Replacing nodes

How to prepare replacement nodes and replace failed nodes
How to prepare replacement nodes and replace failed nodes
Abstract

This document explains how to safely replace a node in a Red Hat OpenShift Container Storage cluster.
Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE ................................................................. 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION ...................................... 4

PREFACE ...................................................................................................................... 5

CHAPTER 1. OPENSHEET CONTAINER STORAGE DEPLOYED USING DYNAMIC DEVICES .......................................................... 6

1.1. OPENSHEET CONTAINER STORAGE DEPLOYED ON AWS .............................. 6
  1.1.1. Replacing an operational AWS node on user-provisioned infrastructure .......... 6
  1.1.2. Replacing an operational AWS node on installer-provisioned infrastructure .... 8
  1.1.3. Replacing a failed AWS node on user-provisioned infrastructure ............... 9
  1.1.4. Replacing a failed AWS node on installer-provisioned infrastructure .......... 11

1.2. OPENSHEET CONTAINER STORAGE DEPLOYED ON VMWARE .................... 12
  1.2.1. Replacing an operational VMware node on user-provisioned infrastructure .... 13
  1.2.2. Replacing an operational VMware node on installer-provisioned infrastructure 15
  1.2.3. Replacing a failed VMware node on user-provisioned infrastructure .......... 16
  1.2.4. Replacing a failed VMware node on installer-provisioned infrastructure ....... 18

1.3. OPENSHEET CONTAINER STORAGE DEPLOYED ON RED HAT VIRTUALIZATION .............................................. 19
  1.3.1. Replacing an operational Red Hat Virtualization node on installer-provisioned infrastructure 20
  1.3.2. Replacing a failed Red Hat Virtualization node on installer-provisioned infrastructure 21

1.4. OPENSHEET CONTAINER STORAGE DEPLOYED ON MICROSOFT AZURE ........ 23
  1.4.1. Replacing operational nodes on Azure installer-provisioned infrastructure .... 23
  1.4.2. Replacing failed nodes on Azure installer-provisioned infrastructure .......... 25

CHAPTER 2. OPENSHEET CONTAINER STORAGE DEPLOYED USING LOCAL STORAGE DEVICES ..................... 27

2.1. REPLACING STORAGE NODES ON BARE METAL INFRASTRUCTURE ......... 27
  2.1.1. Replacing an operational node on bare metal user-provisioned infrastructure .... 27
  2.1.2. Replacing a failed node on bare metal user-provisioned infrastructure ........ 31

2.2. REPLACING STORAGE NODES ON IBM Z OR LINUXONE INFRASTRUCTURE .... 35
  2.2.1. Replacing operational nodes on IBM Z or LinuxONE infrastructure .......... 36
  2.2.2. Replacing failed nodes on IBM Z or LinuxONE infrastructure ................. 37

2.3. REPLACING STORAGE NODES ON VMWARE INFRASTRUCTURE .................. 39
  2.3.1. Replacing an operational node on VMware user-provisioned infrastructure .... 39
  2.3.2. Replacing an operational node on VMware installer-provisioned infrastructure 43
  2.3.3. Replacing a failed node on VMware user-provisioned infrastructure .......... 48
  2.3.4. Replacing a failed node on VMware installer-provisioned infrastructure ...... 52

2.4. REPLACING STORAGE NODES ON RED HAT VIRTUALIZATION INFRASTRUCTURE .... 57
  2.4.1. Replacing an operational node on Red Hat Virtualization installer-provisioned infrastructure 57
  2.4.2. Replacing a failed node on Red Hat Virtualization installer-provisioned infrastructure 62

2.5. REPLACING STORAGE NODES ON IBM POWER SYSTEMS INFRASTRUCTURE ........ 67
  2.5.1. Replacing an operational or failed storage node on IBM Power Systems ....... 67
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:

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.

- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the **Bugzilla** website.
  2. As the Component, use **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**.
For OpenShift Container Storage, node replacement can be performed proactively for an operational node and reactively for a failed node for the following deployments:

- For Amazon Web Services (AWS)
  - User-provisioned infrastructure
  - Installer-provisioned infrastructure
- For VMware
  - User-provisioned infrastructure
  - Installer-provisioned infrastructure
- For Red Hat Virtualization
  - Installer-provisioned infrastructure
- For Microsoft Azure
  - Installer-provisioned infrastructure
- For local storage devices
  - Bare metal
  - VMware
  - Red Hat Virtualization
  - IBM Power Systems
- For replacing your storage nodes in external mode, see Red Hat Ceph Storage documentation.
CHAPTER 1. OPENSIFT CONTAINER STORAGE DEPLOYED USING DYNAMIC DEVICES

1.1. OPENSIFT CONTAINER STORAGE DEPLOYED ON AWS

- To replace an operational node, see:
  - Section 1.1.1, “Replacing an operational AWS node on user-provisioned infrastructure”
  - Section 1.1.2, “Replacing an operational AWS node on installer-provisioned infrastructure”

- To replace a failed node, see:
  - Section 1.1.4, “Replacing a failed AWS node on installer-provisioned infrastructure”
  - Section 1.1.3, “Replacing a failed AWS node on user-provisioned infrastructure”

1.1.1. Replacing an operational AWS node on user-provisioned infrastructure

Perform this procedure to replace an operational node on AWS user-provisioned infrastructure.

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCF) cluster.

Procedure

1. Identify the node that needs to be replaced.

2. Mark the node as unschedulable using the following command:

   ```bash
   $ oc adm cordon <node_name>
   ```

3. Drain the node using the following command:

   ```bash
   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
   ```

   **IMPORTANT**
   This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

4. Delete the node using the following command:

   ```bash
   $ oc delete nodes <node_name>
   ```

5. Create a new AWS machine instance with the required infrastructure. See Platform requirements.
6. Create a new OpenShift Container Platform node using the new AWS machine instance.

7. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

   ```
   $ oc get csr
   ```

8. Approve all required OpenShift Container Platform CSRs for the new node:

   ```
   $ oc adm certificate approve <Certificate_Name>
   ```

9. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.

10. Apply the OpenShift Container Storage label to the new node.

    **From the web user interface**
    
    a. For the new node, click **Action Menu ( ⋮ ) → Edit Labels**
    
    b. Add `cluster.ocs.openshift.io/openshift-storage` and click **Save**.

    **From the command line interface**
    
    - Execute the following command to apply the OpenShift Container Storage label to the new node:

      ```
      $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
      ```

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:

   ```
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Click **Workloads → Pods**, confirm that at least the following pods on the new node are in **Running** state:

   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in **Running** state.

4. Verify that new OSD pods are running on the replacement node.

   ```
   $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
   ```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   For each of the new nodes identified in previous step, do the following:

   a. Create a debug pod and open a chroot environment for the selected host(s).
$ oc debug node/<node name>
$ chroot /host

b. Run "lsblk" and check for the “crypt” keyword beside the ocs-deviceset name(s)

$ lsblk

6. If verification steps fail, contact Red Hat Support.

1.1.2. Replacing an operational AWS node on installer-provisioned infrastructure

Use this procedure to replace an operational node on AWS installer-provisioned infrastructure (IPI).

Procedure

1. Log in to OpenShift Web Console and click Compute → Nodes.

2. Identify the node that needs to be replaced. Take a note of its Machine Name.

3. Mark the node as unschedulable using the following command:

   $ oc adm cordon <node_name>

4. Drain the node using the following command:

   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets

   **IMPORTANT**
   This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

5. Click Compute → Machines. Search for the required machine.

6. Besides the required machine, click the Action menu (⋮) → Delete Machine.

7. Click Delete to confirm the machine deletion. A new machine is automatically created.

8. Wait for new machine to start and transition into Running state.

   **IMPORTANT**
   This activity may take at least 5-10 minutes or more.

9. Click Compute → Nodes, confirm if the new node is in Ready state.

10. Apply the OpenShift Container Storage label to the new node using any one of the following:

    From User interface
    a. For the new node, click Action Menu (⋮) → Edit Labels
b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

From Command line interface

- Execute the following command to apply the OpenShift Container Storage label to the new node:

  ```
  $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
  ```

Verification steps

1. Execute the following command and verify that the new node is present in the output:

  ```
  $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
  ```

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:

   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in Running state.

4. Verify that new OSD pods are running on the replacement node.

   ```
   $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
   ```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in previous step, do the following:
   a. Create a debug pod and open a chroot environment for the selected host(s).

      ```
      $ oc debug node/<node name>
      $ chroot /host
      ```
   b. Run “lsblk” and check for the “crypt” keyword beside the `ocs-deviceset` name(s)

      ```
      $ lsblk
      ```

6. If verification steps fail, contact Red Hat Support.

### 1.1.3. Replacing a failed AWS node on user-provisioned infrastructure

Perform this procedure to replace a failed node which is not operational on AWS user-provisioned infrastructure (UPI) for OpenShift Container Storage.

**Prerequisites**

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
You must be logged into the OpenShift Container Platform (RHOCP) cluster.

Procedure

1. Identify the AWS machine instance of the node that needs to be replaced.
2. Log in to AWS and terminate the identified AWS machine instance.
3. Create a new AWS machine instance with the required infrastructure. See platform requirements.
4. Create a new OpenShift Container Platform node using the new AWS machine instance.
5. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:
   
   ```
   $ oc get csr
   ```

6. Approve all required OpenShift Container Platform CSRs for the new node:
   
   ```
   $ oc adm certificate approve <Certificate_Name>
   ```

7. Click Compute → Nodes, confirm if the new node is in Ready state.
8. Apply the OpenShift Container Storage label to the new node using any one of the following:

   From User interface
   
   a. For the new node, click Action Menu (⋮) → Edit Labels
   
   b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

   From Command line interface

   - Execute the following command to apply the OpenShift Container Storage label to the new node:
     
     ```
     $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
     ```

Verification steps

1. Execute the following command and verify that the new node is present in the output:

   ```
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:

   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in Running state.
4. Verify that new OSD pods are running on the replacement node.

$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   For each of the new nodes identified in previous step, do the following:
   
   a. Create a debug pod and open a chroot environment for the selected host(s).

   $ oc debug node/<node name>
   $ chroot /host

   b. Run "lsblk" and check for the “crypt” keyword beside the ocs-deviceset name(s)

   $ lsblk

6. If verification steps fail, contact Red Hat Support .

1.1.4. Replacing a failed AWS node on installer-provisioned infrastructure

Perform this procedure to replace a failed node which is not operational on AWS installer-provisioned infrastructure (IPI) for OpenShift Container Storage.

Procedure

1. Log in to OpenShift Web Console and click Compute → Nodes.

2. Identify the faulty node and click on its Machine Name.

3. Click Actions → Edit Annotations, and click Add More.

4. Add machine.openshift.io/exclude-node-draining and click Save.

5. Click Actions → Delete Machine, and click Delete.

6. A new machine is automatically created, wait for new machine to start.

   IMPORTANT

   This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

7. Click Compute → Nodes, confirm if the new node is in Ready state.

8. Apply the OpenShift Container Storage label to the new node using any one of the following:

   From User interface
   
   a. For the new node, click Action Menu ( ⋮ ) → Edit Labels

   b. Add cluster.ocs.openshift.io/openshift-storage and click Save.
From Command line interface

- Execute the following command to apply the OpenShift Container Storage label to the new node:

  ```
  $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
  ```

  9. [Optional]: If the failed AWS instance is not removed automatically, terminate the instance from AWS console.

Verification steps

1. Execute the following command and verify that the new node is present in the output:

  ```
  $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
  ```

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:

   - csi-cephfsplugin-*
   - csi-rbdplugin-*

3. Verify that all other required OpenShift Container Storage pods are in Running state.

4. Verify that new OSD pods are running on the replacement node.

  ```
  $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
  ```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   For each of the new nodes identified in previous step, do the following:

   a. Create a debug pod and open a chroot environment for the selected host(s).

      ```
      $ oc debug node/<node name>
      $ chroot /host
      ```

   b. Run "lsblk" and check for the “crypt” keyword beside the ocs-deviceset name(s)

      ```
      $ lsblk
      ```

6. If verification steps fail, contact Red Hat Support.

1.2. OPENSIFHT CONTAINER STORAGE DEPLOYED ON VMWARE

- To replace an operational node, see:

  - Section 1.2.1, "Replacing an operational VMware node on user-provisioned infrastructure"
  - Section 1.2.2, “Replacing an operational VMware node on installer-provisioned infrastructure”
To replace a failed node, see:

- Section 1.2.3, “Replacing a failed VMware node on user-provisioned infrastructure”
- Section 1.2.4, “Replacing a failed VMware node on installer-provisioned infrastructure”

1.2.1. Replacing an operational VMware node on user-provisioned infrastructure

Perform this procedure to replace an operational node on VMware user-provisioned infrastructure (UPI).

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

Procedure

1. Identify the node and its VM that needs to be replaced.

2. Mark the node as unschedulable using the following command:

   ```
   $ oc adm cordon <node_name>
   ```

3. Drain the node using the following command:

   ```
   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
   ```

   **IMPORTANT**
   
   This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

4. Delete the node using the following command:

   ```
   $ oc delete nodes <node_name>
   ```

5. Log in to vSphere and terminate the identified VM.

   **IMPORTANT**
   
   VM should be deleted only from the inventory and not from the disk.

6. Create a new VM on vSphere with the required infrastructure. See Platform requirements.

7. Create a new OpenShift Container Platform worker node using the new VM.

8. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:
9. Approve all required OpenShift Container Platform CSRs for the new node:
   
   $ oc adm certificate approve <Certificate_Name>

10. Click Compute → Nodes, confirm if the new node is in Ready state.

11. Apply the OpenShift Container Storage label to the new node using any one of the following:

    From User interface
    
    a. For the new node, click Action Menu ( ⋮ ) → Edit Labels
    
    b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

    From Command line interface
    
    - Execute the following command to apply the OpenShift Container Storage label to the new node:
      
      $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:

   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:

   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in Running state.

4. Verify that new OSD pods are running on the replacement node.

   $ oc get pods -o wide -n openshift-storage | grep -i new-node-name | grep osd

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   
   For each of the new nodes identified in previous step, do the following:
   
   a. Create a debug pod and open a chroot environment for the selected host(s).
      
      $ oc debug node/<node name>
      $ chroot /host

   b. Run “lsblk” and check for the “crypt” keyword beside the `ocs-deviceset` name(s)
6. If verification steps fail, contact Red Hat Support.

1.2.2. Replacing an operational VMware node on installer-provisioned infrastructure

Use this procedure to replace an operational node on VMware installer-provisioned infrastructure (IPI).

Procedure

1. Log in to OpenShift Web Console and click **Compute → Nodes**.

2. Identify the node that needs to be replaced. Take a note of its **Machine Name**.

3. Mark the node as unschedulable using the following command:

   ```bash
   $ oc adm cordon <node_name>
   ```

4. Drain the node using the following command:

   ```bash
   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
   ```

   **IMPORTANT**
   
   This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

5. Click **Compute → Machines**. Search for the required machine.

6. Besides the required machine, click the **Action menu (...) → Delete Machine**.

7. Click **Delete** to confirm the machine deletion. A new machine is automatically created.

8. Wait for new machine to start and transition into **Running** state.

   **IMPORTANT**
   
   This activity may take at least 5-10 minutes or more.

9. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.

10. Apply the OpenShift Container Storage label to the new node using any one of the following:

    **From User interface**

    a. For the new node, click **Action Menu (...) → Edit Labels**

    b. Add `cluster.ocs.openshift.io/openshift-storage` and click **Save**.

    **From Command line interface**
Execute the following command to apply the OpenShift Container Storage label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:
   - csi-cephfsplugin-*
   - csi-rbdplugin-*

3. Verify that all other required OpenShift Container Storage pods are in Running state.

4. Verify that new OSD pods are running on the replacement node.

```
$ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in previous step, do the following:
   a. Create a debug pod and open a chroot environment for the selected host(s).
   
   ```
   $ oc debug node/<node name>
   $ chroot /host
   ```
   
   b. Run “lsblk” and check for the “crypt” keyword beside the ocs-deviceset name(s)

   ```
   $ lsblk
   ```

6. If verification steps fail, contact Red Hat Support.

**1.2.3. Replacing a failed VMware node on user-provisioned infrastructure**

Perform this procedure to replace a failed node on VMware user-provisioned infrastructure (UPI).

**Prerequisites**

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

**Procedure**

1. Identify the node and its VM that needs to be replaced.
2. Delete the node using the following command:

   $ oc delete nodes <node_name>

3. Log in to vSphere and terminate the identified VM.

   **IMPORTANT**
   VM should be deleted only from the inventory and not from the disk.

4. Create a new VM on vSphere with the required infrastructure. See Platform requirements.

5. Create a new OpenShift Container Platform worker node using the new VM.

6. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

   $ oc get csr

7. Approve all required OpenShift Container Platform CSRs for the new node:

   $ oc adm certificate approve <Certificate_Name>

8. Click **Compute** → **Nodes**, confirm if the new node is in **Ready** state.

9. Apply the OpenShift Container Storage label to the new node using any one of the following:

   **From User interface**
   a. For the new node, click **Action Menu (⋯) → Edit Labels**
   b. Add `cluster.ocs.openshift.io/openshift-storage` and click **Save**.

   **From Command line interface**
   - Execute the following command to apply the OpenShift Container Storage label to the new node:

     $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:

   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1

2. Click **Workloads** → **Pods**, confirm that at least the following pods on the new node are in **Running** state:

   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`
3. Verify that all other required OpenShift Container Storage pods are in **Running** state.

4. Verify that new OSD pods are running on the replacement node.

   ```bash
   $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | grep osd
   ```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   For each of the new nodes identified in previous step, do the following:
   a. Create a debug pod and open a chroot environment for the selected host(s).

      ```bash
      $ oc debug node/<node name>
      $ chroot /host
      ```
   b. Run `lsblk` and check for the “crypt” keyword beside the `ocs-deviceset` name(s)

      ```bash
      $ lsblk
      ```

6. If verification steps fail, contact Red Hat Support.

### 1.2.4. Replacing a failed VMware node on installer-provisioned infrastructure

Perform this procedure to replace a failed node which is not operational on VMware installer-provisioned infrastructure (IPI) for OpenShift Container Storage.

**Procedure**

1. Log in to OpenShift Web Console and click **Compute → Nodes**.

2. Identify the faulty node and click on its **Machine Name**.

3. Click **Actions → Edit Annotations**, and click **Add More**.

4. Add `machine.openshift.io/exclude-node-draining` and click **Save**.

5. Click **Actions → Delete Machine**, and click **Delete**.

6. A new machine is automatically created, wait for new machine to start.

   **IMPORTANT**

   This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

7. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.

8. Apply the OpenShift Container Storage label to the new node using any one of the following:

   **From User interface**

   a. For the new node, click **Action Menu (⋮) → Edit Labels**
b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

From Command line interface

- Execute the following command to apply the OpenShift Container Storage label to the new node:

  ```shell
  $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
  ```

9. [Optional]: If the failed VM is not removed automatically, terminate the VM from vSphere.

Verification steps

1. Execute the following command and verify that the new node is present in the output:

   ```shell
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
   ```

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:
   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in Running state.

4. Verify that new OSD pods are running on the replacement node.

   ```shell
   $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
   ```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   For each of the new nodes identified in previous step, do the following:

   a. Create a debug pod and open a chroot environment for the selected host(s).

   ```shell
   $ oc debug node/<node name>
   $ chroot /host
   ```

   b. Run “lsblk” and check for the “crypt” keyword beside the `ocs-deviceset` name(s)

   ```shell
   $ lsblk
   ```

6. If verification steps fail, contact Red Hat Support.

**1.3. OPENShift CONTAINER STORAGE DEPLOYED ON RED HAT VIRTUALIZATION**

- To replace an operational node, see Section 1.3.1, “Replacing an operational Red Hat Virtualization node on installer-provisioned infrastructure”
To replace a failed node, see Section 2.4.2, “Replacing a failed node on Red Hat Virtualization installer-provisioned infrastructure”

1.3.1. Replacing an operational Red Hat Virtualization node on installer-provisioned infrastructure

Use this procedure to replace an operational node on Red Hat Virtualization installer-provisioned infrastructure (IPI).

Procedure

1. Log in to OpenShift Web Console and click Compute → Nodes.
2. Identify the node that needs to be replaced. Take a note of its Machine Name.
3. Mark the node as unschedulable using the following command:
   
   ```
   $ oc adm cordon <node_name>
   ```
4. Drain the node using the following command:
   
   ```
   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
   ```

   **IMPORTANT**
   
   This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

5. Click Compute → Machines. Search for the required machine.
6. Besides the required machine, click the Action menu ( ⋱ ) → Delete Machine
7. Click Delete to confirm the machine deletion. A new machine is automatically created. Wait for new machine to start and transition into Running state.

   **IMPORTANT**
   
   This activity may take at least 5-10 minutes or more.

8. Click Compute → Nodes, confirm if the new node is in Ready state.
9. Apply the OpenShift Container Storage label to the new node using any one of the following:

   **From User interface**
   
   a. For the new node, click Action Menu ( ⋱ ) → Edit Labels
   b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

   **From Command line interface**
   
   • Execute the following command to apply the OpenShift Container Storage label to the new node:
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""

Verification steps

1. Execute the following command and verify that the new node is present in the output:

   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:
   - csi-cephfsplugin-*
   - csi-rbdplugin-*

3. Verify that all other required OpenShift Container Storage pods are in Running state.

4. Verify that new OSD pods are running on the replacement node.

   $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in previous step, do the following:
   a. Create a debug pod and open a chroot environment for the selected host(s).

      $ oc debug node/<node name>
      $ chroot /host

   b. Run "lsblk" and check for the “crypt” keyword beside the ocs-deviceset name(s)

      $ lsblk

6. If verification steps fail, contact Red Hat Support.

1.3.2. Replacing a failed Red Hat Virtualization node on installer-provisioned infrastructure

Perform this procedure to replace a failed node which is not operational on Red Hat Virtualization installer-provisioned infrastructure (IPI) for OpenShift Container Storage.

Procedure

1. Log in to OpenShift Web Console and click Compute → Nodes.

2. Identify the faulty node. Take a note of its Machine Name.

3. Log in to Red Hat Virtualization Administration Portal and remove the virtual disks associated with mon and OSDs from the failed Virtual Machine.
   This step is required so that the disks are not deleted when the VM instance is deleted as part of the Delete machine step.
IMPORTANT
Do not select the Remove Permanently option when removing the disk(s).

4. In the OpenShift Web Console, click Compute → Machines. Search for the required machine.

5. Click Actions → Edit Annotations, and click Add More.

6. Add machine.openshift.io/exclude-node-draining and click Save.

7. Click Actions → Delete Machine and click Delete.
A new machine is automatically created, wait for new machine to start.

IMPORTANT
This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

8. Click Compute → Nodes, confirm if the new node is in Ready state.

9. Apply the OpenShift Container Storage label to the new node using any one of the following:

   From User interface
   a. For the new node, click Action Menu (⋮) → Edit Labels
   b. Add cluster.ocs.openshift.io/openshift-storage and click Save.

   From Command line interface
   - Execute the following command to apply the OpenShift Container Storage label to the new node:

     $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""

10. (Optional) If the failed VM is not removed automatically, remove the VM from Red Hat Virtualization Administration Portal.

Verification steps
1. Execute the following command and verify that the new node is present in the output:

   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:

   - csi-cephfsplugin-

   - csi-rbdplugin-

3. Verify that all other required OpenShift Container Storage pods are in Running state.
4. Verify that new OSD pods are running on the replacement node.

```
$ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | grep osd
```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
For each of the new nodes identified in previous step, do the following:

   a. Create a debug pod and open a chroot environment for the selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run "lsblk" and check for the “crypt” keyword beside the `ocs-deviceset` name(s)

```
$ lsblk
```

6. If verification steps fail, contact Red Hat Support.

1.4. OPENSPLIT CONTAINER STORAGE DEPLOYED ON MICROSOFT AZURE

To replace an operational node, see Section 1.4.1, “Replacing operational nodes on Azure installer-provisioned infrastructure” To replace a failed node, see Section 1.4.2, “Replacing failed nodes on Azure installer-provisioned infrastructure”

1.4.1. Replacing operational nodes on Azure installer-provisioned infrastructure

Use this procedure to replace an operational node on Azure installer-provisioned infrastructure (IPI).

Procedure

1. Log in to OpenShift Web Console and click **Compute → Nodes**.
2. Identify the node that needs to be replaced. Take a note of its **Machine Name**.
3. Mark the node as unschedulable using the following command:

```
$ oc adm cordon <node_name>
```
4. Drain the node using the following command:

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

**IMPORTANT**

This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

5. Click **Compute → Machines**. Search for the required machine.
6. Besides the required machine, click the Action menu (⋮) → Delete Machine.

7. Click Delete to confirm the machine deletion. A new machine is automatically created.

8. Wait for new machine to start and transition into Running state.

**IMPORTANT**

This activity may take at least 5-10 minutes or more.

9. Click Compute → Nodes, confirm if the new node is in Ready state.

10. Apply the OpenShift Container Storage label to the new node using any one of the following:

    **From User interface**
    a. For the new node, click Action Menu (⋮) → Edit Labels
    b. Add cluster.ocs.openshift.io/openshift-storage and click Save.

    **From Command line interface**
    - Execute the following command to apply the OpenShift Container Storage label to the new node:
      
      ```
      $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
      ```

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:
   
   ```
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage="" | cut -d' ' -f1
   ```

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:
   - csi-cephfplugin-*
   - csi-rbdplugin-*

3. Verify that all other required OpenShift Container Storage pods are in Running state.

4. Verify that new OSD pods are running on the replacement node.
   
   ```
   $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
   ```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in previous step, do the following:
   a. Create a debug pod and open a chroot environment for the selected host(s).
      
      ```
      $ oc debug node/<node_name>
      $ chroot /host
      ```
b. Run “lsblk” and check for the “crypt” keyword beside the ocs-deviceset name(s)

```
$ lsblk
```

6. If verification steps fail, contact Red Hat Support.

1.4.2. Replacing failed nodes on Azure installer-provisioned infrastructure

Perform this procedure to replace a failed node which is not operational on Azure installer-provisioned infrastructure (IPI) for OpenShift Container Storage.

**Procedure**

1. Log in to OpenShift Web Console and click **Compute → Nodes**.

2. Identify the faulty node and click on its **Machine Name**.

3. Click **Actions → Edit Annotations**, and click **Add More**.

4. Add `machine.openshift.io/exclude-node-draining` and click **Save**.

5. Click **Actions → Delete Machine**, and click **Delete**.

6. A new machine is automatically created, wait for new machine to start.

---

**IMPORTANT**

This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

7. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.

8. Apply the OpenShift Container Storage label to the new node using any one of the following:

   **From User interface**
   
   a. For the new node, click **Action Menu (⋮) → Edit Labels**
   
   b. Add `cluster.ocs.openshift.io/openshift-storage` and click **Save**.

   **From Command line interface**
   
   * Execute the following command to apply the OpenShift Container Storage label to the new node:

   ```
   $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

9. [Optional]: If the failed Azure instance is not removed automatically, terminate the instance from Azure console.

**Verification steps**
1. Execute the following command and verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:
   
   - csi-cephfsplugin-*
   - csi-rbdplugin-*

3. Verify that all other required OpenShift Container Storage pods are in Running state.

4. Verify that new OSD pods are running on the replacement node.

```
$ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

For each of the new nodes identified in previous step, do the following:

   a. Create a debug pod and open a chroot environment for the selected host(s).

   ```
   $ oc debug node/<node name>
   $ chroot /host
   ```

   b. Run “lsblk” and check for the “crypt” keyword beside the ocs-deviceset name(s)

   ```
   $ lsblk
   ```

6. If verification steps fail, contact Red Hat Support.
CHAPTER 2. OPENSSHIFT CONTAINER STORAGE DEPLOYED USING LOCAL STORAGE DEVICES

2.1. REPLACING STORAGE NODES ON BARE METAL INFRASTRUCTURE

- To replace an operational node, see Section 2.1.1, “Replacing an operational node on bare metal user-provisioned infrastructure”

- To replace a failed node, see Section 2.1.2, “Replacing a failed node on bare metal user-provisioned infrastructure”

2.1.1. Replacing an operational node on bare metal user-provisioned infrastructure

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.

- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

- If you upgraded to OpenShift Container Storage version 4.8 from a previous version, and have not already created the LocalVolumeDiscovery and LocalVolumeSet objects, do so now by following the procedure described in Post-update configuration changes for clusters backed by local storage.

Procedure

1. Identify the NODE and get labels on the node to be replaced.

   $ oc get nodes --show-labels | grep <node_name>

2. Identify the mon (if any) and OSDs that are running in the node to be replaced.

   $ oc get pods -n openshift-storage -o wide | grep -i <node_name>

3. Scale down the deployments of the pods identified in the previous step. For example:

   $ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
   $ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
   $ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage

4. Mark the node as unschedulable.

   $ oc adm cordon <node_name>

5. Drain the node.

   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
6. Delete the node.

   $ oc delete node <node_name>

7. Get a new bare metal machine with required infrastructure. See Installing a cluster on bare metal.

   **IMPORTANT**

   For information about how to replace a master node when you have installed OpenShift Container Storage on a three-node OpenShift compact bare-metal cluster, see the Backup and Restore guide in the OpenShift Container Platform documentation.

8. Create a new OpenShift Container Platform node using the new bare metal machine.

9. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

   $ oc get csr

10. Approve all required OpenShift Container Platform CSRs for the new node:

    $ oc adm certificate approve <Certificate_Name>

11. Click Compute → Nodes in OpenShift Web Console, confirm if the new node is in Ready state.

12. Apply the OpenShift Container Storage label to the new node using any one of the following:

    **From User interface**

    a. For the new node, click Action Menu (⋮) → Edit Labels

    b. Add cluster.ocs.openshift.io/openshift-storage and click Save.

    **From Command line interface**

    - Execute the following command to apply the OpenShift Container Storage label to the new node:

      $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""

13. Identify the namespace where OpenShift local storage operator is installed and assign it to local_storage_project variable:

    $ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)

    For example:

    $ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
    openshift-local-storage
14. Add a new worker node to `localVolumeDiscovery` and `localVolumeSet`.

   a. Update the `localVolumeDiscovery` definition to include the new node and remove the failed node.

   ```
   # oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
   [...]
   nodeSelector:
   nodeSelectorTerms:
     - matchExpressions:
       - key: kubernetes.io/hostname
         operator: In
         values:
           - server1.example.com
           - server2.example.com
           #- server3.example.com
           - newnode.example.com
   [...]
   
   Remember to save before exiting the editor.
   
   In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.
   
   b. Determine which `localVolumeSet` to edit.

   ```
   # oc get -n $local_storage_project localvolumeset
   NAME    AGE
   localblock  25h
   ```
   
   c. Update the `localVolumeSet` definition to include the new node and remove the failed node.

   ```
   # oc edit -n $local_storage_project localvolumeset localblock
   [...]
   nodeSelector:
   nodeSelectorTerms:
     - matchExpressions:
       - key: kubernetes.io/hostname
         operator: In
         values:
           - server1.example.com
           - server2.example.com
           #- server3.example.com
           - newnode.example.com
   [...]
   
   Remember to save before exiting the editor.
   
   In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.
   
15. Verify that the new `localblock` PV is available.
$ oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s

16. Change to the **openshift-storage** project.

$ oc project openshift-storage

17. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -

18. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.
A status of **Completed** confirms that the OSD removal job succeeded.

# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage

**NOTE**

If **ocs-osd-removal-job** fails and the pod is not in the expected **Completed** state, check the pod logs for further debugging. For example:

# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage

19. Delete the **ocs-osd-removal-job**.

# oc delete -n openshift-storage job ocs-osd-removal-job

Example output:

job.batch "ocs-osd-removal-job" deleted

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:

$ oc nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1

2. Click **Workloads → Pods** confirm that at least the following pods on the new node are in **Running** state:
   - csi-cephfsplugin-*
   - csi-rbdplugin-*

3. Verify that all other required OpenShift Container Storage pods are in Running state. Ensure that the new incremental **mon** is created and is in the Running state.

$ oc get pod -n openshift-storage | grep mon
Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66       2/2     Running
  0          38m
rook-ceph-mon-b-6776bc469b-tzzt8       2/2     Running
  0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh       2/2     Running
  0          4m8s
```

OSD and Mon might take several minutes to get to the Running state.

4. Verify that new OSD pods are running on the replacement node.

```
$ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

For each of the new nodes identified in previous step, do the following:

a. Create a debug pod and open a chroot environment for the selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

b. Run “lsblk” and check for the “crypt” keyword beside the ocs-deviceset name(s)

```
$ lsblk
```

6. If verification steps fail, contact Red Hat Support.

### 2.1.2. Replacing a failed node on bare metal user-provisioned infrastructure

#### Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.

- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

- If you upgraded to OpenShift Container Storage version 4.8 from a previous version, and have not already created the LocalVolumeDiscovery and LocalVolumeSet objects, do so now by following the procedure described in Post-update configuration changes for clusters backed by local storage.

#### Procedure

1. Identify the NODE and get labels on the node to be replaced.

```
$ oc get nodes --show-labels | grep <node_name>
```

2. Identify the mon (if any) and OSDs that are running in the node to be replaced.

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```
3. Scale down the deployments of the pods identified in the previous step. For example:

$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage

4. Mark the node as unschedulable.

$ oc adm cordon <node_name>

5. Remove the pods which are in Terminating state.

$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system("oc -n "$1" delete pods "$2" --grace-period=0 " --force ")}'

6. Drain the node.

$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets

7. Delete the node.

$ oc delete node <node_name>

8. Get a new bare metal machine with required infrastructure. See Installing a cluster on bare metal.

9. Create a new OpenShift Container Platform node using the new bare metal machine.

10. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

$ oc get csr

11. Approve all required OpenShift Container Platform CSRs for the new node:

$ oc adm certificate approve <Certificate_Name>

12. Click Compute → Nodes in OpenShift Web Console, confirm if the new node is in Ready state.

13. Apply the OpenShift Container Storage label to the new node using any one of the following:

   From User interface
   a. For the new node, click Action Menu (⋮) → Edit Labels

   IMPORTANT
   For information about how to replace a master node when you have installed OpenShift Container Storage on a three-node OpenShift compact bare-metal cluster, see the Backup and Restore guide in the OpenShift Container Platform documentation.
b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

**From Command line interface**

- Execute the following command to apply the OpenShift Container Storage label to the new node:
  
  ```bash
  $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
  ```

14. Identify the namespace where OpenShift local storage operator is installed and assign it to `local_storage_project` variable:

  ```bash
  $ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
  ```

  For example:

  ```bash
  $ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
  echo $local_storage_project
  openshift-local-storage
  ```

15. Add a new worker node to `localVolumeDiscovery` and `localVolumeSet`.

  a. Update the `localVolumeDiscovery` definition to include the new node and remove the failed node.

  ```bash
  # oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
  [...] 
  nodeSelector:
  nodeSelectorTerms:
  - matchExpressions:
    - key: kubernetes.io/hostname
      operator: In
      values:
      - server1.example.com
      - server2.example.com
      # server3.example.com
      - newnode.example.com
  [...] 
  ```

  Remember to save before exiting the editor.

  In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

  b. Determine which `localVolumeSet` to edit.

  ```bash
  # oc get -n $local_storage_project localvolumeset
  NAME          AGE
  localblock   25h
  ```

  c. Update the `localVolumeSet` definition to include the new node and remove the failed node.
# oc edit -n $local_storage_project localvolumeset localblock

```yaml
nodeSelector:
  nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
            - server1.example.com
            - server2.example.com
            # - server3.example.com
            - newnode.example.com
```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

16. Verify that the new `localblock` PV is available.

   ```bash
   $ oc get pv | grep localblock | grep Available
   local-pv-551d950     512Gi    RWO    Delete  Available
   localblock     26s
   ```

17. Change to the `openshift-storage` project.

   ```bash
   $ oc project openshift-storage
   ```

18. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

   ```bash
   $ oc process -n openshift-storage ocs-osd-removal \
   -p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
   ```

19. Verify that the OSD was removed successfully by checking the status of the `ocs-osd-removal-job` pod.

   A status of `Completed` confirms that the OSD removal job succeeded.

   ```bash
   # oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
   ```

   **NOTE**

   If `ocs-osd-removal-job` fails and the pod is not in the expected `Completed` state, check the pod logs for further debugging. For example:

   ```bash
   # oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
   ```

20. Delete the `ocs-osd-removal-job`.

   ```bash
   # oc delete -n openshift-storage job ocs-osd-removal-job
   ```

Example output:
Verification steps

1. Execute the following command and verify that the new node is present in the output:
   
   ```bash
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
   ```

2. Click **Workloads → Pods** confirm that at least the following pods on the new node are in **Running** state:
   
   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in Running state. Ensure that the new incremental **mon** is created and is in the Running state.
   
   ```bash
   $ oc get pod -n openshift-storage | grep mon
   ```

   Example output:
   
   ```
   rook-ceph-mon-a-cd575c89b-b6k66         2/2     Running
   0          38m
   rook-ceph-mon-b-6776bc469b-tzzt8        2/2     Running
   0          38m
   rook-ceph-mon-d-5ff5d488b5-7v8xh        2/2     Running
   0          4m8s
   ```

   OSD and Mon might take several minutes to get to the **Running** state.

4. Verify that new OSD pods are running on the replacement node.
   
   ```bash
   $ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
   ```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   
   For each of the new nodes identified in previous step, do the following:
   
   a. Create a debug pod and open a chroot environment for the selected host(s).
      
      ```bash
      $ oc debug node/<node name>
      $ chroot /host
      ```

   b. Run `lsblk` and check for the “crypt” keyword beside the **ocs-deviceset** name(s)
      
      ```bash
      $ lsblk
      ```

6. If verification steps fail, contact Red Hat Support.

---

### 2.2. REPLACING STORAGE NODES ON IBM Z OR LINUXONE INFRASTRUCTURE
You can choose one of the following procedures to replace storage nodes:

- Section 2.2.1, “Replacing operational nodes on IBM Z or LinuxONE infrastructure”
- Section 2.2.2, “Replacing failed nodes on IBM Z or LinuxONE infrastructure”

### 2.2.1. Replacing operational nodes on IBM Z or LinuxONE infrastructure

Use this procedure to replace an operational node on IBM Z or LinuxONE infrastructure.

#### Procedure

1. Log in to OpenShift Web Console.
2. Click **Compute → Nodes**.
3. Identify the node that needs to be replaced. Take a note of its **Machine Name**.
4. Mark the node as unschedulable using the following command:
   ```bash
   $ oc adm cordon <node_name>
   ```
5. Drain the node using the following command:
   ```bash
   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
   ```
   **IMPORTANT**
   This activity may take at least 5-10 minutes. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.
6. Click **Compute → Machines**. Search for the required machine.
7. Besides the required machine, click the **Action menu (⋯) → Delete Machine**.
8. Click **Delete** to confirm the machine deletion. A new machine is automatically created.
9. Wait for the new machine to start and transition into **Running** state.
   **IMPORTANT**
   This activity may take at least 5-10 minutes.
10. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.
11. Apply the OpenShift Container Storage label to the new node using any one of the following:

    **From User interface**
    a. For the new node, click **Action Menu (⋯) → Edit Labels**
    b. Add `cluster.ocs.openshift.io/openshift-storage` and click **Save**.
From command line interface

- Execute the following command to apply the OpenShift Container Storage label to the new node:

  ```shell
  $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
  ```

Verification steps

1. Execute the following command and verify that the new node is present in the output:

  ```shell
  $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
  ```

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:

   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in Running state.

4. Verify that new OSD pods are running on the replacement node.

   ```shell
   $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
   ```

5. (Optional) If data encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   For each of the new nodes identified in previous step, do the following:

   a. Create a debug pod and open a chroot environment for the selected host(s).

      ```shell
      $ oc debug node/<node name>
      $ chroot /host
      ```

   b. Run `lsblk` and check for the “crypt” keyword beside the `ocs-deviceset` name(s)

      ```shell
      $ lsblk
      ```

6. If verification steps fail, contact Red Hat Support.

2.2.2. Replacing failed nodes on IBM Z or LinuxONE infrastructure

Perform this procedure to replace a failed node which is not operational on IBM Z or LinuxONE infrastructure for OpenShift Container Storage.

Procedure

1. Log in to OpenShift Web Console and click Compute → Nodes.

2. Identify the faulty node and click on its Machine Name.

3. Click Actions → Edit Annotations, and click Add More.
4. Add `machine.openshift.io/exclude-node-draining` and click **Save**.

5. Click **Actions** → **Delete Machine**, and click **Delete**.

6. A new machine is automatically created, wait for new machine to start.

   **IMPORTANT**
   
   This activity may take at least 5-10 minutes. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

7. Click **Compute** → **Nodes**, confirm if the new node is in **Ready** state.

8. Apply the OpenShift Container Storage label to the new node using any one of the following:

   **From the web user interface**
   
   a. For the new node, click **Action Menu (⋮) → Edit Labels**
   
   b. Add `cluster.ocs.openshift.io/openshift-storage` and click **Save**.

   **From the command line interface**
   
   - Execute the following command to apply the OpenShift Container Storage label to the new node:
     
     ```bash
     $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
     ```

9. Execute the following command and verify that the new node is present in the output:

   ```bash
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
   ```

10. Click **Workloads** → **Pods**, confirm that at least the following pods on the new node are in **Running** state:

    - `csi-cephfsplugin-*`
    - `csi-rbdplugin-*`

11. Verify that all other required OpenShift Container Storage pods are in **Running** state.

12. Verify that new OSD pods are running on the replacement node.

   ```bash
   $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | grep osd
   ```

13. (Optional) If data encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

    For each of the new nodes identified in previous step, do the following:

    a. Create a debug pod and open a chroot environment for the selected host(s).

    ```bash
    $ oc debug node/<node name>
    $ chroot /host
    ```
b. Run "lsblk" and check for the “crypt” keyword beside the ocs-deviceset name(s)

```
$ lsblk
```

14. If verification steps fail, contact Red Hat Support.

### 2.3. REPLACING STORAGE NODES ON VMWARE INFRASTRUCTURE

- To replace an operational node, see:
  - [Section 2.3.1, “Replacing an operational node on VMware user-provisioned infrastructure”](#)
  - [Section 2.3.2, “Replacing an operational node on VMware installer-provisioned infrastructure”](#)

- To replace a failed node, see:
  - [Section 2.3.3, “Replacing a failed node on VMware user-provisioned infrastructure”](#)
  - [Section 2.3.4, “Replacing a failed node on VMware installer-provisioned infrastructure”](#)

#### 2.3.1. Replacing an operational node on VMware user-provisioned infrastructure

**Prerequisites**

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.

- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

- If you upgraded to OpenShift Container Storage version 4.8 from a previous version, and have not already created the [LocalVolumeDiscovery](#) and [LocalVolumeSet](#) objects, do so now by following the procedure described in [Post-update configuration changes for clusters backed by local storage](#).

**Procedure**

1. Identify the NODE and get labels on the node to be replaced.

   ```
   $ oc get nodes --show-labels | grep <node_name>
   ```

2. Identify the mon (if any) and OSDs that are running in the node to be replaced.

   ```
   $ oc get pods -n openshift-storage -o wide | grep -i <node_name>
   ```

3. Scale down the deployments of the pods identified in the previous step.
   
   For example:
   
   ```
   $ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
   $ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
   $ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
   ```
4. Mark the node as unschedulable.

    $ oc adm cordon <node_name>

5. Drain the node.

    $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets

6. Delete the node.

    $ oc delete node <node_name>

7. Log in to vSphere and terminate the identified VM.

8. Create a new VM on VMware with the required infrastructure. See Supported Infrastructure and Platforms.

9. Create a new OpenShift Container Platform worker node using the new VM.

10. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

    $ oc get csr

11. Approve all required OpenShift Container Platform CSRs for the new node:

    $ oc adm certificate approve <Certificate_Name>

12. Click Compute → Nodes in OpenShift Web Console, confirm if the new node is in Ready state.

13. Apply the OpenShift Container Storage label to the new node using any one of the following:

    **From User interface**
    a. For the new node, click Action Menu ( ⋮ ) → Edit Labels
    b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

    **From Command line interface**
    - Execute the following command to apply the OpenShift Container Storage label to the new node:
      
      $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""

14. Identify the namespace where OpenShift local storage operator is installed and assign it to `local_storage_project` variable:

    $ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)

    For example:
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage

15. Add a new worker node to **localVolumeDiscovery** and **localVolumeSet**.

a. Update the **localVolumeDiscovery** definition to include the new node and remove the failed node.

```bash
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
  nodeSelectorTerms:
    - matchExpressions:
      - key: kubernetes.io/hostname
        operator: In
        values:
          - server1.example.com
          - server2.example.com
          # server3.example.com
          - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

b. Determine which **localVolumeSet** to edit.

```bash
# oc get -n $local_storage_project localvolumeset
NAME          AGE
localblock   25h
```

c. Update the **localVolumeSet** definition to include the new node and remove the failed node.

```bash
# oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
  nodeSelectorTerms:
    - matchExpressions:
      - key: kubernetes.io/hostname
        operator: In
        values:
          - server1.example.com
          - server2.example.com
          # server3.example.com
          - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.
16. Verify that the new `localblock` PV is available.

   ```bash
   $ oc get pv | grep localblock | grep Available
   local-pv-551d950   512Gi   RWO   Delete  Available
   localblock        26s
   ```

17. Change to the `openshift-storage` project.

   ```bash
   $ oc project openshift-storage
   ```

18. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

   ```bash
   $ oc process -n openshift-storage ocs-osd-removal
   -p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
   ```

19. Verify that the OSD was removed successfully by checking the status of the `ocs-osd-removal-job` pod.
   A status of `Completed` confirms that the OSD removal job succeeded.

   ```bash
   # oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
   ```

   **NOTE**

   If `ocs-osd-removal-job` fails and the pod is not in the expected `Completed` state, check the pod logs for further debugging. For example:

   ```bash
   # oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
   ```

20. Delete the `ocs-osd-removal-job`.

   ```bash
   # oc delete -n openshift-storage job ocs-osd-removal-job
   ```

   Example output:

   ```
   job.batch "ocs-osd-removal-job" deleted
   ```

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:

   ```bash
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Click **Workloads → Pods** confirm that at least the following pods on the new node are in `Running` state:
   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in Running state. Ensure that the new incremental `mon` is created and is in the Running state.
$ oc get pod -n openshift-storage | grep mon

Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66     2/2     Running  
  0          38m
rook-ceph-mon-b-6776bc469b-tzzt8    2/2     Running  
  0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh    2/2     Running  
  0          4m8s
```

OSD and Mon might take several minutes to get to the **Running** state.

4. Verify that new OSD pods are running on the replacement node.

```
$ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

For each of the new nodes identified in previous step, do the following:

   a. Create a debug pod and open a chroot environment for the selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

6. If verification steps fail, contact Red Hat Support.

### 2.3.2. Replacing an operational node on VMware installer-provisioned infrastructure

**Prerequisites**

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.

- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

- If you upgraded to OpenShift Container Storage version 4.8 from a previous version, and have not already created the **LocalVolumeDiscovery** and **LocalVolumeSet** objects, do so now by following the procedure described in Post-update configuration changes for clusters backed by local storage.

**Procedure**

1. Log in to OpenShift Web Console and click **Compute → Nodes**.

2. Identify the node that needs to be replaced. Take a note of its **Machine Name**.

3. Get labels on the node to be replaced.
4. Identify the mon (if any) and OSDs that are running in the node to be replaced.

   $ oc get nodes --show-labels | grep <node_name>

5. Scale down the deployments of the pods identified in the previous step. For example:

   $ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
   $ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
   $ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage

6. Mark the node as unschedulable.

   $ oc adm cordon <node_name>

7. Drain the node.

   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets

8. Click **Compute → Machines**. Search for the required machine.

9. Besides the required machine, click the **Action menu (⋯) → Delete Machine**

10. Click **Delete** to confirm the machine deletion. A new machine is automatically created.

11. Wait for the new machine to start and transition into **Running** state.

   **IMPORTANT**

   This activity may take at least 5-10 minutes or more.

12. Click **Compute → Nodes** in OpenShift Web Console, confirm if the new node is in **Ready** state.

13. Physically add a new device to the node.

14. Apply the OpenShift Container Storage label to the new node using any one of the following:

   **From User interface**
   
   a. For the new node, click **Action Menu (⋯) → Edit Labels**
   
   b. Add `cluster.ocs.openshift.io/openshift-storage` and click **Save**.

   **From Command line interface**

   - Execute the following command to apply the OpenShift Container Storage label to the new node:

     $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
15. Identify the namespace where OpenShift local storage operator is installed and assign it to `local_storage_project` variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

16. Add a new worker node to `localVolumeDiscovery` and `localVolumeSet`.

a. Update the `localVolumeDiscovery` definition to include the new node and remove the failed node.

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
  nodeSelectorTerms:
  - matchExpressions:
    - key: kubernetes.io/hostname
      operator: In
      values:
      - server1.example.com
      - server2.example.com
    - key: kubernetes.io/hostname
      operator: In
      values:
      - server3.example.com
    - key: kubernetes.io/hostname
      operator: In
      values:
      - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

b. Determine which `localVolumeSet` to edit.

```
# oc get -n $local_storage_project localvolumeset
NAME          AGE
localblock   25h
```

c. Update the `localVolumeSet` definition to include the new node and remove the failed node.

```
# oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
  nodeSelectorTerms:
  - matchExpressions:
    - key: kubernetes.io/hostname
      operator: In
      values:
      - server1.example.com
      - server2.example.com
```

CHAPTER 2. OPENSOURCE CONTAINER STORAGE DEPLOYED USING LOCAL STORAGE DEVICES
Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

17. Verify that the new `localblock` PV is available.

   ```bash
   $ oc get pv | grep localblock | grep Available
   local-pv-551d950  512Gi  RWO  Delete  Available
   localblock  26s
   ```

18. Change to the `openshift-storage` project.

   ```bash
   $ oc project openshift-storage
   ```

19. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

   ```bash
   $ oc process -n openshift-storage ocs-osd-removal \
   -p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
   ```

20. Verify that the OSD was removed successfully by checking the status of the `ocs-osd-removal-job` pod.

    A status of `Completed` confirms that the OSD removal job succeeded.

    ```bash
    # oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
    ```

    **NOTE**

    If `ocs-osd-removal-job` fails and the pod is not in the expected `Completed` state, check the pod logs for further debugging. For example:

    ```bash
    # oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
    ```

21. Identify the PV associated with the PVC.

    ```bash
    #oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
    local-pv-d6bf175b  1490Gi  RWO  Delete Released  openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock  2d22h  compute-1
    ```

    If there is a PV in `Released` state, delete it.

    ```bash
    # oc delete pv <persistent-volume>
    ```

    For example:

    ```bash
    # oc delete pv local-pv-d6bf175b
    persistentvolume "local-pv-d9c5cbd6" deleted
    ```
22. Identify the **crashcollector** pod deployment.

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

If there is an existing **crashcollector** pod deployment, delete it.

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

23. Delete the **ocs-osd-removal-job**.

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods** confirm that at least the following pods on the new node are in **Running** state:

   - csi-cephfsplugin-*
   - csi-rbdplugin-*

3. Verify that all other required OpenShift Container Storage pods are in Running state. Ensure that the new incremental **mon** is created and is in the Running state.

```
$ oc get pod -n openshift-storage | grep mon
```

Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66         2/2     Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8        2/2     Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh        2/2     Running
0          4m8s
```

OSD and Mon might take several minutes to get to the **Running** state.

4. Verify that new OSD pods are running on the replacement node.

```
$ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd
```

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
For each of the new nodes identified in previous step, do the following:

a. Create a debug pod and open a chroot environment for the selected host(s).

   $ oc debug node/<node name>
   $ chroot /host

b. Run "lsblk" and check for the “crypt” keyword beside the ocs-deviceset name(s)

   $ lsblk

6. If verification steps fail, contact Red Hat Support.

2.3.3. Replacing a failed node on VMware user-provisioned infrastructure

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.
- If you upgraded to OpenShift Container Storage version 4.8 from a previous version, and have not already created the LocalVolumeDiscovery and LocalVolumeSet objects, do so now by following the procedure described in Post-update configuration changes for clusters backed by local storage.

Procedure

1. Identify the NODE and get labels on the node to be replaced.

   $ oc get nodes --show-labels | grep <node_name>

2. Identify the mon (if any) and OSDs that are running in the node to be replaced.

   $ oc get pods -n openshift-storage -o wide | grep -i <node_name>

3. Scale down the deployments of the pods identified in the previous step.

   For example:

   $ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
   $ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
   $ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage

4. Mark the node as unschedulable.

   $ oc adm cordon <node_name>

5. Remove the pods which are in Terminating state.
6. Drain the node.
   ```
   $ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n "$1" delete pods "$2" --grace-period=0 " --force ")}'
   ```

7. Delete the node.
   ```
   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
   $ oc delete node <node_name>
   ```

8. Log in to vSphere and terminate the identified VM.

9. Create a new VM on VMware with the required infrastructure. See [Supported Infrastructure and Platforms](#).

10. Create a new OpenShift Container Platform worker node using the new VM.

11. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

    ```
    $ oc get csr
    ```

12. Approve all required OpenShift Container Platform CSRs for the new node:

    ```
    $ oc adm certificate approve <Certificate_Name>
    ```

13. Click **Compute → Nodes** in OpenShift Web Console, confirm if the new node is in **Ready** state.

14. Apply the OpenShift Container Storage label to the new node using any one of the following:

    **From User interface**
    
    a. For the new node, click **Action Menu ( ⋮ ) → Edit Labels**
    
    b. Add `cluster.ocs.openshift.io/openshift-storage` and click **Save**.

    **From Command line interface**
    
    - Execute the following command to apply the OpenShift Container Storage label to the new node:
      ```
      $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
      ```

15. Identify the namespace where OpenShift local storage operator is installed and assign it to `local_storage_project` variable:

    ```
    $ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
    ```

    For example:
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage

16. Add a new worker node to `localVolumeDiscovery` and `localVolumeSet`.

a. Update the `localVolumeDiscovery` definition to include the new node and remove the failed node.

```shell
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
  nodeSelectorTerms:
  - matchExpressions:
    - key: kubernetes.io/hostname
      operator: In
      values:
    - server1.example.com
    - server2.example.com
    # server3.example.com
    - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

b. Determine which `localVolumeSet` to edit.

```shell
# oc get -n $local_storage_project localvolumeset
NAME    AGE
localblock 25h
```

c. Update the `localVolumeSet` definition to include the new node and remove the failed node.

```shell
# oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
  nodeSelectorTerms:
  - matchExpressions:
    - key: kubernetes.io/hostname
      operator: In
      values:
    - server1.example.com
    - server2.example.com
    # server3.example.com
    - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.
17. Verify that the new **localblock** PV is available.

   ```bash
   $ oc get pv | grep localblock | grep Available
   local-pv-551d9550  512Gi  RWO  Delete  Available
   localblock        26s
   ```

18. Change to the **openshift-storage** project.

   ```bash
   $ oc project openshift-storage
   ```

19. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

   ```bash
   $ oc process -n openshift-storage ocs-osd-removal \[-p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
   ```

20. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.

    A status of **Completed** confirms that the OSD removal job succeeded.

    ```bash
    # oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
    ```

    **NOTE**

    If **ocs-osd-removal-job** fails and the pod is not in the expected **Completed** state, check the pod logs for further debugging. For example:

    ```bash
    # oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
    ```

21. Delete the **ocs-osd-removal-job**.

    ```bash
    # oc delete -n openshift-storage job ocs-osd-removal-job
    ```

    Example output:

    ```bash
    job.batch "ocs-osd-removal-job" deleted
    ```

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:

   ```bash
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Click **Workloads → Pods** confirm that at least the following pods on the new node are in **Running** state:

   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in **Running** state. Ensure that the new incremental **mon** is created and is in the Running state.
$ oc get pod -n openshift-storage | grep mon

Example output:

    rook-ceph-mon-a-cd575c89b-b6k66   2/2   Running
    0    38m
    rook-ceph-mon-b-6776bc469b-tzzt8   2/2   Running
    0    38m
    rook-ceph-mon-d-5ff5d488b5-7v8xh   2/2   Running
    0    4m8s

OSD and Mon might take several minutes to get to the Running state.

4. Verify that new OSD pods are running on the replacement node.

$ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in previous step, do the following:
   a. Create a debug pod and open a chroot environment for the selected host(s).

        $ oc debug node/<node name>
        $ chroot /host

   b. Run “lsblk” and check for the “crypt” keyword beside the ocs-deviceset name(s)

        $ lsblk

6. If verification steps fail, contact Red Hat Support.

2.3.4. Replacing a failed node on VMware installer-provisioned infrastructure

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.

- You must be logged into the OpenShift Container Platform (RHOCPS) cluster.

- If you upgraded to OpenShift Container Storage version 4.8 from a previous version, and have not already created the LocalVolumeDiscovery and LocalVolumeSet objects, do so now by following the procedure described in Post-update configuration changes for clusters backed by local storage.

Procedure

1. Log in to OpenShift Web Console and click Compute → Nodes.

2. Identify the node that needs to be replaced. Take a note of its Machine Name.

3. Get labels on the node to be replaced.
$ oc get nodes --show-labels | grep <node_name>

4. Identify the mon (if any) and OSDs that are running in the node to be replaced.
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>

5. Scale down the deployments of the pods identified in the previous step. For example:
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage

6. Mark the node as unschedulable.
$ oc adm cordon <node_name>

7. Remove the pods which are in Terminating state.
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n "$1" delete pods "$2" --grace-period=0 "--force ")}'

8. Drain the node.
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets

9. Click Compute → Machines. Search for the required machine.

10. Besides the required machine, click the Action menu (⋮) → Delete Machine

11. Click Delete to confirm the machine deletion. A new machine is automatically created.

12. Wait for the new machine to start and transition into Running state.

**IMPORTANT**

This activity may take at least 5-10 minutes or more.

13. Click Compute → Nodes in OpenShift Web Console, confirm if the new node is in Ready state.

14. Physically add a new device to the node.

15. Apply the OpenShift Container Storage label to the new node using any one of the following:

   **From User interface**
   a. For the new node, click Action Menu (⋮) → Edit Labels
   b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

   **From Command line interface**
Execute the following command to apply the OpenShift Container Storage label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

16. Identify the namespace where OpenShift local storage operator is installed and assign it to `local_storage_project` variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
```

```
openshift-local-storage
```

17. Add a new worker node to `localVolumeDiscovery` and `localVolumeSet`.

a. Update the `localVolumeDiscovery` definition to include the new node and remove the failed node.

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
  nodeSelectorTerms:
    - matchExpressions:
      - key: kubernetes.io/hostname
        operator: In
        values:
        - server1.example.com
        - server2.example.com
        #- server3.example.com
        - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

b. Determine which `localVolumeSet` to edit.

```
# oc get -n $local_storage_project localvolumeset
NAME     AGE
localblock 25h
```

c. Update the `localVolumeSet` definition to include the new node and remove the failed node.

```
# oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
  nodeSelectorTerms:
```
- matchExpressions:
  - key: kubernetes.io/hostname
    operator: In
    values:
    - server1.example.com
    - server2.example.com
    # server3.example.com
    - newnode.example.com

Remember to save before exiting the editor.

In the above example, server3.example.com was removed and newnode.example.com is the new node.

18. Verify that the new localblock PV is available.

```bash
$ oc get pv | grep localblock | grep Available
local-pv-551d950  512Gi    RWO    Delete  Available localblock  26s
```

19. Change to the openshift-storage project.

```bash
$ oc project openshift-storage
```

20. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

```bash
$ oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
```

21. Verify that the OSD was removed successfully by checking the status of the ocs-osd-removal-job pod.

A status of Completed confirms that the OSD removal job succeeded.

```bash
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

**NOTE**

If ocs-osd-removal-job fails and the pod is not in the expected Completed state, check the pod logs for further debugging. For example:

```bash
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

22. Identify the PV associated with the PVC.

```bash
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
local-pv-d6bf175b  1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock  2d22h compute-1
```

If there is a PV in Released state, delete it.

```bash
# oc delete pv <persistent-volume>
```
For example:

```
# oc delete pv local-pv-d6bf175b
persistentvolume "local-pv-d9c5cbd6" deleted
```

23. Identify the `crashcollector` pod deployment.

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name
-n openshift-storage
```

If there is an existing `crashcollector` pod deployment, delete it.

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name
-n openshift-storage
```

24. Delete the `ocs-osd-removal-job`.

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

**Verification steps**

1. Execute the following command and verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads** → **Pods** confirm that at least the following pods on the new node are in **Running** state:

   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in Running state. Ensure that the new incremental `mon` is created and is in the Running state.

```
$ oc get pod -n openshift-storage | grep mon
```

Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66         2/2     Running
 0          38m
rook-ceph-mon-b-6776bc469b-tzzt8        2/2     Running
 0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh        2/2     Running
 0          4m8s
```

OSD and Mon might take several minutes to get to the **Running** state.

4. Verify that new OSD pods are running on the replacement node.
5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted. For each of the new nodes identified in previous step, do the following:

a. Create a debug pod and open a chroot environment for the selected host(s).

   $ oc debug node/<node name>
   $ chroot /host

b. Run "lsblk" and check for the "crypt" keyword beside the ocs-deviceset name(s)

   $ lsblk

6. If verification steps fail, contact Red Hat Support.

2.4. REPLACING STORAGE NODES ON RED HAT VIRTUALIZATION INFRASTRUCTURE

- To replace an operational node, see Section 2.4.1, “Replacing an operational node on Red Hat Virtualization installer-provisioned infrastructure”

- To replace a failed node, see Section 2.4.2, “Replacing a failed node on Red Hat Virtualization installer-provisioned infrastructure”

2.4.1. Replacing an operational node on Red Hat Virtualization installer-provisioned infrastructure

Use this procedure to replace an operational node on Red Hat Virtualization installer-provisioned infrastructure (IPI).

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources and disks to the node being replaced.

- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

- If you upgraded to OpenShift Container Storage version 4.8 from a previous version, and have not already created the LocalVolumeDiscovery and LocalVolumeSet objects, do so now by following the procedure described in Post-update configuration changes for clusters backed by local storage.

Procedure

1. Log in to OpenShift Web Console and click Compute → Nodes.

2. Identify the node that needs to be replaced. Take a note of its Machine Name.

3. Get labels on the node to be replaced.

   $ oc get nodes --show-labels | grep <node_name>
4. Identify the mon (if any) and OSDs that are running in the node to be replaced.

   $ oc get pods -n openshift-storage -o wide | grep -i <node_name>

5. Scale down the deployments of the pods identified in the previous step.
   For example:

   $ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
   $ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
   $ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
   --replicas=0 -n openshift-storage

6. Mark the nodes as unschedulable.

   $ oc adm cordon <node_name>

7. Drain the node.

   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets

8. Click **Compute → Machines**. Search for the required machine.

9. Besides the required machine, click the **Action menu (⋮) → Delete Machine**

10. Click **Delete** to confirm the machine deletion. A new machine is automatically created. Wait for the new machine to start and transition into Running state.

    ![IMPORTANT]
    This activity may take at least 5-10 minutes or more.

11. Click **Compute → Nodes** in the OpenShift web console. Confirm if the new node is in **Ready** state.

12. Physically add the new device(s) to the node.

13. Apply the OpenShift Container Storage label to the new node using any one of the following:

    **From User interface**
    a. For the new node, click **Action Menu (⋮) → Edit Labels**
    b. Add `cluster.ocs.openshift.io/openshift-storage` and click **Save**.

    **From Command line interface**
    - Execute the following command to apply the OpenShift Container Storage label to the new node:

      $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""

14. Identify the namespace where OpenShift local storage operator is installed and assign it to **local_storage_project** variable:
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)

For example:

$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)

echo $local_storage_project
openshift-local-storage

15. Add a new worker node to **localVolumeDiscovery** and **localVolumeSet**.

- **a.** Update the **localVolumeDiscovery** definition to include the new node and remove the failed node.

  ```
  # oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
  [...]  
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              # server3.example.com
              - newnode.example.com
  [...]  
  ```

  Remember to save before exiting the editor.

  In the above example, **server3.example.com** was removed and **newnode.example.com** is the new node.

- **b.** Determine which **localVolumeSet** to edit.

  ```
  # oc get -n $local_storage_project localvolumeset
  NAME          AGE
  localblock    25h
  ```

- **c.** Update the **localVolumeSet** definition to include the new node and remove the failed node.

  ```
  # oc edit -n $local_storage_project localvolumeset localblock
  [...]  
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              # server3.example.com
              - newnode.example.com
  [...]  
  ```
Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

16. Verify that the new `localblock` PV is available.

   ```bash
   $ oc get pv | grep localblock | grep Available
   local-pv-551d950  512Gi  RWO  Delete  Available
   localblock    26s
   ```

17. Change to the `openshift-storage` project.

   ```bash
   $ oc project openshift-storage
   ```

18. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

   ```bash
   $ oc process -n openshift-storage ocs-osd-removal
   -p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
   ```

19. Verify that the OSD was removed successfully by checking the status of the `ocs-osd-removal-job` pod.

   A status of `Completed` confirms that the OSD removal job succeeded.

   ```bash
   # oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
   ```

   **NOTE**

   If `ocs-osd-removal-job` fails and the pod is not in the expected Completed state, check the pod logs for further debugging. For example:

   ```bash
   # oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
   ```

20. Identify the PV associated with the PVC.

   ```bash
   # oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
   local-pv-d6bf175b  512Gi  RWO  Delete  Released  openshift-storage/ocs-deviceset-0-data-0-6c5pw  localblock  2d22h  server3.example.com
   ```

   If there is a PV in `Released` state, delete it.

   ```bash
   # oc delete pv <persistent-volume>
   ```

   For example:

   ```bash
   # oc delete pv local-pv-d6bf175b
   persistentvolume "local-pv-d6bf175b" deleted
   ```

21. Identify the `crashcollector` pod deployment.
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage

If there is an existing crashcollector pod, delete it.

$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage

22. Delete the ocs-osd-removal job.

# oc delete -n openshift-storage job ocs-osd-removal-job

Example output:

job.batch "ocs-osd-removal-job" deleted

Verification steps

1. Execute the following command and verify that the new node is present in the output:

$ oc nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1

2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:

- csi-cephfsplugin-*
- csi-rbdplugin-*

3. Verify that all other required OpenShift Container Storage pods are in Running state. Ensure that the new incremental mon is created and is in the Running state.

$ oc get pod -n openshift-storage | grep mon

Example output:

rook-ceph-mon-a-cd575c89b-b6k66         2/2     Running  0  38m
rook-ceph-mon-b-6776bc469b-tzzt8        2/2     Running  0  38m
rook-ceph-mon-d-5ff5d488b5-7v8xh        2/2     Running  0  4m8s

OSD and Mon might take several minutes to get to the Running state.

4. Verify that new OSD pods are running on the replacement node.

$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

For each of the new nodes identified in previous step, do the following:

a. Create a debug pod and open a chroot environment for the selected host(s).
$ oc debug node/<node name>
$ chroot /host

b. Run "lsblk" and check for the “crypt” keyword beside the ocs-deviceset name(s)

$ lsblk

6. If verification steps fail, contact Red Hat Support.

2.4.2. Replacing a failed node on Red Hat Virtualization installer-provisioned infrastructure

Perform this procedure to replace a failed node which is not operational on Red Hat Virtualization installer-provisioned infrastructure (IPI) for OpenShift Container Storage.

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.
- If you upgraded to OpenShift Container Storage version 4.8 from a previous version, and have not already created the LocalVolumeDiscovery and LocalVolumeSet objects, do so now by following the procedure described in Post-update configuration changes for clusters backed by local storage.

Procedure

1. Log in to OpenShift Web Console and click Compute → Nodes.

2. Identify the node that needs to be replaced. Take a note of its Machine Name.

3. Get the labels on the node to be replaced.

   $ oc get nodes --show-labels | grep <node_name>

4. Identify the mon (if any) and OSDs that are running in the node to be replaced.

   $ oc get pods -n openshift-storage -o wide | grep -i <node_name>

5. Scale down the deployments of the pods identified in the previous step.

   For example:

   $ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
   $ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
   $ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage

6. Mark the node as unschedulable.

   $ oc adm cordon <node_name>
7. Remove the pods which are in the **Terminating** state.

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system("oc -n "$1" delete pods "$2" --grace-period=0 " --force ")}'
```

8. Drain the node.

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

9. Click **Compute → Machines**. Search for the required machine.

10. Besides the required machine, click the **Action menu (⋯) → Delete Machine**.

11. Click **Delete** to confirm the machine deletion. A new machine is automatically created. Wait for the new machine to start and transition into Running state.

   **IMPORTANT**

   This activity may take at least 5-10 minutes or more.

12. Click **Compute → Nodes** in the OpenShift web console. Confirm if the new node is in Ready state.

13. Physically add the new device(s) to the node.

14. Apply the OpenShift Container Storage label to the new node using any one of the following:

   **From User interface**
   a. For the new node, click **Action Menu (⋯) → Edit Labels**.
   b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

   **From Command line interface**
   - Execute the following command to apply the OpenShift Container Storage label to the new node:

     ```
     $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
     ```

15. Identify the namespace where OpenShift local storage operator is installed and assign it to `local_storage_project` variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

16. Add a new worker node to `localVolumeDiscovery` and `localVolumeSet`. 
a. Update the `localVolumeDiscovery` definition to include the new node and remove the failed node.

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
  nodeSelectorTerms:
    - matchExpressions:
      - key: kubernetes.io/hostname
        operator: In
        values:
          - server1.example.com
          - server2.example.com
          # server3.example.com
          - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

b. Determine which `localVolumeSet` to edit.

```
# oc get -n $local_storage_project localvolumeset
NAME          AGE
localblock    25h
```

c. Update the `localVolumeSet` definition to include the new node and remove the failed node.

```
# oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
  nodeSelectorTerms:
    - matchExpressions:
      - key: kubernetes.io/hostname
        operator: In
        values:
          - server1.example.com
          - server2.example.com
          # server3.example.com
          - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

17. Verify that the new `localblock` PV is available.

```
$ oc get pv | grep localblock | grep Available
local-pv-551d950  512Gi  RWO  Delete  Available
localblock     26s
```
18. Change to the **openshift-storage** project.

   ```bash
   $ oc project openshift-storage
   ```

19. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

   ```bash
   $ oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
   ```

20. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod. A status of **Completed** confirms that the OSD removal job succeeded.

   ```bash
   # oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
   ```

   **NOTE**

   If **ocs-osd-removal-job** fails and the pod is not in the expected **Completed** state, check the pod logs for further debugging. For example:

   ```bash
   # oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
   ```

21. Identify the PV associated with the PVC.

   ```bash
   # oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
   local-pv-d6bf175b  512Gi  RWO  Delete  Released  openshift-storage/ocs-deviceset-0-data-0-6c5pw  localblock  2d22h  server3.example.com
   ```

   If there is a PV in Released state, delete it.

   ```bash
   # oc delete pv <persistent-volume>
   ```

   For example:

   ```bash
   # oc delete pv local-pv-d6bf175b
   persistentvolume "local-pv-d6bf175b" deleted
   ```

22. Identify the **crashcollector** pod deployment.

   ```bash
   $ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
   ```

   If there is an existing crashcollector pod deployment, delete it.

   ```bash
   $ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
   ```

23. Delete the **ocs-osd-removal** job.

   ```bash
   # oc delete -n openshift-storage job ocs-osd-removal-job
   ```
Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

Verification steps

1. Execute the following command and verify that the new node is present in the output:

   `$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1`

2. Click Workloads → Pods confirm that at least the following pods on the new node are in Running state:

   - `csi-cephfsplugin-*`
   - `csi-rbdplugin-*`

3. Verify that all other required OpenShift Container Storage pods are in Running state. Ensure that the new incremental mon is created and is in the Running state.

   `$ oc get pod -n openshift-storage | grep mon`

   Example output:

   ```
   rook-ceph-mon-a-cd575c89b-b6k66         2/2     Running  0   38m
   rook-ceph-mon-b-6776bc469b-tzzt8        2/2     Running  0   38m
   rook-ceph-mon-d-5ff5d488b5-7v8xh        2/2     Running  0   4m8s
   ```

   OSD and Mon might take several minutes to get to the Running state.

4. Verify that new OSD pods are running on the replacement node.

   `$ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | grep osd`

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   For each of the new nodes identified in previous step, do the following:

   a. Create a debug pod and open a chroot environment for the selected host(s).

      `$ oc debug node/<node name>`

      `$ chroot /host`

   b. Run "lsblk" and check for the “crypt” keyword beside the ocs-deviceset name(s)

      `$ lsblk`

6. If verification steps fail, contact Red Hat Support.
2.5. REPLACING STORAGE NODES ON IBM POWER SYSTEMS INFRASTRUCTURE

For OpenShift Container Storage, node replacement can be performed proactively for an operational node and reactively for a failed node for the IBM Power Systems related deployments.

2.5.1. Replacing an operational or failed storage node on IBM Power Systems

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
- You must be logged into OpenShift Container Platform (RHOC) cluster.
- If you upgraded to OpenShift Container Storage 4.8 from a previous version and have not already created the `LocalVolumeDiscovery` object, do so now following the procedure described in `Post-update configuration changes for clusters backed by local storage`.

Procedure

1. Identify the node and get labels on the node to be replaced.

   ```bash
   $ oc get nodes --show-labels | grep <node_name>
   ```

2. Identify the mon (if any) and object storage device (OSD) pods that are running in the node to be replaced.

   ```bash
   $ oc get pods -n openshift-storage -o wide | grep -i <node_name>
   ```

3. Scale down the deployments of the pods identified in the previous step.

   For example:

   ```bash
   $ oc scale deployment rook-ceph-mon-a --replicas=0 -n openshift-storage
   $ oc scale deployment rook-ceph-osd-1 --replicas=0 -n openshift-storage
   $ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
   ```

4. Mark the node as unschedulable.

   ```bash
   $ oc adm cordon <node_name>
   ```

5. Remove the pods which are in Terminating state

   ```bash
   $ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system("oc -n "$1 " delete pods "$2 " --grace-period=0 " --force ")}'
   ```

6. Drain the node.

   ```bash
   $ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
   ```

7. Delete the node.


10. Check for certificate signing requests (CSRs) related to OpenShift Container Storage that are in Pending state:

```
$ oc get csr
```

11. Approve all required OpenShift Container Storage CSRs for the new node:

```
$ oc adm certificate approve <Certificate_Name>
```

12. Click Compute → Nodes in OpenShift Web Console, confirm if the new node is in Ready state.

13. Apply the OpenShift Container Storage label to the new node using your preferred interface:

   **From User interface**
   
   a. For the new node, click Action Menu (⋮) → Edit Labels.
   
   b. Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

   **From Command line interface**
   
   a. Execute the following command to apply the OpenShift Container Storage label to the new node:

   ```
   $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

14. Identify the namespace where OpenShift local storage operator is installed and assign it to `local_storage_project` variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
```

15. Add a new worker node to `localVolumeDiscovery`.

   a. Update the `localVolumeDiscovery` definition to include the new node and remove the failed node.

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
    nodeSelector: 
    nodeSelectorTerms: 
```
- matchExpressions:
  - key: kubernetes.io/hostname
    operator: In
    values:
      #- worker-0
      - worker-1
      - worker-2
      - worker-3

[...]

Remember to save before exiting the editor.

In the above example, **worker-0** was removed and **worker-3** is the new node.

16. Add a newly added worker node to localVolume.

   a. Determine which `localVolume` to edit.

      ```bash
      # oc get -n $local_storage_project localvolume
      NAME           AGE
      localblock    25h
      ```

   b. Update the `localVolume` definition to include the new node and remove the failed node.

      ```bash
      # oc edit -n $local_storage_project localvolume localblock
      [...]
      nodeSelector:
      nodeSelectorTerms:
        - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              #- worker-0
              - worker-1
              - worker-2
              - worker-3
      [...]
      ```

   Remember to save before exiting the editor.

   In the above example, **worker-0** was removed and **worker-3** is the new node.

17. Verify that the new `localblock` PV is available.

    ```bash
    $ oc get pv | grep localblock
    NAME              CAPACITY   ACCESSMODES RECLAIMPOLICY STATUS     CLAIM
    STORAGECLASS       AGE
    local-pv-3e8964d3  500Gi    RWO         Delete       Bound      ocs-deviceset-localblock-2-data-0-mdbg9 localblock 25h
    local-pv-414755e0  500Gi    RWO         Delete       Bound      ocs-deviceset-localblock-1-data-0-4cslf localblock 25h
    local-pv-b481410   500Gi    RWO         Delete       Available
    localblock         3m24s
    local-pv-5c9b8982  500Gi    RWO         Delete       Bound      ocs-deviceset-localblock-0-data-0-g2mmc localblock 25h
    ```
18. Change to the `openshift-storage` project.

```bash
$ oc project openshift-storage
```

19. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

   a. Identify the PVC as afterwards we need to delete PV associated with that specific PVC.

```bash
$ osd_id_to_remove=1
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} | grep ceph.rook.io/pvc
```

where, `osd_id_to_remove` is the integer in the pod name immediately after the `rook-ceph-osd` prefix. In this example, the deployment name is `rook-ceph-osd-1`.

Example output:

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
```

In this example, the PVC name is `ocs-deviceset-localblock-0-data-0-g2mmc`.

b. Remove the failed OSD from the cluster.

```bash
$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_id_to_remove} | oc create -f -
```

You can remove more than one OSD by adding comma separated OSD IDs in the command. (For example: `FAILED_OSD_IDS=0,1,2`)

**WARNING**

This step results in OSD being completely removed from the cluster. Ensure that the correct value of `osd_id_to_remove` is provided.

20. Verify that the OSD is removed successfully by checking the status of the `ocs-osd-removal-job` pod.  
A status of `Completed` confirms that the OSD removal job succeeded.

```bash
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

**NOTE**

If `ocs-osd-removal-job` fails and the pod is not in the expected `Completed` state, check the pod logs for further debugging. For example:

```bash
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```
21. Delete the PV associated with the failed node.
   
a. Identify the PV associated with the PVC. PVC name should be identical to what we obtained in Step 16(a).
   
   ```bash
   # oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
   local-pv-5c9b8982  500Gi  RWO  Delete  Released  openshift-storage/ocs-deviceset-localblock-0-data-0-g2mmc  localblock  24h  worker-0
   ```
   
b. Delete the PV.
   
   ```bash
   # oc delete pv <persistent-volume>
   
   For example:
   
   ```bash
   # oc delete pv local-pv-5c9b8982
   persistentvolume "local-pv-5c9b8982" deleted
   ```
   
22. Delete the crashcollector pod deployment.
   
   ```bash
   $ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>-n openshift-storage
   ```
   
23. Delete the ocs-osd-removal-job.
   
   ```bash
   # oc delete -n openshift-storage job ocs-osd-removal-job
   ```
   
   Example output:
   
   ```bash
   job.batch "ocs-osd-removal-job" deleted
   ```
   
Verification steps

1. Execute the following command and verify that the new node is present in the output:
   
   ```bash
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```
   
2. Click Workloads → Pods, confirm that at least the following pods on the new node are in Running state:
   
   - csi-cephfsplugin-*
   - csi-rbdplugin-*
   
3. Verify that all other required OpenShift Container Storage pods are in Running state. Ensure that the new incremental mon is created and is in the Running state.
   
   ```bash
   $ oc get pod -n openshift-storage | grep mon
   ```
   
   Example output:
OSD and Mon might take several minutes to get to the Running state.

4. Verify that new OSD pods are running on the replacement node.

   $ oc get pods -o wide -n openshift-storage| egrep -i new-node-name | egrep osd

5. (Optional) If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   For each of the new nodes identified in previous step, do the following:

   a. Create a debug pod and open a chroot environment for the selected host(s).

      $ oc debug node/<node name>
      $ chroot /host

   b. Run “lsblk” and check for the “crypt” keyword beside the ocs-deviceset name(s)

      $ lsblk

6. If verification steps fail, contact Red Hat Support.