0 和 4.7.1 中，只支持 HashiCorp Vault Key/Value (KV) secret engine API，支持版本 1。从 OpenShift Data Foundation 4.7.2 开始，支持 HashiCorp Vault KV secret engine API、版本 1 和 2。从 OpenShift Data Foundation 4.12 开始，Thales CipherTrust Manager 已被作为额外支持的 KMS 被引进。

重要

- KMS 是 StorageClass 加密所必需的，对于集群范围的加密，它是可选的。
- 首先，存储类加密需要一个有效的 Red Hat OpenShift Data Foundation Advanced 订阅。如需更多信息，请参阅 OpenShift Data Foundation 订阅中的知识库文章。

红帽与技术合作伙伴合作，将本文档作为为客户提供服务。但是，红帽不为 Hashicorp 产品提供支持。有关此产品的技术协助，请联系 Hashicorp。

5.3.1. 集群范围的加密

Red Hat OpenShift Data Foundation 支持存储集群中所有磁盘和多云对象网关操作的集群范围加密 (encryption-at-rest)。OpenShift Data Foundation 使用基于 Linux 统一密钥系统 (LUKS) 版本 2 的加密，其密钥大小为 512 位，以及 **aes-xts-plain64** 密码，其中每个设备都有不同的加密密钥。密钥使用 Kubernetes secret 或外部 KMS 存储。两种方法都是互斥的，您不能在方法之间迁移。

对于块存储和文件存储，默认会禁用加密。您可以在部署时为集群启用加密。MultiCloud 对象网关默认支持加密。如需更多信息，请参阅部署指南。

OpenShift Data Foundation 支持使用和不使用密钥管理系统(KMS)进行集群范围加密。使用以下服务供应商支持使用 KMS 进行集群范围加密：

- HashiCorp Vault
- Thales Cipher Trust Manager

安全常见实践需要定期加密密钥轮转。OpenShift Data Foundation 会每周自动轮转 kubernetes secret（非KMS）和 Vault 中存储的加密密钥。但是，在存储集群创建后，必须启用 Vault KMS 的密钥轮转，默认不会发生。如需有关部署指南的更多信息，请参阅部署指南。

注意

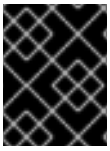
需要有效的 Red Hat OpenShift Data Foundation 高级订阅。要了解 OpenShift Data Foundation 订阅如何工作，请参阅与 OpenShift Data Foundation 订阅相关的知识库文章。

使用 HashiCorp Vault KMS 进行集群范围内的加密提供了两种身份验证方法：

- **令牌** ：此方法允许使用 vault 令牌进行身份验证。在 openshift-storage 命名空间中创建包含 vault 令牌的 kubernetes secret，用于身份验证。如果选择了这个验证方法，那么管理员必须提供 vault 中后端路径（其中存储了加密密钥）的 vault 令牌。
- **Kubernetes** ：此方法允许使用服务账户（serviceaccounts）通过 vault 进行身份验证。如果选择了这种身份验证方法，那么管理员必须提供 Vault 中配置的角色的名称，从而提供对后端路径的访问，然后存储了加密密钥。然后，此角色的值会添加到 **ocs-kms-connection-details** 配置映射中。

注意

除了 HashiCorp Vault KMS，IBM Cloud 平台上的 OpenShift Data Foundation 现在还支持 Hyper Protect Crypto Services(HPCS)Key Management Services(KMS)作为加密解决方案。



重要

红帽与技术合作伙伴合作，将本文档作为为客户提供服务。但是，红帽不为 Hashicorp 产品提供支持。有关此产品的技术协助，请联系 Hashicorp。

5.3.2. 存储类加密

您可以使用外部密钥管理系统 (KMS) 使用存储类加密来加密持久性卷（仅限块）来存储设备加密密钥。永久卷加密仅可用于 RADOS 块设备 (RBD) 持久卷。请参阅 如何使用持久性卷加密创建存储类。

存储类加密在使用 HashiCorp Vault KMS 的 OpenShift Data Foundation 4.7 或更高版本中被支持。存储类加密在使用 HashiCorp Vault KMS 和 Thales CipherTrust Manager KMS 的 OpenShift Data Foundation 4.12 或更高版本中被支持。

注意

需要有效的 Red Hat OpenShift Data Foundation 高级订阅。要了解 OpenShift Data Foundation 订阅如何工作，请参阅与 OpenShift Data Foundation 订阅相关的知识库文章。

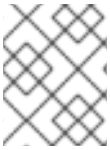
5.3.3. CipherTrust manager

Red Hat OpenShift Data Foundation 版本 4.12 引入了 Thales CipherTrust Manager 作为部署的附加密钥管理系统 (KMS) 供应商。Thales CipherTrust Manager 提供了中央化的密钥生命周期管理功能。CipherTrust Manager 支持密钥管理互操作性协议(KMIP)，它启用了密钥管理系统之间的通信。

CipherTrust Manager 在部署期间被启用。

5.3.4. 通过 Red Hat Ceph Storage 的 messenger 版本 2 协议(msgr2)传输数据加密。

从 OpenShift Data Foundation 版本 4.14 开始，可以使用 Red Hat Ceph Storage 的 messenger 版本 2 协议来加密转换数据。这为您的基础架构提供重要的安全要求。



注意

启用 FIPS 模式时，不应使用 Red Hat Ceph Storage 的 messenger 版本 2 协议。传输中的加密可以使用节点之间的 IPsec 来完成。如需更多信息，请参阅 Transit 中的加密。

在创建集群时，可以在部署期间启用 in-transit 加密。有关在集群创建过程中启用数据加密的说明，请参阅环境的 部署指南。

msgr2 协议支持两种连接模式：

crc

- 与 cephx 建立连接时，提供强大的初始身份验证。
- 提供 crc32c 完整性检查，以防止 bit flips。
- 不能提供对恶意的中间人攻击提供保护。
- 不能阻止对认证后的网络流量进行窃听。

secure

- 与 cephx 建立连接时，提供强大的初始身份验证。
- 提供所有认证后的网络流量的完全加密。
- 提供加密完整性检查。

默认模式是 **crc**。

5.4. TRANSIT 中的加密

您需要启用 IPsec，以便 OVN-Kubernetes Container Network Interface (CNI) 集群网络中的节点之间的所有网络流量都通过加密的隧道进行传输。

默认情况下禁用 IPsec。您可以在安装集群前或安装集群之后启用它。如果您需要在集群安装后启用 IPsec，您必须首先将集群 MTU 大小调整为考虑 IPsec ESP IP 标头的开销。

有关如何配置 IPsec 加密的更多信息，请参阅 OpenShift Container Platform 文档中的配置 网络指南 的 *IPsec 加密*。

第 6 章 订阅

6.1. 订阅服务

Red Hat OpenShift Data Foundation 订阅基于"内核对"，与 Red Hat OpenShift Container Platform 类似的。Red Hat OpenShift Data Foundation 2 核订阅基于 OpenShift Container Platform 运行的系统中 CPU 的逻辑内核数量。

与 OpenShift Container Platform 一样：

- OpenShift Data FoundationOpenShift 订阅可以被叠加，以覆盖更大的主机。
- 内核可以根据需要在多个虚拟机 (VM) 间进行分配。例如，十个 2 核订阅将提供 20 个内核，对于 IBM Power 的 2 核订阅（SMT 级别为 8），提供 2 个内核或 16 个 vCPU，可在任意数量的虚拟机中使用。
- OpenShift Data Foundation 订阅提供高级或标准支持。

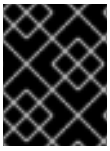
6.2. 灾难恢复订阅要求

Red Hat OpenShift Data Foundation 支持的灾难恢复功能需要满足以下所有先决条件，才能成功实施灾难恢复解决方案：

- 有效的 Red Hat OpenShift Data Foundation 高级授权
- 有效的 Red Hat Advanced Cluster Management for Kubernetes 订阅

任何包含 PV（包括作为源或目标）的 PV 的 Red Hat OpenShift Data Foundation 集群都需要 OpenShift Data Foundation 高级授权。此订阅应该在源和目标集群上处于活跃状态。

要了解 OpenShift Data Foundation 订阅如何工作，请参阅与 OpenShift Data Foundation 订阅相关的知识库文章。



重要

使用 Multus 网络部署的 OpenShift Data Foundation 不支持在区域灾难恢复(Regional-DR)设置中。

6.3. 内核与 VCPU 和超线程

判断特定系统是否消耗一个或多个内核目前取决于该系统是否可用超线程。超线程只是 Intel CPU 的一项功能。访问红帽客户门户，以确定特定系统是否支持超线程。

使用逻辑 CPU 线程进行虚拟化 OpenShift 节点（也称为并发多线程(SMT)用于 Intel CPU 的 AMD EPYC CPU 或超线程），根据分配给该节点的内核/CPU 数量计算其 OpenShift 订阅的核心利用率，但每个订阅都涵盖使用逻辑 CPU 线程时 4 个 vCPU/核心。红帽的订阅管理工具假定所有系统上默认启用逻辑 CPU 线程。

对于启用了超线程的系统，一个超线程等于一个可见的系统内核，内核的计算是 2 个内核到 4 个 vCPU 的比率。因此，2 核订阅涵盖超线程系统中的 4 个 vCPU。一个大型虚拟机 (VM) 可能具有 8 个 vCPU，相当于 4 个订阅内核。当订阅以 2 核作为单位时，您将需要两个 2 核订阅来满足 4 个内核或 8 个 vCPU。

如果没有启用超线程，并且每个可见的系统内核直接与底层物理内核关联，内核的计算为 2 个内核到 2 个 vCPU 的比率。

6.3.1. 用于 IBM Power 的内核数和并发多线程(SMT)

确定特定系统是否消耗一个或多个内核目前取决于配置的并发多线程级别 (SMT)。IBM Power 为每个内核提供并发多线程级别 1、2、4 或 8，每个内核对应于下表中的 vCPU 数量。

表 6.1. 不同的 SMT 级别及其对应的 vCPU

SMT 级别	SMT=1	SMT=2	SMT=4	SMT=8
1 个内核	# vCPUs=1	# vCPUs=2	# vCPUs=4	# vCPUs=8
2 个内核	# vCPUs=2	# vCPUs=4	# vCPUs=8	# vCPUs=16
4 个内核	# vCPUs=4	# vCPUs=8	# vCPUs=16	# vCPUs=32

对于配置 SMT 的系统，用于订阅所需的内核数取决于 SMT 级别。因此，2 核订阅对于 SMT 级别 1 是 2 个 vCPU、对于 SMT 级别 2 是 4 个 vCPU，对于 SMT 级别 4 是 8 个 vCPU，对于 SMT 级别 8 是 16 个 vCPU，如上表所示。一个大型虚拟机 (VM) 可能有 16 个 vCPU，在 SMT 级别 8 中，需要一个 2 核订阅。计算方法是 vCPU 的数量除以 SMT 级别（对于 SMT-8，16 个 vCPU / 8 = 2）。当订阅以 2 核为单位时，您将需要一个 2 核订阅来满足这 2 个内核或 16 个 vCPU。

6.4. 分割内核

需要奇数内核的系统需要消耗整个 2 核订阅。例如，对于被计算为只需要 1 个内核的系统，在注册和订阅后，它会消耗一个整个的 2 核订阅。

当一个使用超线程、具有 2 个 vCPU 的虚拟机 (VM)，其计算的 vCPU 为 1 个时，则需要一个完整的 2 核订阅；一个 2 核订阅不能在两个使用超线程的带有 2 个 vCPU 的虚拟机间分割。如需更多信息，请参阅 内核与 vCPU 和超线程的比较 部分。

建议对虚拟实例进行大小调整，以便它们需要偶数数量的内核。

6.4.1. 用于 IBM Power 的共享处理器池

IBM Power 有共享处理器池的概念。共享处理器池中的处理器可以在集群的节点之间共享。Red Hat OpenShift Data Foundation 所需的聚合计算容量应当是多个内核对。

6.5. 订阅要求

Red Hat OpenShift Data Foundation 组件可以在 OpenShift Container Platform worker 或基础架构节点上运行，您可以将 Red Hat CoreOS (RHCOS) 或 Red Hat Enterprise Linux (RHEL) 8.4 用作主机操作系统。RHEL 7 现已弃用。每个 OpenShift Container Platform 订阅的内核都需要 OpenShift Data Foundation 订阅，比率为 1:1。

当使用基础架构节点时，即使 OpenShift worker 节点不需要任何 OpenShift Container Platform 或 OpenShift Data Foundation 订阅，为所有 worker 节点内核订阅 OpenShift Data Foundation 的这条规则也需要被满足。您可以使用标签来说明节点是 worker 还是基础架构节点。

如需更多信息，请参阅管理和分配存储资源指南中的如何将专用 worker 节点用于 Red Hat OpenShift Data Foundation 章节。

第 7 章 基础架构要求

7.1. 平台要求

Red Hat OpenShift Data Foundation 4.18 只在 OpenShift Container Platform 版本 4.18 及其以后的次版本中被支持。

以前版本的 Red Hat OpenShift Data Foundation 的程序错误修正将会作为程序错误修复版本发布。详情请参阅 Red Hat OpenShift Container Platform 生命周期政策 。

有关外部集群订阅要求，请参阅红帽知识库文章 OpenShift Data Foundation 订阅指南。

有关支持的平台版本的完整列表，请参阅 Red Hat OpenShift Data Foundation Supportability and Interoperability Checker。

7.1.1. Amazon EC2

只支持内部 Red Hat OpenShift Data Foundation 集群。

内部集群必须满足存储设备要求，并且具有通过 aws-ebs 置备程序提供 EBS 存储的存储类。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#storage-device-requirements_rhodf

OpenShift Data Foundation 支持由 Amazon Web Services (AWS) 提供的 **gp2-csi** 和 **gp3-csi** 驱动程序。这些驱动程序提供更好的存储扩展功能，并减少了每月的价格点 (**gp3-csi**)。现在，您可以在选择存储类时选择新驱动程序。如果需要高吞吐量，建议在部署 OpenShift Data Foundation 时使用 **gp3-csi**。

如果您需要每秒的高输入/输出操作(IOPS)，则推荐的 EC2 实例类型为 **D2** 或 **D3**。

7.1.2. 裸机

支持内部集群和使用外部集群。

内部集群必须满足存储设备要求，并且具有通过 Local Storage Operator 提供本地 SSD (NVMe/SATA/SAS、SAN)的存储类。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#storage-device-requirements_rhodf

7.1.3. VMware vSphere

支持内部集群和使用外部集群。

推荐的版本：

- vSphere 7.0 或更高版本
- vSphere 8.0 或更高版本

如需了解更多详细信息，请参阅 VMware vSphere 基础架构要求。

注意

如果 VMware ESXi 不将设备识别为闪存设备，请将其标记为闪存设备。在 Red Hat OpenShift Data Foundation 部署之前，请参考将设备标记为闪存。

另外，内部集群必须满足存储设备要求，并具有提供存储类的存储类。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#storage-device-requirements_rhodf

- VSAN 或 VMFS 数据存储通过 vsphere-volume 置备程序
- VMDK、RDM 或 DirectPath 存储设备通过 Local Storage Operator。

7.1.4. Microsoft Azure

只支持内部 Red Hat OpenShift Data Foundation 集群。

内部集群必须满足存储设备要求，并且具有通过 azure-disk 置备程序提供 azure 磁盘的存储类。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#storage-device-requirements_rhodf

7.1.5. Google Cloud

只支持内部 Red Hat OpenShift Data Foundation 集群。

内部集群必须满足存储设备要求，并且具有通过 gce-pd 置备程序提供 GCE Persistent Disk 的存储类。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#storage-device-requirements_rhodf

7.1.6. Red Hat OpenStack Platform [技术预览]

支持内部 Red Hat OpenShift Data Foundation 集群和使用外部集群。

内部集群必须满足存储设备要求，并且具有通过 Cinder 置备程序提供标准磁盘的存储类。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#storage-device-requirements_rhodf

7.1.7. IBM Power

支持内部 Red Hat OpenShift Data Foundation 集群和使用外部集群。

内部集群必须满足存储设备要求，并且具有通过 Local Storage Operator 提供本地 SSD (NVMe/SATA/SAS、SAN)的存储类。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#storage-device-requirements_rhodf

7.1.8. IBM Z 和 IBM® LinuxONE

支持内部 Red Hat OpenShift Data Foundation 集群。另外，支持在 x86 上运行 Red Hat Ceph Storage 的外部模式。

内部集群必须满足存储设备要求，并且具有通过 Local Storage Operator 提供本地 SSD (NVMe/SATA/SAS、SAN)的存储类。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#storage-device-requirements_rhodf

7.1.9. 带有托管的 control plane (HCP)的 ROSA

只支持内部 Red Hat OpenShift Data Foundation 集群。

内部集群必须满足存储设备要求，并且具有通过 **gp3-csi** 置备程序提供 AWS EBS 卷的存储类。

7.1.10. 任何平台

支持内部集群和使用外部集群。

内部集群必须满足存储设备要求，并且具有通过 Local Storage Operator 提供本地 SSD (NVMe/SATA/SAS、SAN)的存储类。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#storage-device-requirements_rhodf

7.2. 外部模式要求

7.2.1. Red Hat Ceph Storage

要以外部模式检查 Red Hat Ceph Storage (RHCS) 与 Red Hat OpenShift Data Foundation 的支持性和互操作性，请转至 lab Red Hat OpenShift Data Foundation 支持性和互操作性检查器 。

1. 选择 **Service Type** 为 **ODF as Self-Managed Service**。
2. 从下拉菜单中选择适当的 **Version**。
3. 在 Versions 选项卡中，点 **支持的 RHCS 兼容性**选项卡。

有关如何安装 RHCS 集群的说明，请参阅 安装指南。

7.3. 资源要求

Red Hat OpenShift Data Foundation 服务由一组初始的基础服务组成，并可使用附加设备集进行扩展。所有这些 Red Hat OpenShift Data Foundation 服务 pod 都由 OpenShift Container Platform 节点上的 kubernetes 进行调度。以三个节点（每个故障域中一个节点）来扩展集群是满足 pod 放置规则的一种简单方法。

重要

这些要求仅与 OpenShift Data Foundation 服务相关，与这些节点上运行的其他服务、operator 或工作负载无关。

表 7.1. Red Hat OpenShift Data Foundation 的聚合可用资源的要求

部署模式	基础服务	附加设备集
内部	• 30 个 CPU（逻辑） • 72 GiB 内存 • 3 个存储设备	• 6 个 CPU（逻辑） • 15 GiB 内存 • 3 个存储设备

部署模式	基础服务	附加设备集
外部	• 4 个 CPU（逻辑） • 16 GiB 内存	Not applicable

示例：对于内部模式中带有单个设备集的 3 个节点集群，至少需要 3 x 10 = 30 个 CPU 单元。

如需更多信息，请参阅 第 6 章 *订阅* 和 CPU 单元。

有关设计 Red Hat OpenShift Data Foundation 集群的其他指导，请参阅 ODF 大小工具。

CPU 单元

在本节中，1 个 CPU 单元映射到 Kubernetes 的 1 个 CPU 单元的概念。

- 1 个 CPU 单元相当于 1 个非超线程 CPU 内核。
- 2 个 CPU 单元相当于 1 个超线程 CPU 内核。
- Red Hat OpenShift Data Foundation 基于内核的订阅总是成对提供（2 内核）。

表 7.2. IBM Power 的总最低资源要求

部署模式	基础服务
内部	• 48 个 CPU（逻辑） • 192 GiB 内存 • 3 个存储设备，每个设备需要额外 500GB 磁盘
外部	• 24 个 CPU（逻辑） • 48 GiB 内存

示例：对于内部附加设备模式中的 3 个节点集群，至少需要 48（3 x 16）个 CPU 单元，3 x 64 = 192 GB 内存。

7.3.1. IBM Z 和 IBM LinuxONE 基础架构的资源要求

Red Hat OpenShift Data Foundation 服务由一组初始的基础服务组成，并可使用附加设备集进行扩展。

所有这些 Red Hat OpenShift Data Foundation 服务 pod 都由 OpenShift Container Platform 节点上的 kubernetes 调度。以三个节点（每个故障域中一个节点）来扩展集群是满足 pod 放置规则的一种简单方法。

表 7.3. Red Hat OpenShift Data Foundation 的聚会的可用资源的要求（IBM Z 和 IBM® LinuxONE）

部署模式	基础服务	附加设备集	IBM Z 和 IBM® LinuxONE 最低硬件要求
内部	• 30 个 CPU（逻辑） ◦ 3 个具有 10 个 CPU（逻辑）的节点 • 72 GiB 内存 • 3 个存储设备	• 6 个 CPU（逻辑） • 15 GiB 内存 • 3 个存储设备	1 个 IFL
外部	• 4 个 CPU（逻辑） • 16 GiB 内存	Not applicable	Not applicable

CPU

是系统管理程序、IBM Z/VM、内核虚拟机(KVM)或两者中定义的虚拟内核数。

IFL（Linux 集成设施）

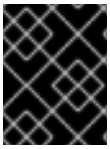
是 IBM Z 和 IBM® LinuxONE 的物理内核。

最低系统环境

- 要运行带有 1 逻辑分区(LPAR)的最小群集，需要在 6 IFL 之上再添加一个 IFL。OpenShift 容器平台使用这些 IFL。

7.3.2. 最低部署资源要求

当不符合标准部署资源要求时，将使用最低配置部署 OpenShift Data Foundation 集群。



重要

这些要求仅与 OpenShift Data Foundation 服务相关，与这些节点上运行的其他服务、operator 或工作负载无关。

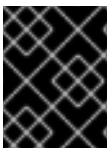
表 7.4. OpenShift Data Foundation 的聚合可用资源的要求

部署模式	基础服务
内部	• 24 个 CPU（逻辑） • 72 GiB 内存 • 3 个存储设备

如果要添加额外的设备集，我们建议将最小部署转换为标准部署。

7.3.3. 紧凑部署资源要求

Red Hat OpenShift Data Foundation 可以安装到三节点 OpenShift 紧凑裸机集群中，所有工作负载都在三个强大的 master 节点上运行。没有 worker 或存储节点。



重要

这些要求仅与 OpenShift Data Foundation 服务相关，与这些节点上运行的其他服务、operator 或工作负载无关。

表 7.5. OpenShift Data Foundation 的聚合可用资源的要求

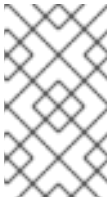
部署模式	基础服务	附加设备集
内部	• 24 个 CPU（逻辑） • 72 GiB 内存 • 3 个存储设备	• 6 个 CPU（逻辑） • 15 GiB 内存 • 3 个存储设备

要在紧凑的裸机集群中配置 OpenShift Container Platform，请参阅配置三节点集群 和 为 Edge Deployment 提供三节点架构。

7.3.4. 仅使用 MCG 部署的资源要求

仅使用 Multicloud Object Gateway(MCG)组件部署的 OpenShift Data Foundation 集群在部署时提供了灵活性，有助于减少资源消耗。

表 7.6. 仅使用 MCG 的聚合资源要求

部署模式	Core	数据库(DB)	端点
内部	• 1 个 CPU • 4 GiB 内存	• 0.5 CPU • 4 GiB 内存	• 1 个 CPU • 2 GiB 内存  **注意** 默认自动扩展介于 1 到 2 之间。

7.3.5. 使用网络文件系统的资源要求

您可以使用网络文件系统(NFS)创建导出，然后可以从 OpenShift 集群外部访问。如果您计划使用这个功能，NFS 服务会消耗 3 个 CPU 和 8Gi 内存。NFS 是可选的，并默认禁用。

NFS 卷可以通过两种方式访问：

- 集群内：被 Openshift 集群内的应用程序 pod 使用。
- 集群外：来自 Openshift 集群之外。

有关 NFS 功能的更多信息，请参阅使用 NFS 创建导出

7.3.6. 性能配置集的资源要求

OpenShift Data Foundation 提供三个性能配置集，以增强集群的性能。您可以根据部署期间的可用资源和所需的性能级别选择其中一个配置集。

表 7.7. 不同性能配置集的建议资源要求

性能配置集	CPU	内存
Swant	24	72 GiB
balanced	30	72 GiB
性能	45	96 GiB

重要

确保根据可用的可用资源选择配置集，因为您可能已经运行其他工作负载。

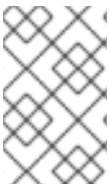
7.4. POD 放置规则

Kubernetes 根据声明性放置规则负责 pod 放置。内部集群的 Red Hat OpenShift Data Foundation 基础服务放置规则总结如下：

- 节点使用 **cluster.ocs.openshift.io/openshift-storage** 密钥标记
- 如果不存在，节点将被排序为伪故障域
- 需要高可用性的组件分散在故障域中
- 每个故障域中必须可以访问存储设备

这会产生以下要求：至少有三个节点，并且节点于三个不同的机架或区域故障域中（如果预先存在的拓扑标签）。

对于额外的设备集，在三个故障域中必须有一个存储设备和消耗它的 pod 的充足资源。可以使用手动放置规则覆盖默认放置规则，但这种方法通常仅适用于裸机部署。



注意

确保 CPU 和 RAM 资源在所有故障域中都保持可用，即使在失败时也是如此。默认情况下，分配为 6 个 CPU 和 15 GB RAM。对此默认配置的任何更改或后备存储的添加都必须在资源计划中考虑。

7.5. 存储设备要求

使用本节了解在规划内部模式部署和升级时可以考虑的不同存储容量要求。我们通常建议每个节点 12 个设备或更少。本建议可确保节点保持低于云供应商动态存储设备附加限制，以及限制使用本地存储设备的节点故障后恢复时间。以三个节点（每个故障域中一个节点）来扩展集群是满足 pod 放置规则 的一种简单方法。

存储节点应至少有两个磁盘，一个用于操作系统，其余磁盘用于 OpenShift Data Foundation 组件。

注意

您只能根据安装时所选的容量递增来扩展存储容量。

7.5.1. 动态存储设备

Red Hat OpenShift Data Foundation 允许选择 0.5 TiB、2 TiB 或 4 TiB 容量作为动态存储设备大小的请求大小。可以每个节点运行的动态存储设备数量取决于节点大小、底层置备程序限制 和资源要求。

7.5.2. 本地存储设备

对于本地存储部署，可以使用任何大小为 16 TiB 或更少的磁盘大小，并且所有磁盘的大小和类型都应相同。可以每个节点运行的本地存储设备数量是节点大小和资源要求的功能。https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation/4.18/html-single/planning_your_deployment/index#resource-requirements_rhodf以三个节点（每个故障域中一个节点）来扩展集群是满足 pod 放置规则 的一种简单方法。

注意

不支持磁盘分区，但 IBM Z 中的基于 DASD 的部署除外。

7.5.3. 容量规划

始终确保可用的存储容量保持领先于消费。如果可用存储容量已完全用尽，则需要更多的干预，而不是仅仅添加容量、删除或迁移内容。

当集群存储容量达到总容量的 75%（接近满）和 85%（满）时，会发出容量警报。始终及时处理容量警告的信息，并定期检查您的存储以确保您不会耗尽存储空间。达到 75% (接近满) 时，释放一些空间或扩展集群。当出现 85%（满）警报时，这表示您已完全耗尽存储空间，并且无法使用标准命令释放空间。如果出现这种情况，请联系红帽客户支持。

下表显示了带有动态存储设备的 Red Hat OpenShift Data Foundation 节点配置示例。

表 7.8. 带有 3 个节点的初始配置示例

存储设备大小	每个节点的存储设备	总容量	可用的存储容量
0.5 TiB	1	1.5 TiB	0.5 TiB
2 TiB	1	6 TiB	2 TiB
4 TiB	1	12 TiB	4 TiB

表 7.9. 带有 30 个节点 (N) 的扩展配置示例

存储设备大小 (D)	每个节点的存储设备 (M)	总容量 (D * M * N)	可用的存储容量 (D*M*N/3)
0.5 TiB	3	45 TiB	15 TiB
2 TiB	6	360 TiB	120 TiB
4 TiB	9	1080 TiB	360 TiB

第 8 章 网络要求

OpenShift Data Foundation 要求至少有一个网络接口，用于集群网络才能至少有 10 千兆网络速度。本节进一步涵盖规划部署的不同网络注意事项。

8.1. IPV6 支持

Red Hat OpenShift Data Foundation 版本 4.12 引入了对 IPv6 的支持。IPv6 只在单一堆栈中被支持，且不能与 IPv4 一起使用。当在 Openshift Container Platform 中打开 IPv6 时，IPv6 是 OpenShift Data Foundation 中的默认行为。

Red Hat OpenShift Data Foundation 版本 4.14 引入了 IPv6 自动检测和配置。使用 IPv6 的集群会自动配置。

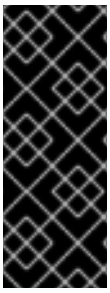
OpenShift Container Platform dual stack with Red Hat OpenShift Data Foundation IPv4 在版本 4.13 及更新的版本中支持。不支持 Red Hat OpenShift Data Foundation IPv6 上的双堆栈。

8.2. 支持多网络插件 (MULTUS)

OpenShift Data Foundation 支持在裸机基础架构上使用多网络插件 Multus 来通过隔离不同类型的网络流量来提高安全性和性能。通过使用 Multus，主机上一个或多个网络接口可能会保留来独占使用 OpenShift Data Foundation。

要使用 Multus，首先运行 Multus 先决条件验证工具。有关使用该工具的说明，请参阅 OpenShift Data Foundation - Multus 先决条件验证工具。有关 Multus 网络的更多信息，请参阅 多个网络。

您可以将 Multus 网络配置为使用 IPv4。您还可以将网络配置为使用 IPv6 是一个技术预览。Multus 网络只能配置为使用 IPv4 或 IPv6。不支持混合模式。



重要

技术预览功能为用户提供了一个对最新的产品创新的试用机会，以便用户可以对其进行测试并提供反馈。但是，Red Hat 服务等级协议不支持这些功能，其功能可能并不完善，且不适用于生产环境。由于红帽会考虑在将来的产品中使用这些技术预览功能，我们将尝试解决客户在使用这些功能时遇到的问题。

如需更多信息，请参阅技术预览功能支持范围。

8.2.1. Multus 先决条件

为了使 Ceph-CSI 与启用了 Multus 的 CephCluster 通信，Kubernetes 主机需要一些设置。

这些先决条件需要了解 Multus 网络的配置方式以及 Rook 如何使用它们。本节将有助于说明可能出现的问题。

必须满足两个基本要求：

- OpenShift 主机必须能够成功路由到 Multus 公共网络。
- Multus 公共网络上的 Pod 必须能够成功路由到 OpenShift 主机。

这两个要求可以进一步划分，如下所示：

- 要将 Kubernetes 主机路由到 Multus 公共网络，每个主机都必须确保以下内容：
  - 主机必须具有连接到 Multus 公共网络的接口（public_network_interface）

- 主机必须具有连接到 Multus 公共网络的接口(public-network-interface)。
- "public-network-interface"必须具有 IP 地址。
- 必须存在一个路由，以通过"public-network-interface"来指示用于 Multus 公共网络上的 pod 的流量。

- 对于 Multus 公共网络上的 pod 路由到 Kubernetes 主机，必须配置公共 NetworkAttachmentDefinition，以确保以下内容：
  - 该定义必须将其 IP 地址管理(IPAM)配置为路由通过网络用于节点的流量。
- 为确保两个网络之间的路由正常工作，分配给节点的 IP 地址可以与 Multus 公共网络上分配给 pod 的任何 IP 地址重叠。
- 通常，NetworkAttachmentDefinition 和节点配置必须使用相同的网络技术(Macvlan)来连接到 Multus 公共网络。

节点配置和 pod 配置是相互相关的且紧密耦合。必须同时计划这两者，而 OpenShift Data Foundation 不支持 Multus 公共网络，而无需两者。

对于这两者，"public-network-interface"必须相同。通常，连接技术(Macvlan)对于两者来说也应该相同。NetworkAttachmentDefinition 中的 IP 范围必须编码为节点上的路由，而 mirror 中，节点的 IP 范围必须在 NetworkAttachmentDefinition 中编码为路由。

有些安装可能不希望为 pod 和节点使用相同的公共网络 IP 地址范围。如果 pod 和节点有不同的范围，则必须执行额外的步骤来确保每个范围路由到另一个范围，以便它们充当一个连续的网络。这些要求需要仔细规划。请参阅 Multus 示例 来帮助理解并实现这些要求。

提示

每个存储节点通常会有十个或更多 OpenShift Data Foundation pod。pod 地址空间通常需要多次大于主机地址空间（或以上）。

OpenShift Container Platform 建议使用 NMState Operator 的 NodeNetworkConfigurationPolicies 作为配置主机要求的好方法。如果需要，也可以使用其他方法。

8.2.1.1. Multus 网络地址空间大小

网络必须具有足够的地址来考虑附加到网络的存储 pod 数量，以及一些额外的空间以考虑故障转移事件。

强烈建议您提前计划将来的存储集群扩展，并估算以后 OpenShift Container Platform 和 OpenShift Data Foundation 集群可以如何增长。为将来的扩展保留地址意味着在扩展过程中意外删除 IP 地址池的风险较低。

分配 25% 比存储集群生命周期一次需要的最大地址总数（或以上）。这有助于降低在故障切换和维护过程中耗尽 IP 地址池的风险。

为了便于编写对应的网络 CIDR 配置，还建议将总计舍入到最接近的 2 指数。

必须规划三个范围：

- 如果使用，公共网络附加定义地址空间必须包含足够的 IP 地址，用于 openshift-storage 命名空间中运行的 ODF pod 总数
- 如果使用，集群网络附加定义地址空间必须包含足够的 IP 地址，用于 openshift-storage 命名空间中运行的 OSD pod 总数

- 如果使用 Multus 公共网络，节点公共网络地址空间必须包含足够的 IP 地址，用于连接到 Multus 公共网络的 OpenShift 节点总数。

注意

如果集群为公共网络附加定义和节点公共网络附加使用统一地址空间，请将这两个要求添加到一起。这相关，例如，如果 DHCP 用于管理公共网络的 IP。

重要

对于具有无效 CIDR 的环境的用户，这是具有两个或者多个不同 CIDR 的网络，自动检测可能只找到单个 CIDR，这意味着 Ceph 守护进程可能无法启动或无法连接到网络。

8.2.1.1.1. 建议

对于大多数机构，以下建议要求：建议使用最后 6.25%(1/16)的保留专用地址空间(192.168.0.0/16)，假设范围的开头正在使用或需要。给出大约最大限制（代表 25% 的开销）。

表 8.1. Multus 建议

网络	网络范围 CIDR	大约最大值
公共网络附加定义	192.168.240.0/21	1,600 个 ODF pod
集群网络附加定义	192.168.248.0/22	800 OSDs
节点公共网络附加	192.168.252.0/23	400 个节点

8.2.1.1.2. 计算

可以确定更详细的地址空间大小，如下所示：

1. 确定未来可能需要的最大 OSD 数量。添加 25%，然后添加 5。将结果向上舍入为最接近的 2 指数。这是集群地址空间大小。
2. 从步骤 1 中计算的未舍入数开始。添加 64，然后增加 25%。将结果向上舍入为最接近的 2 指数。这是 pod 的公共地址空间大小。
3. 确定未来可能需要的最大 OpenShift 节点（包括存储节点）总数。增加了 25%将结果向上舍入为最接近的 2 指数。这是节点的公共地址空间大小。

8.2.1.2. 验证要求是否已满足

配置节点并创建 Multus 公共 NetworkAttachmentDefinition （请参阅 创建网络附加定义）检查节点配置和 NetworkAttachmentDefinition 配置是否兼容。为此，请验证每个节点是否可以通过公共网络 **ping** pod。

启动类似以下示例的 daemonset：

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: multus-public-test
```

```
  namespace: openshift-storage
  labels:
    app: multus-public-test
spec:
 selector:
   matchLabels:
     app: multus-public-test
 template:
   metadata:
     labels:
       app: multus-public-test
     annotations:
       k8s.v1.cni.cncf.io/networks: openshift-storage/public-net #
   spec:
     containers:
       - name: test
         image: quay.io/ceph/ceph:v18 # image known to have ‘ping’ installed
         command:
           - sleep
           - infinity
         resources: {}
```

使用类似以下示例的命令，列出分配给测试 pod 的 Multus 公共网络 IP。此示例命令列出了分配给所有测试 pod 的所有 IP （每个 IP 将有 2 个 IP）。在输出中，手动提取与 Multus 公共网络关联的 IP 很容易。

```
$ oc -n openshift-storage describe pod -l app=multus-public-test | grep -o -E 'Add .* from .*'
Add eth0 [10.128.2.86/23] from ovn-kubernetes
Add net1 [192.168.20.22/24] from default/public-net
Add eth0 [10.129.2.173/23] from ovn-kubernetes
Add net1 [192.168.20.29/24] from default/public-net
Add eth0 [10.131.0.108/23] from ovn-kubernetes
Add net1 [192.168.20.23/24] from default/public-net
```

在上例中，测试 Multus 公共网络上的 pod IP 是：

- 192.168.20.22
- 192.168.20.29
- 192.168.20.23

检查每个节点(NODE)是否可以通过公共网络访问所有测试 pod IP：

```
$ oc debug node/NODE
Starting pod/NODE-debug ...
To use host binaries, run `chroot /host`
Pod IP: ****
If you don't see a command prompt, try pressing enter.


sh-5.1# chroot /host


sh-5.1# ping 192.168.20.22
PING 192.168.20.22 (192.168.20.22) 56(84) bytes of data.
64 bytes from 192.168.20.22: icmp_seq=1 ttl=64 time=0.093 ms
```

```
64 bytes from 192.168.20.22: icmp_seq=2 ttl=64 time=0.056 ms
^C
--- 192.168.20.22 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1046ms
rtt min/avg/max/mdev = 0.056/0.074/0.093/0.018 ms


sh-5.1# ping 192.168.20.29
PING 192.168.20.29 (192.168.20.29) 56(84) bytes of data.
64 bytes from 192.168.20.29: icmp_seq=1 ttl=64 time=0.403 ms
64 bytes from 192.168.20.29: icmp_seq=2 ttl=64 time=0.181 ms
^C
--- 192.168.20.29 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.181/0.292/0.403/0.111 ms


sh-5.1# ping 192.168.20.23
PING 192.168.20.23 (192.168.20.23) 56(84) bytes of data.
64 bytes from 192.168.20.23: icmp_seq=1 ttl=64 time=0.329 ms
64 bytes from 192.168.20.23: icmp_seq=2 ttl=64 time=0.227 ms
^C
--- 192.168.20.23 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1047ms
rtt min/avg/max/mdev = 0.227/0.278/0.329/0.051 ms
```

如果有任何节点没有成功 ping 正在运行的 pod，则无法安全地继续。诊断并修复问题，然后重复此测试。您可能会遇到问题的一些原因包括：

- 主机可能无法正确附加到 Multus 公共网络（通过 Macvlan）
- 主机可能无法正确配置为路由到 pod IP 范围
- 公共 NetworkAttachmentDefinition 可能没有正确配置为路由回主机 IP 范围
- 主机可能具有防火墙规则阻止任一方向的连接
- 网络交换机可能具有防火墙或安全规则阻止连接

推荐的调试步骤：

- 确保节点可以相互 ping 使用公共网络"shim"IP
- 确保 **ip 地址**的输出

8.2.2. Multus 示例

此集群的相关网络计划如下：

- 专用 NIC 为 Multus 公共网络提供 eth0
- macvlan 将 OpenShift Pod 附加到 eth0
- IP 范围 192.168.0.0/16 在示例集群中可用 - pod，节点将在 Multus 公共网络中共享这个 IP 范围
- 节点将获取 IP 范围 192.168.252.0/22 （这最多允许 1024 个 Kubernetes 主机，超过示例组织需要）

- Pod 将把范围的其余部分(192.168.0.1 到 192.168.251.255)）
- 示例机构不想使用 DHCP，除非有必要，节点将在 Multus 网络中具有 IP （通过 eth0），使用 NMState operator's NodeNetworkConfigurationPolicy 资源静态分配
- 在 DHCP 不可用的情况下，W Whereabouts 将用来为 Multus 公共网络分配 IP，因为它可以轻松地开箱即用。
- OpenShift 集群中有 3 个计算节点，其上运行的 OpenShift Data Foundation 也运行：compute-0、compute-1 和 compute-2

节点的网络策略必须配置为路由到 Multus 公共网络上的 pod。

由于 Pod 将通过 Macvlan 连接，因此 Macvlan 不允许主机和 pod 互相路由，因此还必须通过 Macvlan 连接主机。通常，主机必须使用与 pod 相同的技术连接到 Multus 公共网络。Pod 连接在 Network Attachment 定义中配置。

由于主机 IP 范围是整个范围的子集，因此主机无法仅通过 IP 分配路由到 pod。必须向主机添加路由，以允许它们路由到整个 192.168.0.0/16 范围。

NodeNetworkConfigurationPolicy **desiredState** specs 将类似如下：

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: ceph-public-net-shim-compute-0
  namespace: openshift-storage
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ""
    kubernetes.io/hostname: compute-0
 desiredState:
   interfaces:
     - name: odf-pub-shim
       description: Shim interface used to connect host to OpenShift Data Foundation public Multus
network
       type: mac-vlan
       state: up
       mac-vlan:
         base-iface: eth0
         mode: bridge
         promiscuous: true
       ipv4:
         enabled: true
         dhcp: false
         address:
           - ip: 192.168.252.1 # STATIC IP FOR compute-0
             prefix-length: 22
   routes:
     config:
       - destination: 192.168.0.0/16
         next-hop-interface: odf-pub-shim
---
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
```

```
  name: ceph-public-net-shim-compute-1
  namespace: openshift-storage
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ""
    kubernetes.io/hostname: compute-1
 desiredState:
   interfaces:
     - name: odf-pub-shim
       description: Shim interface used to connect host to OpenShift Data Foundation public Multus
network
       type: mac-vlan
       state: up
       mac-vlan:
         base-iface: eth0
         mode: bridge
         promiscuous: true
       ipv4:
         enabled: true
         dhcp: false
         address:
           - ip: 192.168.252.1 # STATIC IP FOR compute-1
             prefix-length: 22
   routes:
     config:
       - destination: 192.168.0.0/16
         next-hop-interface: odf-pub-shim
---
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: ceph-public-net-shim-compute-2 # [1]
  namespace: openshift-storage
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ""
    kubernetes.io/hostname: compute-2 # [2]
 desiredState:
   interfaces: [3]
     - name: odf-pub-shim
       description: Shim interface used to connect host to OpenShift Data Foundation public Multus
network
       type: mac-vlan # [4]
       state: up
       mac-vlan:
         base-iface: eth0 # [5]
         mode: bridge
         promiscuous: true
       ipv4: # [6]
         enabled: true
         dhcp: false
         address:
           - ip: 192.168.252.2 # STATIC IP FOR compute-2 # [7]
             prefix-length: 22
   routes: # [8]
```

```
  config:
    - destination: 192.168.0.0/16 # [9]
      next-hop-interface: odf-pub-shim
```

1. 对于静态 IP 管理，每个节点都必须有一个不同的 NodeNetworkConfigurationPolicy。
2. 为每个策略选择单独的节点来配置静态网络。
3. "shim"接口用于通过与网络附加定义相同的技术将主机连接到 Multus 公共网络。
4. 主机的"shim"类型必须与 Pod 的计划相同，本例中是 **macvlan**。
5. 接口必须与 规划中选择的 Multus 公共网络接口匹配，在本例中是 **eth0**。
6. **ipv4** （或 **ipv6'**）部分在 Multus 公共网络上配置节点 IP 地址。
7. 分配给此节点的 shim 的 IP 必须与计划匹配。本例使用 192.168.252.0/22 用于 Multus 公共网络上的节点 IP。
8. 对于静态 IP 管理，请不要忘记更改每个节点的 IP。
9. **routes** 部分指示节点如何访问 Multus 公共网络上的 pod。
10. 路由目的地必须与为 pod 计划的 CIDR 范围匹配。在这种情况下，使用整个 192.168.0.0/16 范围是安全的，因为它不会影响通过其"shim"接口访问其他节点的能力。通常，这与 Multus 公共 NetworkAttachmentDefinition 中使用的 CIDR 匹配。

公共网络的 NetworkAttachmentDefinition 类似，使用 Whereabouts 的 **exclude** 选项来简化 **范围** 请求。Whereabouts **routes[].dst** 选项确保 pod 通过 Multus 公共网络路由到主机。

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: public-net
  namespace: openshift-storage
spec:
  config: '{
    "cniVersion": "0.3.1",
    "type": "macvlan", # [1]
    "master": "eth0", # [2]
    "mode": "bridge",
    "ipam": {
      "type": "whereabouts", # [3]
      "range": "192.168.0.0/16", # [4]
      "exclude": [
        "192.168.252.0/22" # [5]
      ],
      "routes": [             # [6]
        {"dst": "192.168.252.0/22"} # [7]
      ]
    }
  }'
```

1. 这必须与如何将 pod 附加到 Multus 公共网络的计划匹配。节点必须使用相同的技术 Macvlan 附加。

2. 接口必须与 规划中选择的 Multus 公共网络接口匹配，在本例中是 **eth0**。
3. 本例计划使用 whereabouts 而不是 DHCP 将 IP 分配给 pod。
4. 在本例中，决定 pod 可以在 192.168.0.0/16 范围中分配任何 IP，但分配给节点的范围除外（请参阅 5）。
5. **Whereabouts** 提供了一个 **exclude** 指令，它允许从池中轻松排除为节点分配的范围。这样可使 **范围** 指令保持简单（请参阅 4）。
6. **routes** 部分指示 pod 如何访问 Multus 公共网络中的节点。
7. 路由目的地(**dst**)必须与为节点计划的 CIDR 范围匹配。

8.2.3. holder pod 弃用

由于升级过程中对拥有者 pod 的重复维护影响（当启用 Multus 时存在拥有者 pod），因此拥有者 pod 在 ODF v4.18 版本中已弃用，并在 ODF v4.18 发行版本中删除。此弃用需要在删除拥有者 pod 前完成额外的网络配置操作。在 ODF v4.16 中，启用了 Multus 的集群会根据标准升级步骤升级到 v4.17。当 ODF 集群（启用 Multus）成功升级到 v4.17 后，管理员必须完成 禁用和删除拥有者 pod 文章中记录的步骤。请注意，这个禁用过程会消耗大量时间，但在升级到 v4.17 后，无法立即完成整个过程。在将 ODF 升级到 v4.18 前，务必要完成这个过程。

8.2.4. 使用 Multus 隔离存储流量

默认情况下，Red Hat OpenShift Data Foundation 被配置为使用 Red Hat OpenShift Software Defined Network(SDN)。默认 SDN 执行以下类型流量：

- Pod 到 pod 的流量
- 当存储是 OpenShift Data Foundation 时，pod 到存储的流量称为公共网络流量
- OpenShift Data Foundation 内部复制和重新平衡流量，称为集群网络流量

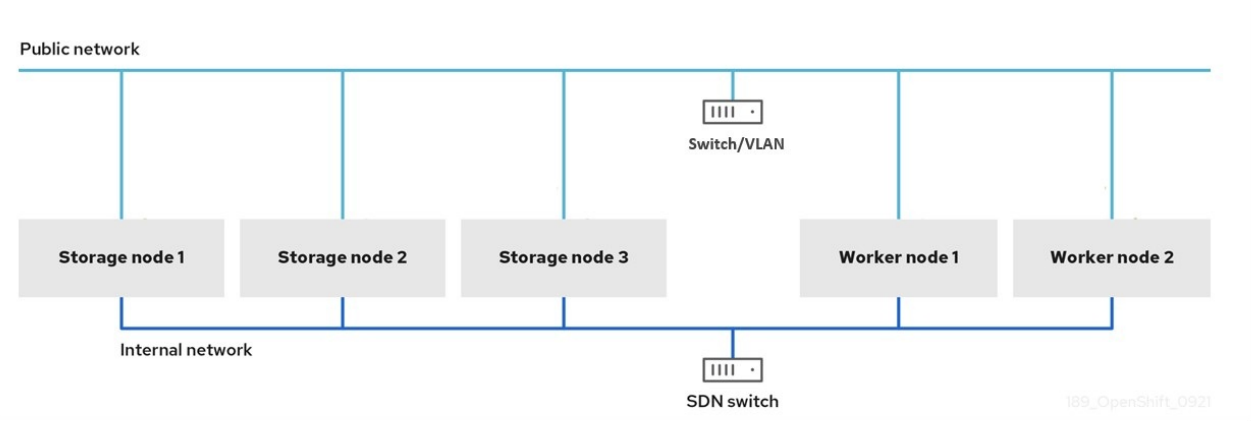
通过三种方法将 OpenShift Data Foundation 与 OpenShift 默认网络隔离：

1. 在主机上为 OpenShift Data Foundation 的公共网络保留网络接口
   - pod 到存储和内部存储复制流量在与 pod 到 pod 网络流量隔离的网络上共存。
   - 当 OpenShift Data Foundation 集群正常运行时，应用程序 pod 可以访问最大公共网络存储带宽。
   - 当 OpenShift Data Foundation 集群从失败时恢复时，应用容器集会因为持续复制和重新平衡流量而降低带宽。
2. 在主机上为 OpenShift Data Foundation 的集群网络保留网络接口
   - pod 到 pod 和 pod-to-storage 流量都继续使用 OpenShift 的默认网络。
   - Pod 到存储带宽会受到 OpenShift Data Foundation 集群的健康状态的影响较低。
   - pod 到 pod 和 pod-to-storage OpenShift Data Foundation 流量可能会在忙碌的 OpenShift 集群中进行网络带宽。
   - 存储内部网络通常具有不使用的带宽超负状态，预留给在故障期间使用。

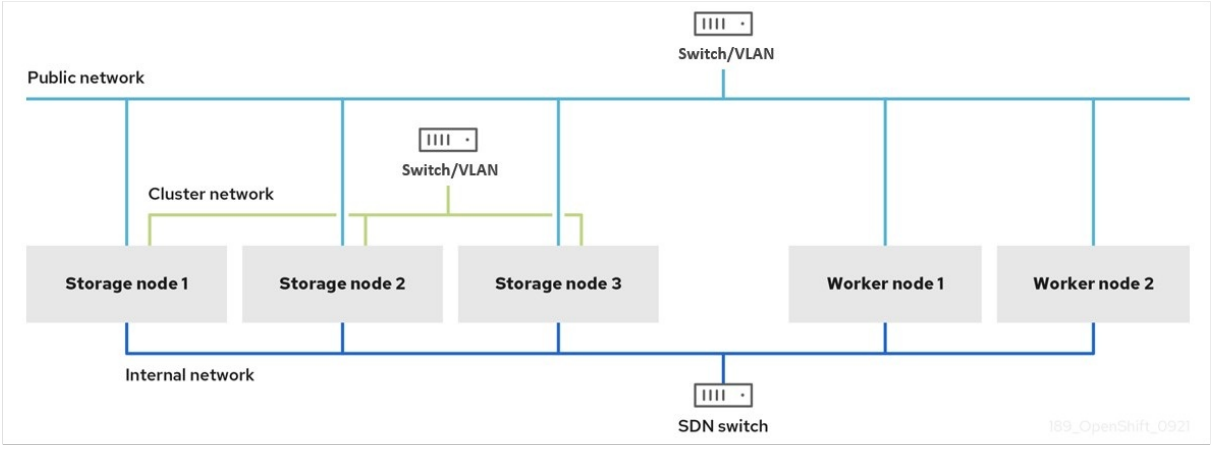
3. 在主机上为 OpenShift Data Foundation 保留两个网络接口：一个用于公共网络，一个用于集群网络

   - pod 到 pod、pod-to-storage 和存储内部流量都是隔离的，资源都没有流量类型。
   - 所有流量类型的服务级别协议可以确保。
   - 在健康运行时，保留更多网络带宽，但跨所有三个网络使用。

双网络接口隔离配置模式示例：



三网络接口完整隔离的配置模式示例：



8.2.5. 何时使用 Multus

当您需要以下内容时，使用 Multus for OpenShift Data Foundation：

改进了带有 ODF 的延迟 - Multus 始终提高了延迟。在近主机网络速度中使用主机接口，并绕过 OpenShift 软件定义的 Pod 网络。您还可以为每个接口执行每个接口级别的 Linux。

改进了 OpenShift Data Foundation 客户端数据流量和内部数据流量的带宽 Dedicated 接口。这些专用接口保留完整的带宽。

改进了 security - Multus 将存储网络流量与应用程序网络流量隔离，以提高安全性。但是，当网络共享一个接口时，带宽或性能可能无法隔离，但您可以使用 QoS 或流量整形来优先选择共享接口的带宽。

8.2.6. Multus 配置

要使用 Multus，您必须在部署 OpenShift Data Foundation 集群前创建网络附加定义(NAD)，该集群稍后附加到集群。如需更多信息，请参阅 创建网络附加定义。

要将额外网络接口附加到 pod，您必须创建配置来定义接口的附加方式。您可以使用 **NetworkAttachmentDefinition** 自定义资源（CR）来指定各个接口。每个 CR 中的 Container Network Interface (CNI)配置定义如何创建该接口。

OpenShift Data Foundation 支持 **macvlan** 驱动程序，其中包括以下功能：

- 每个连接都会获得具有其自身 MAC 地址的父接口的子接口，并与主机网络隔离。
- 使用 CPU 比 Linux 网桥或 **ipvlan** 提供更好的吞吐量。
- 网桥模式几乎总是最佳选择。
- 当网卡(NIC)支持硬件中的虚拟端口/虚拟局域网(VLAN)时，接近主机性能。

OpenShift Data Foundation 支持以下两种类型的 IP 地址管理：

Whereabouts	DHCP
使用 OpenShift/Kubernetes **租期** 来选择每个 Pod 的唯一 IP 地址。	不需要 **range** 字段。
不需要 DHCP 服务器为 Pod 提供 IP。	网络 DHCP 服务器可以为 Multus Pod 以及同一网络上的任何其他主机提供相同的范围。

小心

如果存在 DHCP 服务器，请确保 Multus 配置的 IPAM 没有给出相同的范围，以便网络上的多个 MAC 地址不能具有相同的 IP。

8.2.7. Multus 配置的要求

先决条件

- 用于公共网络的接口必须在每个 OpenShift 存储和工作节点上具有相同的接口名称，接口必须全部连接到同一底层网络。

- 用于集群网络的接口必须在每个 OpenShift 存储节点上具有相同的接口名称，接口必须全部连接到相同的底层网络。集群网络接口不必存在于 OpenShift worker 节点上。
- 用于公共或集群网络的每个网络接口都必须能够至少有 10 千兆网络速度。
- 每个网络都需要单独的虚拟局域网(VLAN)或子网。

如需了解配置基于裸机的 Multus 配置所需的步骤，请参阅创建 Multus 网络。

第 9 章 DISASTER RECOVERY

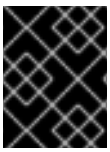
灾难恢复 (DR) 有助于机构在出现中断或紧急情况时恢复业务关键功能或正常操作。OpenShift Data Foundation 为有状态应用程序提供高可用性(HA)和 DR 解决方案，它们被广泛分为两个宽泛：

- **Metro-DR** ：单一区域和跨数据中心保护，无数据丢失。
- **Regional-DR**: 使用最小潜在数据丢失的跨区域保护
- **使用扩展集群进行灾难恢复**：单一 OpenShift Data Foundation 集群在两个不同位置之间扩展，以提供具有灾难恢复功能的存储基础架构。

9.1. METRO-DR

Metropolitan disaster recovery (Metro-DR) 由 Red Hat Advanced Cluster Management for Kubernetes (RHACM)、Red Hat Ceph Storage 和 OpenShift Data Foundation 组件组成，以便在 OpenShift Container Platform 集群中提供应用程序和数据移动性。

这个版本的 Metro-DR 功能在分散的网站之间提供卷持久数据和元数据复制。在公有云中，它们类似于防止可用性区域失败。Metro-DR 可确保在数据中心出现问题时保持业务的连续性，并不会造成数据丢失。此解决方案对 Red Hat Advanced Cluster Management(RHACM)和 OpenShift Data Foundation 高级 SKU 和相关捆绑包授权。



重要

现在，您可以使用 OpenShift Data Foundation 为基于 OpenShift virtualization 技术的工作负载轻松设置 Metropolitan 灾难恢复解决方案。如需更多信息，请参阅知识库文章。

先决条件

- Red Hat OpenShift Data Foundation 支持的灾难恢复功能需要满足以下所有先决条件，才能成功实施灾难恢复解决方案：
  - 有效的 Red Hat OpenShift Data Foundation 高级授权
  - 有效的 Red Hat Advanced Cluster Management for Kubernetes 订阅

  要了解 OpenShift Data Foundation 订阅如何工作，请参阅与 OpenShift Data Foundation 订阅相关的知识库文章。
- 确保主受管集群(Site-1)与活跃 RHACM hub 集群共存，而被动 hub 集群与二级受管集群(Site-2)在一起。或者，活跃的 RHACM hub 集群可以放在中立站点(site-3)中，不受 Site-1 主受管集群或 Site-2 次要集群的故障的影响。在这种情况下，如果使用被动 hub 集群，它可以放在 Site-2 的次要集群。

注意

Metro-DR 的 hub 恢复是一个技术预览功能，受技术预览支持限制。

有关详细解决方案要求，请参阅 Metro-DR 要求，Red Hat Ceph Storage 的部署要求具有仲裁程序 和 RHACM 要求。

9.2. REGIONAL-DR

区域灾难恢复 (Regional-DR) 由 Red Hat Advanced Cluster Management for Kubernetes (RHACM) 和 OpenShift Data Foundation 组件组成，以便在 OpenShift Container Platform 集群中提供应用程序和数据移动性。它以同步数据复制为基础，因此可能会存在潜在的数据丢失，但可为一组广泛的故障提供保护。

Red Hat OpenShift Data Foundation 由 Ceph 作为存储供应商支持，其生命周期由 Rook 管理，并增强了它的功能：

- 启用池以进行镜像。
- 在 RBD 池中自动镜像镜像。
- 提供 csi-addons 以管理每个持久性卷声明镜像。

此 Regional-DR 发行版本支持在不同的地区和数据中心中部署的多集群配置。例如，两个受管集群位于两个不同的区域或数据中心之间的双向复制。此解决方案对 Red Hat Advanced Cluster Management(RHACM)和 OpenShift Data Foundation 高级 SKU 和相关捆绑包授权。

重要

现在，您可以使用 OpenShift Data Foundation 为工作负载轻松设置区域灾难恢复解决方案。如需更多信息，请参阅知识库文章。

先决条件

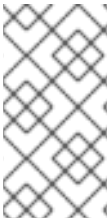
- Red Hat OpenShift Data Foundation 支持的灾难恢复功能需要满足以下所有先决条件，才能成功实施灾难恢复解决方案：
  - 有效的 Red Hat OpenShift Data Foundation 高级授权
  - 有效的 Red Hat Advanced Cluster Management for Kubernetes 订阅

  要了解 OpenShift Data Foundation 订阅如何工作，请参阅与 OpenShift Data Foundation 订阅相关的知识库文章。
- 确保主受管集群(Site-1)与活跃 RHACM hub 集群共存，而被动 hub 集群与二级受管集群(Site-2)在一起。或者，活跃的 RHACM hub 集群可以放在中立站点(site-3)中，不受 Site-1 主受管集群或 Site-2 次要集群的故障的影响。在这种情况下，如果使用被动 hub 集群，它可以放在 Site-2 的次要集群。

有关详细解决方案要求，请参阅 Regional-DR 要求 和 RHACM 要求。

9.3. 使用扩展集群进行灾难恢复

在这种情况下，单个集群将扩展到两个区域，并有第三个区域作为仲裁者的位置。此功能目前用于在 OpenShift Container Platform 内部和同一位置进行部署。对于在多个数据中心上扩展部署，不建议使用这个解决方案。相反，考虑使用 Metro-DR 作为第一个选项，以便在具有低延迟网络的多个数据中心中部署任何数据丢失 DR 解决方案。



注意

扩展集群解决方案是为在包含数据卷的区域之间不超过 10 ms 最大往返时间(RTT)的部署而设计。对于 Arbiter 节点遵循为 etcd 指定的延迟要求，请参阅 Red Hat OpenShift Container Platform 集群的指南 - 部署生成多个站点(Data Centers/Regions)。如果您计划以更高的延迟进行部署，请联系红帽客户支持。

要使用扩展集群，

- 在三个区中，必须至少有无个节点，其中：
  - 每个数据中心区使用两个节点，另一个带一个节点的区域用于仲裁区域（仲裁程序可以在主节点上）。
- 在创建集群前，所有节点必须使用 zone 标签手动标记。
  例如，这些区可以被标记为：
  - **topology.kubernetes.io/zone=arbiter** （master 节点或 worker 节点）
  - **topology.kubernetes.io/zone=datacenter1** (最少两个 worker 节点)
  - **topology.kubernetes.io/zone=datacenter2** (最少两个 worker 节点)

如需更多信息，请参阅为扩展集群配置 OpenShift Data Foundation。

要了解 OpenShift Data Foundation 订阅如何工作，请参阅与 OpenShift Data Foundation 订阅相关的知识库文章。

重要

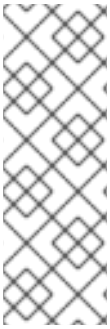
现在，您可以使用 OpenShift Data Foundation 为基于 OpenShift virtualization 技术的工作负载轻松设置灾难恢复功能。如需更多信息，请参阅 OpenShift Container Platform 指南中的 OpenShift Virtualization。

第 10 章 断开连接的环境

断开连接的环境是一个网络受限网络，Operator Lifecycle Manager (OLM) 无法访问需要互联网连接的默认 Operator Hub 和镜像 registry。

红帽支持在受限网络中安装 OpenShift Container Platform 的断开连接的环境中部署 OpenShift Data Foundation。

要在断开连接的环境中安装 OpenShift Data Foundation，请参阅 OpenShift Container Platform *文档中的 Operator 指南的 在断开连接的环境中使用 Operator Lifecycle Manager* 。



注意

在受限网络环境中安装 OpenShift Data Foundation 时，请将自定义网络时间协议 (NTP) 配置应用到节点，因为默认情况下，OpenShift Container Platform 中会假设互联网连接，**chronyd** 被配置为使用 ***.rhel.pool.ntp.org** 服务器。

如需更多信息，请参阅 OpenShift Container Platform 文档中的安装指南的配置 *chrony 时间服务* 部分，请参阅红帽知识库解决方案 A newly deployed OCS 4 cluster status shows as "Degraded ", Why?。

基于 Agent 的安装程序允许您使用镜像 registry 进行断开连接的安装。如需更多信息，请参阅准备使用基于代理的安装程序安装。

OpenShift Data Foundation 的集群日志记录

在修剪 **redhat-operator** 索引镜像时，请为 OpenShift Data Foundation 部署包含以下 operator 捆绑包：

- **ocs-operator**
- **odf-operator**
- **mcg-operator**
- **odf-csi-addons-operator**
- **ocs-client-operator**
- **odf-prometheus-operator**
- **recipe**
- **rook-ceph-operator**
- **cephcsi-operator**
- **odf-dependencies**

只适用于本地存储部署：

- **local-storage-operator**

只适用于区域灾难恢复(Regional-DR)配置或区域灾难恢复(Metro-DR)配置：

- **odf-multicluster-orchestrator**

- **odr-cluster-operator**
- **odr-hub-operator**

重要

确保将 **CatalogSource** 命名为 **redhat-operators**。

升级要求

在断开连接的环境中为 y-stream 升级的 mirror OpenShift Data Foundation 时，请确保以下内容：

- 在镜像配置文件中包含所有 OpenShift Data Foundation Operator 捆绑包。
- 在同一个 operator 目录索引镜像中打包 operator 版本、当前安装的版本和目标升级版本。例如：

```
- name: odf-operator
  channels:
  - name: stable-<target-odf-version>
    minVersion : <target-upgrade-version>-rhodf
    maxVersion : <target-upgrade-version>-rhodf
  - name: stable-<current-odf-version>
    minVersion : <current-version>-rhodf
    maxVersion : <current-version>-rhodf
- name: ocs-operator
  channels:
[...]
```

第 11 章 性能考虑和基准测试

要准确预测性能并识别 OpenShift Data Foundation 集群中的潜在瓶颈，需要基准测试。本 知识库文章解释了执行这些基准并分析结果数据的工具和程序。

第 12 章 IBM POWER 和 IBM Z 支持的功能

表 12.1. IBM Power 和 IBM Z 上支持的和不支持的功能列表

功能	IBM Power	IBM Z
紧凑部署	不支持	不支持
动态存储设备	不支持	支持
扩展的集群 - Arbiter	支持	不支持
Federal Information Processing Standard Publication (FIPS)	不支持	不支持
查看池压缩指标的功能	支持	不支持
多云对象网关(MCG)端点 pod 的自动化扩展	支持	不支持
控制过量置备的警报	支持	不支持
Ceph monitor 空间不足时的警报	支持	不支持
扩展 OpenShift Data Foundation 控制平面，允许可插拔外部存储，如 IBM Flashsystem	不支持	不支持
IPV6 支持	不支持	不支持
Multus	不支持	不支持
multicloud Object Gateway (MCG) 存储桶复制	支持	不支持
对象数据的配额支持	支持	不支持
最小部署	不支持	不支持
使用 Red Hat Advanced Cluster Management(RHACM)的 Region-Disaster Recovery(Regional-DR)	支持	不支持
使用 RHACM 的 Metro-Disaster Recovery(Metro-DR)	支持	支持
用于 Radio 访问网络的单一节点解决方案(RAN)	不支持	不支持

功能	IBM Power	IBM Z
支持网络文件系统 (NFS) 服务	支持	不支持
更改 Multicloud Object Gateway (MCG) 帐户凭证	支持	不支持
Red Hat Advanced Cluster Management 控制台中的多集群监控	支持	不支持
在 Multicloud Object Gateway 生命周期中删除过期的对象	支持	不支持
在任何 Openshift 支持的平台上部署 OpenShift Data Foundation	不支持	不支持
使用裸机基础架构置备 OpenShift Data Foundation 部署	不支持	不支持
使用 IPv4 的 OpenShift 双堆栈和 OpenShift Data Foundation	不支持	不支持
在部署过程中禁用多云对象网关外部服务	不支持	不支持
允许覆盖默认 NooBaa 后备存储的功能	支持	不支持
允许 ocs-operator 部署两个 MGR pod，一个活跃，一个备用 pod	支持	不支持
用于 brownfield 部署的灾难恢复	不支持	支持
自动扩展 RGW	不支持	不支持

第 13 章 后续步骤

要开始部署 OpenShift Data Foundation，您可以使用 OpenShift Container Platform 中的内部模式，或使用外部模式从 OpenShift Container Platform 外运行的集群提供可用服务。

根据您的要求，请转至相应的部署指南。

内部模式

- 使用 Amazon Web 服务部署 OpenShift Data Foundation
- 使用裸机部署 OpenShift Data Foundation
- 使用 VMWare vSphere 部署 OpenShift Data Foundation
- 使用 Microsoft Azure 部署 OpenShift Data Foundation
- 使用 Google Cloud 部署 OpenShift Data Foundation
- 使用 Red Hat OpenStack Platform 部署 OpenShift Data Foundation [技术预览]
- 在 IBM Power 上部署 OpenShift Data Foundation
- 在 IBM Z 上部署 OpenShift Data Foundation
- 在任何平台上部署 OpenShift Data Foundation

外部模式

- 以外部模式部署 OpenShift Data Foundation

内部或外部

有关部署多个集群，请参阅 部署多个 OpenShift Data Foundation 集群。