Red Hat OpenStack Platform 16.1

超融合基础架构指南

在 Red Hat OpenStack Platform overcloud 中了解并配置超融合基础架构

Last Updated: 2023-11-07

Red Hat OpenStack Platform 16.1 超融合基础架构指南

在 Red Hat OpenStack Platform overcloud 中了解并配置超融合基础架构

OpenStack Team
rhos-docs@redhat.com

法律通告

摘要

本文档描述了 Red Hat OpenStack Platform 的超线程，它在同一主机上并置 Compute 和 Ceph Storage 服务。

目录

使开源包含更多

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

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。与我们分享您的成功秘诀。

使用直接文档反馈(DDF)功能

使用 **添加反馈** DDF 功能，用于特定句子、段落或代码块上的直接注释。

1. 以 *Multi-page HTML* 格式查看文档。
2. 请确定您看到文档右上角的 **反馈** 按钮。
3. 用鼠标指针高亮显示您想评论的文本部分。
4. 点 **添加反馈**。
5. 在**添加反馈**项中输入您的意见。
6. 可选：添加您的电子邮件地址，以便文档团队可以联系您以讨论您的问题。
7. 点 **Submit**。

第 1 章 配置和部署 RED HAT OPENSTACK PLATFORM 超融合基础架构

Red Hat OpenStack Platform (RHOSP)超融合基础架构(HCI)由超融合节点组成。服务在这些超融合节点上共存，以优化资源使用量。在 RHOSP HCI 中，计算和存储服务在超融合节点上在一起。您只能使用超融合节点部署 overcloud，或使用常规的 Compute 和 Ceph Storage 节点混合超融合节点。

注意

您必须使用 Red Hat Ceph Storage 作为存储供应商。

提示

- 使用 ceph-ansible 3.2 及更高版本自动调整 Ceph 内存设置。
- 使用 BlueStore 作为 HCI 部署的后端，以利用 BlueStore 内存处理功能。

要在 overcloud 中创建和部署 HCI，集成 overcloud 中的其他功能，如网络功能虚拟化，并确保超融合节点上计算和 Red Hat Ceph Storage 服务的最佳性能，您必须完成以下操作：

1. 为超融合节点准备预定义的自定义 overcloud 角色 **ComputeHCI**。
2. 配置资源隔离。
3. 验证可用的 Red Hat Ceph Storage 软件包。
4. 部署 HCI overcloud。

有关 HCI 配置指导，请参阅 配置指南。

1.1. 先决条件

- 您已部署了 undercloud。有关如何部署 undercloud 的说明，请参阅 Director 安装和使用。
- 您的环境可以置备满足 RHOSP Compute 和 Red Hat Ceph Storage 要求的节点。有关更多信息，请参阅 基本 Overcloud 部署。
- 您已在环境中注册了所有节点。如需更多信息，请参阅 注册节点。
- 您已在环境中标记了所有节点。如需更多信息，请参阅 手动标记节点。
- 您已清理了计划用于计算和 Ceph OSD 服务的节点上的磁盘。如需更多信息，请参阅 清理 Ceph Storage 节点磁盘。
- 您已准备好了 overcloud 节点，以便注册 Red Hat Content Delivery Network 或 Red Hat Satellite 服务器。有关更多信息，请参阅 基于 Ansible 的 Overcloud 注册 。

1.2. 为超融合节点准备 OVERCLOUD 角色

要将节点指定为超融合角色，您需要定义一个超融合角色。Red Hat OpenStack Platform (RHOSP)为超融合节点提供预定义角色 **ComputeHCI**。此角色并置计算和 Ceph 对象存储守护进程(OSD)服务，允许您将它们一起部署到同一超融合节点上。

流程

1. 以 **stack** 用户的身份登录 undercloud。
2. Source **stackrc** 文件：

```
[stack@director ~]$ source ~/stackrc
```

3. 生成包含 **ComputeHCI** 角色的新自定义角色数据文件，以及您要用于 overcloud 的其他角色。以下示例生成角色数据文件 **roles_data_hci.yaml**，其中包括角色 **Controller**, **ComputeHCI**, **Compute**, 和 **CephStorage**：

```
(undercloud)$ openstack overcloud roles \
 generate -o /home/stack/templates/roles_data_hci.yaml \
  Controller ComputeHCI Compute CephStorage
```

注意

在生成的自定义角色数据文件中为 **ComputeHCI** 角色列出的网络包括 Compute 和 Storage 服务所需的网络，例如：

```
- name: ComputeHCI
  description: |
    Compute node role hosting Ceph OSD
  tags:
    - compute
  networks:
    InternalApi:
      subnet: internal_api_subnet
    Tenant:
      subnet: tenant_subnet
    Storage:
      subnet: storage_subnet
    StorageMgmt:
      subnet: storage_mgmt_subnet
```

4. 创建 **network_data.yaml** 文件的本地副本，将可组合网络添加到 overcloud 中。**network_data.yaml** 文件与默认网络环境文件 **/usr/share/openstack-tripleo-heat-templates/environments/*** 交互，将您为 **ComputeHCI** 角色定义的网络与超融合节点关联。有关更多信息，*请参阅高级 Overcloud 自定义指南中的* 添加可组合网络。
5. 要提高 Red Hat Ceph Storage 的性能，请将 **Storage** 和 **StorageMgmt** 网络的 MTU 设置更新为 **9000**（用于巨型帧），位于 **network_data.yaml** 的本地副本中。如需更多信息，请参阅 Director 中配置 MTU 设置，并配置巨型帧。
6. 为超融合节点创建 **计算HCI** overcloud 类别：

```
(undercloud)$ openstack flavor create --id auto \
 --ram <ram_size_mb> --disk <disk_size_gb> \
 --vcpus <no_vcpus> computeHCI
```

- 将 **<ram_size_mb>** 替换为裸机节点的 RAM，以 MB 为单位。
- 将 **<disk_size_gb>** 替换为裸机节点中的磁盘大小（以 GB 为单位）。

- 将 **<no_vcpus>** 替换为裸机节点中的 CPU 数量。

注意

这些属性不可用于调度实例。但是，计算调度程序使用磁盘大小来确定根分区大小。

7. 检索节点列表来识别它们的 UUID：

```
(undercloud)$ openstack baremetal node list
```

8. 使用自定义 HCI 资源对象标记您要指定为超融合的每个裸机节点：

```
(undercloud)$ openstack baremetal node set \
 --resource-class baremetal.HCI <node>
```

将 **<node>** 替换为裸机节点的 ID。

9. 将 **computeHCI** 类别与自定义 HCI 资源类型关联：

```
(undercloud)$ openstack flavor set \
--property resources:CUSTOM_BAREMETAL_HCI=1 \
computeHCI
```

要确定与 Bare Metal 服务节点的资源类对应的自定义资源类的名称，请将资源类转换为大写，用下划线替换所有 punctuation，并使用 **CUSTOM_** 前缀。

注意

类别只能请求一个裸机资源类实例。

10. 设置以下类别属性，以防止计算调度程序使用裸机类别属性来调度实例：

```
(undercloud)$ openstack flavor set \
 --property resources:VCPU=0 \
 --property resources:MEMORY_MB=0 \
 --property resources:DISK_GB=0 computeHCI
```

11. 在 **node-info.yaml** 文件中添加以下参数，以指定超融合和 Controller 节点的数量，以及用于超融合和控制器指定的节点的类别：

```
parameter_defaults:
  OvercloudComputeHCIFlavor: computeHCI
  ComputeHCICount: 3
  OvercloudControlFlavor: baremetal
  ControllerCount: 3
```

其他资源

- 可组合服务和自定义角色
- 检查 roles_data 文件

- 为角色分配节点和类别

1.2.1. 为多磁盘集群定义根磁盘

大多数 Ceph Storage 节点会使用多个磁盘。当节点使用多个磁盘时，director 必须识别根磁盘。默认情况下，director 在置备过程中将 overcloud 镜像写入根磁盘

使用此流程按序列号识别根设备。有关可以用来识别根磁盘的其他属性的更多信息，请参阅 第 1.2.1.1 节 “标识根磁盘的属性”。

流程

1. 从每个节点的硬件内省验证磁盘信息。以下命令显示节点的磁盘信息：

```
(undercloud)$ openstack baremetal introspection data save 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 | jq ".inventory.disks"
```

例如，一个节点的数据可能会显示 3 个磁盘：

```
[
 {
  "size": 299439751168,
  "rotational": true,
  "vendor": "DELL",
  "name": "/dev/sda",
  "wwn_vendor_extension": "0x1ea4dcc412a9632b",
  "wwn_with_extension": "0x61866da04f3807001ea4dcc412a9632b",
  "model": "PERC H330 Mini",
  "wwn": "0x61866da04f380700",
  "serial": "61866da04f3807001ea4dcc412a9632b"
 }
 {
  "size": 299439751168,
  "rotational": true,
  "vendor": "DELL",
  "name": "/dev/sdb",
  "wwn_vendor_extension": "0x1ea4e13c12e36ad6",
  "wwn_with_extension": "0x61866da04f380d001ea4e13c12e36ad6",
  "model": "PERC H330 Mini",
  "wwn": "0x61866da04f380d00",
  "serial": "61866da04f380d001ea4e13c12e36ad6"
 }
 {
  "size": 299439751168,
  "rotational": true,
  "vendor": "DELL",
  "name": "/dev/sdc",
  "wwn_vendor_extension": "0x1ea4e31e121cfb45",
  "wwn_with_extension": "0x61866da04f37fc001ea4e31e121cfb45",
  "model": "PERC H330 Mini",
  "wwn": "0x61866da04f37fc00",
  "serial": "61866da04f37fc001ea4e31e121cfb45"
 }
]
```

2. 在 undercloud 上，为节点设置根磁盘。包括用于定义根磁盘的最合适的硬件属性值。

```
(undercloud)$ openstack baremetal node set --property root_device='{"serial":"
<serial_number>"}' <node-uuid>
```

例如：要将根设备设定为磁盘 2，其序列号为 **61866da04f380d001ea4e13c12e36ad6**，输入以下命令：

```
(undercloud)$ openstack baremetal node set --property root_device='{"serial":
"61866da04f380d001ea4e13c12e36ad6"}' 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0
```

注意

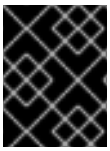
将每个节点的 BIOS 配置为从您选择的根磁盘引导。将引导顺序配置为首先从网络引导，然后从根磁盘引导。

director 识别特定磁盘以用作根磁盘。运行 **openstack overcloud deploy** 命令时，director 置备 overcloud 镜像并将其写入根磁盘。

1.2.1.1. 标识根磁盘的属性

您可以定义多个属性以帮助 director 识别根磁盘：

- **model**（字符串）：设备识别码。
- **vendor**（字符串）：设备厂商。
- **serial**（字符串）：磁盘序列号。
- **hctl**（字符串）：SCSI 的 Host:Channel:Target:Lun。
- **size**（整数）：设备的大小（以 GB 为单位）。
- **wwn**（字符串）：唯一的存储 ID。
- **wwn_with_extension**（字符串）：唯一存储 ID 附加厂商扩展名。
- **wwn_vendor_extension**（字符串）：唯一厂商存储标识符。
- **rotational**（布尔值）：旋转磁盘设备为 true (HDD)，否则为 false (SSD)。
- **name**（字符串）：设备名称，例如：/dev/sdb1。



重要

仅对具有持久名称的设备使用 **name** 属性。不要使用 **name** 来设置任何其他设备的根磁盘，因为此值在节点引导时可能会改变。

1.3. 在超融合节点上配置资源隔离

在超融合节点上查找 Ceph OSD 和计算服务的计算服务会给 Red Hat Ceph Storage 和 Compute 服务之间的资源争用风险，因为不知道同一主机上彼此的存在性。资源争用可能会导致服务降级，这会降低超聚合的益处。

您必须配置 Ceph 和计算服务的资源隔离，以防止争用。

流程

1. 可选：通过在 Compute 环境文件中添加以下参数来覆盖自动生成的 Compute 设置：

```
parameter_defaults:
  ComputeHCIParameters:
    NovaReservedHostMemory: <ram>
    NovaCPUAllocationRatio: <ratio>
```

- 将 **<ram** > 替换为为超融合节点上的 Ceph OSD 服务和实例开销（以 MB 为单位）保留的 RAM 量。
- 将 **<**ratio> 替换为计算调度程序在选择在其上部署实例的计算节点时应使用的比率。如需有关自动生成的计算设置的更多信息，请参阅 自动生成的计算设置的过程，以获取为计算服务保留的 CPU 和内存资源。

2. 要为 Red Hat Ceph Storage 保留内存资源，请在 **/home/stack/templates/storage-container-config.yaml** 中将参数 **is_hci** 设置为 **true** ：

```
parameter_defaults:
  CephAnsibleExtraConfig:
    is_hci: true
```

这允许 **ceph-ansible** 为 Red Hat Ceph Storage 保留内存资源，并通过为 HCI 部署自动调整 **osd_memory_target** 参数设置来减少 Ceph OSD 的内存增长。

警告

红帽不推荐直接覆盖 **ceph_osd_docker_memory_limit** 参数。

注意

从 ceph-ansible 3.2 开始，无论使用了 FileStore 还是 BlueStore 后端，**ceph_osd_docker_memory_limit** 会自动被设置为主机的最大内存。

3. 可选： 默认情况下，**ceph-ansible** 为每个 Ceph OSD 预留一个 vCPU。如果每个 Ceph OSD 需要多个 CPU，请将以下配置添加到 **/home/stack/templates/storage-container-config.yaml** 中：

```
parameter_defaults:
  CephAnsibleExtraConfig:
    ceph_osd_docker_cpu_limit: <cpu_limit>
```

将 **<cpu_limit** > 替换为为每个 Ceph OSD 保留的 CPU 数量。

有关如何根据您的硬件和工作负载调整 CPU 资源的更多信息，请参阅 Red Hat Ceph Storage 硬件选择指南。

4. 可选：通过在 Ceph 环境中添加以下参数来移除 Ceph OSD 时，红帽 Ceph 存储回填和恢复操作的优先级如下：

```
parameter_defaults:
  CephConfigOverrides:
    osd_recovery_op_priority: <priority_value>
    osd_recovery_max_active: <no_active_recovery_requests>
    osd_max_backfills: <max_no_backfills>
```

- 将 **<priority_value>** 替换为与 OSD 客户端 OP 优先级相关的恢复操作的优先级。
- 将 **<no_active_recovery_requests** > 替换为每个 OSD 的活动恢复请求数。
- 将 **<max_no_backfills** > 替换为允许或从一个 OSD 中允许的最大回填数量。
  有关默认 Red Hat Ceph Storage 回填和恢复选项的更多信息，请参阅 Red Hat Ceph Storage回填和恢复操作。

1.3.1. 为 Compute 服务保留的自动生成 CPU 和内存资源的流程

director 提供了一个默认的计划环境文件，用于在部署期间在超融合节点上配置资源限制。此计划环境文件指示 OpenStack 工作流完成以下进程：

1. 检索在检查硬件节点期间所收集的硬件内省数据。
2. 根据这些数据，计算在超融合节点上为计算的最佳 CPU 和内存分配工作负载。
3. 自动生成配置这些约束所需的参数，并为 Compute 保留 CPU 和内存资源。这些参数在 **plan-environment-derived-params.yaml** 文件的 **hci_profile_config** 部分中定义。

注意

每个工作负载配置集中的 **average_guest_memory_size_in_mb** he **average_guest_cpu_utilization_percentage** 参数用于计算 Compute 的 **reserved_host_memory** 和 **cpu_allocation_ratio** 设置的值。

您可以通过在 Compute 环境文件中添加以下参数来覆盖自动生成的 Compute 设置：

自动生成的 nova.conf 参数	计算环境文件覆盖	描述
reserved_host_memory	parameter_defaults: ComputeHCIParameters: NovaReservedHostMemory: 181000	设置应当为 Ceph OSD 服务和超融合节点上的每个客户端实例开销保留多少 RAM。
cpu_allocation_ratio	parameter_defaults: ComputeHCIParameters: NovaCPUAllocationRatio: 8.2	设置计算调度程序在选择在其上部署实例的计算节点时应使用的比率。

这些覆盖应用到所有使用 **ComputeHCI** 角色（即超融合节点）的节点。有关手动确定 **NovaReservedHostMemory** 和 **NovaCPUAllocationRatio** 的最佳值的更多信息，请参阅 OpenStack Workflow Compute CPU 和内存计算器。

提示

您可以使用以下脚本为您的超融合节点计算合适的基准 **NovaReservedHostMemory** 和 **NovaCPUAllocationRatio** 值。

nova_mem_cpu_calc.py

其他资源

- 创建裸机节点硬件的清单

1.3.2. Red Hat Ceph Storage 回填和恢复操作

移除 Ceph OSD 时，Red Hat Ceph Storage 使用回填和恢复操作来重新平衡集群。Red Hat Ceph Storage 这样做会根据放置组策略保留数据的多个副本。这些操作使用系统资源。如果 Red Hat Ceph Storage 集群负载较大，则其性能会断开，因为它会将资源分散到回填和恢复。

为了缓解这个性能在 OSD 移除过程中生效，您可以降低回填和恢复操作的优先级。这样做的代价是，数据副本更长的时间就越少，这样会使数据面临稍高的风险。

下表详述的参数用于配置回填和恢复操作的优先级。

参数	描述	默认值
osd_recovery_op_priority	设置恢复操作的优先级，相对于 OSD 客户端 OP 优先级。	3
osd_recovery_max_active	一次设置每个 OSD 的活动恢复请求数。更多请求可以加快恢复，但请求会增加集群上的负载。如果要降低延迟，则将其设置为 1。	3
osd_max_backfills	设置单个 OSD 允许的最大回填数量。	1

1.4. CEPH STORAGE 的部署前验证

为了帮助避免 overcloud 部署失败，请验证您的服务器上是否存在所需的软件包。

1.4.1. 验证 ceph-ansible 软件包版本

undercloud 包含基于 Ansible 的验证，您可以在部署 overcloud 前运行来识别潜在问题。这些验证可以帮助您避免 overcloud 部署失败，方法是在发生常见问题前识别它们。

流程

验证是否安装了 **ceph-ansible** 软件包的更正版本：

```
$ ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-ansible-installed.yaml
```

1.4.2. 为预置备节点验证软件包

Ceph 只能服务具有一组特定软件包的 overcloud 节点。使用预置备节点时，您可以验证这些软件包是否存在。

有关预置备节点的更多信息，请参阅 使用预置备节点配置基本的 overcloud。

流程

验证服务器是否包含所需的软件包：

```
ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-dependencies-installed.yaml
```

1.5. 部署 HCI OVERCLOUD

完成 HCI 配置后，您必须部署 overcloud。

重要

在部署 Red Hat OpenStack Platform (RHOSP) HCI 环境时，不要启用 Instance HA。如果要使用使用 Red Hat Ceph Storage 的超融合 RHOSP 部署的实例 HA，请联系您的红帽代表。

先决条件

- 您使用单独的基本环境文件或一组文件，用于所有其他 Red Hat Ceph Storage 设置，例如：**/home/stack/templates/storage-config.yaml**。如需更多信息，请参阅自定义存储服务 和 附录 A. Sample 环境文件：创建 Ceph Storage 集群。
- 您已定义了您要分配给基础环境文件中每个角色的节点数量。如需更多信息，请参阅将节点和类别分配给角色。
- 在 undercloud 安装过程中，您可以在 **undercloud.conf** 文件中设置 **generate_service_certificate=false**。否则，必须在部署 overcloud 时注入信任定位符，如 Overcloud 公共端点上启用 SSL/TLS 所述。

流程

- 使用其他环境文件，将新角色和环境文件添加到堆栈中，并部署您的 HCI overcloud：

```
(undercloud)$ openstack overcloud deploy --templates \
  -e [your environment files] \
  -r /home/stack/templates/roles_data_hci.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/network-environment.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/net-single-nic-with-vlans.yaml
  -e /home/stack/templates/storage-config.yaml \
  -e /home/stack/templates/storage-container-config.yaml \
  -n /home/stack/templates/network_data.yaml \
  [-e /home/stack/templates/ceph-backfill-recovery.yaml \ ]
  --ntp-server pool.ntp.org
```

在部署命令中包含 **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml** 来添加部署容器化 Red Hat Ceph 集群的基本环境文件，其中包含所有默认设置。有关更多信息，请参阅使用容器化 Red Hat Ceph 部署 Overcloud 。

注意

如果您的部署使用了单一根输入/输出虚拟化(SR-IOV)，请在部署命令中包括以下选项。

如果您在部署中使用 ML2/OVS 机制驱动程序，请指定以下选项：

```
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-sriov.yaml
-e /home/stack/templates/network-environment.yaml
```

如果您在部署中使用 ML2/OVN 机制驱动程序，请指定以下选项：

```
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovn-sriov.yaml
-e /home/stack/templates/network-environment.yaml
```

提示

您还可以使用 **回答文件** 来指定要在部署中包括的环境文件。如需更多信息，请参阅 *Director 安装和使用指南*中的将 overcloud 部署包含环境文件。

1.5.1. 限制 ceph-ansible 运行所在的节点

您可以通过限制 **ceph-ansible** 运行的节点来减少部署更新时间。当 Red Hat OpenStack Platform (RHOSP)使用 **config-download** 配置 Ceph 时，您可以使用 **--limit** 选项指定节点列表，而不是在整个部署中运行 **config-download** 和 **ceph-ansible**。这个功能很有用，例如，作为扩展 overcloud 的一部分，或者替换失败的磁盘。在这样的情形中，部署只能在您添加到环境中的新节点上运行。

在故障磁盘替换中使用 --limit 的示例

在以下示例中，Ceph 存储节点 **oc0-cephstorage-0** 有一个磁盘故障，因此它收到一个新的工厂清理磁盘。Ansible 需要在 **oc0-cephstorage-0** 节点上运行，以便新磁盘可以用作 OSD，但它不需要在所有其他 Ceph 存储节点上运行。将示例环境文件和节点名称替换为适合您的环境。

流程

1. 以 **stack** 用户身份登录 undercloud 节点，再提供 **stackrc** 凭据文件：

```
# source stackrc
```

2. 完成以下步骤之一，以便使用新磁盘来启动缺少的 OSD。

- 运行堆栈更新，并包含 **--limit** 选项以指定要运行 **ceph-ansible** 的节点：

```
$ openstack overcloud deploy --templates \
  -r /home/stack/roles_data.yaml \
  -n /usr/share/openstack-tripleo-heat-templates/network_data_dashboard.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml \
```

```
-e ~/my-ceph-settings.yaml \
-e <other-environment_files> \
--limit oc0-controller-0:oc0-controller-2:oc0-controller-1:oc0-cephstorage-0:undercloud
```

在本例中，包含 Controller，因为 Ceph mons 需要 Ansible 更改其 OSD 定义。

- 如果 **config-download** 会生成 **ansible-playbook-command.sh** 脚本，您可以使用 **--limit** 选项运行脚本，将指定节点传递给 **ceph-ansible** ：

```
./ansible-playbook-command.sh --limit oc0-controller-0:oc0-controller-2:oc0-controller-1:oc0-cephstorage-0:undercloud
```

警告

在使用 **--limit** 时，您必须始终将 undercloud 包含在限制列表中，否则无法执行 **ceph-ansible**。这是必要的，因为 **ceph-ansible** 执行通过 **external_deploy_steps_tasks** playbook 进行，该 playbook 仅在 undercloud 上运行。

1.6. OPENSTACK WORKFLOW COMPUTE CPU 和内存计算器

OpenStack Workflow 计算 CPU 和内存的最佳设置，并使用结果填充 **NovaReservedHostMemory** 和 **NovaCPUAllocationRatio** 参数。

NovaReservedHostMemory

NovaReservedHostMemory 参数设置主机节点要保留的内存量（以 MB 为单位）。要为超融合节点确定适当的值，假设每个 OSD 使用 3 GB 内存。给定具有 256 GB 内存和 10 个 OSD 的节点，您可以为 Ceph 分配 30 GB 内存，使 Compute 离开 226 GB。如果节点可以托管太多内存，例如，113 个实例各自使用 2 GB 内存。

但是，您仍然需要考虑每个实例对于 *管理程序* 的额外开销。假设这个开销为 0.5 GB，同一节点只能托管 90 个实例，它的帐户将 226 GB 划分到 2.5 GB。为主机节点保留的内存量（即 Compute 服务不应使用的内存）是：

（在 * Ov）+(Os * RA)

其中：

- **在** 中：实例数
- **OV** ：每个实例所需的开销内存量
- **OS**: 节点上的 OSD 数量
- **RA**:每个 OSD 应该拥有的 RAM 量

对于 90 个实例，这是我们(90*0.5)+(10*3)= 75 GB。Compute 服务预期这个值以 MB 为单位，即 75000。

以下 Python 代码提供了这个计算：

```
left_over_mem = mem - (GB_per_OSD * osds)
number_of_guests = int(left_over_mem /
    (average_guest_size + GB_overhead_per_guest))
nova_reserved_mem_MB = MB_per_GB * (
    (GB_per_OSD * osds) +
    (number_of_guests * GB_overhead_per_guest))
```

NovaCPUAllocationRatio

当选择在其上部署实例的 Compute 节点时，计算调度程序使用 **NovaCPUAllocationRatio**。默认情况下，此为 **16.0**（如在 16:1 中相同）。这意味着，如果节点上有 56 个内核，计算调度程序将调度充足的实例在节点上消耗 896 个 vCPU，然后再考虑节点无法托管。

要确定适合超融合节点的 **NovaCPUAllocationRatio**，假设每个 Ceph OSD 至少使用一个核心（除非工作负载是 I/O 密集型，并且没有 SSD）。在有 56 个内核和 10 个 OSD 的节点上，这会为 Compute 保留 46 个内核。如果每个实例使用 100 个接收的 CPU，则比率仅为实例 vCPU 的数量，按内核数除，即 46 / 56 = 0.8。但是，因为实例通常不会每个分配的 CPU 消耗 100 个，因此当确定所需客户机 vCPU 数量时，您可以提高 **NovaCPUAllocationRatio**。

因此，如果预测实例将在其 vCPU 的每 cent （或 0.1）使用 10 个，那么实例的 vCPU 数量可以表示为 46 / 0.1 = 460。当这个值被内核数除(56)时，比率增加到大约 8。

以下 Python 代码提供了这个计算：

```
cores_per_OSD = 1.0
average_guest_util = 0.1 # 10%
nonceph_cores = cores - (cores_per_OSD * osds)
guest_vCPUs = nonceph_cores / average_guest_util
cpu_allocation_ratio = guest_vCPUs / cores
```

1.7. 其他资源

有关 Red Hat OpenStack Platform (RHOSP)的详细信息，请参阅以下指南：

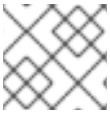
- director 安装和使用 ：本指南提供 RHOSP 环境的端到端部署指南，包括 undercloud 和 overcloud。
- 高级 Overcloud 自定义 ：本指南描述了如何通过 director 配置高级 RHOSP 功能，如如何使用自定义角色。
- 使用容器化 Red Hat Ceph 部署 Overcloud ：本指南描述了如何部署使用 Red Hat Ceph Storage 作为存储供应商的 overcloud。
- 网络指南 ：本指南详细介绍了 RHOSP 网络任务。

第 2 章 扩展超融合节点

要扩展 HCI 节点，适用扩展 Compute 节点或 Red Hat Ceph Storage 节点的相同原则和方法。

2.1. 在 HCI 环境中扩展超融合节点

要在 HCI 环境中扩展超融合节点，请遵循扩展非超融合节点的相同步骤。如需更多信息，请参阅 将节点添加到 overcloud 中。



注意

当您标记新节点时，请务必使用正确的类别。

有关如何通过将 OSD 添加到 Red Hat Ceph Storage 集群来扩展 HCI 节点的信息，请参阅 *Deploying an Overcloud with Containerized Red Hat Ceph* 中的 Adding an OSD to a Ceph Storage node 部分。

2.2. 在 HCI 环境中缩减超融合节点

要在 HCI 环境中缩减超融合节点，您必须在 HCI 节点上重新平衡 Ceph OSD 服务，从 HCI 节点迁移实例，并从 overcloud 中删除 Compute 节点。

流程

1. 在 HCI 节点上禁用和重新平衡 Ceph OSD 服务。这一步是必需的，因为当您删除 HCI 或 Red Hat Ceph Storage 节点时，director 不会自动重新平衡 Red Hat Ceph Storage 集群。
2. 从 HCI 节点迁移实例。如需更多信息，请参阅 *为实例创建指南中的配置 Compute 服务间迁移虚拟机*。
3. 从 overcloud 移除 Compute 节点。如需更多信息，请参阅 删除 Compute 节点。

附录 A. 附加信息

A.1. 配置指南

以下配置指南旨在为创建超融合基础架构环境提供框架。这个指南并不适用于为每个 Red Hat OpenStack Platform 安装提供确定的配置参数。请联系红帽 客户体验与参与团队 以获取适合您的特定环境的具体指导和建议。

- 集群大小并扩展
- 容量规划和大小

A.1.1. 集群大小并扩展

Red Hat Ceph Storage 硬件指南 为优化、吞吐量优化、成本和容量优化的 Ceph 部署场景提供建议。按照最适合您的部署场景的建议，并添加支持计算工作负载所需的 NIC、CPU 和 RAM。

最佳的、小型配置由 7 个节点组成。除非要求在您的环境中优化 IOPS 的性能，并且您使用的是所有闪存存储，否则应使用吞吐量优化部署场景。

可以有三个节点 Ceph Storage 集群配置。在这个配置中，您应该：

- 使用所有闪存存储。
- 在 **ceph.conf** 文件中，将 **replica_count** 参数设置为 3。
- 在 **ceph.conf** 文件中，将 **min_size** 参数设置为 2。

如果节点在这个配置中保留服务，则 IOPS 将继续。要保留 3 个数据副本，对第三个节点的复制会排队，直到它返回到服务。然后，数据被回填到第三个节点。

注意

最多 64 节点的 HCI 配置已被测试。一些 HCI 环境示例已被记录为 128 个节点。大型集群（如这些集群）可通过支持例外和咨询服务参与来考虑。请联系红帽 客户体验与参与团队 以获得指导。

有两个 NUMA 节点的部署可以在一个 NUMA 节点上托管对一个 NUMA 节点和 Ceph OSD 服务的延迟敏感的 Compute 工作负载。如果两个节点上都存在网络接口，并且磁盘控制器位于节点 0 上，请将节点 0 上的网络接口用于 Storage 网络，并在节点 0 上托管 Ceph OSD 工作负载。在节点 1 上托管计算工作负载，并将其配置为在节点 1 上使用网络接口。当为您的部署获取硬件时，请注意，哪些 NIC 将使用哪些节点并尝试在存储和工作负载之间分割它们。

A.1.2. 容量规划和大小

Red Hat Ceph Storage 硬件指南中定义的优化 Ceph 解决方案为大多数不需要优化 IOPS 的部署提供均衡解决方案。除了解决方案提供的配置指南外，在创建环境时请注意以下几点：

- 每个 OSD 分配的 5 GB RAM 可确保 OSD 有足够的操作内存。确保您的硬件可以支持此要求。
- CPU 速度应当与正在使用的存储介质匹配。更快的存储介质的优点（如 SSD）可能会由于支持它们的 CPU 太慢。同样，快速 CPU 可以被更快的存储介质更有效地使用。平衡 CPU 和存储介质速度，以便不再成为其他的瓶颈。