Red Hat OpenShift Data Foundation 4.15

Planning your deployment

Important considerations when deploying Red Hat OpenShift Data Foundation 4.15

Last Updated: 2024-04-22

Red Hat OpenShift Data Foundation 4.15 Planning your deployment

Important considerations when deploying Red Hat OpenShift Data Foundation 4.15

Legal Notice

Abstract

Read this document for important considerations when planning your Red Hat OpenShift Data Foundation deployment.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Do let us know how we can make it better.

To give feedback, create a Bugzilla ticket:

1. Go to the Bugzilla website.
2. In the **Component** section, choose **documentation**.
3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
4. Click **Submit Bug**.

CHAPTER 1. INTRODUCTION TO OPENSHIFT DATA FOUNDATION

Red Hat OpenShift Data Foundation is a highly integrated collection of cloud storage and data services for Red Hat OpenShift Container Platform. It is available as part of the Red Hat OpenShift Container Platform Service Catalog, packaged as an operator to facilitate simple deployment and management.

Red Hat OpenShift Data Foundation services are primarily made available to applications by way of storage classes that represent the following components:

- Block storage devices, catering primarily to database workloads. Prime examples include Red Hat OpenShift Container Platform logging and monitoring, and PostgreSQL.

IMPORTANT

Block storage should be used for any worklaod only when it does not require sharing the data across multiple containers.

- Shared and distributed file system, catering primarily to software development, messaging, and data aggregation workloads. Examples include Jenkins build sources and artifacts, Wordpress uploaded content, Red Hat OpenShift Container Platform registry, and messaging using JBoss AMQ.
- Multicloud object storage, featuring a lightweight S3 API endpoint that can abstract the storage and retrieval of data from multiple cloud object stores.
- On premises object storage, featuring a robust S3 API endpoint that scales to tens of petabytes and billions of objects, primarily targeting data intensive applications. Examples include the storage and access of row, columnar, and semi-structured data with applications like Spark, Presto, Red Hat AMQ Streams (Kafka), and even machine learning frameworks like TensorFlow and Pytorch.

NOTE

Running PostgresSQL workload on CephFS persistent volume is not supported and it is recommended to use RADOS Block Device (RBD) volume. For more information, see the knowledgebase solution ODF Database Workloads Must Not Use CephFS PVs/PVCs .

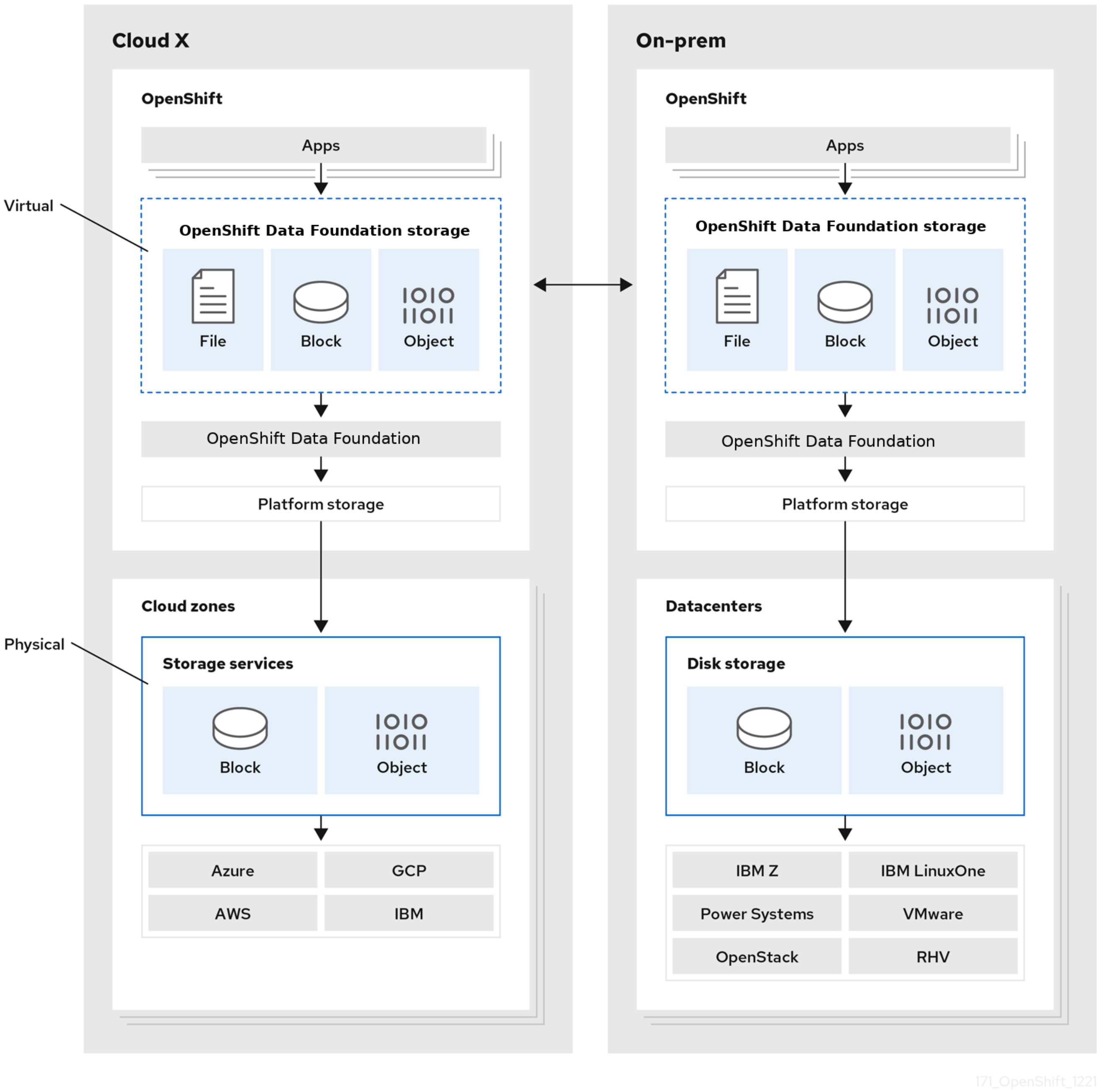
Red Hat OpenShift Data Foundation version 4.x integrates a collection of software projects, including:

- Ceph, providing block storage, a shared and distributed file system, and on-premises object storage
- Ceph CSI, to manage provisioning and lifecycle of persistent volumes and claims
- NooBaa, providing a Multicloud Object Gateway
- OpenShift Data Foundation, Rook-Ceph, and NooBaa operators to initialize and manage OpenShift Data Foundation services.

CHAPTER 2. ARCHITECTURE OF OPENSHIFT DATA FOUNDATION

Red Hat OpenShift Data Foundation provides services for, and can run internally from the Red Hat OpenShift Container Platform.

Figure 2.1. Red Hat OpenShift Data Foundation architecture



Red Hat OpenShift Data Foundation supports deployment into Red Hat OpenShift Container Platform clusters deployed on installer-provisioned or user-provisioned infrastructure.

For details about these two approaches, see OpenShift Container Platform - Installation process.

To know more about interoperability of components for Red Hat OpenShift Data Foundation and Red Hat OpenShift Container Platform, see Red Hat OpenShift Data Foundation Supportability and Interoperability Checker.

For information about the architecture and lifecycle of OpenShift Container Platform, see OpenShift Container Platform architecture.

TIP

For IBM Power, see OpenShift Container Platform - Installation process.

2.1. ABOUT OPERATORS

Red Hat OpenShift Data Foundation comprises of three main operators that codify administrative tasks and custom resources so that you can easily automate the task and resource characteristics. Administrators define the desired end state of the cluster, and the OpenShift Data Foundation operators ensure the cluster is either in that state, or approaching that state, with minimal administrator intervention.

OpenShift Data Foundation operator

This is a meta-operator that draws on other operators in specific tested ways to codify and enforce the recommendations and requirements of a supported Red Hat OpenShift Data Foundation deployment. The rook-ceph and noobaa operators provide the storage cluster resource that wraps these resources.

Rook-ceph operator

This operator automates the packaging, deployment, management, upgrading, and scaling of persistent storage and file, block, and object services. It creates block and file storage classes for all environments, and creates an object storage class and services Object Bucket Claims (OBCs) made against it in on-premises environments.

Additionally, for internal mode clusters, it provides the ceph cluster resource, which manages the deployments and services representing the following:

- Object Storage Daemons (OSDs)
- Monitors (MONs)
- Manager (MGR)
- Metadata servers (MDS)
- RADOS Object Gateways (RGWs) on-premises only

Multicloud Object Gateway operator

This operator automates the packaging, deployment, management, upgrading, and scaling of the Multicloud Object Gateway (MCG) object service. It creates an object storage class and services the OBCs made against it.

Additionally, it provides the NooBaa cluster resource, which manages the deployments and services for NooBaa core, database, and endpoint.

2.2. STORAGE CLUSTER DEPLOYMENT APPROACHES

The growing list of operating modalities is evidence that flexibility is a core tenet of Red Hat OpenShift Data Foundation. This section provides you with information that will help you to select the most appropriate approach for your environments.

You can deploy Red Hat OpenShift Data Foundation either entirely within OpenShift Container Platform (Internal approach) or to make available the services from a cluster running outside of OpenShift Container Platform (External approach).

2.2.1. Internal approach

Deployment of Red Hat OpenShift Data Foundation entirely within Red Hat OpenShift Container Platform has all the benefits of operator based deployment and management. You can use the internal-attached device approach in the graphical user interface (GUI) to deploy Red Hat OpenShift Data Foundation in internal mode using the local storage operator and local storage devices.

Ease of deployment and management are the highlights of running OpenShift Data Foundation services internally on OpenShift Container Platform. There are two different deployment modalities available when Red Hat OpenShift Data Foundation is running entirely within Red Hat OpenShift Container Platform:

- Simple
- Optimized

Simple deployment

Red Hat OpenShift Data Foundation services run co-resident with applications. The operators in Red Hat OpenShift Container Platform manages these applications.

A simple deployment is best for situations where,

- Storage requirements are not clear.
- Red Hat OpenShift Data Foundation services runs co-resident with the applications.
- Creating a node instance of a specific size is difficult, for example, on bare metal.

For Red Hat OpenShift Data Foundation to run co-resident with the applications, the nodes must have local storage devices, or portable storage devices attached to them dynamically, like EBS volumes on EC2, or vSphere Virtual Volumes on VMware, or SAN volumes.

NOTE

PowerVC dynamically provisions the SAN volumes.

Optimized deployment

Red Hat OpenShift Data Foundation services run on dedicated infrastructure nodes. Red Hat OpenShift Container Platform manages these infrastructure nodes.

An optimized approach is best for situations when,

- Storage requirements are clear.
- Red Hat OpenShift Data Foundation services run on dedicated infrastructure nodes.
- Creating a node instance of a specific size is easy, for example, on cloud, virtualized environment, and so on.

2.2.2. External approach

Red Hat OpenShift Data Foundation exposes the Red Hat Ceph Storage services running outside of the OpenShift Container Platform cluster as storage classes.

The external approach is best used when,

- Storage requirements are significant (600+ storage devices).
- Multiple OpenShift Container Platform clusters need to consume storage services from a common external cluster.
- Another team, Site Reliability Engineering (SRE), storage, and so on, needs to manage the external cluster providing storage services. Possibly a pre-existing one.

2.3. NODE TYPES

Nodes run the container runtime, as well as services, to ensure that the containers are running, and maintain network communication and separation between the pods. In OpenShift Data Foundation, there are three types of nodes.

Table 2.1. Types of nodes

Node Type	Description
Master	These nodes run processes that expose the Kubernetes API, watch and schedule newly created pods, maintain node health and quantity, and control interaction with underlying cloud providers.
Infrastructure (Infra)	Infra nodes run cluster level infrastructure services such as logging, metrics, registry, and routing. These are optional in OpenShift Container Platform clusters. In order to separate OpenShift Data Foundation layer workload from applications, ensure that you use infra nodes for OpenShift Data Foundation in virtualized and cloud environments. To create Infra nodes, you can provision new nodes labeled as **infra**. For more information, see How to use dedicated worker nodes for Red Hat OpenShift Data Foundation
Worker	Worker nodes are also known as application nodes since they run applications. When OpenShift Data Foundation is deployed in internal mode, you require a minimal cluster of 3 worker nodes. Make sure that the nodes are spread across 3 different racks, or availability zones, to ensure availability. In order for OpenShift Data Foundation to run on worker nodes, you need to attach the local storage devices, or portable storage devices to the worker nodes dynamically. When OpenShift Data Foundation is deployed in external mode, it runs on multiple nodes. This allows Kubernetes to reschedule on the available nodes in case of a failure.

NOTE

OpenShift Data Foundation requires the same number of subsciptions as OpenShift Container Platform. However, if OpenShift Data Foundation is running on infra nodes, OpenShift does not require OpenShift Container Platform subscription for these nodes. Therefore, the OpenShift Data Foundation control plane does not require additional OpenShift Container Platform and OpenShift Data Foundation subscriptions. For more information, see Chapter 6, *Subscriptions*.

CHAPTER 3. INTERNAL STORAGE SERVICES

Red Hat OpenShift Data Foundation service is available for consumption internally to the Red Hat OpenShift Container Platform that runs on the following infrastructure:

- Amazon Web Services (AWS)
- Bare metal
- VMware vSphere
- Microsoft Azure
- Google Cloud
- Red Hat OpenStack 13 or higher (installer-provisioned infrastructure) [Technology Preview]
- IBM Power
- IBM Z and IBM® LinuxONE

Creation of an internal cluster resource results in the internal provisioning of the OpenShift Data Foundation base services, and makes additional storage classes available to the applications.

CHAPTER 4. EXTERNAL STORAGE SERVICES

Red Hat OpenShift Data Foundation can use IBM FlashSystems or make services from an external Red Hat Ceph Storage cluster available for consumption through OpenShift Container Platform clusters running on the following platforms:

- VMware vSphere
- Bare metal
- Red Hat OpenStack platform (Technology Preview)
- IBM Power
- IBM Z

The OpenShift Data Foundation operators create and manage services to satisfy Persistent Volume (PV) and Object Bucket Claims (OBCs) against the external services. External cluster can serve block, file and object storage classes for applications that run on OpenShift Container Platform. The operators do not deploy or manage the external clusters.

CHAPTER 5. SECURITY CONSIDERATIONS

5.1. FIPS-140-2

The Federal Information Processing Standard Publication 140-2 (FIPS-140-2) is a standard that defines a set of security requirements for the use of cryptographic modules. Law mandates this standard for the US government agencies and contractors and is also referenced in other international and industry specific standards.

Red Hat OpenShift Data Foundation now uses the FIPS validated cryptographic modules. Red Hat Enterprise Linux OS/CoreOS (RHCOS) delivers these modules.

Currently, the Cryptographic Module Validation Program (CMVP) processes the cryptography modules. You can see the state of these modules at Modules in Process List. For more up-to-date information, see the Red Hat Knowledgebase solution RHEL core crypto components.

NOTE

Enable the FIPS mode on the OpenShift Container Platform, before you install OpenShift Data Foundation. OpenShift Container Platform must run on the RHCOS nodes, as the feature does not support OpenShift Data Foundation deployment on Red Hat Enterprise Linux 7 (RHEL 7).

For more information, see *Installing a cluster in FIPS mode* and *Support for FIPS cryptography* of the Installing guide in OpenShift Container Platform documentation.

5.2. PROXY ENVIRONMENT

A proxy environment is a production environment that denies direct access to the internet and provides an available HTTP or HTTPS proxy instead. Red Hat Openshift Container Platform is configured to use a proxy by modifying the proxy object for existing clusters or by configuring the proxy settings in the install-config.yaml file for new clusters.

Red Hat supports deployment of OpenShift Data Foundation in proxy environments when OpenShift Container Platform has been configured according to configuring the cluster-wide proxy.

5.3. DATA ENCRYPTION OPTIONS

Encryption lets you encode your data to make it impossible to read without the required encryption keys. This mechanism protects the confidentiality of your data in the event of a physical security breach that results in a physical media to escape your custody. The per-PV encryption also provides access protection from other namespaces inside the same OpenShift Container Platform cluster. Data is encrypted when it is written to the disk, and decrypted when it is read from the disk. Working with encrypted data might incur a small penalty to performance.

Encryption is only supported for new clusters deployed using Red Hat OpenShift Data Foundation 4.6 or higher. An existing encrypted cluster that is not using an external Key Management System (KMS) cannot be migrated to use an external KMS.

Previously, HashiCorp Vault was the only supported KMS for Cluster-wide and Persistent Volume encryptions. With OpenShift Data Foundation 4.7.0 and 4.7.1, only HashiCorp Vault Key/Value (KV) secret engine API, version 1 is supported. Starting with OpenShift Data Foundation 4.7.2, HashiCorp

Vault KV secret engine API, versions 1 and 2 are supported. As of OpenShift Data Foundation 4.12, Thales CipherTrust Manager has been introduced as an additional supported KMS.

IMPORTANT

- KMS is required for StorageClass encryption, and is optional for cluster-wide encryption.
- To start with, Storage class encryption requires a valid Red Hat OpenShift Data Foundation Advanced subscription. For more information, see the knowledgebase article on OpenShift Data Foundation subscriptions .

Red Hat works with the technology partners to provide this documentation as a service to the customers. However, Red Hat does not provide support for the Hashicorp product. For technical assistance with this product, contact Hashicorp.

5.3.1. Cluster-wide encryption

Red Hat OpenShift Data Foundation supports cluster-wide encryption (encryption-at-rest) for all the disks and Multicloud Object Gateway operations in the storage cluster. OpenShift Data Foundation uses Linux Unified Key System (LUKS) version 2 based encryption with a key size of 512 bits and the **aes-xts-plain64** cipher where each device has a different encryption key. The keys are stored using a Kubernetes secret or an external KMS. Both methods are mutually exclusive and you can not migrate between methods.

Encryption is disabled by default for block and file storage. You can enable encryption for the cluster at the time of deployment. The MultiCloud Object Gateway supports encryption by default. See the deployment guides for more information.

Cluster wide encryption is supported in OpenShift Data Foundation 4.6 without Key Management System (KMS). Starting with OpenShift Data Foundation 4.7, it supports with and without HashiCorp Vault KMS. Starting with OpenShift Data Foundation 4.12, it supports with and without both HashiCorp Vault KMS and Thales CipherTrust Manager KMS.

NOTE

Requires a valid Red Hat OpenShift Data Foundation Advanced subscription. To know how subscriptions for OpenShift Data Foundation work, see knowledgebase article on OpenShift Data Foundation subscriptions.

Cluster wide encryption with HashiCorp Vault KMS provides two authentication methods:

- **Token**: This method allows authentication using vault tokens. A kubernetes secret containing the vault token is created in the openshift-storage namespace and is used for authentication. If this authentication method is selected then the administrator has to provide the vault token that provides access to the backend path in Vault, where the encryption keys are stored.
- **Kubernetes**: This method allows authentication with vault using serviceaccounts. If this authentication method is selected then the administrator has to provide the name of the role configured in Vault that provides access to the backend path, where the encryption keys are stored. The value of this role is then added to the **ocs-kms-connection-details** config map. This method is available from OpenShift Data Foundation 4.10.
Currently, HashiCorp Vault is the only supported KMS. With OpenShift Data Foundation 4.7.0 and 4.7.1, only HashiCorp Vault KV secret engine, API version 1 is supported. Starting with

OpenShift Data Foundation 4.7.2, HashiCorp Vault KV secret engine API, versions 1 and 2 are supported.

NOTE

OpenShift Data Foundation on IBM Cloud platform supports Hyper Protect Crypto Services (HPCS) Key Management Services (KMS) as the encryption solution in addition to HashiCorp Vault KMS.

IMPORTANT

Red Hat works with the technology partners to provide this documentation as a service to the customers. However, Red Hat does not provide support for the Hashicorp product. For technical assistance with this product, contact Hashicorp.

5.3.2. Storage class encryption

You can encrypt persistent volumes (block only) with storage class encryption using an external Key Management System (KMS) to store device encryption keys. Persistent volume encryption is only available for RADOS Block Device (RBD) persistent volumes. See how to create a storage class with persistent volume encryption.

Storage class encryption is supported in OpenShift Data Foundation 4.7 or higher with HashiCorp Vault KMS. Storage class encryption is supported in OpenShift Data Foundation 4.12 or higher with both HashiCorp Vault KMS and Thales CipherTrust Manager KMS.

NOTE

Requires a valid Red Hat OpenShift Data Foundation Advanced subscription. To know how subscriptions for OpenShift Data Foundation work, see knowledgebase article on OpenShift Data Foundation subscriptions.

5.3.3. CipherTrust manager

Red Hat OpenShift Data Foundation version 4.12 introduced Thales CipherTrust Manager as an additional Key Management System (KMS) provider for your deployment. Thales CipherTrust Manager provides centralized key lifecycle management. CipherTrust Manager supports Key Management Interoperability Protocol (KMIP), which enables communication between key management systems.

CipherTrust Manager is enabled during deployment.

5.3.4. Data encryption in-transit via Red Hat Ceph Storage’s messenger version 2 protocol

Starting with OpenShift Data Foundation version 4.14, Red Hat Ceph Storage’s messenger version 2 protocol can be used to encrypt data in-transit. This provides an important security requirement for your infrastructure. In-transit encryption can be enabled during deployment.

5.4. ENCRYPTION IN TRANSIT

You need to enable IPsec so that all the network traffic between the nodes on the OVN-Kubernetes Container Network Interface (CNI) cluster network travels through an encrypted tunnel.

By default, IPsec is disabled. You can enable it either during or after installing the cluster. If you need to enable IPsec after cluster installation, you must first resize your cluster MTU to account for the overhead of the IPsec ESP IP header.

For more information on how to configure the IPsec encryption, see *Configuring IPsec encryption* of the Networking guide in OpenShift Container Platform documentation.

CHAPTER 6. SUBSCRIPTIONS

6.1. SUBSCRIPTION OFFERINGS

Red Hat OpenShift Data Foundation subscription is based on “core-pairs,” similar to Red Hat OpenShift Container Platform. The Red Hat OpenShift Data Foundation 2-core subscription is based on the number of logical cores on the CPUs in the system where OpenShift Container Platform runs.

As with OpenShift Container Platform:

- OpenShift Data Foundation subscriptions are stackable to cover larger hosts.
- Cores can be distributed across as many virtual machines (VMs) as needed. For example, ten 2-core subscriptions will provide 20 cores and in case of IBM Power a 2-core subscription at SMT level of 8 will provide 2 cores or 16 vCPUs that can be used across any number of VMs.
- OpenShift Data Foundation subscriptions are available with Premium or Standard support.

6.2. DISASTER RECOVERY SUBSCRIPTION REQUIREMENT

Disaster Recovery features supported by Red Hat OpenShift Data Foundation require all of the following prerequisites to successfully implement a disaster recovery solution:

- A valid Red Hat OpenShift Data Foundation Advanced entitlement
- A valid Red Hat Advanced Cluster Management for Kubernetes subscription

Any Red Hat OpenShift Data Foundation Cluster containing PVs participating in active replication either as a source or destination requires OpenShift Data Foundation Advanced entitlement. This subscription should be active on both source and destination clusters.

To know how subscriptions for OpenShift Data Foundation work, see knowledgebase article on OpenShift Data Foundation subscriptions.

6.3. CORES VERSUS VCPUS AND HYPERTHREADING

Making a determination about whether or not a particular system consumes one or more cores is currently dependent on whether or not that system has hyperthreading available. Hyperthreading is only a feature of Intel CPUs. Visit the Red Hat Customer Portal to determine whether a particular system supports hyperthreading.

For systems where hyperthreading is enabled and where one hyperthread equates to one visible system core, the calculation of cores is a ratio of 2 cores to 4 vCPUs. Therefore, a 2-core subscription covers 4 vCPUs in a hyperthreaded system. A large virtual machine (VM) might have 8 vCPUs, equating to 4 subscription cores. As subscriptions come in 2-core units, you will need two 2-core subscriptions to cover these 4 cores or 8 vCPUs.

Where hyperthreading is not enabled, and where each visible system core correlates directly to an underlying physical core, the calculation of cores is a ratio of 2 cores to 2 vCPUs.

6.3.1. Cores versus vCPUs and simultaneous multithreading (SMT) for IBM Power

Making a determination about whether or not a particular system consumes one or more cores is currently dependent on the level of simultaneous multithreading configured (SMT). IBM Power provides simultaneous multithreading levels of 1, 2, 4 or 8 for each core which correspond to the number of vCPUs as in the table below.

Table 6.1. Different SMT levels and their corresponding vCPUs

SMT level	SMT=1	SMT=2	SMT=4	SMT=8
1 Core	# vCPUs=1	# vCPUs=2	# vCPUs=4	# vCPUs=8
2 Cores	# vCPUs=2	# vCPUs=4	# vCPUs=8	# vCPUs=16
4 Cores	# vCPUs=4	# vCPUs=8	# vCPUs=16	# vCPUs=32

For systems where SMT is configured the calculation for the number of cores required for subscription purposes depends on the SMT level. Therefore, a 2-core subscription corresponds to 2 vCPUs on SMT level of 1, and to 4 vCPUs on SMT level of 2, and to 8 vCPUs on SMT level of 4 and to 16 vCPUs on SMT level of 8 as seen in the table above. A large virtual machine (VM) might have 16 vCPUs, which at a SMT level 8 will require a 2 core subscription based on dividing the # of vCPUs by the SMT level (16 vCPUs / 8 for SMT-8 = 2). As subscriptions come in 2-core units, you will need one 2-core subscription to cover these 2 cores or 16 vCPUs.

6.4. SPLITTING CORES

Systems that require an odd number of cores need to consume a full 2-core subscription. For example, a system that is calculated to require only 1 core will end up consuming a full 2-core subscription once it is registered and subscribed.

When a single virtual machine (VM) with 2 vCPUs uses hyperthreading resulting in 1 calculated vCPU, a full 2-core subscription is required; a single 2-core subscription may not be split across two VMs with 2 vCPUs using hyperthreading. See section Cores versus vCPUs and hyperthreading for more information.

It is recommended that virtual instances be sized so that they require an even number of cores.

6.4.1. Shared Processor Pools for IBM Power

IBM Power have a notion of shared processor pools. The processors in a shared processor pool can be shared across the nodes in the cluster. The aggregate compute capacity required for a Red Hat OpenShift Data Foundation should be a multiple of core-pairs.

6.5. SUBSCRIPTION REQUIREMENTS

Red Hat OpenShift Data Foundation components can run on either OpenShift Container Platform worker or infrastructure nodes, for which you can use either Red Hat CoreOS (RHCOS) or Red Hat Enterprise Linux (RHEL) 8.4 as the host operating system. RHEL 7 is now deprecated. OpenShift Data Foundation subscriptions are required for every OpenShift Container Platform subscribed core with a ratio of 1:1.

When using infrastructure nodes, the rule to subscribe all OpenShift worker node cores for OpenShift Data Foundation applies even though they don’t need any OpenShift Container Platform or any OpenShift Data Foundation subscriptions. You can use labels to state whether a node is a worker or an infrastructure node.

For more information, see How to use dedicated worker nodes for Red Hat OpenShift Data Foundation in the Managing and Allocating Storage Resources guide.

CHAPTER 7. INFRASTRUCTURE REQUIREMENTS

7.1. PLATFORM REQUIREMENTS

Red Hat OpenShift Data Foundation 4.15 is supported only on OpenShift Container Platform version 4.15 and its next minor versions.

Bug fixes for previous version of Red Hat OpenShift Data Foundation will be released as bug fix versions. For more details, see the Red Hat OpenShift Container Platform Life Cycle Policy .

For external cluster subscription requirements, see the Red Hat Knowledgebase article OpenShift Data Foundation Subscription Guide.

For a complete list of supported platform versions, see the Red Hat OpenShift Data Foundation Supportability and Interoperability Checker.

7.1.1. Amazon EC2

Supports internal Red Hat OpenShift Data Foundation clusters only.

An Internal cluster must meet both, storage device requirements and have a storage class that provides, EBS storage via the aws-ebs provisioner.

OpenShift Data Foundation supports **gp2-csi** and **gp3-csi** drivers that were introduced by Amazon Web Services (AWS). These drivers offer better storage expansion capabilities and a reduced monthly price point (**gp3-csi**). You can now select the new drivers when selecting your storage class. In case a high throughput is required, **gp3-csi** is recommended to be used when deploying OpenShift Data Foundation.

If you need a high input/output operation per second (IOPS), the recommended EC2 instance types are **D2** or **D3**.

7.1.2. Bare Metal

Supports internal clusters and consuming external clusters.

An internal cluster must meet both the storage device requirements and have a storage class that provide local SSD (NVMe/SATA/SAS, SAN) via the Local Storage Operator.

7.1.3. VMware vSphere

Supports internal clusters and consuming external clusters.

Recommended versions:

- vSphere 6.7, Update 2 or later
- vSphere 7.0 or later.

For more details, see the VMware vSphere infrastructure requirements.

NOTE

If VMware ESXi does not recognize its devices as flash, mark them as flash devices. Before Red Hat OpenShift Data Foundation deployment, refer to Mark Storage Devices as Flash.

Additionally, an Internal cluster must meet both the, storage device requirements and have a storage class providing either,

- vSAN or VMFS datastore via the vsphere-volume provisioner
- VMDK, RDM, or DirectPath storage devices via the Local Storage Operator.

7.1.4. Microsoft Azure

Supports internal Red Hat OpenShift Data Foundation clusters only.

An internal cluster must meet both, storage device requirements and have a storage class that provides, an zzure disk via the azure-disk provisioner.

7.1.5. Google Cloud

Supports internal Red Hat OpenShift Data Foundation clusters only.

An internal cluster must meet both, storage device requirements and have a storage class that provides, a GCE Persistent Disk via the gce-pd provisioner.

7.1.6. Red Hat OpenStack Platform [Technology Preview]

Supports internal Red Hat OpenShift Data Foundation clusters and consuming external clusters.

An internal cluster must meet both, storage device requirements and have a storage class that provides a standard disk via the Cinder provisioner.

7.1.7. IBM Power

Supports internal Red Hat OpenShift Data Foundation clusters and consuming external clusters.

An Internal cluster must meet both, storage device requirements and have a storage class providing local SSD (NVMe/SATA/SAS, SAN) via the Local Storage Operator.

7.1.8. IBM Z and IBM® LinuxONE

Supports internal Red Hat OpenShift Data Foundation clusters. Also, supports external mode where Red Hat Ceph Storage is running on x86.

An Internal cluster must meet both, storage device requirements and have a storage class providing local SSD (NVMe/SATA/SAS, SAN) via the Local Storage Operator.

7.2. EXTERNAL MODE REQUIREMENT

7.2.1. Red Hat Ceph Storage

To check the supportability and interoperability of Red Hat Ceph Storage (RHCS) with Red Hat OpenShift Data Foundation in external mode, go to the lab Red Hat OpenShift Data Foundation Supportability and Interoperability Checker.

1. Select **Service Type** as **ODF as Self-Managed Service**.
2. Select appropriate **Version** from the drop down.
3. On the Versions tab, click the **Supported RHCS Compatibility** tab.

For instructions regarding how to install a RHCS cluster, see the installation guide.

7.2.2. IBM FlashSystem

To use IBM FlashSystem as a pluggable external storage on other providers, you need to first deploy it before you can deploy OpenShift Data Foundation, which would use the IBM FlashSystem storage class as a backing storage.

For the latest supported FlashSystem storage systems and versions, see IBM ODF FlashSystem driver documentation.

For instructions on how to deploy OpenShift Data Foundation, see Creating an OpenShift Data Foundation Cluster for external IBM FlashSystem storage.

7.3. RESOURCE REQUIREMENTS

Red Hat OpenShift Data Foundation services consist of an initial set of base services, and can be extended with additional device sets. All of these Red Hat OpenShift Data Foundation services pods are scheduled by kubernetes on OpenShift Container Platform nodes. Expanding the cluster in multiples of three, one node in each failure domain, is an easy way to satisfy the pod placement rules.

IMPORTANT

These requirements relate to OpenShift Data Foundation services only, and not to any other services, operators or workloads that are running on these nodes.

Table 7.1. Aggregate avaliable resource requirements for Red Hat OpenShift Data Foundation only

Deployment Mode	Base services	Additional device Set
Internal	● 30 CPU (logical) ● 72 GiB memory ● 3 storage devices	● 6 CPU (logical) ● 15 GiB memory ● 3 storage devices
External	● 4 CPU (logical) ● 16 GiB memory	Not applicable

Example: For a 3 node cluster in an internal mode deployment with a single device set, a minimum of 3 x 10 = 30 units of CPU are required.

For more information, see Chapter 6, *Subscriptions* and CPU units.

For additional guidance with designing your Red Hat OpenShift Data Foundation cluster, see the ODF Sizing Tool.

CPU units

In this section, 1 CPU Unit maps to the Kubernetes concept of 1 CPU unit.

- 1 unit of CPU is equivalent to 1 core for non-hyperthreaded CPUs.
- 2 units of CPU are equivalent to 1 core for hyperthreaded CPUs.
- Red Hat OpenShift Data Foundation core-based subscriptions always come in pairs (2 cores).

Table 7.2. Aggregate minimum resource requirements for IBM Power

Deployment Mode	Base services
Internal	• 48 CPU (logical) • 192 GiB memory • 3 storage devices, each with additional 500GB of disk
External	• 24 CPU (logical) • 48 GiB memory

Example: For a 3 node cluster in an internal-attached devices mode deployment, a minimum of 3 x 16 = 48 units of CPU and 3 x 64 = 192 GB of memory is required.

7.3.1. Resource requirements for IBM Z and IBM LinuxONE infrastructure

Red Hat OpenShift Data Foundation services consist of an initial set of base services, and can be extended with additional device sets.

All of these Red Hat OpenShift Data Foundation services pods are scheduled by kubernetes on OpenShift Container Platform nodes . Expanding the cluster in multiples of three, one node in each failure domain, is an easy way to satisfy the pod placement rules.

Table 7.3. Aggregate available resource requirements for Red Hat OpenShift Data Foundation only (IBM Z and IBM® LinuxONE)

Deployment Mode	Base services	Additional device Set	IBM Z and IBM® LinuxONE minimum hardware requirements
Internal	• 30 CPU (logical) ◦ 3 nodes with 10 CPUs (logical) each • 72 GiB memory • 3 storage devices	• 6 CPU (logical) • 15 GiB memory • 3 storage devices	1 IFL
External	• 4 CPU (logical) • 16 GiB memory	Not applicable	Not applicable

CPU

Is the number of virtual cores defined in the hypervisor, IBM Z/VM, Kernel Virtual Machine (KVM), or both.

IFL (Integrated Facility for Linux)

Is the physical core for IBM Z and IBM® LinuxONE.

Minimum system environment

- In order to operate a minimal cluster with 1 logical partition (LPAR), one additional IFL is required on top of the 6 IFLs. OpenShift Container Platform consumes these IFLs .

7.3.2. Minimum deployment resource requirements

An OpenShift Data Foundation cluster will be deployed with minimum configuration when the standard deployment resource requirement is not met.

IMPORTANT

These requirements relate to OpenShift Data Foundation services only, and not to any other services, operators or workloads that are running on these nodes.

Table 7.4. Aggregate resource requirements for OpenShift Data Foundation only

Deployment Mode	Base services

Deployment Mode	Base services
Internal	24 CPU (logical) 72 GiB memory 3 storage devices

If you want to add additional device sets, we recommend converting your minimum deployment to standard deployment.

7.3.3. Compact deployment resource requirements

Red Hat OpenShift Data Foundation can be installed on a three-node OpenShift compact bare metal cluster, where all the workloads run on three strong master nodes. There are no worker or storage nodes.

IMPORTANT

These requirements relate to OpenShift Data Foundation services only, and not to any other services, operators or workloads that are running on these nodes.

Table 7.5. Aggregate resource requirements for OpenShift Data Foundation only

Deployment Mode	Base services	Additional device Set
Internal	24 CPU (logical) 72 GiB memory 3 storage devices	6 CPU (logical) 15 GiB memory 3 storage devices

To configure OpenShift Container Platform on a compact bare metal cluster, see Configuring a three-node cluster and Delivering a Three-node Architecture for Edge Deployments.

7.3.4. Resource requirements for MCG only deployment

An OpenShift Data Foundation cluster deployed only with the Multicloud Object Gateway (MCG) component provides the flexibility in deployment and helps to reduce the resource consumption.

Table 7.6. Aggregate resource requirements for MCG only deployment

Deployment Mode	Core	Database (DB)	Endpoint

Deployment Mode	Core	Database (DB)	Endpoint
Internal	• 1 CPU • 4 GiB memory	• 0.5 CPU • 4 GiB memory	• 1 CPU • 2 GiB memory **NOTE** The defaut auto scale is between 1 - 2.

7.3.5. Resource requirements for using Network File system

You can create exports using Network File System (NFS) that can then be accessed externally from the OpenShift cluster. If you plan to use this feature, the NFS service consumes 3 CPUs and 8Gi of Ram. NFS is optional and is disabled by default.

The NFS volume can be accessed two ways:

- In-cluster: by an application pod inside of the Openshift cluster.
- Out of cluster: from outside of the Openshift cluster.

For more information about the NFS feature, see Creating exports using NFS

7.3.6. Resource requirements for performance profiles

OpenShift Data Foundation provides three performance profiles to enhance the performance of the clusters. You can choose one of these profiles based on your available resources and desired performance level during deployment or post deployment.

Table 7.7. Recommended resource requirement for different performance profiles

Performance profile	CPU	Memory
Lean	24	72 GiB
Balanced	30	72 GiB
Performance	45	96 GiB

IMPORTANT

Make sure to select the profiles based on the available free resources as you might already be running other workloads.

7.4. POD PLACEMENT RULES

Kubernetes is responsible for pod placement based on declarative placement rules. The Red Hat OpenShift Data Foundation base service placement rules for Internal cluster can be summarized as follows:

- Nodes are labeled with the **cluster.ocs.openshift.io/openshift-storage** key
- Nodes are sorted into pseudo failure domains if none exist
- Components requiring high availability are spread across failure domains
- A storage device must be accessible in each failure domain

This leads to the requirement that there be at least three nodes, and that nodes be in three distinct rack or zone failure domains in the case of pre-existing topology labels.

For additional device sets, there must be a storage device, and sufficient resources for the pod consuming it, in each of the three failure domains. Manual placement rules can be used to override default placement rules, but generally this approach is only suitable for bare metal deployments.

7.5. STORAGE DEVICE REQUIREMENTS

Use this section to understand the different storage capacity requirements that you can consider when planning internal mode deployments and upgrades. We generally recommend 12 devices or less per node. This recommendation ensures both that nodes stay below cloud provider dynamic storage device attachment limits, and to limit the recovery time after node failures with local storage devices. Expanding the cluster in multiples of three, one node in each failure domain, is an easy way to satisfy pod placement rules.

Storage nodes should have at least two disks, one for the operating system and the remaining disks for OpenShift Data Foundation components.

NOTE

You can expand the storage capacity only in the increment of the capacity selected at the time of installation.

7.5.1. Dynamic storage devices

Red Hat OpenShift Data Foundation permits the selection of either 0.5 TiB, 2 TiB or 4 TiB capacities as the request size for dynamic storage device sizes. The number of dynamic storage devices that can run per node is a function of the node size, underlying provisioner limits and resource requirements.

7.5.2. Local storage devices

For local storage deployment, any disk size of 16 TiB or less can be used, and all disks should be of the same size and type. The number of local storage devices that can run per node is a function of the node size and resource requirements. Expanding the cluster in multiples of three, one node in each failure domain, is an easy way to satisfy pod placement rules.

NOTE

Disk partitioning is not supported.

7.5.3. Capacity planning

Always ensure that available storage capacity stays ahead of consumption. Recovery is difficult if available storage capacity is completely exhausted, and requires more intervention than simply adding capacity or deleting or migrating content.

Capacity alerts are issued when cluster storage capacity reaches 75% (near-full) and 85% (full) of total capacity. Always address capacity warnings promptly, and review your storage regularly to ensure that you do not run out of storage space. When you get to 75% (near-full), either free up space or expand the cluster. When you get the 85% (full) alert, it indicates that you have run out of storage space completely and cannot free up space using standard commands. At this point, contact Red Hat Customer Support.

The following tables show example node configurations for Red Hat OpenShift Data Foundation with dynamic storage devices.

Table 7.8. Example initial configurations with 3 nodes

Storage Device size	Storage Devices per node	Total capacity	Usable storage capacity
0.5 TiB	1	1.5 TiB	0.5 TiB
2 TiB	1	6 TiB	2 TiB
4 TiB	1	12 TiB	4 TiB

Table 7.9. Example of expanded configurations with 30 nodes (N)

Storage Device size (D)	Storage Devices per node (M)	Total capacity (D * M * N)	Usable storage capacity (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

CHAPTER 8. NETWORK REQUIREMENTS

Use this section to understand the different network considerations when planning deployments.

8.1. IPV6 SUPPORT

Red Hat OpenShift Data Foundation version 4.12 introduced the support of IPv6. IPv6 is supported in single stack only, and cannot be used simultaneously with IPv4. IPv6 is the default behavior in OpenShift Data Foundation when IPv6 is turned on in Openshift Container Platform.

Red Hat OpenShift Data Foundation version 4.14 introduces IPv6 auto detection and configuration. Clusters using IPv6 will automatically be configured accordingly.

OpenShift Container Platform dual stack with Red Hat OpenShift Data Foundation IPv4 is supported from version 4.13 and later. Dual stack on Red Hat OpenShift Data Foundation IPv6 is not supported.

8.2. MULTI NETWORK PLUG-IN (MULTUS) SUPPORT

OpenShift Data Foundation supports the ability to use multi-network plug-in Multus on bare metal infrastructures to improve security and performance by isolating the different types of network traffic. By using Multus, one or more network interfaces on hosts can be reserved for exclusive use of OpenShift Data Foundation.

To use Multus, first run the Multus prerequisite validation tool. For instructions to use the tool, see OpenShift Data Foundation - Multus prerequisite validation tool . For more information about Multus networks, see Multiple networks

8.2.1. Segregating storage traffic using Multus

By default, Red Hat OpenShift Data Foundation is configured to use the Red Hat OpenShift Software Defined Network (SDN). The default SDN carries the following types of traffic:

- Pod-to-pod traffic
- Pod-to-storage traffic, known as public network traffic when the storage is OpenShift Data Foundation
- OpenShift Data Foundation internal replication and rebalancing traffic, known as cluster network traffic

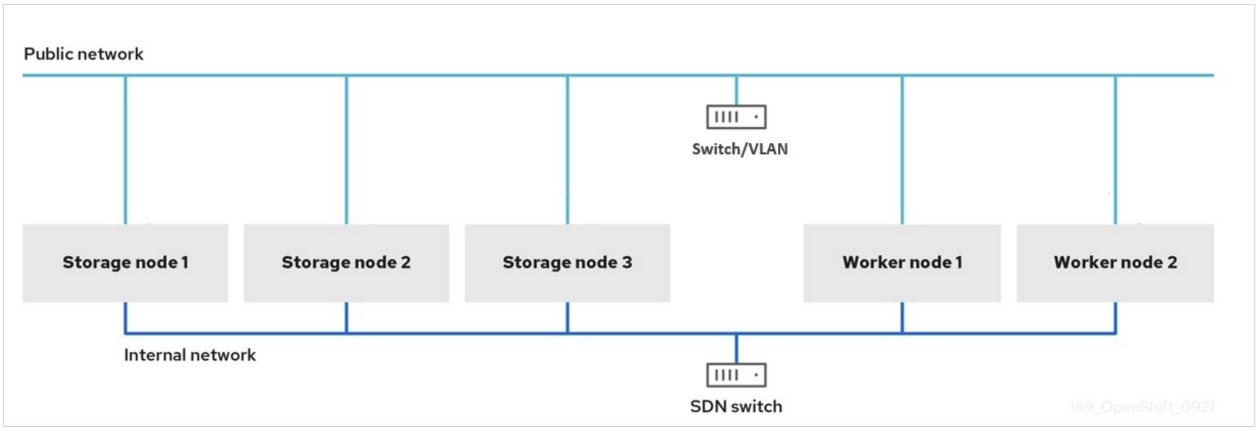
There are three ways to segregate OpenShift Data Foundation from OpenShift default network:

1. Reserve a network interface on the host for the public network of OpenShift Data Foundation
   - Pod-to-storage and internal storage replication traffic coexist on a network that is isolated from pod-to-pod network traffic.
   - Application pods have access to the maximum public network storage bandwidth when the OpenShift Data Foundation cluster is healthy.
   - When the OpenShift Data Foundation cluster is recovering from failure, the application pods will have reduced bandwidth due to ongoing replication and rebalancing traffic.
2. Reserve a network interface on the host for OpenShift Data Foundation’s cluster network

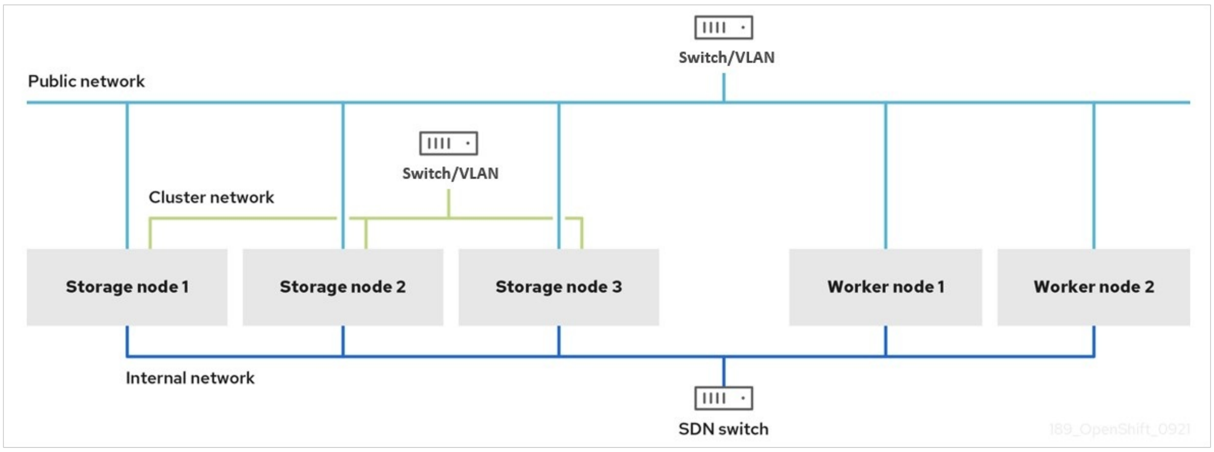
   - Pod-to-pod and pod-to-storage traffic both continue to use OpenShift’s default network.
   - Pod-to-storage bandwidth is less affected by the health of the OpenShift Data Foundation cluster.
   - Pod-to-pod and pod-to-storage OpenShift Data Foundation traffic might contend for network bandwidth in busy OpenShift clusters.
   - The storage internal network often has an overabundance of bandwidth that is unused, reserved for use during failures.

3. Reserve two network interfaces on the host for OpenShift Data Foundation: one for the public network and one for the cluster network
   - Pod-to-pod, pod-to-storage, and storage internal traffic are all isolated, and none of the traffic types will contend for resources.
   - Service level agreements for all traffic types are more able to be ensured.
   - During healthy runtime, more network bandwidth is reserved but unused across all three networks.

Dual network interface segregated configuration schematic example:



Triple network interface full segregated configuration schematic example:



8.2.2. When to use Multus

Use Multus for OpenShift Data Foundation when you need the following:

Improved latency - Multus with ODF always improves latency. Use host interfaces at near-host network speeds and bypass OpenShift's software-defined Pod network. You can also perform Linux per interface level tuning for each interface.

Improved bandwidth - Dedicated interfaces for OpenShift Data Foundation client data traffic and internal data traffic. These dedicated interfaces reserve full bandwidth.

Improved security - Multus isolates storage network traffic from application network traffic for added security. Bandwidth or performance might not be isolated when networks share an interface, however, you can use QoS or traffic shaping to prioritize bandwidth on shared interfaces.

8.2.3. Multus configuration

To use Multus, you must create network attachment definitions (NADs) before deploying the OpenShift Data Foundation cluster, which is later attached to the cluster. For more information, see Creating network attachment definitions.

To attach additional network interfaces to a pod, you must create configurations that define how the interfaces are attached. You specify each interface by using a **NetworkAttachmentDefinition** custom resource (CR). A Container Network Interface (CNI) configuration inside each of these CRs defines how that interface is created.

OpenShift Data Foundation supports two types of drivers. The following tables describes the drivers and their features:

macvlan (recommended)	**ipvlan**

Each connection gets a sub-interface of the parent interface with its own MAC address and is isolated from the host network.	Each connection gets its own IP address and shares the same MAC address.
Uses less CPU and provides better throughput than Linux bridge or **ipvlan**.	**L2** mode is analogous to **macvlan** bridge mode.
Almost always require bridge mode.	**L3** mode is analogous to a router existing on the parent interface. **L3** is useful for Border Gateway Protocol (BGP), otherwise use macvlan for reduced CPU and better throughput.
Near-host performance when network interface card (NIC) supports virtual ports/virtual local area networks (VLANs) in hardware.	If NIC does not support VLANs in hardware, performance might be better than **macvlan**.

OpenShift Data Foundation supports the following two types IP address management:

whereabouts	DHCP
Uses OpenShift/Kubernetes **leases** to select unique IP addresses per Pod.	Does not require **range** field.
Does not require a DHCP server to provide IPs for Pods.	Network DHCP server can give out the same range to Multus Pods as well as any other hosts on the same network.

CAUTION

If there is a DHCP server, ensure Multus configured IPAM does not give out the same range so that multiple MAC addresses on the network cannot have the same IP.

8.2.4. Requirements for Multus configuration

Prerequisites

- The interface used for the public network must have the same interface name on each OpenShift storage and worker node, and the interfaces must all be connected to the same underlying network.
- The interface used for the cluster network must have the same interface name on each OpenShift storage node, and the interfaces must all be connected to the same underlying network. Cluster network interfaces do not have to be present on the OpenShift worker nodes.
- Each network interface used for the public or cluster network must be capable of at least 10 gigabit network speeds.
- Each network requires a separate virtual local area network (VLAN) or subnet.

See Creating Multus networks for the necessary steps to configure a Multus based configuration on bare metal.

CHAPTER 9. DISASTER RECOVERY

Disaster Recovery (DR) helps an organization to recover and resume business critical functions or normal operations when there are disruptions or disasters. OpenShift Data Foundation provides High Availability (HA) & DR solutions for stateful apps which are broadly categorized into two broad categories:

- **Metro-DR**: Single Region and cross data center protection with no data loss.
- **Disaster Recovery with stretch cluster**: Single OpenShift Data Foundation cluster is stretched between two different locations to provide the storage infrastructure with disaster recovery capabilities.
- **Regional-DR**: Cross Region protection with minimal potential data loss.

9.1. METRO-DR

Metropolitan disaster recovery (Metro-DR) is composed of Red Hat Advanced Cluster Management for Kubernetes (RHACM), Red Hat Ceph Storage and OpenShift Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters.

This release of Metro-DR solution provides volume persistent data and metadata replication across sites that are geographically dispersed. In the public cloud these would be similar to protecting from an Availability Zone failure. Metro-DR ensures business continuity during the unavailability of a data center with no data loss. This solution is entitled with Red Hat Advanced Cluster Management (RHACM) and OpenShift Data Foundation Advanced SKUs and related bundles.

IMPORTANT

You can now easily set up Metropolitan disaster recovery solutions for workloads based on OpenShift virtualization technology using OpenShift Data Foundation. For more information, see the knowledgebase article.

Prerequisites

- Disaster Recovery features supported by Red Hat OpenShift Data Foundation require all of the following prerequisites in order to successfully implement a Disaster Recovery solution:
  - A valid Red Hat OpenShift Data Foundation Advanced entitlement
  - A valid Red Hat Advanced Cluster Management for Kubernetes subscription

  To know how subscriptions for OpenShift Data Foundation work, see knowledgebase article on OpenShift Data Foundation subscriptions.
- Ensure that the primary managed cluster (Site-1) is co-situated with the active RHACM hub cluster while the passive hub cluster is situated along with the secondary managed cluster (Site-2). Alternatively, the active RHACM hub cluster can be placed in a neutral site (Site-3) that is not impacted by the failures of either of the primary managed cluster at Site-1 or the secondary cluster at Site-2. In this situation, if a passive hub cluster is used it can be placed with the secondary cluster at Site-2.

NOTE

Hub recovery is a Technology Preview feature and is subject to Technology Preview support limitations.

For detailed solution requirements, see Metro-DR requirements, deployment requirements for Red Hat Ceph Storage stretch cluster with arbiter and RHACM requirements.

9.2. DISASTER RECOVERY WITH STRETCH CLUSTER

In this case, a single cluster is stretched across two zones with a third zone as the location for the arbiter. This feature is currently intended for deployment in the OpenShift Container Platform on-premises and in the same location. This solution is not recommended for deployments stretching over multiple data centers. Instead, consider Metro-DR as a first option for no data loss DR solution deployed over multiple data centers with low latency networks.

NOTE

The stretch cluster solution is designed for deployments where latencies do not exceed 10 ms maximum round-trip time (RTT) between the zones containing data volumes. For Arbiter nodes follow the latency requirements specified for etcd, see Guidance for Red Hat OpenShift Container Platform Clusters - Deployments Spanning Multiple Sites(Data Centers/Regions). Contact Red Hat Customer Support if you are planning to deploy with higher latencies.

To use the stretch cluster,

- You must have a minimum of five nodes across three zones, where:
  - Two nodes per zone are used for each data-center zone, and one additional zone with one node is used for arbiter zone (the arbiter can be on a master node).
- All the nodes must be manually labeled with the zone labels prior to cluster creation. For example, the zones can be labeled as:
  - **topology.kubernetes.io/zone=arbiter** (master or worker node)
  - **topology.kubernetes.io/zone=datacenter1** (minimum two worker nodes)
  - **topology.kubernetes.io/zone=datacenter2** (minimum two worker nodes)

For more information, see Configuring OpenShift Data Foundation for stretch cluster .

To know how subscriptions for OpenShift Data Foundation work, see knowledgebase article on OpenShift Data Foundation subscriptions.

9.3. REGIONAL-DR

Regional disaster recovery (Regional-DR) is composed of Red Hat Advanced Cluster Management for Kubernetes (RHACM) and OpenShift Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters. It is built on Asynchronous data replication and hence could have a potential data loss but provides the protection against a broad set of failures.

Red Hat OpenShift Data Foundation is backed by Ceph as the storage provider, whose lifecycle is managed by Rook and it’s enhanced with the ability to:

- Enable pools for mirroring.
- Automatically mirror images across RBD pools.
- Provides csi-addons to manage per Persistent Volume Claim mirroring.

This release of Regional-DR supports Multi-Cluster configuration that is deployed across different regions and data centers. For example, a 2-way replication across two managed clusters located in two different regions or data centers. This solution is entitled with Red Hat Advanced Cluster Management (RHACM) and OpenShift Data Foundation Advanced SKUs and related bundles.

Prerequisites

- Disaster Recovery features supported by Red Hat OpenShift Data Foundation require all of the following prerequisites in order to successfully implement a Disaster Recovery solution:
    - A valid Red Hat OpenShift Data Foundation Advanced entitlement
    - A valid Red Hat Advanced Cluster Management for Kubernetes subscription

  To know how subscriptions for OpenShift Data Foundation work, see knowledgebase article on OpenShift Data Foundation subscriptions.

- Ensure that the primary managed cluster (Site-1) is co-situated with the active RHACM hub cluster while the passive hub cluster is situated along with the secondary managed cluster (Site-2). Alternatively, the active RHACM hub cluster can be placed in a neutral site (Site-3) that is not impacted by the failures of either of the primary managed cluster at Site-1 or the secondary cluster at Site-2. In this situation, if a passive hub cluster is used it can be placed with the secondary cluster at Site-2.

  NOTE

  Hub recovery is a Technology Preview feature and is subject to Technology Preview support limitations.

For detailed solution requirements, see Regional-DR requirements and RHACM requirements.

CHAPTER 10. DISCONNECTED ENVIRONMENT

Disconnected environment is a network restricted environment where the Operator Lifecycle Manager (OLM) cannot access the default Operator Hub and image registries, which require internet connectivity.

Red Hat supports deployment of OpenShift Data Foundation in disconnected environments where you have installed OpenShift Container Platform in restricted networks.

To install OpenShift Data Foundation in a disconnected environment, see *Using Operator Lifecycle Manager on restricted networks* of the Operators guide in OpenShift Container Platform documentation.

NOTE

When you install OpenShift Data Foundation in a restricted network environment, apply a custom Network Time Protocol (NTP) configuration to the nodes, because by default, internet connectivity is assumed in OpenShift Container Platform and **chronyd** is configured to use the ***.rhel.pool.ntp.org** servers.

For more information, see the Red Hat Knowledgebase solution A newly deployed OCS 4 cluster status shows as "Degraded", Why? and *Configuring chrony time service* of the Installing guide in OpenShift Container Platform documentation.

Red Hat OpenShift Data Foundation version 4.12 introduced the Agent-based Installer for disconnected environment deployment. The Agent-based Installer allows you to use a mirror registry for disconnected installations. For more information, see Preparing to install with Agent-based Installer .

Packages to include for OpenShift Data Foundation

When you prune the **redhat-operator** index image, include the following list of packages for the OpenShift Data Foundation deployment:

- **ocs-operator**
- **odf-operator**
- **mcg-operator**
- **odf-csi-addons-operator**
- **odr-cluster-operator**
- **odr-hub-operator**
- Optional: **local-storage-operator**
  Only for local storage deployments.
- Optional: **odf-multicluster-orchestrator**
  Only for Regional Disaster Recovery (Regional-DR) configuration.

IMPORTANT

Name the **CatalogSource** as **redhat-operators**.

CHAPTER 11. SUPPORTED AND UNSUPPORTED FEATURES FOR IBM POWER AND IBM Z

Table 11.1. List of supported and unsupported features on IBM Power and IBM Z

Features	IBM Power	IBM Z
Compact deployment	Unsupported	Unsupported
Dynamic storage devices	Unsupported	Supported
Stretched Cluster - Arbiter	Supported	Unsupported
Federal Information Processing Standard Publication (FIPS)	Unsupported	Unsupported
Ability to view pool compression metrics	Supported	Unsupported
Automated scaling of Multicloud Object Gateway (MCG) endpoint pods	Supported	Unsupported
Alerts to control overprovision	Supported	Unsupported
Alerts when Ceph Monitor runs out of space	Supported	Unsupported
Extended OpenShift Data Foundation control plane which allows pluggable external storage such as IBM Flashsystem	Unsupported	Unsupported
IPV6 support	Unsupported	Unsupported
Multus	Unsupported	Unsupported
Multicloud Object Gateway (MCG) bucket replication	Supported	Unsupported
Quota support for object data	Supported	Unsupported
Minimum deployment	Unsupported	Unsupported
Regional-Disaster Recovery (Regional-DR) with Red Hat Advanced Cluster Management (RHACM)	Supported	Unsupported

Features	IBM Power	IBM Z
Metro-Disaster Recovery (Metro-DR) multiple clusters with RHACM	Supported	Supported
Single Node solution for Radio Access Network (RAN)	Unsupported	Unsupported
Support for network file system (NFS) services	Supported	Unsupported
Ability to change Multicloud Object Gateway (MCG) account credentials	Supported	Unsupported
Multicluster monitoring in Red Hat Advanced Cluster Management console	Supported	Unsupported
Deletion of expired objects in Multicloud Object Gateway lifecycle	Supported	Unsupported
Agnostic deployment of OpenShift Data Foundation on any Openshift supported platform	Unsupported	Unsupported
Installer provisioned deployment of OpenShift Data Foundation using bare metal infrastructure	Unsupported	Unsupported
Openshift dual stack with OpenShift Data Foundation using IPv4	Unsupported	Unsupported
Ability to disable Multicloud Object Gateway external service during deployment	Unsupported	Unsupported
Ability to allow overriding of default NooBaa backing store	Supported	Unsupported
Allowing ocs-operator to deploy two MGR pods, one active and one standby	Supported	Unsupported
Disaster Recovery for brownfield deployments	Unsupported	Supported

CHAPTER 12. NEXT STEPS

To start deploying your OpenShift Data Foundation, you can use the internal mode within OpenShift Container Platform or use external mode to make available services from a cluster running outside of OpenShift Container Platform.

Depending on your requirement, go to the respective deployment guides.

Internal mode

- Deploying OpenShift Data Foundation using Amazon web services
- Deploying OpenShift Data Foundation using Bare Metal
- Deploying OpenShift Data Foundation using VMWare vSphere
- Deploying OpenShift Data Foundation using Microsoft Azure
- Deploying OpenShift Data Foundation using Google Cloud
- Deploying OpenShift Data Foundation using Red Hat OpenStack Platform [Technology Preview]
- Deploying OpenShift Data Foundation on IBM Power
- Deploying OpenShift Data Foundation on IBM Z

External mode

- Deploying OpenShift Data Foundation in external mode