Red Hat OpenShift Data Foundation 4.12

Monitoring OpenShift Data Foundation

View cluster health, metrics, or set alerts.

Last Updated: 2024-03-06

Red Hat OpenShift Data Foundation 4.12 Monitoring OpenShift Data Foundation

View cluster health, metrics, or set alerts.

Legal Notice

Abstract

Read this document for instructions on monitoring Red Hat OpenShift Data Foundation using the Block and File, and Object dashboards.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 4

CHAPTER 1. CLUSTER HEALTH 5
1.1. VERIFYING OPENSHIFT DATA FOUNDATION IS HEALTHY 5
1.2. STORAGE HEALTH LEVELS AND CLUSTER STATE 5
1.2.1. Block and File dashboard indicators 5
1.2.2. Object dashboard indicators 6
1.2.3. Alert panel 6

CHAPTER 2. MULTICLUSTER STORAGE HEALTH 7
2.1. ENABLING MULTICLUSTER DASHBOARD ON HUB CLUSTER 7
2.2. VERIFYING MULTICLUSTER STORAGE HEALTH ON HUB CLUSTER 8

CHAPTER 3. METRICS 9
3.1. METRICS IN THE BLOCK AND FILE DASHBOARD 9
3.2. METRICS IN THE OBJECT DASHBOARD 11
3.3. POOL METRICS 14
3.4. ENABLING METADATA ON RBD AND CEPHFS VOLUMES 15

CHAPTER 4. ALERTS 24
4.1. SETTING UP ALERTS 24

CHAPTER 5. REMOTE HEALTH MONITORING 25
5.1. ABOUT TELEMETRY 25
5.2. INFORMATION COLLECTED BY TELEMETRY 25

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. CLUSTER HEALTH

1.1. VERIFYING OPENSHIFT DATA FOUNDATION IS HEALTHY

Storage health is visible on the Block and File and Object dashboards.

Procedure

1. In the OpenShift Web Console, click **Storage → Data Foundation**.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
3. Check if the **Status** card has a green tick in the **Block and File** and the **Object** tabs. Green tick indicates that the cluster is healthy.

See Section 1.2, "Storage health levels and cluster state" for information about the different health states and the alerts that appear.

1.2. STORAGE HEALTH LEVELS AND CLUSTER STATE

Status information and alerts related to OpenShift Data Foundation are displayed in the storage dashboards.

1.2.1. Block and File dashboard indicators

The **Block and File** dashboard shows the complete state of OpenShift Data Foundation and the state of persistent volumes.

The states that are possible for each resource type are listed in the following table.

Table 1.1. OpenShift Data Foundation health levels

State	Icon	Description
UNKNOWN	?	OpenShift Data Foundation is not deployed or unavailable.
Green Tick	✓	Cluster health is good.
Warning	⚠	OpenShift Data Foundation cluster is in a warning state. In internal mode, an alert will be displayed along with the issue details. Alerts are not displayed for external mode.

State	Icon	Description
Error	❗	OpenShift Data Foundation cluster has encountered an error and some component is nonfunctional. In internal mode, an alert is displayed along with the issue details. Alerts are not displayed for external mode.

1.2.2. Object dashboard indicators

The **Object** dashboard shows the state of the Multicloud Object Gateway and any object claims in the cluster.

The states that are possible for each resource type are listed in the following table.

Table 1.2. Object Service health levels

State	Description
Green Tick ✓	Object storage is healthy.
Multicloud Object Gateway is not running	Shown when NooBaa system is not found.
All resources are unhealthy	Shown when all NooBaa pools are unhealthy.
Many buckets have issues	Shown when >= 50% of buckets encounter error(s).
Some buckets have issues	Shown when >= 30% of buckets encounter error(s).
Unavailable	Shown when network issues and/or errors exist.

1.2.3. Alert panel

The Alert panel appears below the **Status** card in both the **Block and File** dashboard and the **Object** dashboard when the cluster state is not healthy.

Information about specific alerts and how to respond to them is available in Troubleshooting OpenShift Data Foundation.

CHAPTER 2. MULTICLUSTER STORAGE HEALTH

To view the overall storage health status across all the clusters with OpenShift Data Foundation and manage its capacity, you must first enable the multicluster dashboard on the Hub cluster.

2.1. ENABLING MULTICLUSTER DASHBOARD ON HUB CLUSTER

You can enable the multicluster dashboard on the install screen either before or after installing ODF Multicluster Orchestrator version 4.12 with the console plugin.

Prerequisites

- Ensure that you have installed OpenShift Container Platform version 4.12 and have administrator privileges.
- Ensure that you have installed Multicluster Orchestrator 4.12 operator with plugin for console enabled.
- Ensure that you have installed Red Hat Advanced Cluster Management for Kubernetes 2.6 from Operator Hub. For instructions on how to install, see Installing RHACM.
- Ensure you have enabled observability on RHACM. See Enabling observability guidelines.

Procedure

1. Create the configmap file named **observability-metrics-custom-allowlist.yaml** and add the name of the custom metric to the **metrics_list.yaml** parameter.
   You can use the following YAML to list the OpenShift Data Foundation metrics on Hub cluster. For details, see Adding custom metrics.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: observability-metrics-custom-allowlist
  Namespace: open-cluster-management-observability
data:
  metrics_list.yaml: |
    names:
      - odf_system_health_status
      - odf_system_map
      - odf_system_raw_capacity_total_bytes
      - odf_system_raw_capacity_used_bytes
    matches:
      - __name__="csv_succeeded",exported_namespace="openshift-storage",name=~"odf-
operator.*"
```

2. Run the following command in the **open-cluster-management-observability** namespace:

```
# oc apply -n open-cluster-management-observability -f observability-metrics-custom-
allowlist.yaml
```

After **observability-metrics-custom-allowlist** yaml is created, RHACM will start collecting the listed OpenShift Data Foundation metrics from all the managed clusters.

If you want to exclude specific managed clusters from collecting the observability data, add the following cluster label to your clusters: **observability: disabled**.

3. To view the multicluster health, see chapter verifying multicluster storage dashboard .

2.2. VERIFYING MULTICLUSTER STORAGE HEALTH ON HUB CLUSTER

Prerequisites

Ensure that you have enabled multicluster monitoring. For instructions, see chapter Enabling multicluster dashboard.

Procedure

1. In the OpenShift web console of Hub cluster, ensure **All Clusters** is selected.
2. Navigate to **Data Services** and click **Storage System**.
3. On the Overview tab, verify that there are green ticks in front of **OpenShift Data Foundation** and **Systems**. This indicates that the operator is running and all storage systems are available.
4. In the Status card,
   a. Click **OpenShift Data Foundation** to view the operator status.
   b. Click **Systems** to view the storage system status.
5. The **Storage system capacity** card shows the following details:
   - Name of the storage system
   - Cluster name
   - Graphical representation of total and used capacity in percentage
   - Actual values for total and used capacity in TiB

CHAPTER 3. METRICS

3.1. METRICS IN THE BLOCK AND FILE DASHBOARD

You can navigate to the Block and File dashboard in the OpenShift Web Console as follows:

1. Click **Storage → Data Foundation**.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
3. Click the **Block and File** tab.

The following cards on the Block and File dashboard provides the metrics based on deployment mode (internal or external):

Details card

The Details card shows the following:

- Service Name
- Cluster name
- The name of the Provider on which the system runs (example: **AWS**, **VSphere**, **None** for Bare metal)
- Mode (deployment mode as either Internal or External)
- OpenShift Data Foundation operator version.

Inventory card

The Inventory card shows the number of active nodes, PVCs and PVs backed by OpenShift Data Foundation provisioner. On the left hand side of the card, total number of storage nodes, PVCs and PVs are displayed. While on the corresponding right hand side of the card, number of storage nodes in Not Ready state, count of PVCs in Pending state and PVs in Released state are shown.

> **NOTE**
>
> For external mode, the number of nodes will be 0 by default, since there are no dedicated nodes for OpenShift Data Foundation.

Status card

This card shows whether the cluster is up and running without any errors or is experiencing some issues.
For internal mode, Data Resiliency indicates the status of data re-balancing in Ceph across the replicas. When the internal mode cluster is in a warning or error state, the Alerts section is shown along with the relevant alerts.

For external mode, Data Resiliency and alerts will not be displayed

Raw Capacity card

This card shows the total raw storage capacity which includes replication, on the cluster.

- **Used** legend indicates space used raw storage capacity on the cluster
- **Available** legend indicates the available raw storage capacity on the cluster.

NOTE

This card is not applicable for external mode clusters.

Used Capacity Breakdown card

This card shows the actual amount of non-replicated data stored in the cluster and its distribution. You can choose between Projects, Storage Classes and Pods from the drop-down menu on the top of the card. These options are for filtering the data shown in the graph. The graph displays the used capacity for only the top five entities, based on usage. The aggregate usage of the remaining entities is displayed as Other.

Option	Display
Projects	The aggregated capacity of each project which is using the OpenShift Data Foundation and how much is being used.
Storage Classes	The aggregate capacity based on OpenShift Data Foundation based storage classes.
Pods	All the pods trying to use the PVC backed by OpenShift Data Foundation provisioner.

For external mode, see the **Capacity breakdown card**.

Capacity breakdown card

This card is only applicable for external mode clusters. In this card, you can view graphic breakdown of capacity per project, storage classes and pods. You can choose between Projects, Storage Classes and Pods from the drop-down menu on the top of the card. These options are for filtering the data shown in the graph. The graph displays the used capacity for only the top five entities, based on usage. The aggregate usage of the remaining entities is displayed as Other.

Utilization card

The card shows used capacity, input/output operations per second, latency, throughput, and recovery information for the internal mode cluster.

For external mode, this card shows only the used and requested capacity details for that cluster.

Storage Efficiency card

This card shows the compression ratio that represents a compressible data effectiveness metric inclusive of all compression-enabled pools. It also shows the savings metric that represents the actual disk capacity saved inclusive of all compression-enabled pools and associated replicas.

Activity card

This card shows what activities are happening or have recently happened in the OpenShift Data Foundation cluster. The card is separated into two sections:

- **Ongoing**: Displays the progress of ongoing activities related to rebuilding of data resiliency and upgrading of OpenShift Data Foundation operator.
- **Recent Events**: Displays the list of events that happened in the **openshift-storage** namespace.

3.2. METRICS IN THE OBJECT DASHBOARD

You can navigate to the Object dashboard in the OpenShift Web Console as follows:

1. Click **Storage → Data Foundation**.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
3. Click the **Object** tab.

The following metrics are available in the Object dashboard:

Details card

This card shows the following information:

- **Service Name**: The Multicloud Object Gateway (MCG) service name.
- **System Name**: The Multicloud Object Gateway and RADOS Object Gateway system names. The Multicloud Object Gateway system name is also a hyperlink to the MCG management user interface.
- **Provider**: The name of the provider on which the system runs (example: **AWS**, **VSphere**, **None** for Baremetal)
- **Version**: OpenShift Data Foundation operator version.

Storage Efficiency card

In this card you can view how the MCG optimizes the consumption of the storage backend resources through deduplication and compression and provides you with a calculated efficiency ratio (application data vs logical data) and an estimated savings figure (how many bytes the MCG did not send to the storage provider) based on capacity of bare metal and cloud based storage and egress of cloud based storage.

Buckets card

Buckets are containers maintained by the MCG and RADOS Object Gateway to store data on behalf of the applications. These buckets are created and accessed through object bucket claims (OBCs). A specific policy can be applied to bucket to customize data placement, data spill-over, data resiliency, capacity quotas, and so on.

In this card, information about object buckets (OB) and object bucket claims (OBCs) is shown separately. OB includes all the buckets that are created using S3 or the user interface(UI) and OBC includes all the buckets created using YAMLs or the command line interface (CLI). The number displayed on the left of the bucket type is the total count of OBs or OBCs. The number displayed on the right shows the error count and is visible only when the error count is greater than zero. You can click on the number to see the list of buckets that has the warning or error status.

Resource Providers card

This card displays a list of all Multicloud Object Gateway and RADOS Object Gateway resources that are currently in use. Those resources are used to store data according to the buckets policies and can be a cloud-based resource or a bare metal resource.

Status card

This card shows whether the system and its services are running without any issues. When the system is in a warning or error state, the alerts section is shown and the relevant alerts are displayed there. Click the alert links beside each alert for more information about the issue. For information about health checks, see Cluster health.
If multiple object storage services are available in the cluster, click the service type (such as **Object Service** or **Data Resiliency**) to see the state of the individual services.

Data resiliency in the status card indicates if there is any resiliency issue regarding the data stored through the Multicloud Object Gateway and RADOS Object Gateway.

Capacity breakdown card

In this card you can visualize how applications consume the object storage through the Multicloud Object Gateway and RADOS Object Gateway. You can use the **Service Type** drop-down to view the capacity breakdown for the Multicloud Gateway and Object Gateway separately. When viewing the Multicloud Object Gateway, you can use the **Break By** drop-down to filter the results in the graph by either **Projects** or **Bucket Class**.

Performance card

In this card, you can view the performance of the Multicloud Object Gateway or RADOS Object Gateway. Use the **Service Type** drop-down to choose which you would like to view.
For Multicloud Object Gateway accounts, you can view the I/O operations and logical used capacity. For providers, you can view I/O operation, physical and logical usage, and egress.

The following tables explain the different metrics that you can view based on your selection from the drop-down menus on the top of the card:

Table 3.1. Indicators for Multicloud Object Gateway

Consumer types	Metrics	Chart display
Accounts	I/O operations	Displays read and write I/O operations for the top five consumers. The total reads and writes of all the consumers is displayed at the bottom. This information helps you monitor the throughput demand (IOPS) per application or account.
Accounts	Logical Used Capacity	Displays total logical usage of each account for the top five consumers. This helps you monitor the throughput demand per application or account.

Consumer types	Metrics	Chart display
Providers	I/O operations	Displays the count of I/O operations generated by the MCG when accessing the storage backend hosted by the provider. This helps you understand the traffic in the cloud so that you can improve resource allocation according to the I/O pattern, thereby optimizing the cost.
Providers	Physical vs Logical usage	Displays the data consumption in the system by comparing the physical usage with the logical usage per provider. This helps you control the storage resources and devise a placement strategy in line with your usage characteristics and your performance requirements while potentially optimizing your costs.
Providers	Egress	The amount of data the MCG retrieves from each provider (read bandwidth originated with the applications). This helps you understand the traffic in the cloud to improve resource allocation according to the egress pattern, thereby optimizing the cost.

For the RADOS Object Gateway, you can use the **Metric** drop-down to view the **Latency** or **Bandwidth**.

- **Latency**: Provides a visual indication of the average GET/PUT latency imbalance across RADOS Object Gateway instances.
- **Bandwidth**: Provides a visual indication of the sum of GET/PUT bandwidth across RADOS Object Gateway instances.

Activity card

This card displays what activities are happening or have recently happened in the OpenShift Data Foundation cluster. The card is separated into two sections:

- **Ongoing**: Displays the progress of ongoing activities related to rebuilding of data resiliency and upgrading of OpenShift Data Foundation operator.
- **Recent Events**: Displays the list of events that happened in the **openshift-storage** namespace.

3.3. POOL METRICS

The Pool metrics dashboard provides information to ensure efficient data consumption, and how to enable or disable compression if less effective.

Viewing pool metrics

To view the pool list:

1. Click **Storage → Data Foundation**
2. In the **Storage systems** tab, select the storage system and then click **BlockPools**.

When you click on a pool name, the following cards on each Pool dashboard is displayed along with the metrics based on deployment mode (internal or external):

Details card

The Details card shows the following:

- Pool Name
- Volume type
- Replicas

Status card

This card shows whether the pool is up and running without any errors or is experiencing some issues.

Mirroring card

When the mirroring option is enabled, this card shows the mirroring status, image health, and last checked time-stamp. The mirroring metrics are displayed when cluster level mirroring is enabled. The metrics help to prevent disaster recovery failures and notify of any discrepancies so that the data is kept intact.
The mirroring card shows high-level information such as:

- Mirroring state as either enabled or disabled for the particular pool.
- Status of all images under the pool as replicating successfully or not.
- Percentage of images that are replicating and not replicating.

Inventory card

The Inventory card shows the number of storage classes and Persistent Volume Claims.

Compression card

This card shows the compression status as enabled or disabled as the case may be. It also displays the storage efficiency details as follows:

- Compression eligibility that indicates what portion of written compression-eligible data is compressible (per ceph parameters)
- Compression ratio of compression-eligible data
- Compression savings provides the total savings (including replicas) of compression-eligible data

For information on how to enable or disable compression for an existing pool, see Updating an existing pool.

Raw Capacity card

This card shows the total raw storage capacity which includes replication, on the cluster.

- **Used** legend indicates storage capacity used by the pool
- **Available** legend indicates the available raw storage capacity on the cluster

Performance card

In this card, you can view the usage of I/O operations and throughput demand per application or account. The graph indicates the average latency or bandwidth across the instances.

3.4. ENABLING METADATA ON RBD AND CEPHFS VOLUMES

You can set the persistent volume claim (PVC), persistent volume (PV), and Namespace names in the RADOS block device (RBD) and CephFS volumes for monitoring purposes. This enables you to read the RBD and CephFS metadata to identify the mapping between the OpenShift Container Platform and RBD and CephFS volumes.

To enable RADOS block device (RBD) and CephFS volume metadata feature, you need to set the **CSI_ENABLE_METADATA** variable in the **rook-ceph-operator-config configmap**. By default, this feature is disabled. If you enable the feature after upgrading from a previous version, the existing PVCs will not contain the metadata. Also, when you enable the metadata feature, the PVCs that were created before enabling will not have the metadata.

Prerequisites

- Ensure to install **ocs_operator** and create a **storagecluster** for the operator.
- Ensure that the **storagecluster** is in **Ready** state.

```
$ oc get storagecluster
NAME                 AGE   PHASE   EXTERNAL   CREATED AT             VERSION
ocs-storagecluster   57m   Ready              2022-08-30T06:52:58Z   4.12.0
```

Procedure

1. Edit the **rook-ceph** operator **ConfigMap** to mark **CSI_ENABLE_METADATA** to **true**.

```
$ oc patch cm rook-ceph-operator-config -n openshift-storage -p $'data:\n
"CSI_ENABLE_METADATA":  "true"'
configmap/rook-ceph-operator-config patched
```

2. Wait for the respective CSI CephFS plugin provisioner pods and CSI RBD plugin pods to reach the **Running** state.

> **NOTE**
>
> Ensure that the **setmetadata** variable is automatically set after the metadata feature is enabled. This variable should not be available when the metadata feature is disabled.

```
$ oc get pods | grep csi

csi-cephfsplugin-b8d6c                        2/2     Running     0          56m
csi-cephfsplugin-bnbg9                        2/2     Running     0          56m
csi-cephfsplugin-kqdw4                        2/2     Running     0          56m
csi-cephfsplugin-provisioner-7dcd78bb9b-q6dxb  5/5     Running     0          56m
csi-cephfsplugin-provisioner-7dcd78bb9b-zc4q5  5/5     Running     0          56m
csi-rbdplugin-776dl                           3/3     Running     0          56m
csi-rbdplugin-ffl52                           3/3     Running     0          56m
csi-rbdplugin-jx9mz                           3/3     Running     0          56m
csi-rbdplugin-provisioner-5f6d766b6c-694fx     6/6     Running     0          56m
csi-rbdplugin-provisioner-5f6d766b6c-vzv45     6/6     Running     0          56m
```

Verification steps

- To verify the metadata for RBD PVC:

  1. Create a PVC.

     ```
     $ cat <<EOF | oc create -f -
     apiVersion: v1
     kind: PersistentVolumeClaim
     metadata:
       name: rbd-pvc
     spec:
       accessModes:
         - ReadWriteOnce
       resources:
         requests:
           storage: 1Gi
       storageClassName: ocs-storagecluster-ceph-rbd
     EOF
     ```

  2. Check the status of the PVC.

     ```
     $ oc get pvc | grep rbd-pvc
     rbd-pvc                           Bound    pvc-30628fa8-2966-499c-832d-a6a3a8ebc594   1Gi
     RWO            ocs-storagecluster-ceph-rbd   32s
     ```

  3. Verify the metadata in the Red Hat Ceph Storage command-line interface (CLI).
     For information about how to access the Red Hat Ceph Storage CLI, see the How to access Red Hat Ceph Storage CLI in Red Hat OpenShift Data Foundation environment article.

     ```
     [sh-4.x]$ rbd ls ocs-storagecluster-cephblockpool

     csi-vol-7d67bfad-2842-11ed-94bd-0a580a830012
     csi-vol-ed5ce27b-2842-11ed-94bd-0a580a830012

     [sh-4.x]$ rbd image-meta ls ocs-storagecluster-cephblockpool/csi-vol-ed5ce27b-2842-
     11ed-94bd-0a580a830012
     ```

     There are four metadata on this image:

     ```
     Key                               Value
     ```

```
csi.ceph.com/cluster/name         6cd7a18d-7363-4830-ad5c-f7b96927f026
csi.storage.k8s.io/pv/name        pvc-30628fa8-2966-499c-832d-a6a3a8ebc594
csi.storage.k8s.io/pvc/name       rbd-pvc
csi.storage.k8s.io/pvc/namespace  openshift-storage
```

- To verify the metadata for RBD clones:

1. Create a clone.

```
$ cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rbd-pvc-clone
spec:
  storageClassName: ocs-storagecluster-ceph-rbd
  dataSource:
    name: rbd-pvc
    kind: PersistentVolumeClaim
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
EOF
```

2. Check the status of the clone.

```
$ oc get pvc | grep rbd-pvc
rbd-pvc                          Bound    pvc-30628fa8-2966-499c-832d-a6a3a8ebc594   1Gi
RWO            ocs-storagecluster-ceph-rbd   15m
rbd-pvc-clone                    Bound    pvc-0d72afda-f433-4d46-a7f1-a5fcb3d766e0   1Gi
RWO            ocs-storagecluster-ceph-rbd   52s
```

3. Verify the metadata in the Red Hat Ceph Storage command-line interface (CLI).
   For information about how to access the Red Hat Ceph Storage CLI, see the How to access Red Hat Ceph Storage CLI in Red Hat OpenShift Data Foundation environment article.

```
[sh-4.x]$ rbd ls ocs-storagecluster-cephblockpool
csi-vol-063b982d-2845-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012-temp
csi-vol-7d67bfad-2842-11ed-94bd-0a580a830012
csi-vol-ed5ce27b-2842-11ed-94bd-0a580a830012

[sh-4.x]$ rbd image-meta ls ocs-storagecluster-cephblockpool/csi-vol-063b982d-2845-
11ed-94bd-0a580a830012
There are 4 metadata on this image:

Key                               Value
csi.ceph.com/cluster/name         6cd7a18d-7363-4830-ad5c-f7b96927f026
csi.storage.k8s.io/pv/name        pvc-0d72afda-f433-4d46-a7f1-a5fcb3d766e0
csi.storage.k8s.io/pvc/name       rbd-pvc-clone
csi.storage.k8s.io/pvc/namespace  openshift-storage
```

- To verify the metadata for RBD Snapshots:

1. Create a snapshot.

```
$ cat <<EOF | oc create -f -
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: rbd-pvc-snapshot
spec:
  volumeSnapshotClassName: ocs-storagecluster-rbdplugin-snapclass
  source:
    persistentVolumeClaimName: rbd-pvc
EOF
volumesnapshot.snapshot.storage.k8s.io/rbd-pvc-snapshot created
```

2. Check the status of the snapshot.

```
$ oc get volumesnapshot
NAME               READYTOUSE   SOURCEPVC   SOURCESNAPSHOTCONTENT
RESTORESIZE   SNAPSHOTCLASS                            SNAPSHOTCONTENT
CREATIONTIME   AGE
rbd-pvc-snapshot   true         rbd-pvc                             1Gi          ocs-storagecluster-
rbdplugin-snapclass   snapcontent-b992b782-7174-4101-8fe3-e6e478eb2c8f   17s
18s
```

3. Verify the metadata in the Red Hat Ceph Storage command-line interface (CLI).
   For information about how to access the Red Hat Ceph Storage CLI, see the How to access Red Hat Ceph Storage CLI in Red Hat OpenShift Data Foundation environment article.

```
[sh-4.x]$ rbd ls ocs-storagecluster-cephblockpool
csi-snap-a1e24408-2848-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012-temp
csi-vol-7d67bfad-2842-11ed-94bd-0a580a830012
csi-vol-ed5ce27b-2842-11ed-94bd-0a580a830012

[sh-4.x]$ rbd image-meta ls ocs-storagecluster-cephblockpool/csi-snap-a1e24408-2848-
11ed-94bd-0a580a830012
There are 4 metadata on this image:

Key                                        Value
csi.ceph.com/cluster/name                  6cd7a18d-7363-4830-ad5c-f7b96927f026
csi.storage.k8s.io/volumesnapshot/name          rbd-pvc-snapshot
csi.storage.k8s.io/volumesnapshot/namespace    openshift-storage
csi.storage.k8s.io/volumesnapshotcontent/name  snapcontent-b992b782-7174-4101-
8fe3-e6e478eb2c8f
```

- Verify the metadata for RBD Restore:

1. Restore a volume snapshot.

```
$ cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rbd-pvc-restore
```

```
spec:
  storageClassName: ocs-storagecluster-ceph-rbd
  dataSource:
    name: rbd-pvc-snapshot
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
EOF
persistentvolumeclaim/rbd-pvc-restore created
```

2. Check the status of the restored volume snapshot.

```
$ oc get pvc | grep rbd
db-noobaa-db-pg-0                 Bound    pvc-615e2027-78cd-4ea2-a341-fdedd50c5208
50Gi       RWO            ocs-storagecluster-ceph-rbd   51m
rbd-pvc                           Bound    pvc-30628fa8-2966-499c-832d-a6a3a8ebc594   1Gi
RWO            ocs-storagecluster-ceph-rbd   47m
rbd-pvc-clone                     Bound    pvc-0d72afda-f433-4d46-a7f1-a5fcb3d766e0   1Gi
RWO            ocs-storagecluster-ceph-rbd   32m
rbd-pvc-restore                   Bound    pvc-f900e19b-3924-485c-bb47-01b84c559034   1Gi
RWO            ocs-storagecluster-ceph-rbd   111s
```

3. Verify the metadata in the Red Hat Ceph Storage command-line interface (CLI).
   For information about how to access the Red Hat Ceph Storage CLI, see the How to access Red Hat Ceph Storage CLI in Red Hat OpenShift Data Foundation environment article.

```
[sh-4.x]$ rbd ls ocs-storagecluster-cephblockpool
csi-snap-a1e24408-2848-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012-temp
csi-vol-5f6e0737-2849-11ed-94bd-0a580a830012
csi-vol-7d67bfad-2842-11ed-94bd-0a580a830012
csi-vol-ed5ce27b-2842-11ed-94bd-0a580a830012

[sh-4.x]$ rbd image-meta ls ocs-storagecluster-cephblockpool/csi-vol-5f6e0737-2849-
11ed-94bd-0a580a830012
There are 4 metadata on this image:

Key                               Value
csi.ceph.com/cluster/name         6cd7a18d-7363-4830-ad5c-f7b96927f026
csi.storage.k8s.io/pv/name        pvc-f900e19b-3924-485c-bb47-01b84c559034
csi.storage.k8s.io/pvc/name       rbd-pvc-restore
csi.storage.k8s.io/pvc/namespace  openshift-storage
```

- To verify the metadata for CephFS PVC:

1. Create a PVC.

```
cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
```

```
metadata:
  name: cephfs-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ocs-storagecluster-cephfs
EOF
```

2. Check the status of the PVC.

```
oc get pvc | grep cephfs
cephfs-pvc                       Bound    pvc-4151128c-86f0-468b-b6e7-5fdfb51ba1b9   1Gi
RWO            ocs-storagecluster-cephfs     11s
```

3. Verify the metadata in the Red Hat Ceph Storage command-line interface (CLI).
For information about how to access the Red Hat Ceph Storage CLI, see the How to access Red Hat Ceph Storage CLI in Red Hat OpenShift Data Foundation environment article.

```
$ ceph fs volume ls
[
    {
        "name": "ocs-storagecluster-cephfilesystem"
    }
]

$ ceph fs subvolumegroup ls ocs-storagecluster-cephfilesystem
[
    {
        "name": "csi"
    }
]

$ ceph fs subvolume ls ocs-storagecluster-cephfilesystem --group_name csi
[
    {
        "name": "csi-vol-25266061-284c-11ed-95e0-0a580a810215"
    }
]

$ ceph fs subvolume metadata ls ocs-storagecluster-cephfilesystem csi-vol-25266061-
284c-11ed-95e0-0a580a810215 --group_name=csi --format=json

{
    "csi.ceph.com/cluster/name": "6cd7a18d-7363-4830-ad5c-f7b96927f026",
    "csi.storage.k8s.io/pv/name": "pvc-4151128c-86f0-468b-b6e7-5fdfb51ba1b9",
    "csi.storage.k8s.io/pvc/name": "cephfs-pvc",
    "csi.storage.k8s.io/pvc/namespace": "openshift-storage"
}
```

- To verify the metadata for CephFS clone:

1. Create a clone.

```
$ cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cephfs-pvc-clone
spec:
  storageClassName: ocs-storagecluster-cephfs
  dataSource:
    name: cephfs-pvc
    kind: PersistentVolumeClaim
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
EOF
persistentvolumeclaim/cephfs-pvc-clone created
```

2. Check the status of the clone.

```
$ oc get pvc | grep cephfs
cephfs-pvc                  Bound    pvc-4151128c-86f0-468b-b6e7-5fdfb51ba1b9   1Gi
RWO         ocs-storagecluster-cephfs     9m5s
cephfs-pvc-clone                  Bound    pvc-3d4c4e78-f7d5-456a-aa6e-4da4a05ca4ce   1Gi
RWX         ocs-storagecluster-cephfs     20s
```

3. Verify the metadata in the Red Hat Ceph Storage command-line interface (CLI).
   For information about how to access the Red Hat Ceph Storage CLI, see the How to access Red Hat Ceph Storage CLI in Red Hat OpenShift Data Foundation environment article.

```
[rook@rook-ceph-tools-c99fd8dfc-6sdbg /]$ ceph fs subvolume ls ocs-storagecluster-
cephfilesystem --group_name csi
[
    {
        "name": "csi-vol-5ea23eb0-284d-11ed-95e0-0a580a810215"
    },
    {
        "name": "csi-vol-25266061-284c-11ed-95e0-0a580a810215"
    }
]

[rook@rook-ceph-tools-c99fd8dfc-6sdbg /]$ ceph fs subvolume metadata ls ocs-
storagecluster-cephfilesystem csi-vol-5ea23eb0-284d-11ed-95e0-0a580a810215 --
group_name=csi --format=json

{
    "csi.ceph.com/cluster/name": "6cd7a18d-7363-4830-ad5c-f7b96927f026",
    "csi.storage.k8s.io/pv/name": "pvc-3d4c4e78-f7d5-456a-aa6e-4da4a05ca4ce",
    "csi.storage.k8s.io/pvc/name": "cephfs-pvc-clone",
    "csi.storage.k8s.io/pvc/namespace": "openshift-storage"
}
```

- To verify the metadata for CephFS volume snapshot:

  1. Create a volume snapshot.

```
$ cat <<EOF | oc create -f -
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: cephfs-pvc-snapshot
spec:
  volumeSnapshotClassName: ocs-storagecluster-cephfsplugin-snapclass
  source:
    persistentVolumeClaimName: cephfs-pvc
EOF
volumesnapshot.snapshot.storage.k8s.io/cephfs-pvc-snapshot created
```

2. Check the status of the volume snapshot.

```
$ oc get volumesnapshot
NAME                READYTOUSE   SOURCEPVC    SOURCESNAPSHOTCONTENT
RESTORESIZE   SNAPSHOTCLASS                          SNAPSHOTCONTENT
CREATIONTIME   AGE
cephfs-pvc-snapshot   true         cephfs-pvc                          1Gi          ocs-
storagecluster-cephfsplugin-snapclass   snapcontent-f0f17463-d13b-4e13-b44e-
6340bbb3bee0   9s             9s
```

3. Verify the metadata in the Red Hat Ceph Storage command-line interface (CLI).
   For information about how to access the Red Hat Ceph Storage CLI, see the How to access Red Hat Ceph Storage CLI in Red Hat OpenShift Data Foundation environment article.

```
$ ceph fs subvolume snapshot ls ocs-storagecluster-cephfilesystem csi-vol-25266061-
284c-11ed-95e0-0a580a810215 --group_name csi
[
    {
        "name": "csi-snap-06336f4e-284e-11ed-95e0-0a580a810215"
    }
]

$ ceph fs subvolume snapshot metadata ls ocs-storagecluster-cephfilesystem csi-vol-
25266061-284c-11ed-95e0-0a580a810215 csi-snap-06336f4e-284e-11ed-95e0-
0a580a810215 --group_name=csi --format=json

{
    "csi.ceph.com/cluster/name": "6cd7a18d-7363-4830-ad5c-f7b96927f026",
    "csi.storage.k8s.io/volumesnapshot/name": "cephfs-pvc-snapshot",
    "csi.storage.k8s.io/volumesnapshot/namespace": "openshift-storage",
    "csi.storage.k8s.io/volumesnapshotcontent/name": "snapcontent-f0f17463-d13b-4e13-
b44e-6340bbb3bee0"
}
```

- To verify the metadata of the CephFS Restore:

  1. Restore a volume snapshot.

```
$ cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cephfs-pvc-restore
```

```
spec:
  storageClassName: ocs-storagecluster-cephfs
  dataSource:
    name: cephfs-pvc-snapshot
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
EOF
persistentvolumeclaim/cephfs-pvc-restore created
```

2. Check the status of the restored volume snapshot.

```
$ oc get pvc | grep cephfs
cephfs-pvc                   Bound    pvc-4151128c-86f0-468b-b6e7-5fdfb51ba1b9   1Gi
RWO            ocs-storagecluster-cephfs     29m
cephfs-pvc-clone               Bound    pvc-3d4c4e78-f7d5-456a-aa6e-4da4a05ca4ce   1Gi
RWX            ocs-storagecluster-cephfs     20m
cephfs-pvc-restore              Bound    pvc-43d55ea1-95c0-42c8-8616-4ee70b504445
1Gi        RWX            ocs-storagecluster-cephfs     21s
```

3. Verify the metadata in the Red Hat Ceph Storage command-line interface (CLI).
   For information about how to access the Red Hat Ceph Storage CLI, see the How to access Red Hat Ceph Storage CLI in Red Hat OpenShift Data Foundation environment article.

```
$ ceph fs subvolume ls ocs-storagecluster-cephfilesystem --group_name csi
[
    {
        "name": "csi-vol-3536db13-2850-11ed-95e0-0a580a810215"
    },
    {
        "name": "csi-vol-5ea23eb0-284d-11ed-95e0-0a580a810215"
    },
    {
        "name": "csi-vol-25266061-284c-11ed-95e0-0a580a810215"
    }
]

$ ceph fs subvolume metadata ls ocs-storagecluster-cephfilesystem csi-vol-3536db13-
2850-11ed-95e0-0a580a810215 --group_name=csi --format=json

{
    "csi.ceph.com/cluster/name": "6cd7a18d-7363-4830-ad5c-f7b96927f026",
    "csi.storage.k8s.io/pv/name": "pvc-43d55ea1-95c0-42c8-8616-4ee70b504445",
    "csi.storage.k8s.io/pvc/name": "cephfs-pvc-restore",
    "csi.storage.k8s.io/pvc/namespace": "openshift-storage"
}
```

CHAPTER 4. ALERTS

4.1. SETTING UP ALERTS

For internal Mode clusters, various alerts related to the storage metrics services, storage cluster, disk devices, cluster health, cluster capacity, and so on are displayed in the Block and File, and the object dashboards. These alerts are not available for external Mode.

NOTE

It might take a few minutes for alerts to be shown in the alert panel, because only firing alerts are visible in this panel.

You can also view alerts with additional details and customize the display of Alerts in the OpenShift Container Platform.

For more information, see Managing alerts.

CHAPTER 5. REMOTE HEALTH MONITORING

OpenShift Data Foundation collects anonymized aggregated information about the health, usage, and size of clusters and reports it to Red Hat via an integrated component called Telemetry. This information allows Red Hat to improve OpenShift Data Foundation and to react to issues that impact customers more quickly.

A cluster that reports data to Red Hat via Telemetry is considered a *connected cluster*.

5.1. ABOUT TELEMETRY

Telemetry sends a carefully chosen subset of the cluster monitoring metrics to Red Hat. These metrics are sent continuously and describe:

- The size of an OpenShift Data Foundation cluster
- The health and status of OpenShift Data Foundation components
- The health and status of any upgrade being performed
- Limited usage information about OpenShift Data Foundation components and features
- Summary info about alerts reported by the cluster monitoring component

This continuous stream of data is used by Red Hat to monitor the health of clusters in real time and to react as necessary to problems that impact our customers. It also allows Red Hat to roll out OpenShift Data Foundation upgrades to customers so as to minimize service impact and continuously improve the upgrade experience.

This debugging information is available to Red Hat Support and engineering teams with the same restrictions as accessing data reported via support cases. All connected cluster information is used by Red Hat to help make OpenShift Data Foundation better and more intuitive to use. None of the information is shared with third parties.

5.2. INFORMATION COLLECTED BY TELEMETRY

Primary information collected by Telemetry includes:

- The size of the Ceph cluster in bytes : **"ceph_cluster_total_bytes"**,
- The amount of the Ceph cluster storage used in bytes : **"ceph_cluster_total_used_raw_bytes"**,
- Ceph cluster health status : **"ceph_health_status"**,
- The total count of object storage devices (OSDs) : **"job:ceph_osd_metadata:count"**,
- The total number of OpenShift Data Foundation Persistent Volumes (PVs) present in the Red Hat OpenShift Container Platform cluster : **"job:kube_pv:count"**,
- The total input/output operations per second (IOPS) (reads+writes) value for all the pools in the Ceph cluster : **"job:ceph_pools_iops:total"**,
- The total IOPS (reads+writes) value in bytes for all the pools in the Ceph cluster : **"job:ceph_pools_iops_bytes:total"**,

- The total count of the Ceph cluster versions running : **"job:ceph_versions_running:count"**

- The total number of unhealthy NooBaa buckets : **"job:noobaa_total_unhealthy_buckets:sum"**,

- The total number of NooBaa buckets : **"job:noobaa_bucket_count:sum"**,

- The total number of NooBaa objects : **"job:noobaa_total_object_count:sum"**,

- The count of NooBaa accounts : **"noobaa_accounts_num"**,

- The total usage of storage by NooBaa in bytes : **"noobaa_total_usage"**,

- The total amount of storage requested by the persistent volume claims (PVCs) from a particular storage provisioner in bytes: **"cluster:kube_persistentvolumeclaim_resource_requests_storage_bytes:provisioner:sum"**,

- The total amount of storage used by the PVCs from a particular storage provisioner in bytes: **"cluster:kubelet_volume_stats_used_bytes:provisioner:sum"**.

Telemetry does not collect identifying information such as user names, passwords, or the names or addresses of user resources.