



Red Hat Enterprise Linux OpenStack Platform 7

发行注记

Red Hat Enterprise Linux OpenStack Platform 的发行信息

OpenStack 文档团队

Red Hat Enterprise Linux OpenStack Platform 7 发行注记

Red Hat Enterprise Linux OpenStack Platform 的发行信息

OpenStack 文档团队
Red Hat 出版部
rhos-docs@redhat.com

法律通告

Copyright © 2015 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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, 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 Software Collections 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

本文档包括了这个 Red Hat Enterprise Linux OpenStack Platform 发行版本的主要功能、改进和已知的问题。

目录

第 1 章 简介	2
1.1. 关于本发行版本	2
1.2. 要求	2
1.3. 部署限制	3
1.4. 认证的驱动和插件	3
1.5. 认证的虚拟机操作系统	3
1.6. 支持的虚拟机监控程序	3
1.7. Content Delivery Network (CDN) 频道	3
1.8. 产品支持	4
1.9. 产品文档	5
第 2 章 主要的新功能	7
2.1. RHEL OpenStack Platform Director	7
2.2. Block Storage	7
2.3. Compute	8
2.4. Identity	8
2.5. Image Service	9
2.6. Object Storage	9
2.7. OpenStack 集成测试套件	10
2.8. OpenStack 网络	10
2.9. 技术预览	11
第 3 章 发行信息	13
3.1. 改进	13
3.2. 发行注记	16
3.3. 已知问题	18
3.4. 过时的功能	19
第 4 章 技术备注	20
4.1. RHEA-2015:1548 — Red Hat Enterprise Linux OpenStack Platform 功能增强公告	20
4.2. RHEA-2015:1549 — Red Hat Enterprise Linux OpenStack Platform director 发行版本	37
附录 A. 修订历史	56

第 1 章 简介

Red Hat Enterprise Linux OpenStack Platform 提供了一个在 Red Hat Enterprise Linux 上构建私有或公共 IaaS (Infrastructure-as-a-Service, 基础设施即服务) 云服务的平台。它为部署使用云技术的计算环境提供了扩展性和容错性。

当前的红帽发行版本是基于 OpenStack Kilo 的, 它可以使您的物理硬件转换为具有以下特性的私有云、公共云或混合云平台:

- ✧ 完全分布的对象存储
- ✧ 持久性的块级别存储
- ✧ 虚拟机设置引擎和镜像存储
- ✧ 验证和授权机制
- ✧ 集成的网络
- ✧ 普通用户和管理员使用的、基于 web 的图形用户界面。

Red Hat Enterprise Linux OpenStack Platform IaaS 云通过一组相互合作的服务实现, 这些服务可以控制云的计算资源、存储资源和网络资源。管理员可以通过一个基于 web 的接口对云进行管理 (控制、设置和自动化 OpenStack 资源)。另外, OpenStack 还包括了一个扩展的 API, 云的最终用户可以使用这些 API。

1.1. 关于本发行版本

这个 Red Hat Enterprise Linux OpenStack Platform 发行版本是基于 OpenStack "Kilo" 的, 它包括了 Red Hat Enterprise Linux OpenStack Platform 特有的新功能、已知问题以及相关问题的解决方案。

本发行注册只包括了与 Red Hat Enterprise Linux OpenStack Platform 相关的信息。OpenStack "Kilo" 本身的发行注册可以从以下资源获得: <https://wiki.openstack.org/wiki/ReleaseNotes/Kilo>

Red Hat Enterprise Linux OpenStack Platform 同时使用了其它红帽产品的组件, 相关信息可以从以下资源获得:

<https://access.redhat.com/site/support/policy/updates/openstack/platform/>

如需试用 Red Hat Enterprise Linux OpenStack Platform, 请通过以下网址注册:

<http://www.redhat.com/openstack/>.



注意

Red Hat Enterprise Linux OpenStack Platform 可以使用 Red Hat Enterprise Linux High Availability Add-On (<http://www.redhat.com/products/enterprise-linux-add-ons/high-availability/>)。 <https://access.redhat.com/site/solutions/509783> 包括了 Red Hat Enterprise Linux OpenStack Platform 可以使用的软件包版本信息。

1.2. 要求

这个版本的 Red Hat Enterprise Linux OpenStack Platform 在 Red Hat Enterprise Linux 7.1 上被支持。

Red Hat Enterprise Linux OpenStack Platform 仪表板 (dashboard) 是一个基于 web 的用户界面，用户可以通过它来管理 OpenStack 资源和服务。这个版本中的仪表板支持以下网络浏览器的最新发行版：

- Chrome
- Firefox
- Firefox ESR
- Internet Explorer 11 和更高版本 (需要禁用 *Compatibility Mode*)

1.3. 部署限制

如需获得 RHEL OpenStack Platform 的部署限制列表，请参阅 [Deployment Limits for Red Hat Enterprise Linux OpenStack Platform](#)。

1.4. 认证的驱动和插件

如需获得 RHEL OpenStack Platform 中认证的驱动和插件列表，请参阅 [Component, Plug-In, and Driver Support in RHEL OpenStack Platform](#)。

1.5. 认证的虚拟机操作系统

如需获得认证的 RHEL OpenStack Platform 虚拟机操作系统，请参阅 [Certified Guest Operating Systems in Red Hat Enterprise Linux OpenStack Platform and Red Hat Enterprise Virtualization](#)。

1.6. 支持的虚拟机监控程序

Red Hat Enterprise Linux OpenStack Platform 只支持 **libvirt** 驱动 (在 Compute 节点上使用 KVM 作为虚拟机监控程序) 和 VMware vCenter 虚拟机监控程序驱动。如需了解配置 VMware vCenter 驱动的信息，请参阅 *VMware Integration Guide*。

在这个 Red Hat Enterprise Linux OpenStack Platform 发行版本中，ironic 被完全支持。通过 Ironic，您可以使用常用的技术 (如 PXE 引导和 IPMI) 来部署 (provision) 裸机，并支持使用可插入的驱动程序来实现与特定厂商相关的功能。

红帽不支持其它 Compute 虚拟化驱动，如 VMware "direct-to-ESX" hypervisor 和非 KVM libvirt hypervisor。

1.7. Content Delivery Network (CDN) 频道

本节包括了部署 Red Hat Enterprise Linux OpenStack Platform 所需的频道和软件仓库信息。

您可以通过 Content Delivery Network (CDN) 安装 Red Hat Enterprise Linux OpenStack Platform 7。您需要配置 **subscription-manager** 来使用正确的频道。

运行以下命令启用 CDN 频道：

```
# subscription-manager repos --enable=[reponame]
```

运行以下命令禁用 CDN 频道：

```
# subscription-manager repos --disable=[reponame]
```

表 1.1. 所需频道

频道	软件仓库名
Red Hat Enterprise Linux 7 Server (RPMS)	rhel-7-server-rpms
Red Hat Enterprise Linux 7 Server - RH Common (RPMS)	rhel-7-server-rh-common-rpms
Red Hat Enterprise Linux High Availability (for RHEL 7 Server)	rhel-ha-for-rhel-7-server-rpms
Red Hat Enterprise Linux OpenStack Platform 7.0 (RPMS)	rhel-7-server-openstack-7.0-rpms
Red Hat Enterprise Linux OpenStack Platform Director 7.0 (RPMS)	rhel-7-server-openstack-7.0-director-rpms

表 1.2. 可选频道

频道	软件仓库名
Red Hat Enterprise Linux 7 Server - Optional	rhel-7-server-optional-rpms
Red Hat Enterprise Linux OpenStack Platform 7.0 Files	rhel-7-server-openstack-7.0-files
Red Hat Enterprise Linux OpenStack Platform 7.0 Operational Tools	rhel7-server-openstack-7.0-optools-rpms

禁用频道

为了使 Red Hat Enterprise Linux OpenStack Platform 7 可以正常工作，请禁用下表中的频道。

表 1.3. 禁用频道

频道	软件仓库名
Red Hat CloudForms Management Engine	"cf-me-"
Red Hat Enterprise Virtualization	"rhel-7-server-rhev"
Red Hat Enterprise Linux 7 Server - Extended Update Support	"*-eus-rpms"



警告

Red Hat Enterprise Linux OpenStack Platform 软件仓库中的一些软件包和 Extra Packages for Enterprise Linux (EPEL) 软件仓库提供的软件包有冲突。在启用了 EPEL 软件包仓库的系统上使用 Red Hat Enterprise Linux OpenStack Platform 不被支持。

1.8. 产品支持

可用资源包括：

客户门户网站

红帽客户门户网站提供了丰富的资源帮助您规划、实施和维护您的 OpenStack 系统。这些资源包括：

- ✱ 知识库文档和问题解答。

- 技术概要。
- 产品文档。
- 客户问题管理。

通过 <https://access.redhat.com/> 访问客户门户网站。

邮件列表

红帽为用户提供了以下与 OpenStack 相关的公共邮件列表

- **rhsa-announce** 邮件列表提供了红帽产品（包括 Red Hat Enterprise Linux OpenStack Platform）的安全补丁程序发行通知。

请通过 <https://www.redhat.com/mailman/listinfo/rhsa-announce> 订阅这个邮件列表。

- **rhos-list** 邮件列表提供了一个讨论安装、运行和使用红帽发行的 OpenStack 产品的论坛。

请通过 <https://www.redhat.com/mailman/listinfo/rhos-list> 订阅这个邮件列表。

1.9. 产品文档

在这个 RHEL OpenStack Platform 版本中，以前以 KBase 文档形式发布的内容被重新根据 RHEL OpenStack Platform 的组件和管理任务进行组合，并编辑为一系列新的指南。现在，这些文档包括 **html**、**html-single**、**epub** 和 **pdf** 格式。使用 Packstack 部署工具试用 RHEL OpenStack Platform 的内容将不再出现在产品文档页中，但使用这个工具的一些情景介绍包括在 RHEL OpenStack Platform 产品页中。

产品文档包括：

[Architecture Guide](#)

对 RHEL OpenStack Platform 的每个主组件的介绍，以及一组使用情景示例。这个指南的内容包括了 RHEL OpenStack Platform 6 中的 *Component Overview* 的内容，以及设计 RHEL OpenStack Platform 环境时需要考虑的内容和相关示例。

[Command-Line Interface Reference](#)

RHEL OpenStack Platform 命令行客户端的参考文档，包括了可用的命令、语法和选项列表。

[Configuration Reference](#)

RHEL OpenStack Platform 环境中每个组件的配置选项的参考文档。

[Dell EqualLogic Back End Guide](#)

介绍了如果配置 RHEL OpenStack Platform 来使用一个或多个 Dell EqualLogic 后端。

[Director Installation and Usage](#)

使用新的 RHEL OpenStack Platform director 实施和管理 RHEL OpenStack Platform 环境的详细信息。

[Installation Reference](#)

手工配置组件的基本步骤的参考文档。这个文档是以前版本中的 *Deploying OpenStack: Learning Environments (Manual Setup)* 文档的改进版。

[Introduction to the OpenStack Dashboard](#)

对 RHEL OpenStack Platform dashboard 主要用户接口项的介绍。

[Instances and Images Guide](#)

这个文档包括了对创建和管理镜像、使用实例以及相关概念（如卷和容器）的介绍。

[Logging, Monitoring, and Troubleshooting Guide](#)

与 RHEL OpenStack Platform 中的日志、如果配置和使用 telemetry 服务、解决一般问题的方法以及 dashboard 中的 Red Hat Access 标签页相关的信息。

[Migrating Instances](#)

与在 RHEL OpenStack Platform 环境中的 hypervisor 间迁移运行实例和静态实例相关的信息。

[Networking Guide](#)

RHEL OpenStack Platform 中的网络服务的指南信息，包括对网络服务的介绍、传统的网络概念和软件定义的网络间的关系。另外，它还包括了对高级的网络概念、网络操作以及在使用网络时需要考虑的因素的介绍。

[OpenStack Data Processing](#)

配置和使用数据处理服务来部署和扩展 Hadoop 集群以用来处理大量数据的信息。

[Package Manifest](#)

本 RHEL OpenStack Platform 发行版本以及其后的主要维护发行版本的完整软件包列表。

[Users and Identity Management Guide](#)

创建和使用用户、项目和角色，以及集成外部目录的指南。

[VMware Integration Guide](#)

RHEL OpenStack Platform 和 VMware 进行集成的信息，包括如何把虚拟机从 VMware 迁移到 RHEL OpenStack Platform、与 VMware vCenter 进行集成的信息以及 VMware NSX 和 OpenStack Networking 间的集成信息。

第 2 章 主要的新功能

本节介绍了这个 Red Hat Enterprise Linux OpenStack Platform 发行版本中包括的主要新功能。

2.1. RHEL OpenStack Platform Director

Red Hat Enterprise Linux OpenStack Platform director 是一个用来部署 OpenStack 环境以及管理产品生命周期的新工具。它是基于 TripleO 项目的，可以提供以下功能：

实施的一致性

Red Hat Enterprise Linux OpenStack Platform director 提供了一个使用 OpenStack 服务和 API 安装 OpenStack 环境的方法。这个方法会使用一个底层的 OpenStack 实例进行安装，然后再管理其它实例，OpenStack 实例通过一组镜像来构建配置文件。director 使用 CLI 或 GUI 实现这个方法。另外，这个方法还需要使用 OpenStack 的 IroniC 服务来部署裸机。

除了安装、配置和管理任务外，director 还提供了一组在安装完成后进行的自动化基准指标数据和健康检查。另外，它还还为 RAID、BIOS 和网络接口提供了就绪状态的配置。

生命周期管理

Red Hat Enterprise Linux OpenStack Platform director 提供了对环境进行扩展的工具。director 可以在当前使用的 Red Hat Enterprise Linux OpenStack Platform 版本环境中应用新的程序更新，并且可以把 Red Hat Enterprise Linux OpenStack Platform 升级到更新的版本。

director 还可以和以下红帽产品进行集成：

- ✱ Red Hat Ceph Storage
- ✱ Red Hat Satellite 6
- ✱ 红帽其它的云基础架构产品（Red Hat CloudForms 和 Red Hat Enterprise Virtualization）

更快的版本更新

红帽计划每两个月进行一次 Red Hat Enterprise Linux OpenStack Platform director 版本更新。

操作可见性（技术预览）

Red Hat Enterprise Linux OpenStack Platform director 可以作为 OpenStack 环境的一个特定的日志工具。它还包括了对可用性和性能监测的提示信息。

2.2. Block Storage

增量备份

Block Storage 服务现在支持增量备份。使用这个功能，您可以只对前一次备份做增量备份，而不需要每次都进行完全备份。这就可以在您的卷数量和大小增加时，提供更好的备份扩展性。

Block Storage 现在还支持基于快照的备份。您可以在卷仍然附加在实例上时对它们进行备份。

NFS 和 POSIX 备份

Cinder 备份支持使用 NFS 和 POSIX 提供的数据仓库作为备份目标。

私有卷类型

在创建一个卷类型时，您可以把它设为私有。私有卷类型只能在明确添加到的项目中被访问。如果

一个私有卷没有被添加到任何项目中，只有具有 admin 角色的用户才可以使用它。私有卷可以用来对卷的访问进行限制。

增强的 iSCSI 多路径支持

Block Storage 服务现在可以为 Compute 服务访问一个卷提供所有有效的门户 (portal)、IQN 和 LUN 信息。这将允许 Compute 服务为每个目标建立多个会话，从而可以在主连接出现问题时实现故障转移的功能。

Consistency 组

现在，Block Storage 服务允许您设置 Consistency 组。使用它，您可以把多个卷组成一个单独的项，从而可以同时对这些卷进行操作（如创建快照），而不用分别对每个卷进行操作。目前，在 consistency 组中支持的唯一操作是创建快照。

请注意，这个功能只能通过 IBM Storwize 驱动实现。

2.3. Compute

支持在使用 QEMU Guest Agent 进行镜像快照时静默 (quiesce) 文件系统。

以前，在对活跃的实例进行快照时，需要使用 fsfreeze 手工静默文件系统。在这个版本中，Compute 的 libvirt 驱动会在进行快照时自动请求 QEMU Guest Agent 来冻结文件系统（以及安装了 fsfreeze-hook 的应用）。对静默文件系统功能的支持实现了块设备一级的自动调度快照功能。

这个功能只在安装了 QEMU Guest Agent (qemu-ga)，并且镜像元数据启用了代理 (hw_qemu_guest_agent=yes) 的情况下才有效。

OpenStack Bare Metal Provisioning (ironic)

Bare Metal Provisioning 服务现在在 Compute 中被支持（使用 Compute 和 OpenStack Networking 服务配置）。这个服务和 Compute 服务集成在一起（和部署虚拟机的方式相同），为 'bare-metal-to-trusted-tenant' 用例提供了一个解决方案。例如，在 OpenStack 云中：

- Hadoop 集群可以在裸机上进行部署。
- Hyperscale 和 HPC 集群可以被部署。
- 可以使用用于对虚拟机敏感的应用程序的数据库。

这个服务由 Bare Metal Provisioning API、一个 Conductor、数据库和硬件驱动组成，并可以利用常见的技术，如 PXE、IPMI 和 DHCP。在这个版本中，Bare Metal Provisioning 服务还可以接受使用 flavor 的 extra-spec 关键字定义的能力。

2.4. Identity

分级结构的多租户技术

Red Hat Enterprise Linux OpenStack Platform 现在增加了对对象所有权的分级结构支持。现在，您可以修改 RHEL OpenStack Platform 的机构组织形式，在 Identity 中创建嵌套的项目。

使用 SAML 的联邦 (Federation)

联邦的 Identity 在 Identity Providers (IdP) 和 OpenStack Cloud 提供给最终用户的服务间创建了信任关系。它提供了使用存在的凭证安全访问不同云资源（如服务器、卷和数据库）的一个方法。这些资源可以分散在由不同云授权的不同端点中，用户只需要使用一组凭证来访问它们，而不需要再部署其它的凭证，或进行多次登录。用户和组的凭证由用户的 Identity Provider 提供。

联邦的用户信息没有包括在 Identity 服务的后端中（例如，SQL 驱动）。Identity Provider 有责任来验证用户，并使用 SAML 声明（SAML assertion）把验证的结果通知给 Identity 服务。SAML 声明中包括了一个 Identity Provider 所提供的用户信息。Identity 服务在 SAML 声明和由 Identity 服务所创建的 Keystone 用户组间创建映射关系。

使用 Keystone 和 SAML 的 Web SSO

现在，RHEL OpenStack Platform 为用户提供了一个验证方法：用户可以使用一个存在的 Identity Provider (IdP)，通过网络浏览器访问一个单点登录页来进行身份验证。

2.5. Image Service

镜像转换

允许您在导入镜像时，根据 store 的情况动态转换镜像。（qcow/raw）。

导入流程中的一个插件提供了转换功能。根据部署者的配置，您可以激活或取消激活这个插件。作为一个部署者，您需要为部署指定首选的镜像格式。当前 qemu-img 转换工具所支持的格式是 **raw** 和 **qcow2**。

镜像内省 (introspection)

元数据摘要信息可以帮助管理员更好地了解如何覆盖特定的元数据。

多个镜像格式都会在镜像数据中包括它的元数据。现在，通过镜像内省功能可以获得镜像的元数据。例如，您可以通过一个 **vmdk** 格式镜像的元数据了解到镜像的磁盘类型是 **“streamOptimized”**。使用 Image 服务的内省功能可以减少管理员的负担。另外，元数据信息对镜像的使用者也会有帮助，因为根据镜像的磁盘类型的不同，Compute 的工作流程也会有所不同。

当前，可用的、和 Image 服务相关的元数据项是 **disk_format** 和 **virtual_size**。

2.6. Object Storage

复合型令牌 (Composite Token) 和服务账户 (Service Account)

以前，数据存储在在一个专用的服务帐户中（单独的项目）；或存储在最终用户的帐户中（多个项目）。如果数据存储在服务帐户中，容器或服务用户删除就可能会有一些问题，密码和令牌也会比较脆弱。如果数据存储在最终用户帐户中，当用户使用和服务相同的名称时，就会出现问

题，Object Storage 服务需要处理用户破坏了令牌完整性的问题。

- 这些问题的解决办法：
- Object Storage 现在把服务数据保存在一个和最终用户的 'normal' 帐户相独立的帐户中，但它仍然和最终用户的项目相关联。对服务帐户的访问由复合型令牌管理。
 - 复合型令牌在存储数据时使用两个验证令牌：一个来自于 Object Storage 服务，另一个来自于最终用户。当这两个令牌被设定后，如果没有服务和用户双方的确认，数据将无法被改变。这个机制可以防止在缺少管理操作的情况下，数据被最终用户删除；另外，也可以防止服务替代最终用户做出决定。

对全局分布式集群的高效复制

以前，在全局分布式集群（globally distributed cluster）中，副本（replica）会在每个区域中被推到所有节点上。在这个版本中，Object Storage 服务现在只把副本推到每个区域的一个远程节点上（根据管理规则），然后，这个远程节点再把副本推到每个区域中的主节点（primary mode）上。这个方法可以减少在区域间的数据流量，从而可以获得稳定的性能，并可以降低副本延迟的机会，以及数据传输的成本。

2.7. OpenStack 集成测试套件

现在，Red Hat Enterprise Linux OpenStack Platform 包括了集成测试套件（tempest）。这个套件包括了一组测试，可以被用来测试您的 OpenStack 环境，保证您的云可以正常工作。它会把独立的 OpenStack 模块进行组合，作为一个组来进行测试，从而可以完整地测试云的功能。

集成测试套件提供了以下功能：

完成的测试

这个套件包括了 API 测试、场景测试和压力测试。另外，这个套件中还包括了可以用来对套件本身代码进行测试的单元测试。您可以运行整个测试套件（所有测试在一个目录中），也可以只运行其中的一个测试。

配置

您可以手工配置套件，也可以使用脚本从测试环境中抓取信息（查询云环境）来创建所需的资源或凭证。

可扩展

这个套件可以对任何规格的 OpenStack 云进行测试。它可以在云的计算节点和存储节点上操作实例或卷，对它们进行测试，然后再终止它们。

公共接口

这个套件只针对于公共接口（您的 OpenStack 端点）运行，它不会使用私人接口以及只针对于实施的特定接口。测试不会直接针对数据库或虚拟机管理程序（hypervisor）运行。

可选的验证

测试可以通过一个普通用户运行，也可以通过一个全局的 admin 用户运行，还可以使用其它用户的凭据运行。

2.8. OpenStack 网络

ML2 和 Open vSwitch 的端口安全性

在默认情况下，OpenStack Networking 会使用反欺骗（anti-spoofing）防火墙规则，虚拟机将无法使用没有在它们自己的网络端口中配置的 MAC 或 IP 地址进行通讯。在 Red Hat Enterprise Linux OpenStack Platform 7 中，使用新的 'port-security-enabled' 参数，可以在每个端口上启用或禁用安全组功能。项目管理员可以对防火墙在网络拓扑中的位置进行“颗粒控制（granular control）”。

L3 高可用性的改进

查看 HA 路由的状态 - 现在，管理员可以查看每个节点上的高可用性路由的状态（当前活跃的实例运行在哪个节点）。这个新功能可以用来确认，一个路由只在一个节点上处于活跃状态。

支持外部网络的多子网 - 现在，HA 路由可以为所有连接的子网分配浮动 IP 地址。

LBaaS v2 API

LBaaS 版本 2.0 增加了负载均衡部署的稳定性，它包括了对 SSL/TLS 中断的支持。这个版本包括了对 LBaaS 架构和 HAProxy 参考插件的重新设计。

技术预览 - VLANs 和 VXLAN/GRE 的 DVR 集成

Red Hat Enterprise Linux OpenStack Platform 7.0 (kilo) 添加了在使用分布式路由时，对 VLAN 和 VXLAN/GRE 之间连接的支持。现在，DVR 中的 VLANs 和 VXLAN/GRE 通道间可以进行相互连接。

IPv6 支持

在 RHEL OpenStack Platform 7 中，核心的 OpenStack 服务可以在 IPv6 网络中进行。当前，RHEL OpenStack Platform director 还无法通过基于 IPv6 的网络进行部署和管理。

2.9. 技术预览

分布式虚拟路由、DNS-as-a-Service 和 erasure coding 作为技术预览包括在 Red Hat Enterprise Linux OpenStack Platform 7 中。



注意

关于技术预览的支持范围的更多信息，请参考 [Technology Preview Features Support Scope](#)。

Database-as-a-Service

使用 OpenStack 的 Database-as-a-Service，用户可以方便地在 OpenStack Compute 实例中部署单租户数据库。Database-as-a-Service 的机制可以为用户减少实施、使用、管理、监控和扩展数据库时所需的管理消耗。

分布式虚拟路由

分布式虚拟路由（Distributed Virtual Routing，简称 DVR）允许用户在 Compute 节点上直接放置 L3 路由。这可以在不首先通过一个网络节点路由的情况下，在 Compute 节点间（East-West）进行网络通信。没有浮动 IP 地址的实例仍然需要通过 Network 节点路由 SNAT 网络数据。

DNS-as-a-Service (DNSaaS)

Red Hat Enterprise Linux OpenStack Platform 7 包括了 DNS-as-a-Service (DNSaaS，也被称为 Designate) 作为一个技术预览。DNSaaS 包括了一个域和记录管理的 REST API，它是多租户的，并与 OpenStack Identity Service (keystone) 集成来进行用户验证。DNSaaS 提供了一个和 Compute (nova) 以及 OpenStack Networking (neutron) 的事件通知进行集成的机制，从而可以自动产生 DNS 记录。此外，DNSaaS 还包括了 PowerDNS 和 Bind9 集成的支持功能。

Erasure Coding (EC)

作为一个技术预览，Object Storage 服务现在为带有大量数据但不会被经常访问的设备提供了一个 EC 存储策略类型。EC 存储策略使用自己的 ring 和可配置的参数集来管理数据的可用性，同时减少相关的成本和存储需求（只需要 triple-replication 容量的一半）。因为 EC 需要更多的 CPU 和网络资源，所以把 EC 作为一个策略实现可以把与集群的 EC 容量相关联的所有存储设备进行隔离。

File Share Service

OpenStack File Share Service 提供了一个在 OpenStack 中简单地部署和管理共享文件系统的方法。这些共享的文件系统可以被实例安全地使用（挂载）。通过 File Share Service，对部署的共享进行管理的任务（如设置配额、配置访问权限、创建快照以及执行其它管理任务）变得更加稳定。

OpenStack File Share Service 基于上游社区的 [Manila](#) 项目。如需了解更多关于实施和测试 OpenStack File Share Service 的信息，请参阅 <https://access.redhat.com/articles/1555463>。

Firewall-as-a-Service

Firewall-as-a-Service (FWaaS) 插件为 OpenStack Networking (neutron) 添加了边界防火墙（perimeter firewall）管理功能。FWaaS 使用 iptables 在一个项目的所有虚拟路由上应用防火墙规则，并支持在一个项目中使用一个防火墙策略和逻辑防火墙实例。FWaaS 在网络边界进行操作，它会对 OpenStack Networking (neutron) 的路由进行过滤。这一点和安全组有所不同，安全组在实例

一级进行操作。

操作工具

现在提供了可以用来进行故障排除的新的日志和监控工具程序。通过一个中央化的、简单易用的分析和搜索 dashboard，故障排除任务变得简单，并增加了可用服务检查、阈值警告管理、图形化数据收集和表示的新功能。

第 3 章 发行信息

本发行注记包括了在部署 Red Hat OpenStack 时需要考虑的信息，如技术预览项、推荐的最佳方案、已知问题、过时的功能等。

在本 Red Hat OpenStack 发行版本的产品支持周期内，每个更新版本的备注都会包括在相应的公告或 *Red Hat Enterprise Linux OpenStack Platform Technical Notes* 中。相应文档包括在以下网页中：

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform

3.1. 改进

这个 Red Hat Enterprise Linux OpenStack Platform 发行版本包括了以下改进：

BZ#1228419

`director` 提供了一个把不同的网络服务在独立的子网和 VLAN 中进行分离的方法。相关的信息包括在 Red Hat Enterprise Linux OpenStack Platform 7.0 Director 安装和使用指南中。

BZ#1230450

这个发行版本为管理默认 `overcloud neutron` 租户网络行为提供了 4 个参数：`--neutron-network-type`、`--neutron-tunnel-types`、`--neutron-disable-tunneling` 和 `--neutron-network-vlan-ranges`。它的默认设置是为 `overcloud` 租户网络提供 GRE 类型和通道 (`tunnel`)。

如果把 `--neutron-network-type` 设置为 `'vlan'`，您将可以通过设置 `--neutron-network-vlan-ranges` 参数来管理分配给 `overcloud Neutron` 租户网络的 VLAN ID 的范围（默认值是 `"datacenter:1:1000"`，其中的 `"datacenter"` 是 VLAN 要被分配到的 `Neutron` 物理网络的名称）。如果把网络类型只设置为 `'vlan'`，`--neutron-disable-tunneling` 需要被同时指定（参见以下示例）。

如果把 `--neutron-network-type` 设置为 `'gre'` 或 `'vxlan'`（或两个都被设置），则相关的 `--neutron-tunnel-types` 参数也需要被启用（参见以下示例）。另外，如果把网络类型设置为 `'gre'` 或 `'vxlan'`，您还可以通过设置相关的 `--neutron-tunnel-id-ranges` 和 `--neutron-vni-ranges` 参数来分别管理 GRE 和 VXLAN 的通道 ID。

示例：

* 为 `overcloud` 租户网络指定 `'vxlan'` 类型：

```
openstack overcloud deploy --plan overcloud --control-scale 3 --
compute-scale 2 --neutron-network-type "vxlan" --neutron-tunnel-
types "vxlan"
```

* 为 `overcloud` 租户网络指定 `'gre'` 和 `'vxlan'` 类型：

```
openstack overcloud deploy --plan overcloud --control-scale 3 --
compute-scale 2 --neutron-network-type "gre,vxlan" --neutron-tunnel-
types "gre,vxlan"
```

* 为 `overcloud` 租户网络指定 `'vlan'` 类型，并同时设置 VLAN 的范围：

```
openstack overcloud deploy --plan overcloud --control-scale 3 --
compute-scale 2 --neutron-network-type "vlan" --neutron-disable-
tunneling --neutron-network-vlan-ranges "datacenter:3:3000"
```

如果在部署时没有指定这些参数，它们会被设置为以下的默认值。这些默认值可能会不适用于

具体的部署环境：

```
--neutron-network-type # default 'gre'
--neutron-tunnel-types # default 'gre'
--neutron-disable-tunneling # defaults to tunneling being enabled
--neutron-network-vlan-ranges # 'datacenter:1:1000'
```

在部署完成后，您可以检查控制器节点（controller node）的 Neutron 配置日志来确定以下是否已被配置：

```
grep -rni 'tunnel_types\|network_type\|enable_tunneling\|vlan_ranges'
/etc/neutron/*
/etc/neutron/plugin.ini:14:tenant_network_types = vlan
/etc/neutron/plugin.ini:72:network_vlan_ranges =datacenter:3:3000
/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini:54:enable_tun
neling=False
/etc/neutron/plugins/ml2/ml2_conf.ini:14:tenant_network_types = vlan
/etc/neutron/plugins/ml2/ml2_conf.ini:72:network_vlan_ranges
=datacenter:3:3000
```

另外，您可以在 overcloud Neutron 上创建一个网络并检查它。例如，在 undercloud 上：

```
$ source overcloudrc
$ neutron net-create "default"
$ neutron net-show default
...
| provider:network_type      | vlan
|
| provider:physical_network | datacenter
|
| provider:segmentation_id  | 3
|
...
```

BZ#[1232439](#)

在这个发行版本中，增加了可以为独立的网络分配特定 IP 地址范围的功能。

现在，可以通过在 Overcloud Orchestration 模板中设置以下参数来指定 IP 地址分配：

```
* InternalApiAllocationPools
* StorageAllocationPools
* StorageMgmtAllocationPools
* TenantAllocationPools
* ExternalAllocationPools
```

BZ#[1232461](#)

在这个发行版本中，您可以配置 Compute VNC 代理网络。通过这个功能，您可以分离不同的网络，并可以指定哪个服务需要在哪里运行。

现在，通过设置 Overcloud Orchestration 模板中的 ServiceNetMap -> NovaVncProxyNetwork 参数，可以指定哪个网络可以作为 Compute VNC 代理使用。

BZ#1232485

在这个发行版本中，Heat 模板增加了 VLAN 标识符和 OVS 绑定选项，这可以减少重复的配置。Heat 模板现在包括了以下参数：BondInterfaceOvsOptions、StorageNetworkVlanID、StorageMgmtNetworkVlanID、InternalApiNetworkVlanID、TenantNetworkVlanID 和 ExternalNetworkVlanID。

BZ#1237145

在这个发行版本中，为 Block Storage 服务增加了 NFS 后端，从而使它可以提供更多的 Block Storage 后端选择。

现在，Overcloud Block Storage 服务可以被配置为带有以下参数的 NFS 后端：

- * CinderEnableNfsBackend
- * CinderNfsMountOptions
- * CinderNfsServers

BZ#1237235

OpenStack Dashboard (Horizon) 现在成为了高可用性架构的一部分。现在，Dashboard 是一个由 Pacemaker 管理的资源。

BZ#1240631

在这个发行版本中，可以设置 `neutron_tunnel_id_ranges` 和 `neutron_vni_ranges` 参数来分别管理 GRE 和 VXLAN 的通道 ID。通过 overcloud Neutron，它们可以应用到 overcloud 租户网络。例如：

```
# openstack overcloud deploy --plan overcloud --control-scale 3 --
compute-scale 1 --neutron-tunnel-id-ranges "1:1111,2:2222" --
neutron-vni-ranges "3:33,5:55,100:999" --neutron-tunnel-types
"gre,vxlan"
```

如果没有指定，`tunnel_id_ranges` 和 `neutron_vni_ranges` 的默认值都会是 "1:1000"，这可能不适用于特定的部署环境。

如果使用了以上示例中的设置，您可以在部署完成后，在控制器节点上的相关 neutron 配置文件中进行检查和验证，例如：

```
# grep -rni 'vni_ranges\|id_ranges\|tunnel_types' /etc/neutron/*
/etc/neutron/plugin.ini:78:tunnel_id_ranges =1:1111,2:2222
/etc/neutron/plugin.ini:85:vni_ranges =3:33,5:55,100:999
/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini:77:tunnel_types
=gre,vxlan
/etc/neutron/plugins/ml2/ml2_conf.ini:78:tunnel_id_ranges
=1:1111,2:2222
/etc/neutron/plugins/ml2/ml2_conf.ini:85:vni_ranges
=3:33,5:55,100:999
```

您也可以在部署完成后，通过创建一个 overcloud 网络进行测试，例如：

```
# source overcloudrc # from the undercloud box for example
# neutron net-create --provider:network_type "vxlan" "foo"
```

这会创建一个类型为 vxlan 的网络。您可以检查它是否从指定的 vni_ranges 中接收到一个分段 ID：

```
# neutron net-show foo
| provider:network_type      | vxlan
| provider:segmentation_id   | 3
```

同样，您也可以检查 GRE 网络是否被分配了适当的分段 ID。

BZ#[1242967](#)

除了使用数据库中的计划，现在 director 还可以使用 Heat 模板集合来创建一个 Overcloud。这个功能提供了一个在运行 "openstack overcloud stack update" 命令来更新 Overcloud 中的软件包时直接使用 Heat 模板的方法。如果您使用模板集合而不是计划创建 Overcloud，在运行 "openstack overcloud stack update" 命令时使用 "--templates" 参数替代 "--plan" 参数。

BZ#[1242973](#)

现在，director 提供了一个参数来在更新 Overcloud 节点上的软件包时接收格外的 Heat 环境文件。Overcloud 分离的网络在创建以及以后的更新过程中需要额外的环境文件，如果没有这些环境文件，用户将无法在带有分离网络的 Overcloud 中使用 "openstack overcloud update stack" 命令。现在，用户可以在 "openstack overcloud update stack" 命令中使用 "-e" 参数来解决这个问题。

BZ#[1242989](#)

在这个发行版本中，用户可以在删除 overcloud 节点时直接使用 Heat 模板。现在，如果在部署 overcloud 时没有使用 Tuskar，则可以在运行 "openstack overcloud node delete" 命令时使用 "--templates" 参数。

BZ#[1242990](#)

在这个发行版本中增加了一个新功能 - 在从 overcloud 中删除一个节点时可以接受额外的 Heat 环境文件。当在一个分离的网络中进行 overcloud 更新时，额外的环境文件需要被传递给 Heat。现在，"openstack overcloud node delete" 命令可以被用来在一个分离的网络中删除 overcloud 节点。

3.2. 发行注记

本节介绍了与这个发行版本相关的重要信息，包括推荐的实用方法以及 Red Hat Enterprise Linux OpenStack Platform 的显著改变。您在进行部署时需要对这些方面加以考虑。

BZ#[1203160](#)

当把 Red Hat Enterprise Linux OpenStack Platform 从版本 6 完全升级到版本 7 后（所有节点都运行版本 7 的代码），您需要启动一个后台的 PCI 设备 NUMA 节点信息迁移操作，把这些信息从旧的位置迁移到新的位置。版本 7 的 conductor 节点会在需要的时候自动进行这个操作，但是其它处于空闲状态的数据需要在后台进行迁移。这个操作需要在版本 8 发行前完成，因为版本 8 将不再支持旧的位置。使用 'nova-manage migrate-rhos-6-pci-device-data' 来进行这个操作。
请注意，这只适用于使用 Compute 的 PCI pass-through 功能的用户。

BZ#1229372

DHCP 提供的路由信息和静态路由信息都出现在路由表中，但是 DHCP 路由的 metric 值会是 100。这意味着 metric 值为 0 的静态路由一直会被使用。如果需要使用多个 DHCP 连接，您可以把其中的一个或多个配置为忽略 DHCP 服务器提供的路由信息（使用 "defroute: false" 声明）。

BZ#1230966

Redis 需要使用一个独立的 VIP。在部署带有网络隔离的 Redis 时，director 会默认自动把 Redis VIP 放置在 Internal API VIP。操作者无法使用 ServiceNetMap 参数把 Redis 移到其它网络中。

BZ#1233916

Overcloud 节点会包括不正确的同步系统时间，这会在 HA Controller 集群中产生一些错误。一个临时解决办法是，在运行 "openstack overcloud deploy" 命令时使用 --ntp-server 命令行参数。这个参数会在每个 Overcloud 节点的 /etc/ntp.conf 文件中配置 ntp 服务器，从而保证了正确的同步系统时间，并使 Overcloud 可以被成功部署。

BZ#1238217

没有 CLI 参数可以被用来设置 NeutronExternalNetworkBridge，这会在分配浮动 IP 时出现问题。当前设置这个参数的唯一方法是通过网络隔离的自定义环境文件进行。例如：

```
parameters:
  # Set to "br-ex" if External is on native VLAN
  Controller-1::NeutronExternalNetworkBridge: ""

parameter_defaults:
  # Set to "br-ex" if External is on native VLAN
  NeutronExternalNetworkBridge: ""
```

如果浮动 IP 网络在一个 VLAN 上，把两个参数都设为 ''；如果在一个 br-ex 网桥的 native VLAN 上，则把它们都设置为 'br-ex'。这个配置将使 Neutron 桥接映射在环境中可以正常工作。相关信息包括在 Red Hat Enterprise Linux OpenStack Platform 7 Director 安装和使用指南中。

BZ#1240824

到数据库的连接数量会受到 Controller 的数量，以及每个 Controller 所具有的内核的限制。在一个有 3 个 controller，每个 controller 有多于 12 个内核的 HA 环境中，数据库的连接数量可能会达到 max_connections 的默认值 (1024)，这可能会出现对服务的请求没有响应的问题。作为一个临时的解决办法，您可以使用以下命令增加 max_connections 的值：

```
$ openstack management plan set [tuskar_plan_uuid] -P "Controller-1::MysqlMaxConnections=4096"
```

使用实际的计划 UUID 替换 [tuskar_plan_uuid]，计划的 UUID 可以使用以下命令找到：

```
$ openstack management plan list
```

要在部署时使用 `--templates` 参数来增加 `max_connections` 的值，为部署命令提供包括以下内容的额外自定义环境文件

```
parameters:
  MySQLMaxConnections: 4096
```

把它添加到 `deploy` 命令中：

```
$ openstack management deploy --plan overcloud -e
/path/to/custom_environment_file.yaml
```

BZ#[1241610](#)

在部署一个 Overcloud 时，Tuskar 会修改 Heat 套件 (Heat stack) 的顶级参数的名称。这会在 Heat 对套件进行验证时出现一个错误：

```
ERROR: The Parameter (NeutronExternalNetworkBridge) was not defined
in template.
```

这个问题的一个临时解决办法是，使用 `"tuskar plan-update"` 修改参数；或在环境文件中使用修改过的参数名：

```
parameters:
  Controller-1::NeutronExternalNetworkBridge: ""
```

Overcloud 部署会使用正确的参数值。

请注意：参数需要在 `"parameters:"` 项而不是 `"parameter_defaults:"` 项中定义。否则，Tuskar 的导出环境 `environment.yaml` 中设置的值会覆盖这个值。

3.3. 已知问题

BZ#[1250043](#)

当 File Share Service 后端使用 `'gluster_native'` 驱动时，如果一个共享对应的 Red Hat Gluster Storage 共享卷不支持快照功能，则不要创建这个共享的快照。这样做会导致操作失败，`'openstack-manila-share'` 服务会产生一个带有 `KeyError` 的回溯，而不会产生有用的错误信息。

如需了解创建支持快照功能的 Red Hat Gluster Storage 共享卷（可以避免以上问题）的信息，请参阅 <https://access.redhat.com/articles/1555463>。

BZ#[1250130](#)

`'manila list'` 命令显示 File Share 服务中的所有共享的信息。这个命令同时显示每个共享的 Export Location 项，它包括了在一个实例中组成挂载点项的信息。但是，这个项的信息以下面的格式被显示：

```
user@host:/vol
```

其中的 `'user@'` 前缀是不需要的，在组成挂载点项时应该被忽略。

3.4. 过时的功能

以下所列出的功能已不再被支持，或不会在以后的版本中被支持。

BZ#[1210479](#)

在实施过程中不再使用 `deploy-baremetal-overcloudrc` 文件。现在，使用 `"openstack overcloud deploy"` 命令以及相关的命令行参数。

BZ#[1227940](#)

在 `instack.answers` 中为 Undercloud 设置 `deployment_mode` 配置的功能以前是一个技术预览 (tech preview)，现在这个功能已被删除。`deployment_mode` 允许选择一个 `"scale"` 模式来把不同的角色部署到特定节点中。这个功能现在被节点标签 (node tagging) 功能所替代，它是 Automated Health Check (AHC) 工具的一部分。

BZ#[1233201](#)

在这个发行版本中，删除了一个重复的、用于扩展部署的命令。现在，可以使用 `Tuskar` 或 `部署命令` 来直接扩展部署。

第 4 章 技术备注

本章的内容是对 Red Hat Enterprise Linux OpenStack Platform "Kilo" errata advisories（通过 Content Delivery Network 获得）内容的补充。

4.1. RHEA-2015:1548 — Red Hat Enterprise Linux OpenStack Platform 功能增强公告

本节中包括的内容由公告 RHEA-2015:1548 提供。如需了解更多与本公告相关的信息，请参阅 <https://access.redhat.com/errata/RHEA-2015:1548.html>。

4.1.1. crudini

BZ#[1223624](#)

在这个版本以前，更新配置文件时独立的锁定文件会被使用。另外，命令项在更新时无法被正确同步。

因此，如果在进行处理的过程中出现故障，将可能会导致在以后的配置更新操作中出现死锁问题；或在一些特别的情况下，会出现被破坏（空）的配置文件。

这个版本在 'crudini' 工具程序中添加了更加稳定的锁定和同步机制，从而解决了上面提到的、在系统出现故障时会导致的问题。

4.1.2. mariadb-galera

BZ#[1211088](#)

这个 rebase 软件包包括了版本 5.5.42 的一个显著的错误修正：

- * 以下错误已被解决：对于一个设置了自动增加的主键（auto-incrementing primary key）的表，当一个正常的 Galera 节点使用 INSERT 命令时，另外一个当前已被移出集群的 Galera 也在使用同一个表中的 INSERT 命令时，会产生一个 "DUPLICATE PRIMARY KEY" 错误。这会导致，在一个 Galera 故障转移正在进行时，OpenStack 应用无法创建新数据库记录的问题。

4.1.3. openstack-ceilometer

BZ#[1232163](#)

在以前的 'alarm-history' 版本中，没有包括提示（alarm）的严重性在什么时间被改变（如从 'low' 变为 'critical'）的信息。它所包括的信息只显示了一个改变已发生，而没有其它任何详细信息。

在这个版本中包括了显示严重性改变详情的代码。

'alarm-history' 现在会在 alarm-history 的输出中包括严重性改变的详细信息。

BZ#[1240532](#)

在以前的版本中，当一个 ceilometer 轮询扩展（polling extension）无法被加载时，会在日志中记录一个 ERROR 信息。因为在某些情况下，加载模块失败是所期望的结果（如一个扩展模块是可选的，或它的依赖模块不可用），所以这个错项信息在某些时候会产生误导。

现在，这个事件的日志信息级别已改为 WARN，从而表明它不是一个严重的错误。

4.1.4. openstack-cinder

BZ#1133175

在这个版本中，增加了对 NetApp Cmode 和 7mode iSCSI 驱动的扩展卷管理和非管理的支持。这为使用这些驱动提供了新的功能。

BZ#1133177

在这个版本中，增加了对 NetApp e-series 驱动的管理/非管理卷的支持。您可以使用 '--source-name' 参数指定没有由 Block Storage 管理机制管理的卷的输入。

BZ#1156682

这个版本为 cinder-backup 服务增加了 NFS 后端。现在，可以把卷备份到一个 NFS 存储后端。

BZ#1159142

这个版本为 'cinder-manage db' 增加了一个功能：现在可以安全地从 Cinder 数据库中永久删除 (purge) 老的、已被删除的 ("deleted") 数据。这将可以减少对数据库存储空间的使用，并可以增加虚拟机的性能。

BZ#1200986

在以前的版本中，SQLAlchemy 对象会错误地被多个 'cinder-volume' 进程共享。因此，在使用多后端 Block Storage 时，SQLAlchemy 连接会失败。这会在卷服务中导致与数据库相关的错误。现在，在 fork 'cinder-volume' 子进程时，SQLAlchemy 连接会被重新初始化。因此，多后端可以正常运行。

BZ#1208767

在以前的版本中，从一个镜像创建卷会失败。在一个带有大量扇区的虚拟磁盘中，扇区数量会在某些情况下被错误处理，造成转换 QEMU 镜像失败 ("invalid argument" 错误)。

现在，一个更新的 QEMU-img 版本解决了这个问题。从镜像创建卷现在可以正常完成。

4.1.5. openstack-glance

BZ#1118578

现在，Image Service 的日志功能进行了改进，它可以为用户提供更好的信息。另外，敏感的信息也被从日志中删除，日志信息的级别设置也更准确。这个改变只对操作者可见。

BZ#1151300

在这个版本中，可以通过对 'glance-*' 进程发送 SIGHUP 信号来动态地重新加载 Image 服务的配置信息。这个信号会要求进程重新读取配置文件并加载新的配置信息。因此，现在不再需要重启整个的 Image 服务来引用配置改变。

BZ#1155388

在这个版本中，底层的异步任务引擎已改变。现在，它基于 `taskflow` 库。虽然这没有改变 API 或工作流程，但它增加了以下的配置选项：

```
[taskflow_executor]
engine_mode = serial # or parallel
```

BZ#[1164520](#)

以前，`glance-manage` 工具程序是通过 '`glance-api.conf`' 或 '`glance-registry.conf`' 进行配置的。在这个版本中包括了一个名为 '`glance-manage.conf`' 的新配置文件，使用它可以配置 `glance-manage`。您仍然可以使用 '`glance-api.conf`' 或 '`glance-registry.conf`' 来配置 `glance-manage`，但 '`glance-manage.conf`' 中的设置会被优先使用。

BZ#[1168371](#)

在以前的版本中，Image 服务的 '`swift`' 存储会在一个单一的容器中存储所有镜像。虽然这个方法可以正常工作，但是会在大型部署中产生一个性能瓶颈。

在这个版本中，可以使用多个 Object Storage 容器来存储 '`glance`' 镜像。要使用这个功能，您需要把 '`swift_store_multiple_containers_seed`' 设为大于 '`0`' 的值。您可以通过启用 '`swift_user_multi_tenant`' 参数来禁止使用多容器，这会使这些容器基于每个租户进行分离。

BZ#[1170475](#)

在这个版本中，`glance_store` 库支持更多的存储功能。您可以对在特定 store 中可以进行什么操作进行更精细的控制。这个版本包括了以下访问控制：

- `READ_ACCESS`：一般的读访问
- `WRITE_ACCESS`：一般的写访问
- `RW_ACCESS`：`READ_ACCESS` 和 `WRITE_ACCESS`
- `READ_OFFSET`：从一个 `offset` 中读所有位（包括 `READ_ACCESS`）
- `WRITE_OFFSET`：对一个 `offset` 写所有位（包括 `WRITE_ACCESS`）
- `RW_OFFSET`：`READ_OFFSET` 和 `WRITE_OFFSET`
- `READ_CHUNK`：读所需的位长度（包括 `READ_ACCESS`）
- `WRITE_CHUNK`：写所需的位长度（包括 `WRITE_ACCESS`）
- `RW_CHUNK`：`READ_CHUNK` 和 `WRITE_CHUNK`
- `READ_RANDOM`：`READ_OFFSET` 和 `READ_CHUNK`
- `WRITE_RANDOM`：`WRITE_OFFSET` 和 `WRITE_CHUNK`
- `RW_RANDOM`：`RW_OFFSET` 和 `RW_CHUNK`
- `DRIVER_REUSABLE`：驱动是无状态的，它的实例可以被安全地重新使用

BZ#[1170476](#)

在这个版本中，添加了一个全新的 Image 服务搜索功能 API，它提高了列出和搜索操作的性能，特别是现在可以和 UI 进行交流。

用户可以使用搜索 API 进行搜索查询，并返回查询的结果。查询的条件可以是作为参数的一个简单查询字符串，也可以是一个请求的内容（body）。所有的查询 API 可以应用于一个索引中的多个类型，也可以应用于支持多索引语法的多索引。

请注意：在 RHEL OpenStack Platform 8 (Liberty) 发行版本中，这个功能会从 Image 服务中删除。

BZ#1189811

在以前的版本中，每个到 `policy.enforce` 的调用都会传递一个空的字典（dictionary）作为目标。因为目标总是一个空字典，操作者将无法使用 `policy.json` 文件中的、特定租户的限制规则。如果您试图限制一些操作来使镜像的所有者（带有正确租户 ID 的用户）可以执行操作，检查的过程会失败。这是因为目标是正确的，但它是一个空的字典。

在这个版本中，您可以把带有镜像的 `ImageTargeton` 实例传递给 `enforcer`，从而使这些规则可以被使用并被强制执行。您可以基于租户（例如，`owner: %(tenant)`）来为镜像的所有者正确地分配权限。如果没有这个前缀，`Image` 服务中唯一可以正常工作的检查是 `RoleCheck`（例如，`role: admin`）。

BZ#1198911

在这个版本中，可以对列出操作的结果根据多个过滤选项进行过滤，并以不同的顺序排序。例如：

```
/images?sort=status:asc,name:asc,created_at:desc
```

以上会返回一系列镜像，它们会根据状态、名称和创建时间进行排序（排序顺序分别是升序、升序和降序）。

BZ#1201116

在这个版本中，可以对列出操作的结果根据多个过滤选项进行过滤，并以不同的顺序排序。例如：

```
/images?sort=status:asc,name:asc,created_at:desc
```

以上会返回一系列镜像，它们会根据状态、名称和创建时间进行排序（排序顺序分别是升序、升序和降序）。

4.1.6. openstack-heat**BZ#1042222**

Orchestration 服务现在包括了一个 `"OS::Heat::Stack"` 资源类型。这个原生的 OpenStack 资源被用来在一个模板中明确地创建一个子堆栈。`"OS::Heat::Stack"` 资源类型包括了带有一个 `'region_name'` 子属性的 `'context'` 属性，允许 Orchestration 服务在不同的区域中管理堆栈。

BZ#1053078

现在，当它们的规则被改变后，`AWS::EC2::SecurityGroup` 类型资源可以被更新。这和 CloudFormation 中的 `AWS::EC2::SecurityGroup` 相一致。以前，当它们的规则被改变后，安全组会被替代。

BZ#1108981

现在，Heat 支持 hook，用户可以使用它在特定点上暂停堆栈操作，把自己的操作加入到 Heat 的流程中。hook 通过堆栈的环境文件附加到资源中，当前支持的 hook 类型是 `'pre-create'` 和 `'pre-update'`。

BZ#1122774

现在, `OS::Nova::Server` 资源类型包括了一个 `'console_urls'` 属性。这可以使用户从资源中获得服务器的控制台（如 VNC 控制台）的 URL。

BZ#1142563

现在, 当使用 `Orchestration` API 对资源进行查询时, 用户可以要求在输出中包括资源的一个或多个属性值。通过这个功能, 用户在不修改堆栈模板的情况下就可以随时在输出中获得任何资源的数据, 这将有助于对系统进行故障排除。

BZ#1143805

现在, `OS::Cinder::Volume` 资源类型包括了一个 `'scheduler_hints'` 属性, 使用它, 可以在创建卷时把 `scheduler hint` 传递给 `Block Storage` 服务。这需要 `Block Storage` API 的版本是 `v2`。

BZ#1144230

现在, `heat-manage` 命令包括了一个 `"heat-manage service-list"` 子命令。这个子命令会显示活动的 `"heat-engine"` 进程的信息, 如它们在哪里运行、它们当前的状态。

BZ#1149959

现在, `OS::Neutron::Port` 资源类型支持一个 `'binding:vnic_type'` 属性, 具有适当权限的用户可以使用它来为 `OpenStack Networking` 指定 `VNIC` 类型。

BZ#1156671

现在, `AWS::AutoScaling::AutoScalingGroup` 资源类型支持一个 `'InstanceId'` 属性。使用它, 可以使一个 `autoscaling` 组的启动配置通过克隆一个存在的服务器获得, 而不是克隆一个 `AWS::AutoScaling::LaunchConfiguration` 资源。

BZ#1159598

现在, `AWS::AutoScaling::LaunchConfiguration` 资源类型支持一个 `'InstanceId'` 属性。使用它, 可以使一个 `autoscaling` 组的启动配置通过克隆一个存在的服务器获得。

BZ#1212625

以前, 在堆栈更新时, 如果一个环境的 `'files'` 项的内容发生了变化, `Orchestration` 服务会通过综合新文件和旧的堆栈定义来决定当前的状态。这么做的目的是把以前的状态和新文件以及新模板进行比较。

作为结果, `Orchestration` 服务没有发现所包括的文件的变化, 那些因文件变化而需要进行的更新将不会发生。另外, 如果在堆栈更新过程中以前指定的文件被从环境中删除, 堆栈更新会失败（尽管以后使用相同数据进行的更新将会成功）。

在这个版本中, `Orchestration` 服务使用带有旧文件的旧堆栈与新模板和新文件进行比较。当在更新过程中编辑了在环境中包括的文件, 更新操作仍然可以正常工作。

BZ#1218692

在以前的版本中，Orchestration 服务无法识别对一个模板资源（如基于堆栈但没有被明确指定的资源）中的模板绝对路径的改变。这会导致一个问题：当资源模板被重命名，或被删除时，基于嵌套堆栈的模板资源无法被更新。

在这个版本中，Orchestration 服务可以发现这种变化，从而使嵌套的堆栈可以被更新。

4.1.7. openstack-ironic

BZ#1151691

现在，Bare Metal 可以通过使用 `iLO client python` 库来支持 HP ProLiant Services 的管理接口，这将允许 Bare Metal 执行管理操作（如获取/设置一个启动设备）。

BZ#1153875

现在，Bare Metal 服务可以使用 `cloud-init` 和相似的早期初始化工具在实例中插入用户数据。在以前，需要设置一个元数据服务来实现这个功能。

在这个版本中，Bare Metal 可以在部署过程中在本地磁盘（被标识为 `'config-2'` 的设备）上插入实例元数据。然后，再配置早期初始化工具来找到这个设备，并从中获取相关数据。

BZ#1154485

现在，Bare Metal 服务可以使用 UEFI (<http://www.uefi.org>) 的 Secure Boot 功能部署节点。Secure Boot 可以保证节点只引导信任的软件。

因此，整个引导链可以在引导时被验证。然后，您可以配置节点来只引导经过授权的镜像，从而达到增强安全性的目的。

BZ#1154927

现在，Bare Metal 实例包括了一个名为 `'maintenance_reason'` 的新项，它可以被用来解释一个节点处于维护状态的原因。

BZ#1165499

现在，Bare Metal 服务支持 Fujitsu iRMC（集成的远程管理控制器）硬件。因此，Bare Metal 现在可以管理这种系统的电源状态。

BZ#1198904

现在，所有 Ironic 驱动都支持通过 IPA ramdisk 进行部署。IPA 是使用 Python 语言开发的，它比 BASH ramdisk 支持更多的功能，并作为一个服务运行。因此，在一般情况下，通过 IPA 部署的节点更易进行部署、故障排除和管理。

BZ#1230142

以前，DRAC 卡中的 WSMAN 接口可以在 11g 硬件和 12g 硬件间改变。因此，当在 11g 硬件上使用 DRAC 驱动时，``get_boot_device`` 和 ``set_boot_device`` 调用在 OpenStack Bare Metal Provisioning (Ironic) 上会失败。

在这个版本中，DRAC 启动会检查 Lifecycle 控制器的版本，并根据版本的不同使用相应的方法来管理引导设备。
因此，`get_boot_device` 和 `set_boot_device` 操作可以在 11g 节点上正常工作。

BZ#1230163

在任何时候，Compute 服务都应该可以删除一个实例。但是，Bare Metal 实例只能在特定阶段（处于 'DEPLOYWAIT' 状态）被停止。因此，当 Compute 服务试图删除一个没有处于 DEPLOYWAIT 状态的 Bare Metal 实例时，操作会失败。实例可能会被固定在特定的状态，并需要对数据库进行修改才可以解决这个问题。

在这个版本中，虽然 Bare Metal 服务在没有处于 DEPLOYWAIT 状态时仍然无法终止实例，但 Bare Metal 实例不再会被固定在特定状态。

BZ#1231327

以前，OpenStack Bare Metal Provisioning (Ironic) 中的 DRAC 驱动会错误地把任务状态 'completed with errors' 作为 'in-progress' 状态。这会导致 `get_boot_device` 和 `set_boot_device` 操作失败，因为它们需要没有状态为 in-progress 的任务存在。
这个版本解决了这个问题。它把 'completed with errors' 加入到成功完成状态列表中。因此，`get_boot_device` 和 `set_boot_device` 可以在 DRAC 卡上具有状态为 'completed with errors' 的任务时，仍然正常工作。

BZ#1231331

以前，DRAC `vendor_passthru interface` 缺少了 `pass_bootloader_install_info` 这个方法。当本地引导被启用时，PXE 部署任务会失败。
在这个版本中，标准 PXE 接口中的 `pass_bootloader_install_info` 被添加到 `DRAC vendor_passthru` 中。当本地引导被启用时，部署可以成功。

BZ#1233452

在以前的版本中，OpenStack Bare Metal Provisioning (Ironic) 操作（如 'Power off'）会过长地锁定一个节点。
作为结果，一些操作会失败，因为节点仍然被认为处于锁定状态。
在这个版本中，重新尝试的超时时间被调整为 2 分钟。因此，不会出现节点锁定错误。

4.1.8. openstack-keystone

BZ#1110589

现在，Identity Service (keystone) 可以重新授权信任关系。具有信任权利的信任者 (trustee) 可以创建另外一个信任关系来把他们的角色授权给其它用户。另外，一个计时器会列出信任可以被重新授权的次数。
这个功能允许一个信任者把包括在令牌中的角色重新授权给其它信任者。初始创建信任的用户可以控制一个信任是否可以被重新授权。

BZ#1121844

现在，Identity Service (keystone) 允许明确请求无范围令牌 (unscoped

token)。

在以前的版本中，一个被分配了默认项目的用户无法获取一个无范围令牌。如果这个用户在请求令牌时没有指定范围，它的范围会被自动定为默认项目。

在这个版本中，所有用户（包括指定了默认项目的用户）都可以获得无范围令牌。

BZ#[1165505](#)

在这个版本中，Identity Service (keystone) 通过在项目资源中指定 'parent_id' 来创建一个项目的分级结构。

以前，Identity 服务只支持平面的项目模式，而分级结构的项目会有更大的灵活性，可以更好地与实际的组织结构相对应。

现在，项目可以定义一个父项目，这就实现了项目的分级结构。

BZ#[1189633](#)

现在，Identity 服务允许无范围联邦令牌 (unscoped federation token) 使用 'token' 验证方法获得一个有范围令牌 (scoped token)。

在使用 Identity 服务的 federation 扩展时，会返回一个无范围联邦令牌作为初始验证。然后，它可以被交换为一个有范围令牌。在以前的版本中，无范围联邦令牌需要使用 'saml2' 或 'mapped' 验证来获取一个有范围令牌。这和把普通的无范围令牌交换为有范围令牌的过程（使用 'token'）不一致。

现在，为无范围联邦令牌交换一个有范围令牌的过程也使用 'token' 验证方法，这和普通无范围令牌进行交换的方法保持了一致。

BZ#[1189639](#)

现在，Identity 服务对重新设定令牌范围的操作进行限制，它只允许无范围令牌被交换为有范围令牌。

Identity 服务允许一个存在的令牌使用 'token' 验证方法来获得一个新令牌。在以前的版本中，如果用户具有一个有效的、范围是某个项目的令牌，这个用户就可以使用这个令牌获得范围为另外一个项目的有效令牌。这将允许具有用户令牌的用户访问这个用户可以访问的任何项目，而不是只可以访问令牌范围内的项目。而从提高安全性的角度来看，应该不允许这个行为。

现在，添加了一个新的 'allow_rescope_scoped_token' 配置选项，使用它可以限制重新设定令牌范围的操作。只有在启用了这个选项时，使用无范围令牌进行验证的重设令牌范围的操作才被允许。

BZ#[1196013](#)

现在，Identity 服务对一个名为 'fernet' 的新令牌格式提供了试验性的支持。

Identity 服务当前所支持的令牌格式需要签发的令牌持久存在于数据库的表中，随着时间的推移，这个数据库表可能变得很大，这就需要对数据库进行相应的调整并执行删除任务来保证 Identity 服务的性能。新的 'fernet' 令牌格式被设计为可以不使用数据库表，从而避免了表数据过大的问题。'fernet' 令牌格式当前还是作为一个试验功能被支持。

4.1.9. openstack-neutron

BZ#[1108790](#)

在以前的版本中，当在一个 Open vSwitch (OVS) 代理上手工切换通道源 IP 地址时，其它代理会为这个代理保持开放两个通道：一个是它的旧 IP 地址，一个是它的新 IP 地址。作为结果，在运行 OVS 代理的云中的所有虚拟机监控程序 (hypervisor) 上会积累过多的元数据。

现在，为了解决这个问题，Network 节点会监测到主机上的 IP 地址发生改变的情况，记录下相应信息，并把 IP 地址的改变通知给其它代理。

BZ#[1152579](#)

在以前的版本中，当附加到一个 LBaaS 池中的子网被计划外删除时，OpenStack Dashboard LBaaS 池的详情页无法正确处理这个变化。

作为结果，如果您创建了一个网络、子网、路由和负载均衡设备 (load balancer)，然后删除了网络、子网和路由但保留了负载均衡设备，OpenStack Dashboard LBaaS 详情页会返回代码为 500 的错误。

在这个版本中解决了这个问题，如果出现以上情况，LBaaS 详情页会正确处理，并显示一个警告信息。

BZ#[1153446](#)

在这个版本中，管理员可以查看每个节点（运行活动实例）的高可用性路由的状态。

在以前的版本中，管理员无法查看高可用性路由的状态信息，这为维护（如把一个 HA 路由从一个代理移到另外一个代理上；或评估把一个节点设置为维护模式将会造成的影响）造成了困难。

这个新功能也被作为一个健全检查来保证路由确实只在一个节点上是活跃的。现在，管理员可以在高可用性路由上运行 'neutron l3-agent-list-hosting-router <router_id>' 命令来查看活跃的实例运行于哪个节点。

BZ#[1158729](#)

现在，带有分布式路由的 OpenStack Networking 部署允许租户创建他们自己的、带有 VLAN 段的网络。

在以前的版本中，分布式路由只支持通道网络 (tunnel network)，而许多部署倾向使用 VLAN 租户网络。

在这个版本中，分布式路由可以支持通道网络和 VLAN 网络。

BZ#[1213148](#)

Red Hat Enterprise Linux OpenStack Platform 7 使用 libreswan 替代了 openswan，但 OpenStack Networking (neutron) openswan VPNaaS 驱动无法和 libreswan 一起正常工作。

在这个版本中，您可以在 vpnagent.ini 中启用 libreswan 的驱动：

```
[vpnagent]
vpn_device_driver=neutron_vpnaas.services.vpn.device_drivers.libreswan_ipsec.LibreswanDrive
```

作为结果，VPNaaS 现在可以正常工作。

BZ#[1221034](#)

因为 'python-neutron-fwaas' 软件包中的一个已知问题（缺少了数据库升级的 'versions' 目录），防火墙即服务 (Firewall-as-a-Service, 简称 FWaaS) 可能无法正常工作。

另外，在不同的发行版本间升级数据库 schema 当前可能还无法正常工作。

BZ#1221076

因为 'python-neutron-fwaas' 软件包中的一个已知问题（缺少了数据库升级的 'versions' 目录），防火墙即服务（Firewall-as-a-Service，简称 FWaaS）可能无法正常工作。
另外，在不同的发行版本间升级数据库 schema 当前可能还无法正常工作。

BZ#1227633

在以前的版本中，dnsmasq 不会把租赁（lease）信息保存在具有持久性的存储设备中。当它被重启时，租赁信息会丢失。这是因为删除了 dnsmasq 的 '--dhcp-script' 选项所造成的（BZ#1202392）。
作为结果，实例会被长时间固定在网络引导过程中。另外，NACK 信息会被记录在 dnsmasq 日志中。
这个发行版本解决了以上问题，它删除了验证选项，在对 DHCPREQUEST 进行响应时不会把 NAKs 发送到其它服务器。通过这个改变，日志文件中将不再包括 DHCPNAK 信息。

BZ#1228096

在 Kilo 中，Neutron 服务会使用 rootwrap 守护进程执行外部命令（如 'ip' 或 'sysctl'）。守护进程会预先缓存 rootwrap 过滤器，从而可以极大地提高代理的整体性能。

对于 RHEL-OSP7，rootwrap 进程被默认启用。如果您不想使用它，而是使用其它 root 权限分离机制（如 'sudo'），则需要在 neutron.conf 文件的 [agent] 段中设置 'root_helper_daemon =' 来禁用这个守护进程。

4.1.10. openstack-neutron-lbaas**BZ#1228227**

在以前的版本中，'neutron-lbaasv2-agent' 服务缺少了 .service 文件。
因此，在 systemd 的控制下无法启动代理。
在这个版本中，添加了 .service 文件。
作为结果，'systemctl start neutron-lbaasv2-agent' 命令现在可以启动服务。

4.1.11. openstack-nova**BZ#1041068**

现在，您可以使用 VMware vSAN 数据存储。这些存储允许您为实例使用虚拟机监控程序本地存储的同时使用 vMotion。

BZ#1052804

现在，您可以使用 VMware 存储策略来管理如何把存储配置给不同的实例。使用这个功能，您可以在带有多个数据存储（data store）的 VMware 基础架构的环境中为实例分配最佳存储。

BZ#1085989

在以前的版本中，Compute 数据库少了一个对 virtual_interfaces 表的索引。因此，

当增加了大量对这个表的操作时，会需要大量时间来进行处理，从而会出现超时错误。

在这个版本中，加上了对 `virtual_interfaces` 表的索引，从而提高了对这个表进行操作的性能。

BZ#1193287

增加了对已经被分配了一个主机 PCI 设备的客户机 (guest) 的智能 NUMA 节点放置功能，现在，PCI I/O 设备 (如网卡) 可以更紧密地关联到某个处理器。因为访问直接附加到某个处理器上的内存会和访问直接附加到同一个服务器上的其它处理器上的内存有不同的性能和延迟特征，所以这个功能对提升系统性能有好处。在这个版本中，Openstack 客户机的放置操作可以被优化，从而可以保证绑定了特定 PCI 设备的客户机被调度为在与这个客户机的 pCPU 和内存分配相关联的 NUMA 节点上运行。例如，如果一个客户机的资源需要包括在一个单一 NUMA 节点中，则这个客户机的所有资源都将会和同一个 NUMA 节点相关联。

BZ#1203160

当把 Red Hat Enterprise Linux OpenStack Platform 从版本 6 完全升级到版本 7 后 (所有节点都运行版本 7 的代码)，您需要启动一个后台的 PCI 设备 NUMA 节点信息迁移操作，把这些信息从旧的位置迁移到新的位置。版本 7 的 conductor 节点会在需要的时候自动进行这个操作，但是其它处于空闲状态的数据需要在后台进行迁移。这个操作需要在版本 8 发行前完成，因为版本 8 将不再支持旧的位置。使用 `'nova-manage migrate-rhos-6-pci-device-data'` 来进行这个操作。请注意，这只适用于使用 Compute 的 PCI pass-through 功能的用户。

BZ#1226438

在以前的版本中，当试图在一个由 `staypuft/openstack-foreman-installer` 配置的 `nova-network` 计算节点上启动实例时，会出现一个错误。这是因为在安装程序 (installer) 中缺少了 `conntrack-tools` 软件包。

现在，在 `openstack-nova.spec` 中添加了相应内容来为 `nova-network` 的服务安装 `conntrack-tools` 软件包。Nova-network 现在可以配置网络，从而避免了以上错误的产生。

BZ#1228295

在以前的版本中，当到 Cinder iSCSI 卷的主路径失效时，这个卷将无法被附加到实例上，即使 Compute 和 Block Storage 后端的多路径功能已经被启动。这意味着，云系统用户无法附加一个卷，或无法从一个卷引导服务器。

在这个版本中，主机有一个单独的配置选择，它可以被用来指定块设备的流量是否由一个独立的网络处理，从而可以实现通过从路径 (secondary path) 附加卷。

BZ#1229655

当部署一个使用 IPv6 的 OpenStack 环境时，因为 `websocketproxy` 无法验证原始的头数据 (header)，VNC 控制台会加载失败并产生一个异常 - `"handler exception: Origin header does not match this host."`。

在这个版本中，`websocketproxy` 代码已被更新来处理 IPv6。现在，当所有服务都被配置为使用 IPv6 时，用户可以成功地连接到 VNC 控制台。

BZ#1230237

在以前的版本中，因为更新端口绑定失败，会导致在使用 neutron 时，撤离 (evacuate) nova 中的虚拟机操作失败。这个问题同样会出现在使用 FloatingIP 的 nova-network 中。因此，由于创建所需的虚拟接口失败，会导致虚拟机无法被撤离。

在这个版本中，nova 可以在以上所提到的两种网络设置中正确配置虚拟机。现在，用户可以成功进行虚拟机撤离操作。

BZ#[1230485](#)

在以前的版本中，libvirt 驱动使用 libguestfs 进行特定的虚拟机检查和修改任务。但是，libguestfs 是一个外部的程序库，它无法被 eventlet 的 monkey patch 更新。因此，在进行 libguestfs API 调用时，eventlet greenthreads 没有运行，从而导致 nova-compute 服务在调用的整个过程中都处于无响应的状态。在系统安装或更新后，对 libguestfs 的初始调用会持续几秒钟，在这个期间，openstack-nova-compute 将无法提供正常的服务。

在这个版本中，对 libguestfs 的调用会被放入到一个独立的、非 Eventlet 的 threadpool 中。这些调用会异步执行，从而不会影响到 openstack-nova-compute 的运行。

BZ#[1242502](#)

在以前的版本中使用了不正确数据版本，从而造成 PCI 设备数据模式以不正确的格式发送。这就使得当有任何 PCI-passthrough 设备处于白名单 (whitelist) 中时，openstack-nova-compute 服务无法启动。

在这个版本中，使用了正确的数据版本，从而允许 openstack-nova-compute 启动并注册白名单中的 PCI 设备。

4.1.12. openstack-packstack

BZ#[1185652](#)

这个功能增加了 Packstack 对 IPv6 的支持，Packstack 可以在网络相关的参数中（如 CONFIG_CONTROLLER_HOST、CONFIG_COMPUTE_HOSTS 和 CONFIG_NETWORK_HOSTS）使用 IPv6 地址。

4.1.13. openstack-puppet-modules

BZ#[1231918](#)

在以前的版本中，puppet-neutron 不允许自定义 neutron 的 dhcp_domain 设置。因此，undercloud DHCP 会向 overcloud 节点提供无效的域后缀。在这个版本中，可以对 neutron dhcp_domain 进行设置，默认是一个空的后缀。

BZ#[1236057](#)

在以前的版本中，Telemetry 服务的 HAProxy 配置使用不正确的检查（HAProxy 配置没有有效的检查，并错误地使用 SSL 检查而不是 TCP 检查），这会导致 Telemetry 服务在 HA 部署中失败。

在这个版本中，修正了检查错误，确保 Telemetry 服务可以在 HA 部署中被正确地平衡，并启动。

BZ#[1244358](#)

当在 undercloud 中部署启用了 SSL 的 Bare Metal 和 Telemetry 服务时，Director 会使用错误的 HAProxy 设置。这会导致一些节点无法进行注册。

为了解决这个问题，可以在安装 undercloud 后，在 /etc/haproxy/haproxy.cfg 文件的 Bare Metal 和 Telemetry 项中注释掉 'option ssl-hello-chk'。

4.1.14. openstack-sahara

BZ#[1149055](#)

现在，在 HDP 2.0.6 插件中添加了 namenode 高可用性作为一个支持的选项。用户可以通过传递一个带有法定数量的 zookeeper 服务器和 journalnodes，以及最少 2 个 namenode 的集群，来表明在 HA 模式中需要产生一个集群。例如：

```
"cluster_configs": {
  "HDFSHA": {
    "hdfs.nnha": true
  }
}
```

BZ#[1155378](#)

在这个版本中，Sahara API 现在可以完全支持 HTTPS 协议。

BZ#[1158163](#)

在这个版本以前，Sahara 的 'distributed' 模式功能是一个 alpha 测试的功能，Red Hat Enterprise Linux OpenStack Platform 没有单独打包或支持 'sahara-api' 和 'sahara-engine' 进程。

在这个版本中，'distributed' 模式功能已被认为是稳定的了，RHEL OpenStack Platform 为 'sahara-api' 和 'sahara-engine' 服务提供了 systemd 单元文件。作为结果，用户可以使用分布模式运行带有独立 API 和引擎节点集群的 Sahara。

BZ#[1164087](#)

现在，通过使用与 API 项的名称相匹配的 GET 参数，可以根据任何项名对 Sahara 对象进行查询。

BZ#[1189500](#)

在这个版本中，增加了一个 CLI，使用它可以为每个主要的插件配置默认的集群模板。部署默认模板可以方便最终用户对 Sahara 的使用。

作为结果，管理员现在可以添加共享的默认模板，并可以被用户直接使用。

BZ#[1189504](#)

Sahara 的集成测试被重新设计来使用简单的、基于 YAML 的配置来定义 "scenarios"。

BZ#1189511

以前, 任何 Linux 发行版本中的 Cloudera 都没有打包 `cm_api` 程序库。因为以前的 CDH 插件都需要依赖这个软件包, 所以在以前的版本中, CDH 无法作为默认的插件被启用。现在, `cm_api` 程序库的一个子集被添加到 Sahara 的代码中, CDH 可以正常工作并被默认启用。

BZ#1192290

在以前的版本中, 集群创建中的一些进程会无限期地进行轮询。现在, 在集群创建和处理过程中增加了超时设置, 当集群操作进行的时间超过了正常时间, 用户会看到相关的错误信息。

BZ#1194532

一个新端点 (endpoint) 被添加到 Sahara 来允许根据 Sahara 所支持的每个插件和版本的任务类型进行查询。这些信息可以用于 UI 的显示和过滤, 以及 CLI 和 REST API 用户。

BZ#1214817

在这个版本以前, Sahara 的 "distributed" 模式功能是一个 alpha 测试的功能, Red Hat Enterprise Linux OpenStack Platform 没有单独打包或支持 `sahara-api` 和 `sahara-engine` 进程。在这个版本中, 'distributed' 模式功能已被认为是稳定的了, RHEL OpenStack Platform 为 `sahara-api` 和 `sahara-engine` 服务提供了 `systemd` 单元文件。现在, 用户可以使用分布模式运行带有独立 API 和引擎节点集群的 Sahara。

BZ#1231923

在以前的版本中, 虽然插件和 `sahara-image-elements` 软件包都不会使用 Extra Packages for Enterprise Linux (EPEL) 仓库, 但 HDP 插件仍然会在集群产生的过程中安装这个仓库。这就在 HDP 集群创建中产生了一个可能造成潜在错误的、不需要的步骤, 对这些集群进行更新时可能会使用不被支持的软件包。现在, 这个仓库不再会被 HDP 插件安装。

BZ#1231974

现在增加了一个 `logrotate` 文件来强制执行当前 Red Hat OpenStack 标准中对日志文件大小限制, 使用它可以防止日志文件在被重复使用前变得过大。

BZ#1238700

在以前的版本中, 当 HDP 的 NameNode HA 可以正常工作后, Sahara 仍然会为所有任务把 Oozie 指向一个单一的 NameNode IP。因此, Oozie 和 Sahara 的 EDP 只在一个特定节点活跃时 (处于 A/P HA 模式) 才可以正常工作。在这个版本中, Oozie 会被重新指向一个名称解析服务器而不是一个 `amenode`, 从而解决了以上问题。现在, 无论哪个 NameNode 是活跃的, Oozie 和 EDP 的任务都可以正常工作。

4.1.15. openstack-selinux**BZ#1233154**

在以前的版本中，Neutron 会被绑定到不允许使用的端口上，所以 SELinux 会使 Neutron 无法工作。在这个版本中，Neutron 可以被连接到非保留的端口上进行运行，而不再会出现问题。

BZ#[1240647](#)

在以前的版本中，Neutron VPN 代理使用错误的上下文 (context) 启动，SELinux 会禁止 VPN 代理运行。在这个版本中，Neutron VPN 代理具有正确的上下文，它可以在 enforcing 模式中稳定运行。

4.1.16. python-django-horizon

BZ#[1101375](#)

现在，通过在 OpenStack dashboard 用户界面中为实例选择一个新的 flavor，可以对 OpenStack Trove 实例的大小进行调整。

BZ#[1107490](#)

dashboard 中的 'API Access' 页 ('Project > Compute > Access & Security > API Access') 现在可以提供更详细的、与用户凭证相关的信息。要查看这些信息，点 'View Credentials'。一个弹出界面会显示用户名、项目名、项目 ID、验证 URL、S3 URL、EC2 URL、EC2 访问和密钥。

BZ#[1107924](#)

在 OpenStack dashboard 的 'Volumes' 标签页中添加了创建 Block Storage (cinder) 卷转移的选项。卷转移是指把所有权从一个项目移到另一个项目。一个赠与者 (donor) 创建一个卷转移，并获得结果转移 ID 和验证密钥，然后把这些信息通过电子邮件或短信传递给接受者 (recipient)。接受者接受这个转移，提供转移 ID 和验证密钥。现在，所有者权限就从赠与者转移给接受者，相应的卷将不再对赠与者可见。

请注意，进行卷转移的 Block Storage API 有以下限制，它们会影响到 UI 的设计：

1. 当创建一个卷转移时，您不能指定谁会成为接受者，任何具有转移 ID 和验证密钥的人都可以声明成为接受者。因此，dashboard UI 不会提示输入接受者。
2. 当前的卷转移只对赠与者可见，其它项目中的用户无法看到这些转移。因为当前的转移不可见，所以 UI 没有包括一个项目表来查看和接受卷转移。而转移信息会被添加到卷的详情中，这对赠与者可见，卷的状态明确反映了一个转移已经被创建。另外，UI 同样无法为接受者显示一个包括可以接受的转移下拉列表。
3. 赠与者只有在响应创建转移的阶段可以看到验证密钥，在创建完成后，即使是赠与者也无法恢复验证密钥。因此，赠与者必须获得转移 ID 和验证密钥信息来发送给接受者。在转移被创建后，一个包括了这些信息的额外表格会被马上创建，赠与者从中获得相关信息。

BZ#[1112481](#)

OpenStack Dashboard 现在使用 Block Storage (cinder) 版本 2 作为它的首选版本。

当请求 Block Storage 客户端时，如果没有特别指定，会使用 cinder 版本 2。

BZ#[1114804](#)

现在，您可以使用 dashboard 来查看、导入和关联可以被不同资源 (images、artifacts、volumes、flavors、aggregates) 使用的元数据定义。

BZ#1121848

在 OpenStack Dashboard 中，实例的详情页现在可以显示主机节点。这些信息可以帮助用户进行系统诊断。

BZ#1124672

在这个版本中，为 OpenStack Dashboard 添加了部分的 Domain Admins 支持。另外，当使用 Identity Service (keystone) 版本 3 时，一个新创建的用户不需要指定主项目。

BZ#1143807

现在，您可以通过 dashboard 启用和禁用计算节点（在 'Admin > Hypervisors > Compute Host' 界面中点计算节点的 'Actions' 列）。

禁用一个计算主机后，调度器 (scheduler) 将无法使用这个主机启动实例。

BZ#1150839

OpenStack dashboard 的 'Volumes' 标签页中添加了 'Manage/Unmanage' 选项。'Manage' 会获取一个已存在的、在 OpenStack 外创建的卷，使它可以被使用。'Unmanage' 会使卷在 OpenStack 中不可见，但并不删除实际的卷。

BZ#1156678

dashboard 中的 OpenStack Orchestration 服务 (heat) 的用户接口选项已被改进。例如，用户现在可以检查、挂起、恢复和预览堆栈。

BZ#1162436

现在，可以对显示的 Data Processing 服务的结果数据进行过滤，从而可以使用户只看相关的数据。

BZ#1162961

现在，可以通过 dashboard 把一个卷标记为 'Bootable'。

BZ#1166490

现在，OpenStack dashboard 可以使用自定义的主题。一个新设置 ('CUSTOM_THEME_PATH') 被添加到 /etc/openstack_dashboard/local_settings 文件中。这个主题文件夹包括一个 _variables.scss 文件和一个 _styles.scss 文件。_variables.scss 文件包括了所有可以被用来设计图形用户界面的 bootstrap 变量以及和 Horizon 相关的变量；_styles.scss 文件包括了额外的图形界面显示分隔的信息。

BZ#1170470

现在，SRIOV 可以在 OpenStack dashboard 中进行配置。在 'Port Details' 标签页中包括了更多的信息；在进行端口创建和更新时，可以选择端口类型。

BZ#1170471

现在，可以在 OpenStack Dashboard (horizon) 中查看加密卷的加密元数据。一个用来显示加密元数据的功能被添加，用户可以通过点“加密”栏中的 "Yes" 来查看加密元数据。

BZ#1186380

当通过 dashboard 上传一个镜像时，您可以选择 OVA 作为它的格式。在以前的版本中，无法选择 OVA。

BZ#1189711

现在，dashboard 提供了向导程序来创建和配置 OpenStack Data Processing 所需的组件。这些向导程序可以帮助用户进行集群创建和任务执行操作。要使用这些向导程序，点 'Project > Data Processing > Guides'。

BZ#1189716

现在，在 OpenStack Dashboard 中添加了 ceilometer IPMI 计费器 (meter) 6 个 ipmi 计费器被从 ceilometer 中导出，'list_ipmi' 和 '_get_ipmi_meters_info' 可以被用来获取计费器的数据。

BZ#1190312

现在，可以通过 dashboard 查看 Orchestration 服务主机的详情（点 'Admin > System > System Information > Orchestration Services'）。这个页只在 Orchestration 服务已被部署的情况下才可见。

4.1.17. python-glance-store

BZ#1236055

现在，基于 Ceph 的 ephemeral 磁盘快照可以使用 RBD 快照和克隆。在这个版本中，数据会在 Ceph 服务器中处理，而不需要在不同节点中进行传递，这可以提高 Ceph 快照的性能。

4.1.18. python-ironicclient

BZ#1212134

以前，当节点处于 `locked` 状态时，OpenStack Bare Metal Provisioning (Ironic) 中的特定操作会失败。在这个版本中，为 Ironic 客户端添加了一个 `retry` 功能。现在，特定的操作会需要较长时间完成，但不会因为 `node locked` 造成运行失败。

4.1.19. python-openstackclient

BZ#1194779

现在，python-openstackclient 软件包和上游社区的版本 1.0.3 代码进行了同步。它增强并改进了对 Identity service 服务 v3 API 的支持。

4.1.20. qemu-kvm-rhev

BZ#[1216130](#)

以前，在一个带有大量扇区的虚拟磁盘中，扇区数量会在某些情况下被错误处理，造成转换 QEMU 镜像失败（"invalid argument" 错误）。在这个版本中，这个错误已被解决。

BZ#[1240402](#)

以前，因为不正确的可移植内存边界，当一个虚拟磁盘具有大量 I/O 负载时，QEMU 仿真器可能会被非正常终止。在这个版本中，QEMU 的线程可以被正确同步，从而解决了以上的问题。

4.1.21. sahara-image-elements

BZ#[1155241](#)

这个软件包允许用户创建可以在 RHEL OpenStack Platform 7 中使用的 HDP 2.0.6 和 CDH 5.3.0 镜像。

BZ#[1231934](#)

以前，创建 CDH 镜像在某些时候会失败，这是因为镜像创建的 wrapper 脚本在某些系统上为创建 CDH 镜像指定了太小的空间。现在，为创建 CDH 镜像增加了创建空间，镜像可以被成功创建。

4.1.22. sos

BZ#[1232720](#)

当在 Pacemaker 节点上使用 sosreport 工具程序时，MariaDB MySQL 服务器的一个日志文件不会被正确收集。在这个版本中，相应的代码已被修正，日志文件可以被正确收集。

BZ#[1240667](#)

以前，sosreport 工具程序的多个 OpenStack 插件会不正确地以明文形式收集密码，使用 sosreport 所产生的压缩文件会包括未加密的可读密码。在这个版本中，为 sosreport OpenStack 插件的所有密码增加了混淆（obfuscation）处理，sosreport tarball 中的密码将不再以明文的形式出现。

4.2. RHEA-2015:1549 — Red Hat Enterprise Linux OpenStack Platform director 发行版本

RHEA-2015:1549 公告解决了本节中所描述的软件错误。如需了解更多与这个公告相关的信息，请参阅 <https://access.redhat.com/errata/RHEA-2015:1549.html>。

4.2.1. diskimage-builder

BZ#[1230823](#)

diskimage-builder 中缺少了 PyYAML 的一个依赖程序，这会导致构建镜像失败。在这个版本中，增加了缺少的依赖程序，现在镜像可以被成功构建。

4.2.2. instack

BZ#[1210479](#)

在实施过程中不再使用 deploy-baremetal-overcloudrc 文件。现在，使用 "openstack overcloud deploy" 命令以及相关的命令行参数。

4.2.3. instack-undercloud

BZ#[1205825](#)

RabbitMQ 配置中的一个重复的行会导致 Undercloud 安装失败。在这个版本中，删除了这个重复的行，Undercloud 可以被成功安装。

BZ#[1229296](#)

openstack-ironic-discoverd 服务会在启动时检查是否存在 openstack-ironic-api。因为启动顺序的原因，openstack-ironic-discoverd 会在重启时无法监测到 openstack-ironic-api。在这个版本中，openstack-ironic-discoverd 启动代码中删除了对 openstack-ironic-api 的检查，在重启时，所有服务都可以被成功启动。

4.2.4. openstack-ironic

BZ#[1190481](#)

Ironic/Heat systemd 服务文件中的一个小错误导致日志文件重定向错误。在这个版本中，Ironic/Heat service 文件已被修正，日志会被重定向到正确的文件。

4.2.5. openstack-ironic-discoverd

BZ#[1227755](#)

ironic-discoverd 的 edeploy 插件会收集过度的信息，并存在一个 SQL blob 列中。因为这个列中的数据可能会过多，发现 (Discovery) 服务会在向 Ironic 保存 edeploy 数据时出现错误。在这个版本中，edeploy 插件会在 Undercloud 中把数据保存在一个 Swift 对象中的，在使用 edeploy 插件时，发现服务不会运行失败。

4.2.6. openstack-tripleo-heat-templates

BZ#[1232269](#)

Hostname 在 Overcloud 节点中被不正确地配置。这意味着，Pacemaker 不能解析集群成员的名称。在这个版本中，缩短了 hostname，这意味着，Nova 和 Neutron 现在不再需要附加无效的域名。

BZ#[1232439](#)

在这个发行版本中，增加了可以为独立的网络分配特定 IP 地址范围的功能。

现在，可以通过在 Overcloud Orchestration 模板中设置以下参数来指定 IP 地址分配：

```
* InternalApiAllocationPools
* StorageAllocationPools
* StorageMgmtAllocationPools
* TenantAllocationPools
* ExternalAllocationPools
```

BZ#[1232461](#)

在这个发行版本中，您可以配置 Compute VNC 代理网络。通过这个功能，您可以分离不同的网络，并可以指定哪个服务需要在哪里运行。

现在，通过设置 Overcloud Orchestration 模板中的 ServiceNetMap -> NovaVncProxyNetwork 参数，可以指定哪个网络可以作为 Compute VNC 代理使用。

BZ#[1232485](#)

在这个发行版本中，Heat 模板增加了 VLAN 标识符和 OVS 绑定选项，这可以减少重复的配置。Heat 模板现在包括了以下参数：BondInterfaceOvsOptions、StorageNetworkVlanID、StorageMgmtNetworkVlanID、InternalApiNetworkVlanID、TenantNetworkVlanID 和 ExternalNetworkVlanID。

BZ#[1232747](#)

在过去的版本中，HAProxy 无法正确配置 Horizon listener，这会导致在公共 VIP 中 Horizon 无效。在这个版本中，启用了 HAProxy Horizon listener，Horizon dashboard 现在在公共 VIP 中有效。

BZ#[1232797](#)

Ceilometer 的 backend_url 使用了一个不正确的 redis VIP。在这个版本中，Ceilometer backend_url 被设置为由 Heat 提供的服务，并不会在实施过程中构建。现在，Ceilometer 使用正确的 IP 地址。

BZ#[1232938](#)

因为 socket 绑定冲突，novncproxy 会启动失败。这意味着 novncproxy 服务无效。在这个版本中，novncproxy 服务只被绑定到控制器的 internal_api 地址。novncproxy 现在可以成功启动。

BZ#[1233061](#)

以前，当 neutron-server 首先运行时，一个竞争情况会在 neutron 数据库初始化过程中出现。当两个控制器在同时启用 neutron-server 时，启动 neutron-server 和代理的操作会在输掉竞争的控制器节点上失败，因此 Neutron 服务会在相应的控制器节点上无效。相关日志的内容和以下相似：

```
DBDuplicateEntry: (IntegrityError) (1062, "Duplicate entry
'datacentre-1' for key 'PRIMARY'") 'INSERT INTO ml2_vlan_allocations
```

```
(physical_network, vlan_id, allocated) VALUES (%s, %s, %s)'
(('datacentre', 1, 0),
```

在这个版本中，Neutron 服务器会在一个节点上短暂启动后马上暂停，主 pacemaker 在允许其它 puppet 或 pacemaker 配置发生前，会先让初始的数据库配置发生。这样做的结果就是，Neutron 服务可以在所有控制器节点上启动，而不会出现任何错误。

BZ#1233283

以前，mongodb 节点列表只能在一个 Controller 节点上被正确创建，这意味着，其它所有 Controller 节点中的 Ceilometer 的 mongodb 节点列表都为空。在这个版本中，mongodb 节点列表可以在所有 Controller 节点上被正确创建，Ceilometer 具有正确的 mongodb 节点列表。

BZ#1234637

以前，一个不正确的 HAProxy 后端配置会导致 HAProxy 把 glance-registry 服务请求转发给已经下线的节点。在这个版本中，glance-registry 服务的更新过程会被监控，从而确保可以发现已经下线的节点。

BZ#1234817

以前，Galera 的 HAProxy listener 会和 ctlplane 地址进行绑定。这意味着，在使用带有网络分离的 Openstack 时，客户端无法访问 Galera 服务。在这个版本中，HAProxy Galera listener 的绑定地址被改为 internal_api 网络中的 VIP。现在，客户端可以访问带有独立网络的 Openstacks 上的 Galera 服务。

BZ#1235408

HAProxy 过去不会使用 clustercheck 检查 MariaDB 的后端状态。这会导致 HAProxy 转发请求到 MariaDB 节点的操作由 TCP 检查，而没有和 Galera 集群进行同步。在这个版本中会使用 clustercheck 检查 MariaDB 的后端状态。现在，HAProxy 可以正确地把请求转发到 MariaDB 节点。

BZ#1235421

以前，distro-specific hieradata 文件没有应用到 Openstack 节点。在这个版本中提供了用于在所有 Openstack 节点上分发的静态 RedHat.yaml。

BZ#1235454

以前，mariadb 服务在系统引导时启动，这会导致在重启后，Pacemaker 的 mariadb 资源会失败。在这个版本中，引导时不会自动启动 mariadb 服务。现在，mariadb 完全由 Pacemaker 资源控制。

BZ#1235703

Keystone 服务通过 Pacemaker 启动，并作为 Pacemaker 中的 Ceilometer 资源的一个 systemd 的依赖。这会在启动两个版本的 Keystone 服务间产生冲突，从而导致 Pacemaker Keystone 资源启动失败。在这个版本中添加了一个 Pacemaker 的管制机制，它会暂停 Ceilometer 资源，直到 Keystone 资源已被启动。现在，Keystone 会在启动 Ceilometer 前启动，并且不再通过 systemd 启动。

BZ#1235848

因为缺少了一个 `ControlPlaneNetwork` 参数，带有 `director` 计划的实施和 `Heat` 基于模板的实施不完全相同。这会导致在使用网络隔离时，基于计划的 `Overcloud` 实施会失败。在这个版本中包括了到 `ControlPlaneNetwork` 参数的路径。现在，`Overcloud` 部署可以正常工作。

BZ#1236374

过去，`Heat` 服务会在不相关的 `redis` VIP 重新设定位置时重新启动。在 `Pacemaker` 中，`Heat` 资源会重后失败，这是因为它依赖于 `Ceilometer` 资源，而 `Ceilometer` 资源会因为集群失败造成无法在重后设定位置的 `redis` VIP 上重后。在这个版本中，当 `Ceilometer` 重后时会停止 `Heat` 服务。

BZ#1236407

以前，在启用了网络隔离的 `Overclouds` 中，`Pacemaker` 会把 `redis` master 设置为 master 无法访问的网络中的一个主机名。这意味着 `redis` 节点无法加入集群。在这个版本中，当部署启用了网络隔离时，`Pacemaker` 的主机名会根据 `internal_api` 地址进行解析。

BZ#1238117

以前，`OpenStack` 使用在控制器节点上启用的 `NeutronScale` puppet 资源，并把 `neutron` 代理的 "host" 项进行重写 ("`neutron-n-0`" 代表控制器 0，"`neutron-n-1`" 代表控制器 1)。当相关的 `neutron-scale` 资源由 `pacemaker` 启动时，这个重写操作会在部署的结束阶段进行。常见于 VM 环境中，`neutron` 会报告 L3 HA 没有足够的 L3 代理，这和 `overcloud` 的 `neutron agent-list` 命令不一致。因此，在一些情况下，`Neutron` 会产生一个错误信息，声明没有足够的 L3 代理来提供 HA（默认最少需要 2 个 L3 代理）。`Overcloud` 上的 "`neutron agent-list`" 命令会显示不一致的代理信息，例如，每个代理都有重复的项，一个是在主机 "`overcloud-controller-1.localdomain`" 上的原始代理（通常显示为 "`xxx`"），另一个是主机 "`neutron-n-1`" 上的“较新的”代理（最终状态会是 "`:-`"）。在其它情况中，如果只有一个控制器，代理重新命名会导致一个 `neutron` 代理 (`openvswitch`) 失败，它下面的其它代理将会无法启动，最终导致没有 L3、元数据或 `dhcp` 代理。

这个问题已被解决。现在，会确保原生 (native) `neutron` L3 High Availability 被使用，并且为原生 `neutron` HA 在每个网络上启用足够的 DHCP 代理。以前，在所有情况下都只静态地设置两个 DHCP 代理。现在，在 `tripleo heat` 模板中添加了一个参数来设置它（默认值是 '3'），并在 `oscplugin` 中使用它。`NeutronScale` 资源本身被从 `tripleo heat` 模板中删除 (`overcloud` 控制器 puppet manifest 保存在这个模板中)。作为结果，在使用这个版本后进行的部署中，控制器节点不会包括 `neutron-scale` 资源。这可以使用以下命令进行验证：

```
1. 在控制器节点上运行：
# pcs status | grep -n neutron -A 1
```

您应该看不到任何 "`neutron-scale`" 克隆设置或资源定义。

```
2. 在 undercloud 上运行：
$ source overcloudrc
$ neutron agent-list
```

所有的 neutron 代理都应该显示位于一个主机上。这个主机的名字与 "overcloud-controller-0.localdomain" 或 "overcloud-controller-2.localdomain" 类似，而不是 "neutron-n-0" 或 "neutron-n-2"。

BZ#1238336

过去，因为控制器节点不共享 consoleauth 令牌，所以会导致进行验证请求时失败。在这个版本中，memcached 可以共享 consoleauth 令牌，验证请求可以成功进行。

BZ#1240631

在这个发行版本中，可以设置 neutron_tunnel_id_ranges 和 neutron_vni_ranges 参数来分别管理 GRE 和 VXLAN 的通道 ID。通过 overcloud Neutron，它们可以应用到 overcloud 租户网络。例如：

```
# openstack overcloud deploy --plan overcloud --control-scale 3 --
compute-scale 1 --neutron-tunnel-id-ranges "1:1111,2:2222" --
neutron-vni-ranges "3:33,5:55,100:999" --neutron-tunnel-types
"gre,vxlan"
```

如果没有指定，tunnel_id_ranges 和 neutron_vni_ranges 的默认值都会是 "1:1000"，这可能不适用于特定的部署环境。

如果使用了以上示例中的设置，您可以在部署完成后，在控制器节点上的相关 neutron 配置文件中进行检查和验证，例如：

```
# grep -rni 'vni_ranges\|id_ranges\|tunnel_types' /etc/neutron/*
/etc/neutron/plugin.ini:78:tunnel_id_ranges =1:1111,2:2222
/etc/neutron/plugin.ini:85:vni_ranges =3:33,5:55,100:999
/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini:77:tunnel_types
=gre,vxlan
/etc/neutron/plugins/ml2/ml2_conf.ini:78:tunnel_id_ranges
=1:1111,2:2222
/etc/neutron/plugins/ml2/ml2_conf.ini:85:vni_ranges
=3:33,5:55,100:999
```

您也可以在部署完成后，通过创建一个 overcloud 网络进行测试，例如：

```
# source overcloudrc # from the undercloud box for example
# neutron net-create --provider:network_type "vxlan" "foo"
```

这会创建一个类型为 vxlan 的网络。您可以检查它是否从指定的 vni_ranges 中接收到一个分段 ID：

```
# neutron net-show foo
| provider:network_type      | vxlan
| provider:segmentation_id   | 3
```

同样，您也可以检查 GRE 网络是否被分配了适当的分段 ID。

BZ#1241231

以前，在没有协调的情况下创建 overcloud Keystone admin 租户会在一个有多个控制器的部署中出现问题。Heat 堆栈的创建会在 ControllerNodesPostDeployment 资源

上失败，Keystone 会返回一个 409 ERROR: openstack Conflict occurred attempting to store project - Duplicate Entry (HTTP 409)。Puppet 的运行会在此时失败。在这个版本中，admin 租户会首先在 pacemaker master 节点上创建，因此，带有多个控制器的部署可以正确地创建 overcloud keystone admin 用户。在部署完成后，您可以通过使用 admin 用户访问任何 overcloud 服务来进行验证：

```
# on the undercloud system, for example:
$ source overcloudrc
$ keystone user-list
```

BZ#1241610

在部署一个 Overcloud 时，Tuskar 会修改 Heat 套件 (Heat stack) 的顶级参数的名称。这会在 Heat 对套件进行验证时出现一个错误：

```
ERROR: The Parameter (NeutronExternalNetworkBridge) was not defined
in template.
```

这个问题的一个临时解决办法是，使用 "tuskar plan-update" 修改参数；或在环境文件中使用修改过的参数名：

```
parameters:
  Controller-1::NeutronExternalNetworkBridge: ""
```

Overcloud 部署会使用正确的参数值。

请注意：参数需要在 "parameters:" 项而不是 "parameter_defaults:" 项中定义。否则，Tuskar 的导出环境 environment.yaml 中设置的值会覆盖这个值。

BZ#1244013

以前，无论 Cinder publicurl 端点 (endpoint) 是否可以被连接，Compute 节点都会查询 Keystone。这意味着，专用的 Compute 节点无法和 Cinder API 进行通讯。在这个版本中，publicurl 端点被改为了 Compute 端口可以访问的 internalurl 端点。

BZ#1244019

以前，无论 Nova 和 Swift publicurl 端点 (endpoint) 是否可以被连接，Compute 节点都会查询 Keystone。这意味着，专用的 Cinder Storage 节点无法与 Nova 和 Swift API 进行通讯。在这个版本中，publicurl 端点被改为 internalurl 端点，现在，Cinder Storage 节点可以和 Nova 和 Swift API 进行通讯。

BZ#1244226

在以前的版本中，allow_overlapping_ips Neutron 被设置为默认的值，Neutron 中的 allow_overlapping_ips 被禁用，这会无法在相同的范围内定义多个网络。在这个版本中，allow_overlapping_ips 被设置为 "true"，这将允许定义相互间有覆盖范围的多个网络。

4.2.7. openstack-tripleo-image-elements

BZ#1235994

以前，默认的 Ceph scale 被设置为 1。这意味着，即使用户不希望创建 Ceph 节点，一

个 Ceph 节点也会被创建。

在这个版本中，默认的 Ceph scale 设置是 0，Ceph 不再会被默认部署。

4.2.8. openstack-tripleo-puppet-elements

BZ#[1229302](#)

以前，os-apply-config 命令创建的 /etc/puppet/hieradata 没有设置访问权限。这个目录中的文件包括了密码和令牌信息，从而可能造成对 OpenStack 安装的非授权访问。在这个版本中，/etc/puppet/hieradata 的所有者被设置为 root，它的权限是 0700。现在，只有 root 用户可以访问 /etc/puppet/hieradata，这增加了系统的安全性。

BZ#[1233916](#)

Overcloud 节点会包括不正确的同步系统时间，这会在 HA Controller 集群中产生一些错误。一个临时解决办法是，在运行 "openstack overcloud deploy" 命令时使用 --ntp-server 命令行参数。这个参数会在每个 Overcloud 节点的 /etc/ntp.conf 文件中配置 ntp 服务器，从而保证了正确的同步系统时间，并使 Overcloud 可以被成功部署。

4.2.9. openstack-tuskar

BZ#[1205281](#)

以前，迁移到一个 puppet 部署意味着 boot-stack tripleo-image-element 不再创建初始的 Tuskar 数据库。因此，在成功安装 undercloud 后，Tuskar 服务并没有被正确配置。在这个版本中，undercloud 会正确安装并配置 Tuskar。现在，当成功安装完 undercloud 后，您可以正常使用 Tuskar 服务。

BZ#[1220651](#)

以前，tuskar 服务的配置参数 auth_strategy 被默认设置为 "noauth"。这会导致对 tuskar 管理集合和角色（包括模板，以及密码等敏感参数设置）的不受限制的访问。在这个版本中，默认的设置是进行 keystone 验证。现在，到 tuskar 服务的、没有被验证的 http 请求将返回 HTTP 401 Unauthorized 错误。在 Undercloud 上使用以下命令进行验证：

```
$ curl -v localhost:8585/v2/plans
```

4.2.10. openstack-tuskar-ui

BZ#[1197857](#)

以前，执行批量操作的代码并不检查一个节点是否已被选择。操作会假设节点列表不为空。当 DEBUG 被禁用时，代码会产生一个未捕获的异常 (uncaught exception)，并产生一个 "Something went wrong" 信息。

在这个版本中，代码会检查是否最少有一个节点被选项，在没有选择任何节点时，会显示一个详细的错误信息。现在，当执行批量操作时如果没有选择节点，一个信息会要求您需要首先选择一些节点。

BZ#1227013

以前，节点详情 URL 名称的指定不正确。在这个版本中，节点详情 URL 的名称已被修正。现在，节点详情链接可以被正确处理，用户可以正确地访问节点详情页。

BZ#1232329

以前，在 Undercloud 的过程中，角色不能被添加到计划中。因此，不能使用 OpenStack Dashboard 对角色进行编辑。

在这个版本中，角色会在 Undercloud 的安装过程中被添加到计划中。因此，现在可以通过 Dashboard 编辑角色。

BZ#1236360

以前，因为需要和外部 API 服务（如 keystone、heat、ironic 和 tuskar）进行交流，director 的用户界面会需要大量时间来加载页面。在这个版本中添加了所有外部服务调用的缓存，这减少了实际调用的数量，并减少了页面加载的时间。现在，当访问用户界面时，页面的加载时间会大大缩短。

BZ#1245192

以前，在创建和初始相关端点（endpoint）前，用户接口会尝试连接相应的 Overcloud keystone-api。当用户接口无法找到端点时，Overcloud 就无法从用户接口进行初始化。在这个版本中，当 Overcloud 还没有被初始化时，用户接口可以被正确地识别。以上错误将不会再发生。

4.2.11. python-rdomanager-oscplugin**BZ#1229795**

以前，post-install 验证的功能没有在 CLI 中实现，它只包括在已过时的 OpenStack Deployment (triple0) 脚本中。因此，python-rdomanager-oscplugin 中没有包括这个命令。

在这个版本中，post-install 验证功能已被实现，'openstack overcloud validate' 命令现在包括在 CLI 中。

BZ#1229796

以前，DRAC ready-state 配置没有在 CLI 中实现，它只包括在已过时的 OpenStack Deployment (triple0) 脚本中。因此，python-rdomanager-oscplugin 中没有包括这个命令。

在这个版本中，DRAC ready-state 功能已被实现，'openstack baremetal configure ready state' 命令现在包括在 CLI 中。

BZ#1230450

这个发行版本为管理默认 overcloud neutron 租户网络行为提供了 4 个参数：--neutron-network-type、--neutron-tunnel-types、--neutron-disable-tunneling 和 --neutron-network-vlan-ranges。它的默认设置是为 overcloud 租户网络提供 GRE 类型和通道（tunnel）。

如果把 `--neutron-network-type` 设置为 `'vlan'`，您将可以通过设置 `--neutron-network-vlan-ranges` 参数来管理分配给 overcloud Neutron 租户网络的 VLAN ID 的范围（默认值是 `"datacenter:1:1000"`，其中的 `"datacenter"` 是 VLAN 要被分配到的 Neutron 物理网络的名称）。如果把网络类型只设置为 `'vlan'`，`--neutron-disable-tunneling` 需要被同时指定（参见以下示例）。

如果把 `--neutron-network-type` 设置为 `'gre'` 或 `'vxlan'`（或两个都被设置），则相关的 `--neutron-tunnel-types` 参数也需要被启用（参见以下示例）。另外，如果把网络类型设置为 `'gre'` 或 `'vxlan'`，您还可以通过设置相关的 `--neutron-tunnel-id-ranges` 和 `--neutron-vni-ranges` 参数来分别管理 GRE 和 VXLAN 的通道 ID。

示例：

* 为 overcloud 租户网络指定 `'vxlan'` 类型：

```
openstack overcloud deploy --plan overcloud --control-scale 3 --
compute-scale 2 --neutron-network-type "vxlan" --neutron-tunnel-
types "vxlan"
```

* 为 overcloud 租户网络指定 `'gre'` 和 `'vxlan'` 类型：

```
openstack overcloud deploy --plan overcloud --control-scale 3 --
compute-scale 2 --neutron-network-type "gre,vxlan" --neutron-tunnel-
types "gre,vxlan"
```

* 为 overcloud 租户网络指定 `'vlan'` 类型，并同时设置 VLAN 的范围：

```
openstack overcloud deploy --plan overcloud --control-scale 3 --
compute-scale 2 --neutron-network-type "vlan" --neutron-disable-
tunneling --neutron-network-vlan-ranges "datacenter:3:3000"
```

如果在部署时没有指定这些参数，它们会被设置为以下的默认值。这些默认值可能会不适用于具体的部署环境：

```
--neutron-network-type # default 'gre'
--neutron-tunnel-types # default 'gre'
--neutron-disable-tunneling # defaults to tunneling being enabled
--neutron-network-vlan-ranges # 'datacenter:1:1000'
```

在部署完成后，您可以检查控制器节点（controller node）的 Neutron 配置日志来确定以下是否已被配置：

```
grep -rni 'tunnel_types\|network_type\|enable_tunneling\|vlan_ranges'
/etc/neutron/*
/etc/neutron/plugin.ini:14:tenant_network_types = vlan
/etc/neutron/plugin.ini:72:network_vlan_ranges =datacenter:3:3000
/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini:54:enable_tun
neling=False
/etc/neutron/plugins/ml2/ml2_conf.ini:14:tenant_network_types = vlan
/etc/neutron/plugins/ml2/ml2_conf.ini:72:network_vlan_ranges
=datacenter:3:3000
```

另外，您可以在 overcloud Neutron 上创建一个网络并检查它。例如，在 undercloud 上：

```
$ source overcloudrc
$ neutron net-create "default"
$ neutron net-show default
...
```

```
| provider:network_type      | vlan
|
| provider:physical_network | datacenter
|
| provider:segmentation_id   | 3
|
| ...
```

BZ#1232851

过去, Unified CLI 会在 OpenStack overcloud 部署还没有完成时就出现超时, 这会导致用户错误地接收到一个部署失败的信息。

在这个版本中, 超时的值被增加到一个小时。现在, 部署过程应该可以在超时时间内完成。

BZ#1233201

在这个发行版本中, 删除了一个重复的、用于扩展部署的命令。现在, 可以使用 Tuskar 或部署命令来直接扩展部署。

BZ#1234343

在以前的 KVM PXE 代码中有一个错误, 它在有多个节点同时使用 PXE 引导时显示失败, 从而造成一些节点无法连接到 DHCP。

在这个版本中, 睡眠 (sleep) 值被增加, 允许节点处于反省状态 (introspection)。现在, DHCP 将不再会是一个问题, 只是反省状态时间会稍长。

BZ#1234673

以前, 当使用 'openstack overcloud deploy' 命令时, 一些命令行参数不会被代码使用, 从而使用户无法设置这些参数。

在这个版本中, 这些参数会被正确使用, 并被发送到 Orchestration 服务。

BZ#1236073

过去, OpenStack Overcloud 验证使用上游的文件来运行 OpenStack Integration Test Suite (tempest), 而不是使用 redhat-openstack/tempest 中的中游工具。这造成了输出格式的不同。

在这个版本中, OpenStack Overcloud 验证使用中游工具运行 tempest, 输出格式与期望的相同。

BZ#1236399

以前, CLI 没有提供修改特定 Neutron 参数的方法。在这个版本中, deploy 命令添加了新的选择, 用户可以在 "Openstack overcloud deploy" 命令中使用它们来配置这些参数。

BZ#1237068

以前, Tuskar API 需要所有参数都是一个字符串。但是, 当传递的一些参数是整数时, 这

些参数就无法被 API 接受。

在这个版本中，所有参数在被发送给 API 前，都会被转换为字符串。现在，API 可以介绍所有参数，部署可以正常进行。

BZ#1240101

过去，在 `/usr/share/openstack-tripleo-heat-templates/` 中使用“硬代码”来指定到 Heat 模板集合的路径。这会导致，在使用自定义 Heat 模板（如创建一个本地副本并对它进行定制）时出现问题。在这个版本中，使用一个输入变量（通过 `--template` 参数指定）来替代它。现在，用户可以创建一个带有自定义 Heat 模板集合的 Overcloud。

BZ#1240461

以前，在重新实效 OpenStack Overcloud 时，所需的参数不会被正确地传递给 Orchestration 服务，这会产生一个 `UPGRADE_FAILED` 信息。

在这个版本中，在 Orchestration 更新参数中添加了一个 `'Existing=True'` 参数。现在，Overcloud 可以被成功部署。

BZ#1240679

以前，在部署过程中，虽然 ironic 日志显示有可用的节点，heat 引擎日志会因为没有找到有效的主机而返回一个非零的错误。这是因为，在 `"openstack baremetal introspection"` 命令完成后，director 没有把节点设置为 `"available"`。在这个版本中，当 `introspection` 完成后节点被设置为 `"available"`。现在，director 在实施 Overcloud 时可以看到节点。

BZ#1242967

除了使用数据库中的计划，现在 director 还可以使用 Heat 模板集合来创建一个 Overcloud。这个功能提供了一个在运行 `"openstack overcloud stack update"` 命令来更新 Overcloud 中的软件包时直接使用 Heat 模板的方法。如果您使用模板集合而不是计划创建 Overcloud，在运行 `"openstack overcloud stack update"` 命令时使用 `"--templates"` 参数替代 `"--plan"` 参数。

BZ#1242973

现在，director 提供了一个参数来在更新 Overcloud 节点上的软件包时接收格外的 Heat 环境文件。Overcloud 分离的网络在创建以及以后的更新过程中需要额外的环境文件，如果没有这些环境文件，用户将无法在带有分离网络的 Overcloud 中使用 `"openstack overcloud update stack"` 命令。现在，用户可以在 `"openstack overcloud update stack"` 命令中使用 `"-e"` 参数来解决这个问题。

BZ#1242989

在这个发行版本中，用户可以在删除 overcloud 节点时直接使用 Heat 模板。现在，如果在部署 overcloud 时没有使用 Tuskar，则可以在运行 `"openstack overcloud node delete"` 命令时使用 `"--templates"` 参数。

BZ#1242990

在这个发行版本中增加了一个新功能 - 在从 overcloud 中删除一个节点时可以接受额外的 Heat 环境文件。当在一个分离的网络中进行 overcloud 更新时，额外的环境文件需要被传递给 Heat。现在，"openstack overcloud node delete" 命令可以被用来在一个分离的网络中删除 overcloud 节点。

BZ#1243016

以前，在 deploy 命令中存在一些功能相同的重复的参数。在这个版本中，这些重复的参数（--plan-uuid、--use-tripleo-heat-templates、--extra-template）已被删除。

BZ#1243365

以前，超时值没有被设置，这意味着在一个小时之后，部署就会出现超时。在这个版本中增加了一个超时参数，可以被用来设置一个自定义的超时值。它的默认值是 4 小时。

BZ#1243846

以前，Heat 模板缺少一个特定的参数来配置 Neutron L3 Agent。因此，director 无法正确配置 L3 Agent。在这个版本中，为模板添加了以前缺少的 Heat 参数。现在，director 可以正确配置 L3 Agent。

BZ#1244913

以前，当实施只有一个控制器时，'L3HA' 会作为 'True' 传递给 overcloud Neutron 配置。因为 Neutron 需要最少 2 个 L3 代理来实现原生 L3 HA，在实施成功完成后设置 overcloud 租户网络会报告错误。在实施成功完成后创建租户 Neutron 网络或路由时，Neutron 会返回以下错误：

```
Not enough l3 agents available to ensure HA. Minimum required 2,
available 1.
```

在这个版本中，这个错误已被解决。当进行带有一个控制器的实施时，控制器节点的 /etc/neutron/neutron.conf 文件中的 'L3HA' 是 'False'；当实施有 2 个控制器节点时，它的值是 'True'。

4.2.12. python-tuskarclient

BZ#1228433

以前，python-tuskarclient 不支持指定一个 Tuskar API 服务器自定义 CA 证书来验证 SSL 证书。因为服务器使用的证书对于客户是未知的，所有连接会失败。在这个版本中，python-tuskarclient 增加了一个 --os-cacert 选项，使用它可以指定 CA 证书路径，从而可以和 API 服务器进行正常交流。

4.2.13. rhel-osp-director

BZ#1225016

以前, 在 Overcloud 中, `/etc/glance/glance-api.conf` 文件中的 `glance_store` 项被设置为 `stores=glance.store.filesystem.Store`。这会导致因为不同的 `store` 类型造成镜像上传错误。在这个版本中, `glance` 配置被改为使用 `glance.store.http.Store` 作为 `store` 参数, 并包括了 `store` 类型使用的后端。

BZ#1225621

过去, 配置文件中的不正确的 DHCP 选项顺序会导致引导失败。在这个版本中, 使用 DHCP 选项标签 (tag) 来提供正确的顺序。现在, 机器可以使用 iPXE ROM 进行键条式加载引导, 然后调用 HTTP URL 继续引导的过程。这可以保证整个引导过程可以正确进行。

BZ#1226097

过去, `grub` 配置会把内核参数设置为把控制台重新定向到一个可能并不存在的串口上。这会导致节点引导失败。在这个版本中, 把控制台重新定向到串口的功能在默认情况下被禁用。现在, 节点可以被成功引导。

BZ#1227421

上游的 Nova 会在主机名中默认加上 `"novalocal"`, 这会导致 nova 主机名和 Puppet 所期望的相关主机名不符, 从而使在创建 Overcloud 时, Heat 堆栈中的 `ControllerNodesPostDeployment` 出现 `CREATE_FAILED` 错误。在这个版本中, 上游的默认设备被覆盖, 主机名中不再会被自动添加 `"novalocal"` 后缀。现在, 部署后的 Puppet 配置会在 Overcloud Heat 堆栈中报告一个 `CREATE_COMPLETE` 信息。

BZ#1227940

在 `instack.answers` 中为 Undercloud 设置 `deployment_mode` 配置的功能以前是一个技术预览 (tech preview), 现在这个功能已被删除。`deployment_mode` 允许选择一个 `"scale"` 模式来把不同的角色部署到特定节点中。这个功能现在被节点标签 (node tagging) 功能所替代, 它是 Automated Health Check (AHC) 工具的一部分。

BZ#1228419

`director` 提供了一个把不同的网络服务在独立的子网和 VLAN 中进行分离的方法。相关的信息包括在 Red Hat Enterprise Linux OpenStack Platform 7.0 Director 安装和使用指南中。

BZ#1229372

DHCP 提供的路由信息和静态路由信息都出现在路由表中, 但是 DHCP 路由的 `metric` 值会是 100。这意味着 `metric` 值为 0 的静态路由一直会被使用。如果需要使用多个 DHCP 连接, 您可以把其中的一个或多个配置为忽略 DHCP 服务器提供的路由信息 (使用 `"defroute: false"` 声明)。

BZ#1230840

以前, OpenStack Integration Test Suite (tempest) 没有提供针对于部署的值, 从而会导致以下模板测试失败。

在这个版本中, `'openstack overcloud validate'` 命令中添加了一个 `'--deployer-input'` 参数, 管理员可以使用它提供一个包括针对于部署的值的文件 (`tempest.conf`)。使用 `'--deployer-input filename'` 参数可以减少测试结果失败的情况。

BZ#1230966

Redis 需要使用一个独立的 VIP。在部署带有网络隔离的 Redis 时, director 会默认自动把 Redis VIP 放置在 Internal API VIP。操作者无法使用 ServiceNetMap 参数把 Redis 移到其它网络中。

BZ#1233860

以前, director 使用一个带有网络和浮动 IP 选项的文件来进行部署。但是, 早期版本的 director 会忽略网络选项。在这个版本中, 使用 "openstack overcloud director" 命令的一组命令行参数来替代这个文件。现在, 用来配置网络和浮动 IP 地址的选项包括在 CLI 工具中。

BZ#1234856

以前, 'management_plan.uuid' 变量名中的一个错误会造成属性错误, 从而可能导致所有 Tuskar 计划的部署失败。在这个版本中, 这个变量名已被改正。

BZ#1235476

以前, 当部署一个 Overcloud 时, director 会把 Public VIP 服务放置到 Provisioning 网络的 "ctlplane" 中。这意味着, 您无法从 Overcloud 以外访问这些服务。在这个版本更新中, Heat 模板会把 Public VIP 放置到一个 External 网络中。

BZ#1235624

以前, 当部署一个 Overcloud 时, director 会把 Public VIP 服务放置到 Provisioning 网络的 "ctlplane" 中。这意味着, 您无法从 Overcloud 以外访问 horizon 和 keystone 服务。在这个版本更新中, Heat 模板会把 Public VIP 放置到一个 External 网络中。

BZ#1235667

以前, 当 ironic-discoverd 发出一个带有已在节点上配置的、相同的引导设备的 'set_boot_device' 请求时, OpenStack Bare Metal Provisioning (ironic) 中的 DRAC 驱动会试图对 BIOS 进行一个没有任何改变的配置任务, 这会导致部署失败。

在这个版本中, 如果请求中的目标引导设备和当前的相同, OpenStack Bare Metal Provisioning 中的 DRAC 驱动会忽略这个请求。因此, 部署可以成功完成。

BZ#1235908

以前, 当 Orchestration 部署 Overcloud 时 keystone 令牌会过期, 这会导致部署失败 (验证错误)。

在这个版本中, 令牌超时值已被增加, 部署 Overcloud 可以成功完成。

BZ#1236251

以前, External 网络上没有设置默认的路由。这意味着, 您只能从和 Controller 相同的子网中访问 Horizon 和 Public API。在这个版本更新中, Heat 模板包括了 ExternalInterfaceDefaultRoute 参数, 这可以确保 External 接口上有一个默认的路由。

BZ#1236578

以前, NeutronScale 资源对 Controller 节点上的 neutron 代理进行了重命名, 这会导致与 "neutron agent-list" 命令的结果不一致, Neutron 会报告一个“没有足够 L3 代理来实现 L3 HA”错误。在这个版本中, NeutronScale 资源被从 Overcloud Heat 模板和计划中删除。NeutronScale 不会再出现在 "neutron agent-list" 命令结果中, Neutron 不会报告任何错误。

BZ#1237064

以前, undercloud 的 admin 用户没有服务租户的 'swiftoperator' 角色。这会导致 CloudForms Management Engine (CFME) 无法访问 swift 对象 (因为 CFME 使用 admin 用户)。

在这个版本更新中, undercloud 的 admin 用户被分配了服务租户的 'swiftoperator' 角色。现在, 通过在 API 请求中指定服务租户, admin 用户可以访问 swift 对象。

BZ#1237144

以前, NeutronScale 资源对 Controller 节点上的 neutron 代理进行了重命名, 这会导致与 "neutron agent-list" 命令的结果不一致, Neutron 会报告一个“没有足够 L3 代理来实现 L3 HA”错误。在这个版本中, NeutronScale 资源被从 Overcloud Heat 模板和计划中删除。NeutronScale 不会再出现在 "neutron agent-list" 命令结果中, Neutron 不会报告任何错误。

BZ#1237145

在这个发行版本中, 为 Block Storage 服务增加了 NFS 后端, 从而使它可以提供更多的 Block Storage 后端选择。

现在, Overcloud Block Storage 服务可以被配置为带有以下参数的 NFS 后端:

- * CinderEnableNfsBackend
- * CinderNfsMountOptions
- * CinderNfsServers

BZ#1237150

以前, glance 后端以“硬代码”的形式设置为使用 swift, 这意味着您无法选择使用其它文件系统类型, 如 NFS。在这个版本中, 增加了 glance 使用 NFS 后端的功能。您可以使用以下参数来把 Overcloud 的 glance 服务配置为使用 NFS 后端:

- * GlanceBackend
- * GlanceFilePcmkManage
- * GlanceFilePcmkDevice
- * GlanceFilePcmkOptions

BZ#1237235

OpenStack Dashboard (Horizon) 现在成为了高可用性架构的一部分。现在, Dashboard 是一个由 Pacemaker 管理的资源。

BZ#[1237329](#)

以前, Pacemaker 和 ironic 会相互争夺对电源管理的控制, 这会导致隔离 (fencing) 功能出现问题。在这个版本中, /etc/ironic/ironic.conf 中默认设置了 force_power_state_during_sync=False。这可以防止在同步的过程中 ironic 自动恢复节点的电源状态。现在, Pacemaker 可以成功地隔离节点。

BZ#[1238217](#)

没有 CLI 参数可以被用来设置 NeutronExternalNetworkBridge, 这会在分配浮动 IP 时出现问题。当前设置这个参数的唯一方法是通过网络隔离的自定义环境文件进行。例如:

```
parameters:
  # Set to "br-ex" if External is on native VLAN
  Controller-1::NeutronExternalNetworkBridge: ""

parameter_defaults:
  # Set to "br-ex" if External is on native VLAN
  NeutronExternalNetworkBridge: ""
```

如果浮动 IP 网络在一个 VLAN 上, 把两个参数都设为 ''; 如果在一个 br-ex 网桥的 native VLAN 上, 则把它们都设置为 'br-ex'。这个配置将使 Neutron 桥接映射在环境中可以正常工作。相关信息包括在 Red Hat Enterprise Linux OpenStack Platform 7 Director 安装和使用指南中。

BZ#[1238434](#)

以前, ipxe-bootimgs 软件包没有包括在默认的 Red Hat Enterprise Linux 仓库 (repository) 中, 它只包括在 Red Hat Enterprise Linux - Optional 仓库中。这意味着, 没有 Optional 频道的部署会失败。在这个版本中, 这个软件包被加入到 director 频道中。现在, 在没有启用 Optional 频道的情况下, 部署也可正常工作。

BZ#[1238750](#)

以前, NeutronScale 资源对 Controller 节点上的 neutron 代理进行了重命名, 这会导致与 "neutron agent-list" 命令的结果不一致, Neutron 会报告一个“没有足够 L3 代理来实现 L3 HA”错误。在这个版本中, NeutronScale 资源被从 Overcloud Heat 模板和计划中删除。NeutronScale 不会再出现在 "neutron agent-list" 命令结果中, Neutron 不会报告任何错误。

BZ#[1238844](#)

以前, Overcloud 对它的 Heat 组件配置不正确, 缺少对 heat_stack_user_role、stack_domain_admin 和 stack_domain_admin_password 的设置。在这个版本中, user 和 admin 角色在 /etc/heat/heat.conf 中被正确配置。

BZ#[1238862](#)

以前, 部署的 Overclouds 把 Heat CloudFormation API 配置为使用 auth_url 指

向 localhost。但是，Keystone 不会监听 localhost，这会导致 Heat CloudFormation API 无法使用。在这个版本中，`/etc/heat/heat.conf` 中的 `auth_url` 选项被设置为指向 Keystone 在 Internal API 网络中监听的 IP 地址。Heat CloudFormation API 现在可以正常工作。

BZ#1240449

以前，Overcloud 为 heat 服务配置了 `instance_user=heat-admin`。这意味着，到 heat 部署的虚拟机的 SSH 通讯需要使用 heat-admin 用户。在这个版本中，`instance_user` 被设置为空。现在，可以使用默认的镜像用户 SSH 到虚拟机。

BZ#1240824

到数据库的连接数量会受到 Controller 的数量，以及每个 Controller 所具有的内核的限制。在一个有 3 个 controller，每个 controller 有多于 12 个内核的 HA 环境中，数据库的连接数量可能会达到 `max_connections` 的默认值（1024），这可能会出现对服务的请求没有响应的问题。作为一个临时的解决办法，您可以使用以下命令增加 `max_connections` 的值：

```
$ openstack management plan set [tuskar_plan_uuid] -P "Controller-1::MysqlMaxConnections=4096"
```

使用实际的计划 UUID 替换 `[tuskar_plan_uuid]`，计划的 UUID 可以使用以下命令找到：

```
$ openstack management plan list
```

要在部署时使用 `--templates` 参数来增加 `max_connections` 的值，为部署命令提供包括以下内容的额外自定义环境文件

```
parameters:
  MysqlMaxConnections: 4096
```

把它添加到 `deploy` 命令中：

```
$ openstack management deploy --plan overcloud -e
/path/to/custom_environment_file.yaml
```

BZ#1241131

过去，不是所有节点都被配置为可以访问所需要的服务器（如 NTP、DNS），这会导致因为没有访问 NTP 服务器，节点间出现不同步的现象。在这个版本中，Undercloud 被设置为非控制器节点（non-Controller node）的网关。这样，非控制器节点就可以访问外部服务（如 DNS 和 NTP），从而避免了节点不同步现象的出现。

BZ#1241422

以前，SELinux 在 Ceph OSD 节点上被设置为 `enforcing` 模式。但是，根据 Ceph 的官方文件，SELinux 在 Ceph OSD 节点上应该被设置为 `permissive` 模式。在这个版本中，Ceph OSD 节点上的 SELinux 被设置为 `permissive`。

BZ#1242052

以前，Pacemaker 服务的启动时间超时值被设置为 20 秒。有些时候，启动时间会超过这个限制，从而导致部署停止。在这个版本中，这个值被增加到 60 秒。现在，Pacemaker 服务可以正确启动，部署可以完成。

附录 A. 修订历史

修订 7.0.0-1.1	Thu Aug 6 2015	Red Hat Localization Services
与 XML 源 7.0.0-1 版本同步的翻译文件		
修订 7.0.0-1	August 5, 2015	RHEL OpenStack Platform Docs Team
添加到认证插件的 kbase 链接。		
修订 7.0.0-0	Mon Jun 9 2015	RHEL OpenStack Platform Docs Team
Red Hat Enterprise Linux OpenStack Platform 7.0 的初始修订版本		