



# Red Hat OpenStack Platform 16.1

## Deployment Recommendations for Specific Red Hat OpenStack Platform Services

Maximizing the performance of the Red Hat OpenStack Platform Telemetry and  
Object Storage services



# Red Hat OpenStack Platform 16.1 Deployment Recommendations for Specific Red Hat OpenStack Platform Services

---

Maximizing the performance of the Red Hat OpenStack Platform Telemetry and Object Storage services

OpenStack Team  
rhos-docs@redhat.com

## Legal Notice

Copyright © 2020 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.

## Abstract

You can address many performance issues by following these recommendations when deploying Red Hat OpenStack Platform with director.

---

## Table of Contents

<b>CHAPTER 1. OVERVIEW .....</b>	<b>3</b>
1.1. REASONS TO OPTIMIZE YOUR OVERCLOUD	3
<b>CHAPTER 2. CONFIGURATION RECOMMENDATIONS FOR THE TELEMETRY SERVICE .....</b>	<b>4</b>
2.1. CONFIGURING THE TELEMETRY SERVICE ON A SMALL, TEST OVERCLOUD	4
2.2. CONFIGURING THE TELEMETRY SERVICE ON A LARGE, PRODUCTION OVERCLOUD	4
<b>CHAPTER 3. CONFIGURATION RECOMMENDATIONS FOR THE OBJECT STORAGE SERVICE (SWIFT) ..</b>	<b>7</b>
3.1. DISK RECOMMENDATION FOR THE OBJECT STORAGE SERVICE	7
3.2. TOPOLOGY RECOMMENDATION FOR THE OBJECT STORAGE SERVICE	7
3.2.1. Defining dedicated Object Storage nodes	7
3.3. PARTITION POWER RECOMMENDATION FOR THE OBJECT STORAGE SERVICE	8



# CHAPTER 1. OVERVIEW

## 1.1. REASONS TO OPTIMIZE YOUR OVERCLOUD

If you are planning to scale to or deploy a large overcloud, optimize your overcloud to prevent any potential performance issues as its workload increases. By following these recommendations, you can prevent scale from affecting the performance of the Telemetry service and the Object Storage service within the overcloud.

## CHAPTER 2. CONFIGURATION RECOMMENDATIONS FOR THE TELEMETRY SERVICE

Because the Red Hat OpenStack Platform (RHOSP) Telemetry service is CPU-intensive, telemetry is not enabled by default in RHOSP 16.1. However, by following these deployment recommendations, you can avoid performance degradation if you enable telemetry.

These procedures—one for a small, test overcloud and one for a large, production overcloud—contain recommendations that maximize Telemetry service performance.

### 2.1. CONFIGURING THE TELEMETRY SERVICE ON A SMALL, TEST OVERCLOUD

When you enable the Red Hat OpenStack Platform (RHOSP) Telemetry service on small, test overclouds, you can improve its performance by using a file back end.

#### Prerequisites

- The overcloud deployment on which you are configuring the Telemetry service is *not* a production system.
- The overcloud is a small deployment that supports fewer than 100 instances, with a maximum of 12 physical cores on each Controller node, or 24 cores with hyperthreading enabled.
- The overcloud deployment has high availability *disabled*.

#### Procedure

1. Add the following to **parameter\_defaults** in your **/usr/share/openstack-tripleo-heat-templates/environments/enable-legacy-telemetry.yaml** environment file and replace <FILE> with the name of the gnocchi configuration file:

```
parameter_defaults:
  GnocchiBackend: <FILE>
```

2. Add the **enable-legacy-telemetry.yaml** file to your **openstack overcloud deploy** command:

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

#### Additional resources

- [Modifying the Overcloud Environment](#) in the *Director Installation and Usage* guide

### 2.2. CONFIGURING THE TELEMETRY SERVICE ON A LARGE, PRODUCTION OVERCLOUD

When you enable the Red Hat OpenStack Platform (RHOSP) Telemetry service on a large production overcloud, you can improve its performance by deploying the Telemetry service on a dedicated node.



The Telemetry service uses whichever RHOSP object store that has been chosen as its storage back end. If you do not enable Red Hat Ceph Storage, the Telemetry service uses the the RHOSP Object Storage service (swift). By default, RHOSP director colocates the Object Storage service with the Telemetry service on the Controller.

## Prerequisites

- The overcloud on which you are deploying the Telemetry service is a large, production overcloud.

## Procedure

1. To set dedicated telemetry nodes, remove the telemetry services from the Controller role. Create an Orchestration service (heat) custom environment file by copying `/usr/share/openstack-tripleo-heat-templates/roles_data.yaml` to `/home/stack/templates/roles_data.yaml`.
2. In `/home/stack/templates/roles_data.yaml`, remove the following lines from the **ServicesDefault** list of the Controller role:

```
- 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. Add the following snippet, and save **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
```

4. In the **/home/stack/storage-environment.yaml** file, set the number of dedicated nodes for the Telemetry service.

For example, add **TelemetryCount: 3** to the **parameter\_defaults** to deploy three dedicated telemetry nodes:

```
parameter_defaults:
  TelemetryCount: *3*
```

You now have a custom telemetry role.

With this role, you can define a new flavor to tag and assign specific telemetry nodes.

5. When you deploy your overcloud, include **roles\_data.yaml** and **storage-environment.yaml** to the list of your templates and environment files that **openstack overcloud deploy** command calls:

```
$ 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 \
  [...]
```

6. If you cannot allocate dedicated nodes to the Telemetry service, and you still need to use the Object Storage service as its back end, configure the Object Storage service on the Controller node. Locating the Object Storage service on the Controller lowers the overall storage I/O.

## Additional resources

- [Creating a New Role](#) in the *Advanced Overcloud Customization* guide
- [Configuration recommendations for the Object Storage service \(swift\)](#)
- [Modifying the Overcloud Environment](#) in the *Director Installation and Usage* guide

## CHAPTER 3. CONFIGURATION RECOMMENDATIONS FOR THE OBJECT STORAGE SERVICE (SWIFT)

If you choose not to deploy Red Hat OpenStack Platform (RHOSP) with Red Hat Ceph Storage, RHOSP director deploys the RHOSP Object Storage service (swift). The Object Store service is the object store for several OpenStack services, including the RHOSP Telemetry service and RabbitMQ. Here are several recommendations to improve your RHOSP performance when using the Telemetry service with the Object Storage service.

### 3.1. DISK RECOMMENDATION FOR THE OBJECT STORAGE SERVICE

Use one or more separate disks for the Red Hat OpenStack Platform (RHOSP) Object Storage service.

By default, RHOSP director uses the directory `/srv/node/d1` on the system disk for the Object Storage service. On the Controller this disk is also used by other services, and the disk could become a performance bottleneck after the Telemetry service starts recording events in an enterprise setting.

The following example is a excerpt from an RHOSP Orchestration service (heat) custom environment file. On each Controller node, the Object Storage service uses two separate disks. The entirety of both disks contains an XFS file system:

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

**SwiftRawDisks** defines each storage disk on the node. This example defines both **sdb** and **sdc** disks on each Controller node.



#### IMPORTANT

When configuring multiple disks, ensure that the Bare Metal service (ironic) uses the intended root disk.

#### Additional resources

- [Defining the Root Disk for Nodes](#) in the *Director Installation and Usage* guide.

### 3.2. TOPOLOGY RECOMMENDATION FOR THE OBJECT STORAGE SERVICE

Define dedicated nodes for the Red Hat OpenStack Platform (RHOSP) Object Storage service. Doing this prevents any disk I/O by the RHOSP Telemetry service from impacting any other services on the Controller node.

#### 3.2.1. Defining dedicated Object Storage nodes

Dedicating a node to the Red Hat OpenStack Platform (RHOSP) Object Storage service improves performance.

#### Procedure

1. Create a custom **roles\_data.yaml** file (based on the default `/usr/share/openstack-tripleo-heat-templates/roles_data.yaml`).

2. Edit the custom **roles\_data.yaml** file by removing the Object Storage service entry from the Controller node.

Specifically, remove the following line from the **ServicesDefault** list of the **Controller** role:

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

3. Use the **ObjectStorageCount** resource in your custom environment file to set how many dedicated nodes to allocate for the Object Storage service.

For example, add **ObjectStorageCount: 3** to the **parameter\_defaults** in your environment file to deploy three dedicated object storage nodes:

```
parameter_defaults:
  ObjectStorageCount: 3
```

4. To apply this configuration, deploy the overcloud, adding **roles\_data.yaml** to the stack along with your other environment files:

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

#### Additional resources

- [Composable Services and Custom Roles](#) in the *Advanced Overcloud Customization* guide
- [Adding and Removing Services from Roles](#) in the *Advanced Overcloud Customization* guide
- [Replacing Object Storage Nodes](#) in the *Director Installation and Usage* guide
- [Modifying the Overcloud Environment](#) in the *Director Installation and Usage* guide

### 3.3. PARTITION POWER RECOMMENDATION FOR THE OBJECT STORAGE SERVICE

When using separate Red Hat OpenStack Platform (RHOSP) Object Storage service nodes, use a higher partition power value.

The Object Storage service distributes data across disks and nodes using modified *hash rings*. There are three rings by default: one for accounts, one for containers, and one for objects. Each ring uses a fixed parameter called *partition power*. This parameter sets the maximum number of partitions that can be created.

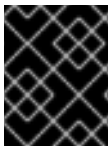
The partition power parameter is important and can only be changed for new containers and their objects. As such, it is important to set this value before *initial deployment*.

The default partition power value is **10** for environments that RHOSP director deploys. This is a reasonable value for smaller deployments, especially if you only plan to use disks on the Controller nodes for the Object Storage service.

The following table helps you to select an appropriate partition power if you use three replicas:

**Table 3.1. Appropriate partition power values per number of available disks**

Partition Power	Maximum number of disks
10	~ 35
11	~ 75
12	~ 150
13	~ 250
14	~ 500



### IMPORTANT

Setting an excessively high partition power value (for example, **14** for only 40 disks) negatively impacts replication times.

To set the partition power, use the following resource:

```
parameter_defaults:  
  SwiftPartPower: 11
```

### TIP

You can also configure an additional object server ring for new containers. This is useful if you want to add more disks to an Object Storage service deployment that initially uses a low partition power.

### Additional resources

- [Configure an Object Storage Ring](#) in the *Storage Guide*
- [The Rings](#) in swift upstream documentation
- [Modifying the Overcloud Environment](#) in the *Director Installation and Usage* guide