



# Red Hat OpenStack Services on OpenShift 18.0

## 部署超融合基础架构环境

在 OpenShift 中为 Red Hat OpenStack Services 部署超融合基础架构(HCI)环境



# Red Hat OpenStack Services on OpenShift 18.0 部署超融合基础架构环境

---

在 OpenShift 中为 Red Hat OpenStack Services 部署超融合基础架构(HCI)环境

OpenStack Team  
rhos-docs@redhat.com

## 法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

在 OpenShift 环境中为您的 Red Hat OpenStack Services 部署超融合基础架构(HCI)环境。

---

# 目录

对红帽文档提供反馈 .....	3
<b>第 1 章 配置超融合基础架构环境 .....</b>	<b>4</b>
1.1. DATA PLANE 节点服务列表	4
1.2. 配置数据平面节点网络	4
1.3. 在 DATA PLANE 节点上配置和部署 RED HAT CEPH STORAGE	9
1.4. 配置数据平面以使用并置 RED HAT CEPH STORAGE 服务器	12



## 对红帽文档提供反馈

我们感谢您对文档提供反馈信息。与我们分享您的成功秘诀。

### 在 JIRA 中提供文档反馈

使用 [Create Issue](#) 表单在 OpenShift (RHOSO)或更早版本的 Red Hat OpenStack Platform (RHOSP)上提供有关 Red Hat OpenStack Services 文档的反馈。当您为 RHOSO 或 RHOSP 文档创建问题时，这个问题将在 RHOSO Jira 项目中记录，您可以在其中跟踪您的反馈的进度。

要完成 [Create Issue](#) 表单，请确保您已登录到 JIRA。如果您没有红帽 JIRA 帐户，您可以在 <https://issues.redhat.com> 创建一个帐户。

1. 点击以下链接打开 **Create Issue** 页面：[Create Issue](#)
2. 完成 **Summary** 和 **Description** 字段。在 **Description** 字段中，包含文档 URL、章节号以及问题的详细描述。不要修改表单中的任何其他字段。
3. 点 **Create**。

# 第 1 章 配置超融合基础架构环境

这部分论述了如何部署超融合基础架构(HCI)环境。HCI 环境包含托管 Ceph Storage 和计算服务的数据平面节点。

通过完成以下高级别任务来创建 HCI 环境：

1. 配置数据平面节点网络。
2. 在 data plane 节点上安装 Red Hat Ceph Storage。
3. 在 OpenShift (RHOSO)上配置 Red Hat OpenStack Services，以使用 Red Hat Ceph Storage 集群。

## 1.1. DATA PLANE 节点服务列表

创建 **OpenStackDataPlaneNodeSet** CR 来配置 data plane 节点。在创建 **OpenStackDataPlaneDeployment** CR 时，**openstack-operator** 会协调 **OpenStackDataPlaneNodeSet** CR。

这些 CR 有一个服务列表，类似以下示例：

```
apiVersion: dataplane.openstack.org/v1beta1
kind: OpenStackDataPlaneNodeSet
spec:
  ...
  services:
    - configure-network
    - validate-network
    - install-os
    - configure-os
    - run-os
    - ovn
    - libvirt
    - nova
```

仅配置 services 列表中的服务。

在配置存储网络和 NTP 后，在配置 Compute 服务前，必须将 Red Hat Ceph **Storage** 部署到 data plane 节点上。这意味着您必须编辑服务列表并对 CR 进行其他更改。在本节中，您可以编辑服务列表以完成 HCI 环境的配置。

## 1.2. 配置数据平面节点网络

您必须配置 data plane 节点网络，以适应 Red Hat Ceph Storage 网络要求。

### 先决条件

- control plane 部署已完成，但尚未修改以使用 Ceph Storage。
- data plane 节点已置备有操作系统。
- data plane 节点可以通过 Ansible 可以使用的 SSH 密钥访问。
- data plane 节点具有可供用作 Ceph OSD 的磁盘。



- 至少三个可用的数据平面节点。Ceph Storage 集群必须至少有三个节点以确保冗余。

## 流程

1. 创建一个 **OpenStackDataPlaneNodeSet** CRD 文件来代表 data plane 节点。



### 注意

暂时不要在 Red Hat OpenShift 中创建 CR。

2. 在 **configure-os** 服务前将 **ceph-hci-pre** 服务添加到列表中，并在 **run-os** 后删除所有其他服务列表。

以下是编辑列表的示例：

```
apiVersion: dataplane.openstack.org/v1beta1
kind: OpenStackDataPlaneNodeSet
spec:
  ...
  services:
    - download-cache
    - bootstrap
    - configure-network
    - validate-network
    - install-os
    - ceph-hci-pre
    - configure-os
    - ssh-known-hosts
    - run-os
    - reboot-os
```



### 注意

请注意，您要从列表中删除的服务。您稍后会将它们重新添加到列表中。

3. (可选) **ceph-hci-pre** 服务使用 **edpm\_ceph\_hci\_pre edpm-ansible** 角色在网络配置后准备 EDPM 节点来托管 Red Hat Ceph Storage 服务。默认情况下，此角色的 **edpm\_ceph\_hci\_pre\_enabled\_services** 参数仅包含 RBD、RGW 和 NFS 服务。如果其他服务（如 Dashboard）部署有 HCI 节点；它们必须添加到 **edpm\_ceph\_hci\_pre\_enabled\_services** 参数列表中。有关此角色的更多信息，请参阅 [edpm\\_ceph\\_hci\\_pre](#) 角色。

4. 配置 Red Hat Ceph Storage **cluster\_network**，以用于 OSD 之间的存储管理流量。修改 CR 以设置 **edpm-ansible** 变量，以便 **edpm\_network\_config** 角色配置一个存储管理网络，Ceph 用作 **cluster\_network**。

以下示例有 3 个节点。它假定存储管理网络范围为 **172.20.0.0/24**，并且它位于 **VLAN23** 上。为 **cluster\_network** 添加了粗体行：

```
apiVersion: dataplane.openstack.org/v1beta1
kind: OpenStackDataPlaneNodeSet
metadata:
  name: openstack-edpm
  namespace: openstack
spec:
  env:
```

```

- name: ANSIBLE_FORCE_COLOR
  value: "True"
networkAttachments:
- ctlplane
nodeTemplate:
  ansible:
    ansiblePort: 22
    ansibleUser: cloud-admin
    ansibleVars:
      edpm_ceph_hci_pre_enabled_services:
        - ceph_mon
        - ceph_mgr
        - ceph_osd
        - ceph_rgw
        - ceph_nfs
        - ceph_rgw_frontend
        - ceph_nfs_frontend
      edpm_fips_mode: check
      edpm_iscsid_image: {{ registry_url }}/openstack-iscsid:{{ image_tag }}
      edpm_logrotate_cron_image: {{ registry_url }}/openstack-cron:{{ image_tag }}
      edpm_network_config_hide_sensitive_logs: false
      edpm_network_config_os_net_config_mappings:
        edpm-compute-0:
          nic1: 52:54:00:1e:af:6b
          nic2: 52:54:00:d9:cb:f4
        edpm-compute-1:
          nic1: 52:54:00:f2:bc:af
          nic2: 52:54:00:f1:c7:dd
        edpm-compute-2:
          nic1: 52:54:00:dd:33:14
          nic2: 52:54:00:50:fb:c3
      edpm_network_config_template: |
        ---
        {% set mtu_list = [ctlplane_mtu] %}
        {% for network in nodeset_networks %}
        {{ mtu_list.append(lookup(vars, networks_lower[network] ~ _mtu)) }}
        {%- endfor %}
        {% set min_viable_mtu = mtu_list | max %}
        network_config:
        - type: ovs_bridge
          name: {{ neutron_physical_bridge_name }}
          mtu: {{ min_viable_mtu }}
          use_dhcp: false
          dns_servers: {{ ctlplane_dns_nameservers }}
          domain: {{ dns_search_domains }}
          addresses:
            - ip_netmask: {{ ctlplane_ip }}/{{ ctlplane_cidr }}
          routes: {{ ctlplane_host_routes }}
          members:
            - type: interface
              name: nic2
              mtu: {{ min_viable_mtu }}
              # force the MAC address of the bridge to this interface
              primary: true
        {% for network in nodeset_networks %}
        - type: vlan

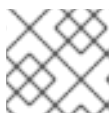
```

```

mtu: {{ lookup(vars, networks_lower[network] ~ _mtu) }}
vlan_id: {{ lookup(vars, networks_lower[network] ~ _vlan_id) }}
addresses:
- ip_netmask:
    {{ lookup(vars, networks_lower[network] ~ _ip) }}/{{ lookup(vars,
networks_lower[network] ~ _cidr) }}
    routes: {{ lookup(vars, networks_lower[network] ~ _host_routes) }}
    {% endfor %}
edpm_neutron_metadata_agent_image: {{ registry_url }}/openstack-neutron-metadata-
agent-ovn:{{ image_tag }}
edpm_nodes_validation_validate_controllers_icmp: false
edpm_nodes_validation_validate_gateway_icmp: false
edpm_selinux_mode: enforcing
edpm_sshd_allowed_ranges:
- 192.168.122.0/24
- 192.168.111.0/24
edpm_sshd_configure_firewall: true
enable_debug: false
gather_facts: false
image_tag: current-podified
neutron_physical_bridge_name: br-ex
neutron_public_interface_name: eth0
service_net_map:
  nova_api_network: internalapi
  nova_libvirt_network: internalapi
storage_mgmt_cidr: "24"
storage_mgmt_host_routes: []
storage_mgmt_mtu: 9000
storage_mgmt_vlan_id: 23
storage_mtu: 9000
timesync_ntp_servers:
- hostname: pool.ntp.org
ansibleSSHPrivateKeySecret: dataplane-ansible-ssh-private-key-secret
managementNetwork: ctlplane
networks:
- defaultRoute: true
  name: ctlplane
  subnetName: subnet1
- name: internalapi
  subnetName: subnet1
- name: storage
  subnetName: subnet1
- name: tenant
  subnetName: subnet1
nodes:
edpm-compute-0:
  ansible:
    host: 192.168.122.100
    hostName: compute-0
  networks:
    - defaultRoute: true
      fixedIP: 192.168.122.100
      name: ctlplane
      subnetName: subnet1
    - name: internalapi
      subnetName: subnet1

```

```
- name: storage
  subnetName: subnet1
- name: storagemgmt
  subnetName: subnet1
- name: tenant
  subnetName: subnet1
edpm-compute-1:
ansible:
  host: 192.168.122.101
  hostName: compute-1
networks:
- defaultRoute: true
  fixedIP: 192.168.122.101
  name: ctlplane
  subnetName: subnet1
- name: internalapi
  subnetName: subnet1
- name: storage
  subnetName: subnet1
- name: storagemgmt
  subnetName: subnet1
- name: tenant
  subnetName: subnet1
edpm-compute-2:
ansible:
  host: 192.168.122.102
  hostName: compute-2
networks:
- defaultRoute: true
  fixedIP: 192.168.122.102
  name: ctlplane
  subnetName: subnet1
- name: internalapi
  subnetName: subnet1
- name: storage
  subnetName: subnet1
- name: storagemgmt
  subnetName: subnet1
- name: tenant
  subnetName: subnet1
preProvisioned: true
services:
- bootstrap
- configure-network
- validate-network
- install-os
- ceph-hci-pre
- configure-os
- ssh-known-hosts
- run-os
- reboot-os
```



### 注意

不需要将存储管理网络添加到 networkAttachments 键。

## 5. 应用 CR :

```
$ oc apply -f <dataplane_cr_file>
```

- 将 `<dataplane_cr_file >` 替换为您的文件的名称。

**注意**

Ansible 在创建 **OpenStackDataPlaneDeployment** CRD 之前，不会配置或验证网络。

6. 创建 **OpenStackDataPlaneDeployment** CRD，如在 OpenShift *上部署 Red Hat OpenStack Services on OpenShift* 指南中的创建 [data plane](#) 所述，它定义了 **OpenStackDataPlaneNodeSet** CRD 文件，以便 Ansible 配置 data plane 节点上的服务。

## 7. 要确认网络配置，请完成以下步骤：

- a. SSH 到数据平面节点。
- b. 使用 `ip a` 命令显示配置网络。
- c. 确认存储网络位于配置网络列表中。

### 1.2.1. Red Hat Ceph Storage MTU 设置

此流程中的示例将 storage 和 storage\_mgmt 网络的 MTU 从 1500 改为 9000。MTU 9000 称为巨型帧。尽管不强制提高 MTU，但使用巨型帧来提高存储性能。如果使用巨型帧，则必须将数据路径中的所有网络交换机端口配置为支持巨型帧。还必须为使用 OpenShift 上运行的存储网络的服务进行 MTU 更改。

要更改连接到 data plane 节点的 OpenShift 服务的 MTU，请更新基本接口和 VLAN 接口的节点网络配置策略(NNCP)。如果主 NAD 接口已具有所需的 MTU，则不需要更新网络附加定义(NAD)。如果底层接口的 MTU 设为 9000，但没有为其顶部的 VLAN 接口指定它，那么它将默认为底层接口中的值。

如果 MTU 值不一致，则问题可能会导致 Red Hat Ceph Storage 集群无法访问仲裁，或者不支持使用 CephX 协议进行身份验证。如果 MTU 已更改，并且您观察到这些类型的问题，请验证使用巨型帧的网络的所有主机都可以与 ping 命令进行通信，例如：

```
$ ping -M do -s 8972 172.20.0.100
```

## 1.3. 在 DATA PLANE 节点上配置和部署 RED HAT CEPH STORAGE

使用 **cephadm** 实用程序为 HCI 环境配置和部署 Red Hat Ceph Storage。

### 1.3.1. cephadm 实用程序

使用 **cephadm** 实用程序在 data plane 节点上配置和部署 Red Hat Ceph Storage。在继续操作前，必须将 **cephadm** 软件包部署到至少一个数据平面节点上。**edpm-ansible** 不会部署 Red Hat Ceph Storage。

有关部署 Red Hat Ceph Storage 的更多信息和流程，请参阅 [Red Hat Ceph Storage 安装指南中的 Red Hat Ceph Storage 安装](#)。

### 1.3.2. 配置和部署 Red Hat Ceph Storage

通过编辑配置文件和使用 **cephadm** 实用程序来配置和部署 Red Hat Ceph Storage。

## 流程

1. 编辑 Red Hat Ceph Storage 配置文件。
2. 添加 **Storage** 和 **Storage Management** 网络范围。Red Hat Ceph Storage 使用 **Storage** 网络作为 Red Hat Ceph Storage **public\_network**，**Storage Management** 网络作为 **cluster\_network**。  
以下示例是 **Storage** 网络范围为 **172.18.0.0/24** 的配置文件条目，**Storage Management** 网络范围为 **172.20.0.0/24**：

```
[global]
public_network = 172.18.0.0/24
cluster_network = 172.20.0.0/24
```

3. 在 Compute 服务和 Ceph OSD 服务之间添加并置界限。应在并置计算服务和 Ceph OSD 服务之间设置界限，以减少 CPU 和内存争用。  
以下是设置了这些边界的 Ceph 配置文件条目的示例：

```
[osd]
osd_memory_target_autotune = true
osd_numa_auto_affinity = true
[mgr]
mgr/cephadm/autotune_memory_target_ratio = 0.2
```

在本例中，**osd\_memory\_target\_autotune** 参数设置为 **true**，以便 OSD 守护进程根据 **osd\_memory\_target** 选项调整内存消耗。**autotune\_memory\_target\_ratio** 默认为 0.7。这意味着系统总 RAM 的 70% 是非自动调优 Ceph 守护进程所消耗的内存的起点。剩余的内存存在 OSD 之间划分；假设所有 OSD 的 **osd\_memory\_target\_autotune** 设置为 **true**。对于 HCI 部署，您可以将 **mgr/cephadm/autotune\_memory\_target\_ratio** 设置为 0.2，以便计算服务有更多内存可用。

有关服务并置的更多信息，请参阅 [在 NUMA 节点的 HCI 环境中并置服务](#)。



### 注意

如果需要在部署后调整这些值，请使用 **ceph config set osd <key> <value>** 命令。

4. 使用 data plane 节点上编辑的配置文件部署 Ceph Storage：  
**\$ cephadm bootstrap --config <config\_file> --mon-ip <data\_plane\_node\_ip> --skip-monitoring-stack**
  - 将 **<config\_file>** 替换为 Ceph 配置文件的名称。
  - 将 **<data\_plane\_node\_ip & gt;** 替换为安装 Red Hat Ceph Storage 的 data plane 节点的 Storage 网络 IP 地址。



## 注意

**cephadm bootstrap** 命令中使用 **--skip-monitoring-stack** 选项，以跳过监控服务的部署。如果之前已作为上述其他流程的一部分部署了监控服务，则确保 Red Hat Ceph Storage 部署成功完成。

如果还没有部署监控服务，请参阅 [Red Hat Ceph Storage](#) 文档来了解启用监控服务的信息和流程。

5. 在第一个 EDPM 节点上启动 Red Hat Ceph Storage 集群后，请参阅 [Red Hat Ceph Storage 安装指南中的 Red Hat Ceph Storage 安装](#)，将其他 EDPM 节点添加到 Ceph 集群。

### 1.3.2.1. 在 HCI 环境中为 NUMA 节点并置服务

双 NUMA 节点系统可以在一个 NUMA 节点上托管对延迟敏感的 Compute 服务工作负载，另一个 NUMA 节点上的 Ceph OSD 工作负载。要将 Ceph OSD 配置为使用未由计算服务使用的特定 NUMA 节点，请使用以下 Ceph OSD 配置之一：

- **osd\_numa\_node** 将关联性设置为 NUMA 节点(none 为 -1)。
- **osd\_numa\_auto\_affinity** 会自动将关联性设置为存储和网络匹配的 NUMA 节点。

如果 NUMA 节点和磁盘控制器都位于 NUMA 节点 0 上，请执行以下操作：

1. 将 NUMA 节点 0 上的网络接口用于存储网络
2. 在 NUMA 节点 0 上托管 Ceph OSD 工作负载。
3. 在 NUMA 节点 1 上托管 Compute 服务工作负载，并使用 NUMA 节点 1 上的网络接口。

将 **osd\_numa\_auto\_affinity** 设置为 **true**，如初始 Ceph 配置文件中所示。或者，将 **osd\_numa\_node** 直接设置为 0，然后清除 **osd\_numa\_auto\_affinity** 参数，使其默认为 **false**。

当 OSD 离线时，超融合集群回填可能会减慢回填过程。在交换速度较慢的恢复时，回填活动对并置计算服务(nova)工作负载的影响较低。Red Hat Ceph Storage 有以下默认值来控制回填活动的速度。

- **osd\_recovery\_op\_priority = 3**
- **osd\_max\_backfills = 1**
- **osd\_recovery\_max\_active\_hdd = 3**
- **osd\_recovery\_max\_active\_ssd = 10**

### 1.3.3. 确认 Red Hat Ceph Storage 部署

在继续操作前，确认 Red Hat Ceph Storage 已部署。

#### 流程

1. 使用 SSH 连接到数据平面节点。
2. 查看 Red Hat Ceph Storage 集群的状态：

```
$ cephadm shell -- ceph -s
```

### 1.3.4. 确认 Red Hat Ceph Storage 调整

在继续操作前，请确保 Red Hat Ceph Storage 已正确调优。

#### 流程

1. 使用 SSH 连接到数据平面节点。
2. 使用以下命令验证整个 Red Hat Ceph Storage 调优：

```
$ ceph config dump | grep numa
$ ceph config dump | grep autotune
$ ceph config dump | get mgr
```

3. 使用以下命令验证 OSD 的调优：

```
$ ceph config get <osd_number> osd_memory_target
$ ceph config get <osd_number> osd_memory_target_autotune
$ ceph config get <osd_number> osd_numa_auto_affinity
```

- 将 **<osd\_number>** 替换为 OSD 的数量。例如，若要引用 OSD 11，可使用 **osd.11**。

4. 使用以下命令验证 OSD 的默认回填值：

```
$ ceph config get <osd_number> osd_recovery_op_priority
$ ceph config get <osd_number> osd_max_backfills
$ ceph config get <osd_number> osd_recovery_max_active_hdd
$ ceph config get <osd_number> osd_recovery_max_active_ssd
```

- 将 **<osd\_number>** 替换为 OSD 的数量。例如，若要引用 OSD 11，可使用 **osd.11**。

### 1.4. 配置数据平面以使用并置 RED HAT CEPH STORAGE 服务器

虽然 Red Hat Ceph Storage 集群与 data plane 节点上的 Compute 服务物理并置，但它被视为逻辑分开。在 data plane 节点可以使用前，必须将 Red Hat Ceph Storage 配置为存储解决方案。

#### 先决条件

- 完成 [集成 Red Hat Ceph Storage](#) 中的步骤。

#### 流程

1. 编辑 **OpenStackDataPlaneNodeSet** CR。
2. 要为 Compute 服务(nova)定义 **cephx** 密钥和配置文件，请使用 **extraMounts** 参数。以下是为此目的使用 **extraMounts** 参数的示例：

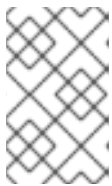
```
apiVersion: dataplane.openstack.org/v1beta1
kind: OpenStackDataPlaneNodeSet
spec:
  ...
  nodeTemplate:
    extraMounts:
      - extraVolType: Ceph
```



```
volumes:
- name: ceph
  secret:
    secretName: ceph-conf-files
mounts:
- name: ceph
  mountPath: "/etc/ceph"
  readOnly: true
```

3. 在 CR 中找到 **服务列表**。
4. 编辑 **服务列表**，以恢复 [配置 data plane 节点网络](#) 中描述的所有服务。恢复完整 **服务列表** 可让剩余的作业完成 HCI 环境的配置。  
 以下是一个完整 **服务列表** 的示例，其额外服务以粗体显示：

```
apiVersion: dataplane.openstack.org/v1beta1
kind: OpenStackDataPlaneNodeSet
spec:
  ...
  services:
    - bootstrap
    - configure-network
    - validate-network
    - install-os
    - ceph-hci-pre
    - configure-os
    - ssh-known-hosts
    - run-os
    - reboot-os
    - install-certs
    - ceph-client
    - ovn
    - neutron-metadata
    - libvirt
    - nova-custom-ceph
```

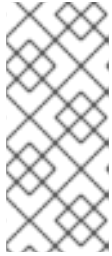


### 注意

除了恢复默认服务列表外，**ceph-client** 服务也会在 **run-os** 服务后添加。**ceph-client** 服务将 EDPM 节点配置为 Red Hat Ceph Storage 服务器的客户端。此服务分发客户端连接到 Red Hat Ceph Storage 服务器所需的文件。

5. 创建一个 **ConfigMap**，将 **reserved\_host\_memory\_mb** 参数设置为适合您配置的值。  
 以下是用于此目的的 ConfigMap 示例：

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: reserved-memory-nova
data:
  04-reserved-memory-nova.conf: |
    [DEFAULT]
    reserved_host_memory_mb=75000
```



### 注意

可以设置 **reserved\_host\_memory\_mb** 参数的值，以便计算服务调度程序不会将内存提供给同一服务器上的 Ceph OSD 所需的虚拟机。除了虚拟机监控程序的默认保留内存外，示例还为每个主机保留 5 GB 的 OSD 为 10 个 OSD。在 IOPS 优化的集群中，您可以通过为每个 OSD 保留更多内存来提高性能。5 GB 数字作为起点提供，必要时可以进一步调整。

- 通过编辑 **OpenStackDataPlaneService/nova-custom-ceph** 文件，将 **reserved-memory-nova** 添加到 **configMaps** 列表中：

```
kind: OpenStackDataPlaneService
<...>
spec:
  configMaps:
  - ceph-nova
  - reserved-memory-nova
```

- 应用 CR 更改。

```
$ oc apply -f <dataplane_cr_file>
```

- 将 **<dataplane\_cr\_file >** 替换为您的文件的名称。



### 注意

Ansible 在创建 **OpenStackDataPlaneDeployment** CRD 之前，不会配置或验证网络。

- 创建 **OpenStackDataPlaneDeployment** CRD，如在 OpenShift 上部署 *Red Hat OpenStack Services on OpenShift* 指南中的创建 [data plane](#) 所述，它定义了 **OpenStackDataPlaneNodeSet** CRD 文件，以便 Ansible 配置 data plane 节点上的服务。