



Red Hat OpenStack Platform 16.2

特定 Red Hat OpenStack Platform 服务的部署建 议

最大化 Red Hat OpenStack Platform Telemetry 和 Object Storage 服务的性能

Red Hat OpenStack Platform 16.2 特定 Red Hat OpenStack Platform 服务的部署建议

最大化 Red Hat OpenStack Platform Telemetry 和 Object Storage 服务的性能

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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

在使用 director 部署 Red Hat OpenStack Platform 时，您可以按照这些建议解决许多性能问题。

目录

第 1 章 优化 OVERCLOUD 的原因	3
第 2 章 TELEMETRY 服务的配置建议	4
2.1. 在小型测试 OVERCLOUD 上配置 TELEMETRY 服务	4
2.2. 在大型生产 OVERCLOUD 上配置 TELEMETRY 服务	4
第 3 章 OBJECT STORAGE 服务(SWIFT)的配置建议.	7
3.1. 对象存储服务的磁盘建议	7
3.2. 定义专用对象存储节点	7
3.3. 对象存储服务的分区电源建议	8

第1章 优化 OVERCLOUD 的原因

如果您计划扩展到或部署大型 overcloud，请优化 overcloud，以防止在工作负载增加时造成潜在的性能问题。通过遵循这些建议，您可以防止扩展影响 Telemetry 服务和 overcloud 中的对象存储服务的性能。

第 2 章 TELEMETRY 服务的配置建议

因为 Red Hat OpenStack Platform (RHOSP) Telemetry 服务是 CPU 密集型，所以 RHOSP 16.2 中不默认启用遥测。但是，通过遵循这些部署建议，如果启用遥测功能，您可以避免性能下降。

这些流程 - 一个小、测试 overcloud，另一个用于大型生产 overcloud - 包含最大化 Telemetry 服务性能的建议。

2.1. 在小型测试 OVERCLOUD 上配置 TELEMETRY 服务

当您在小的情况下启用 Red Hat OpenStack Platform (RHOSP) Telemetry 服务时，测试 overcloud，您可以使用文件后端提高性能。

先决条件

- 您要配置 Telemetry 服务的 overcloud 部署 *不是* 生产系统。
- overcloud 是一个小型部署，支持少于 100 个实例，每个 Controller 节点上最多有 12 个物理内核，或启用了超线程的 24 个内核。
- overcloud 部署 *禁用了* 高可用性。

流程

1. 在 `/usr/share/openstack-tripleo-heat-templates/environments/enable-legacy-telemetry.yaml` 环境文件中添加以下到 `parameter_defaults` :

```
parameter_defaults:
  GnocchiBackend: file
```

2. 将 `enable-legacy-telemetry.yaml` 文件添加到 `openstack overcloud deploy` 命令中 :

```
openstack overcloud deploy \
-e /home/stack/environment.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/enable-legacy-telemetry.yaml \
[...]
```

其他资源

- *Director 安装和使用指南*中的 [修改 Overcloud 环境](#)

2.2. 在大型生产 OVERCLOUD 上配置 TELEMETRY 服务

当您在大型生产 overcloud 上启用 Red Hat OpenStack Platform (RHOSP) Telemetry 服务时，您可以通过在专用节点上部署 Telemetry 服务来提高性能。

Telemetry 服务使用选择哪个 RHOSP 对象存储作为其存储后端。如果没有启用 Red Hat Ceph Storage，Telemetry 服务将使用 RHOSP Object Storage 服务(swift)。默认情况下，RHOSP director 将对象存储服务与控制器上的 Telemetry 服务并置。



警告

将 RHOSP Object Storage (swift)用于时间序列数据库(gnocchi)存储，仅支持小型和非生产环境。

先决条件

- 您要在其上部署 Telemetry 服务的 overcloud 是一个大型生产 overcloud。

流程

1. 要设置专用遥测节点，请从 Controller 角色中删除遥测服务。
通过将 `/usr/share/openstack-tripleo-heat-templates/roles_data.yaml` 复制到 `/home/stack/templates/roles_data.yaml`，创建一个编排服务(heat)自定义环境文件。
2. 在 `/home/stack/templates/roles_data.yaml` 中，从 Controller 角色的 **ServicesDefault** 列表中删除以下行：

```
- OS::TripleO::Services::CeilometerAgentCentral
- OS::TripleO::Services::CeilometerAgentNotification
- OS::TripleO::Services::GnocchiApi
- OS::TripleO::Services::GnocchiMetricd
- OS::TripleO::Services::GnocchiStatsd
- OS::TripleO::Services::AodhApi
- OS::TripleO::Services::AodhEvaluator
- OS::TripleO::Services::AodhNotifier
- OS::TripleO::Services::AodhListener
- OS::TripleO::Services::PankoApi
- OS::TripleO::Services::CeilometerAgentIpmi
```

3. 添加以下片断并保存 **roles_data.yaml**：

```
- name: Telemetry
  ServicesDefault:
    - OS::TripleO::Services::CACerts
    - OS::TripleO::Services::CertmongerUser
    - OS::TripleO::Services::Kernel
    - OS::TripleO::Services::Ntp
    - OS::TripleO::Services::Timezone
    - OS::TripleO::Services::Snmp
    - OS::TripleO::Services::Sshd
    - OS::TripleO::Services::Securetty
    - OS::TripleO::Services::TripleoPackages
    - OS::TripleO::Services::TripleoFirewall
    - OS::TripleO::Services::SensuClient
    - OS::TripleO::Services::FluentdClient
    - OS::TripleO::Services::AuditD
    - OS::TripleO::Services::Collectd
    - OS::TripleO::Services::MySQLClient
    - OS::TripleO::Services::Docker
    - OS::TripleO::Services::CeilometerAgentCentral
```

```

- OS::TripleO::Services::CeilometerAgentNotification
- OS::TripleO::Services::GnocchiApi
- OS::TripleO::Services::GnocchiMetricd
- OS::TripleO::Services::GnocchiStatsd
- OS::TripleO::Services::AodhApi
- OS::TripleO::Services::AodhEvaluator
- OS::TripleO::Services::AodhNotifier
- OS::TripleO::Services::AodhListener
- OS::TripleO::Services::PankoApi
- OS::TripleO::Services::CeilometerAgentIpmi

```

- 在 `/home/stack/node-info.yaml` 文件中，设置 Telemetry 服务的专用节点数量。
例如，将 **TelemetryCount: 3** 添加到 **parameter_defaults** 中，以部署三个专用遥测节点：

```

parameter_defaults:
  TelemetryCount: 3

```

现在，您有一个自定义遥测角色。

使用此角色，您可以定义用于标记和分配特定遥测节点的新类别。

- 部署 overcloud 时，将 **roles_data.yaml** 和 **storage-environment.yaml** 包含在 **openstack overcloud deploy** 命令调用的模板和环境文件列表中：

```

$ openstack overcloud deploy \
  -r /home/stack/templates/roles_data.yaml \
  -e /home/stack/templates/storage-environment.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/enable-legacy-telemetry.yaml \
  [...]

```

- 如果无法为对象存储分配专用节点，请将对象存储配置为在 Controller 节点上运行较低的总体存储 I/O。

其他资源

- 高级 Overcloud 自定义指南中的创建新角色 https://access.redhat.com/documentation/zh-cn/red_hat_openstack_platform/16.2/html-single/advanced_overcloud_customization/index#proc_creating-a-new-role_composable-services-and-custom-roles
- Object Storage 服务(swift)的配置建议.
- Director 安装和使用指南中的 [修改 Overcloud 环境](#)

第 3 章 OBJECT STORAGE 服务(SWIFT)的配置建议.

如果您选择不使用 Red Hat Ceph Storage 部署 Red Hat OpenStack Platform (RHOSP), RHOSP director 将部署 RHOSP Object Storage 服务(swift)。对象存储服务是多个 OpenStack 服务的对象存储, 包括 RHOSP 遥测服务和 RabbitMQ。以下是在将 Telemetry 服务与对象存储服务搭配使用时提高 RHOSP 性能的几个建议。

3.1. 对象存储服务的磁盘建议

为 Red Hat OpenStack Platform (RHOSP)对象存储服务使用一个或多个单独的本地磁盘。

默认情况下, RHOSP director 将系统磁盘上的 `/srv/node/d1` 目录用于对象存储服务。在 Controller 上, 其他服务也会使用此磁盘, 磁盘可能会在 Telemetry 服务开始记录企业设置中的事件后成为性能瓶颈。

以下示例是 RHOSP 编排服务(heat)自定义环境文件摘录。在每个 Controller 节点上, 对象存储服务使用两个独立磁盘。两个磁盘的完整都包含一个 XFS 文件系统:

```
parameter_defaults:
  SwiftRawDisks: {"sdb": {}, "sdc": {}}
```

SwiftRawDisks 定义节点上的每个存储磁盘。这个示例在每个 Controller 节点上同时定义了 **sdb** 和 **sdc** 磁盘。



重要

在配置多个磁盘时, 请确保 Bare Metal 服务(ironic)使用预期的根磁盘。

其他资源

- *Director 安装和使用指南*中的 [为多磁盘集群定义根磁盘](#)。

3.2. 定义专用对象存储节点

将节点指定到 Red Hat OpenStack Platform (RHOSP)对象存储服务可提高性能。

流程

1. 创建一个自定义 **roles_data.yaml** 文件 (基于默认的 `/usr/share/openstack-tripleo-heat-templates/roles_data.yaml`)。
2. 通过从 Controller 节点中删除 Object Storage 服务条目来编辑自定义 **roles_data.yaml** 文件。具体来说, 从 **Controller** 角色的 **ServicesDefault** 列表中删除以下行:

```
- OS::TripleO::Services::SwiftStorage
```

3. 使用自定义环境文件中的 **ObjectStorageCount** 资源, 设置要为对象存储服务分配的专用节点数量。例如, 将 **ObjectStorageCount: 3** 添加到环境文件中的 **parameter_defaults** 中, 以部署三个专用对象存储节点:

```
parameter_defaults:
  ObjectStorageCount: 3
```

4. 要应用此配置，请部署 overcloud，将 `roles_data.yaml` 添加到堆栈和其他环境文件：

```
(undercloud) $ openstack overcloud deploy --templates \
-e [your environment files]
-e /home/stack/templates/roles_data.yaml
```

其他资源

- [高级 Overcloud 自定义指南中的可组合服务和自定义角色](#)
- [高级 Overcloud 自定义指南中的从角色添加和删除服务](#)
- [Director 安装和使用指南中的修改 Overcloud 环境](#)

3.3. 对象存储服务的分区电源建议

使用单独的 Red Hat OpenStack Platform (RHOSP) Object Storage 服务节点时，使用较高的分区电源值。

对象存储服务使用修改后的 *哈希环* 在磁盘和节点之间分发数据。默认有三个环：一个用于帐户，一个用于容器，另一个用于对象。每个环使用一个称为 *分区电源* 的固定参数。此参数设置可创建的最大分区数。

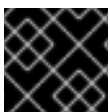
分区电源参数非常重要，只能为新容器及其对象更改。因此，*在初始部署前* 设置这个值非常重要。

对于 RHOSP director 部署的环境，默认分区 power 值为 **10**。对于较小的部署来说，这是合理的值，特别是您只计划在 Controller 节点上为对象存储服务使用磁盘。

如果使用三个副本，下表可帮助您选择适当的分区电源：

表 3.1. 每个可用磁盘数量的适当分区电源值

分区 Power	磁盘的最大数量
10	~ 35
11	~ 75
12	~ 150
13	~ 250
14	~ 500



重要

设置过度高的分区电源值（例如，**14** 代表 40 个磁盘）对复制时间造成负面影响。

要设置分区电源，请使用以下资源：

```
parameter_defaults:
  SwiftPartPower: 11
```

■

提示

您还可以为新容器配置额外的对象服务器 ring。如果要在最初使用低分区电源的对象存储服务部署中添加更多磁盘，这非常有用。

其他资源

- [存储指南中的对象存储环](#)
- [swift 上游文档中的 Rings](#)
- [Director 安装和使用指南中的 修改 Overcloud 环境](#)