



Red Hat Virtualization 4.4

数据仓库指南

如何使用 Red Hat Virtualization 的数据仓库(Data Warehouse)功能

Red Hat Virtualization 4.4 数据仓库指南

如何使用 Red Hat Virtualization 的数据仓库(Data Warehouse)功能

Red Hat Virtualization Documentation Team

Red Hat Customer Content Services

rhev-docs@redhat.com

法律通告

Copyright © 2023 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

本文档提供有关 Red Hat Virtualization 数据仓库的信息和流程。

目录

前言	3
第 1 章 安装和配置数据仓库	4
1.1. 配置数据仓库概述	4
1.2. 在一个单独的计算机上安装和配置数据仓库	4
第 2 章 将数据仓库迁移到 9 月的机器	10
2.1. 将数据仓库数据库迁移到独立机器	10
2.2. 将数据仓库服务迁移到独立机器	12
2.3. 更改数据仓库采样规模	16
第 3 章 关于历史记录数据库	18
3.1. 历史记录数据库概述	18
3.2. 跟踪配置历史	18
3.3. 记录工资历史	19
3.4. OVIRT-ENGINE-DWHD.CONF 中的数据仓库服务的应用程序设置	19
3.5. 跟踪标签历史记录	19
3.6. 允许只读访问 HISTORY 数据库	20
3.7. 统计历史视图	21
3.8. 配置历史视图	34
附录 A. 法律通知	50

前言

Red Hat Virtualization Manager 包括一个数据仓库，用于收集有关主机、虚拟机和存储的监控数据。必须在同一台机器上或单独服务器上安装和配置包含数据库和服务的数据（包括数据库和服务）。

Red Hat Virtualization 安装创建两个数据库：

- 管理器（Manager）数据库（引擎）是 Red Hat Virtualization Manager 使用的主要数据存储。有关虚拟化环境的信息，如状态、配置和性能会存储在此数据库中。
- Data Warehouse 数据库(**ovirt_engine_history**)包含配置信息和统计数据，这些配置信息与管理器数据库有关。Manager 数据库中的配置数据会每分钟检查，更改也会复制到数据仓库数据库中。跟踪对数据库的更改提供了有关数据库中对象的信息。这可让您分析和提高 Red Hat Virtualization 环境的性能并解决困难。

要计算使用 **ovirt_engine_history** 数据库所使用的空间和资源估算，请使用 [RHV Manager History Database Size Calculator](#) 工具。估计结果基于实体数量和您选择保留历史记录记录的时间长度。

第 1 章 安装和配置数据仓库

1.1. 配置数据仓库概述

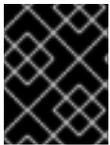
您可以在与 Manager 相同的机器上安装和配置数据仓库，或者在可访问 Manager 的单独机器上安装和配置：

在 Manager 计算机上安装和配置数据仓库

此配置只需要单一注册的计算机，而且是最简单的配置，但会增加 Manager 机器的需求。需要访问数据仓库服务的用户需要访问 Manager 机器本身。请参阅 [使用本地数据库安装 Red Hat Virtualization 作为一个独立的 Manager 中的配置 Red Hat Virtualization Manager](#)。

在单独的机器上安装和配置数据仓库

此配置需要两个注册的计算机。它减少了管理器机器上的负载，并避免在该机器上潜在的 CPU 和内存共享冲突。管理员还可以允许用户访问数据仓库计算机，而无需授予 Manager 计算机的访问权限。有关该配置的更多信息，请参阅 [在独立机器中安装和配置数据仓库](#)。



重要

建议您将数据仓库部署中的所有机器的系统时区设置为 UTC。这样可确保数据收集不会因为您的本地时区的不同而中断：例如，从夏天的时间冬天时间的变化。

要计算使用 `ovirt_engine_history` 数据库所使用的空间和资源估算，请使用 [RHV Manager History Database Size Calculator](#) 工具。估计结果基于实体数量和您选择保留历史记录记录的时间长度。



重要

`engine-setup` 中预期的行为如下：

- 安装数据仓库软件包、运行 `engine-setup` 并回答 **No** 以配置数据仓库：

```
Configure Data Warehouse on this host (Yes, No) [Yes]: No
```

- 再次运行 `engine-setup`；设置不再提供配置数据仓库的选项。

要强制 `engine-setup` 重新显示选项，请运行 `engine-setup --reconfigure-optional-components`。

要配置当前安装的数据仓库软件包，并阻止设置应用启用的存储库中找到的软件包更新，请添加 `--offline` 选项。

1.2. 在一个单独的计算机上安装和配置数据仓库

这部分论述了在独立于 Red Hat Virtualization Manager 的机器上安装和配置数据仓库服务。在单独的计算机上安装数据仓库有助于减少管理器计算机的负载。



注意

红帽只支持安装数据仓库数据库、数据仓库服务和 Grafana，它们都与彼此相同，尽管您可以在独立的机器上分别安装这些组件。

先决条件

- Red Hat Virtualization Manager 安装在一个独立的机器上。
- 运行 Red Hat Enterprise Linux 8 的物理服务器或虚拟机。
- Manager 数据库密码。
- 从数据仓库计算机访问 Manager 数据库的 TCP 端口 5432。
- 如果您选择从数据仓库服务单独安装数据仓库数据库，您必须首先设置数据库。要手动安装远程数据仓库数据库，请参阅[准备远程 PostgreSQL 数据库](#)。您必须包含有关数据库机器的以下信息：
 - FQDN
 - 可以访问数据库的端口（默认为5432）
 - 数据库名称
 - 数据库用户
 - 数据库密码
 - 您必须通过编辑 `postgresql.conf` 文件手动授予访问权限。编辑 `/var/lib/pgsql/data/postgresql.conf` 文件并修改 `listen_addresses` 行，使其与以下内容匹配：

```
listen_addresses = '*'
```

如果该行不存在或已被注释掉，请手动添加。

启用 Red Hat Virtualization Manager 存储库

您需要使用 Red Hat Subscription Manager 登录并注册数据仓库，附加 **Red Hat Virtualization Manager** 订阅并启用 Manager 存储库。

流程

1. 使用 Content Delivery Network 注册您的系统，在提示时输入您的客户门户网站用户名和密码：

```
# subscription-manager register
```



注意

如果您使用 IPv6 网络，请使用 IPv6 转换机制来访问 Content Delivery Network 和 subscription Manager。

2. 查找 **Red Hat Virtualization Manager** 订阅池并记录池 ID：

```
# subscription-manager list --available
```

3. 使用池 ID 将订阅附加到系统：

```
# subscription-manager attach --pool=pool_id
```



注意

查看当前附加的订阅：

```
# subscription-manager list --consumed
```

列出所有启用的软件仓库：

```
# dnf repolist
```

4. 配置存储库：

```
# subscription-manager repos \
  --disable='*' \
  --enable=rhel-8-for-x86_64-baseos-eus-rpms \
  --enable=rhel-8-for-x86_64-appstream-eus-rpms \
  --enable=rhv-4.4-manager-for-rhel-8-x86_64-rpms \
  --enable=fast-datapath-for-rhel-8-x86_64-rpms \
  --enable=jb-eap-7.4-for-rhel-8-x86_64-rpms \
  --enable=openstack-16.2-cinderlib-for-rhel-8-x86_64-rpms \
  --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

5. 将 RHEL 版本设置为 8.6:

```
# subscription-manager release --set=8.6
```

6. 启用 **pki-deps** 模块。

```
# dnf module -y enable pki-deps
```

7. 启用 **postgresql** 模块的版本 12。

```
# dnf module -y enable postgresql:12
```

8. 启用 **nodejs** 模块的版本 14:

```
# dnf module -y enable nodejs:14
```

9. 同步安装的软件包，将它们更新至最新可用版本。

```
# dnf distro-sync --nobest
```

其它资源

有关模块和模块流的详情，请参考 [安装](#)、[管理](#)和[删除用户空间组件中的以下部分](#)。

- [模块流](#)
- [安装软件包前选择流](#)
- [重置模块流](#)
- [切换到更新的流](#)

在一个单独的机器中安装数据仓库

流程

1. 登录到您要安装数据库的机器。

2. 请确定所有软件包都为最新版本：

```
# dnf upgrade --nobest
```

3. 安装 **ovirt-engine-dwh-setup** 软件包：

```
# dnf install ovirt-engine-dwh-setup
```

4. 运行 **engine-setup** 命令开始安装：

```
# engine-setup
```

5. 回答 **Yes** 在该机器上安装数据仓库：

```
Configure Data Warehouse on this host (Yes, No) [Yes]:
```

6. 回答 **Yes** 在该机器上安装 Grafana：

```
Configure Grafana on this host (Yes, No) [Yes]:
```

7. 按 **Enter** 接受自动探测的主机名，或者输入备用主机名并按 **Enter**：

```
Host fully qualified DNS name of this server [autodetected hostname]:
```

8. 按 **Enter** 键自动配置防火墙，或者键入 **No** 并按 **Enter** 来维护现有设置：

```
Setup can automatically configure the firewall on this system.
Note: automatic configuration of the firewall may overwrite current settings.
Do you want Setup to configure the firewall? (Yes, No) [Yes]:
```

如果您选择自动配置防火墙，且没有防火墙管理器处于活动状态，系统会提示您从支持的选项列表中选择您所选的防火墙管理器。输入防火墙管理器的名称，然后按 **Enter**。即使只列出了一个选项，也是如此。

9. 输入 Manager 机器的完全限定域名，然后按 **Enter**:

```
Host fully qualified DNS name of the engine server []:
```

10. 按 **Enter** 以允许设置通过 SSH 在 Manager 上签署证书：

```
Setup will need to do some actions on the remote engine server. Either automatically, using
ssh as root to access it, or you will be prompted to manually perform each such action.
Please choose one of the following:
1 - Access remote engine server using ssh as root
2 - Perform each action manually, use files to copy content around
(1, 2) [1]:
```

- 按 **Enter** 接受默认 SSH 端口，或者输入备用端口号，然后按 **Enter** 键：

```
ssh port on remote engine server [22]:
```

- 输入 Manager 机器的 root 密码：

```
root password on remote engine server manager.example.com:
```

- 指定是否在此机器上托管数据仓库数据库（本地），或在另一台机器上（远程）：



注意

红帽只支持安装数据仓库数据库、数据仓库服务和 Grafana，它们都与彼此相同，尽管您可以在独立的机器上分别安装这些组件。

```
Where is the DWH database located? (Local, Remote) [Local]:
```

- 如果您选择 **Local**，则 **engine-setup** 脚本可以自动配置数据库（包括添加用户和数据库）或者它连接到预配置的本地数据库：

```
Setup can configure the local postgresql server automatically for the DWH to run. This may conflict with existing applications.
```

```
Would you like Setup to automatically configure postgresql and create DWH database, or prefer to perform that manually? (Automatic, Manual) [Automatic]:
```

- 如果您通过按 **Enter** 来选择 **Automatic**，则不需要进一步的操作。
- 如果选择 **Manual**，请为手动配置的本地数据库输入以下值：

```
DWH database secured connection (Yes, No) [No]:
```

```
DWH database name [ovirt_engine_history]:
```

```
DWH database user [ovirt_engine_history]:
```

```
DWH database password:
```

- 为 Manager 数据库机器输入完全限定域名和密码。如果您要在安装 Manager 数据库的同一机器上安装 Data Warehouse 数据库，则使用相同的 FQDN。按 **Enter** 键接受其他字段中的默认值：

```
Engine database host []: engine-db-fqdn
```

```
Engine database port [5432]:
```

```
Engine database secured connection (Yes, No) [No]:
```

```
Engine database name [engine]:
```

```
Engine database user [engine]:
```

```
Engine database password: password
```

- 选择数据仓库将保留所收集数据的时长：

```
Please choose Data Warehouse sampling scale:
```

```
(1) Basic
```

```
(2) Full
```

```
(1, 2)[1]:
```

Full 使用默认值用于数据存储设置（[Application Settings for the Data Warehouse service in ovirt-engine-dwhd.conf](#)，在远程主机上安装 Data Warehouse 时这是推荐的设置。

Basic 会将 `DWH_TABLES_KEEP_HOURLY` 的值减小到 **720**, `DWH_TABLES_KEEP_DAILY` 的值减小到 **0**, 从而减轻了 Manager 集群的负载。在同一机器上安装 Manager 和数据仓库时, 请使用 **Basic**。

16. 确认安装设置 :

```
Please confirm installation settings (OK, Cancel) [OK]:
```

17. 在 Data Warehouse 配置完成后, 在 Red Hat Virtualization Manager 中重启 **ovirt-engine** 服务 :

```
# systemctl restart ovirt-engine
```

18. (可选) 将 [SSL](#) 设置为安全数据库连接。

第 2 章 将数据仓库迁移到 9 月的机器

本节论述了如何将数据仓库数据库和服务从 Red Hat Virtualization Manager 机器迁移到单独的机器。在单独的计算机上托管数据仓库服务可减少每台计算机的负载，并避免与其他进程共享 CPU 和内存资源导致的潜在冲突。



注意

红帽只支持安装数据仓库数据库、数据仓库服务和 Grafana，它们都与彼此相同，尽管您可以在独立的机器上分别安装这些组件。

您有以下迁移选项：

- 您可以从管理器计算机迁移数据仓库服务，并将其与现有数据仓库数据库 (`ovirt_engine_history`) 连接。
- 您可以从 Manager 机器迁移数据仓库数据库，然后迁移数据仓库服务。

2.1. 将数据仓库数据库迁移到独立机器

在迁移数据仓库服务之前，迁移数据仓库数据库 (`ovirt_engine_history`)。使用 `engine-backup` 创建数据库备份，并在新数据库计算机上恢复它。有关 `engine-backup` 的更多信息，请运行 `engine-backup --help`。



注意

红帽只支持安装数据仓库数据库、数据仓库服务和 Grafana，它们都与彼此相同，尽管您可以在独立的机器上分别安装这些组件。

新数据库服务器必须安装了 Red Hat Enterprise Linux 8。

在新数据库服务器上启用所需的存储库。

2.1.1. 启用 Red Hat Virtualization Manager 存储库

您需要使用 Red Hat Subscription Manager 登录并注册数据仓库，附加 **Red Hat Virtualization Manager** 订阅并启用 Manager 存储库。

流程

1. 使用 Content Delivery Network 注册您的系统，在提示时输入您的客户门户网站用户名和密码：

```
# subscription-manager register
```



注意

如果您使用 IPv6 网络，请使用 IPv6 转换机制来访问 Content Delivery Network 和 subscription Manager。

2. 查找 **Red Hat Virtualization Manager** 订阅池并记录池 ID：

```
# subscription-manager list --available
```

- 使用池 ID 将订阅附加到系统：

```
# subscription-manager attach --pool=pool_id
```



注意

查看当前附加的订阅：

```
# subscription-manager list --consumed
```

列出所有启用的软件仓库：

```
# dnf repolist
```

- 配置存储库：

```
# subscription-manager repos \
  --disable='*' \
  --enable=rhel-8-for-x86_64-baseos-eus-rpms \
  --enable=rhel-8-for-x86_64-appstream-eus-rpms \
  --enable=rhv-4.4-manager-for-rhel-8-x86_64-rpms \
  --enable=fast-datapath-for-rhel-8-x86_64-rpms \
  --enable=jb-eap-7.4-for-rhel-8-x86_64-rpms \
  --enable=openstack-16.2-cinderlib-for-rhel-8-x86_64-rpms \
  --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

- 将 RHEL 版本设置为 8.6:

```
# subscription-manager release --set=8.6
```

- 启用 **postgresql** 模块的版本 12。

```
# dnf module -y enable postgresql:12
```

- 启用 **nodejs** 模块的版本 14:

```
# dnf module -y enable nodejs:14
```

- 同步安装的软件包，将它们更新至最新可用版本。

```
# dnf distro-sync --nobest
```

其它资源

有关模块和模块流的详情，请参考 [安装、管理和删除用户空间组件中的以下部分](#)。

- [模块流](#)
- [安装软件包前选择流](#)
- [重置模块流](#)

- [切换到更新的流](#)

2.1.2. 将数据仓库数据库迁移到独立机器

流程

1. 在 Manager 中创建数据仓库数据库和配置文件的备份：

```
# engine-backup --mode=backup --scope=grafanadb --scope=dwhdb --scope=files --
file=file_name --log=log_file_name
```

2. 将备份文件从 Manager 复制到新机器：

```
# scp /tmp/file_name root@new.dwh.server.com:/tmp
```

3. 在新机器上安装 **engine-backup**：

```
# dnf install ovirt-engine-tools-backup
```

4. 安装 PostgreSQL 服务器软件包：

```
# dnf install postgresql-server postgresql-contrib
```

5. 初始化 PostgreSQL 数据库，启动 **postgresql** 服务，并确保该服务在引导时启动：

```
# su - postgres -c 'initdb'
# systemctl enable postgresql
# systemctl start postgresql
```

6. 在新计算机上恢复数据仓库数据库。file_name 是从 Manager 复制的备份文件。

```
# engine-backup --mode=restore --scope=files --scope=grafanadb --scope=dwhdb --
file=file_name --log=log_file_name --provision-dwh-db --restore-permissions
```

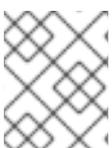
数据仓库数据库现在托管在与管理器托管的独立计算机上。成功恢复数据仓库数据库后，提示您运行 **engine-setup** 命令。在运行此命令之前，请先迁移数据仓库服务。

2.2. 将数据仓库服务迁移到独立机器

您可以将 Red Hat Virtualization Manager 上安装并配置的数据仓库服务迁移到单独的机器。在单独的计算机上托管数据仓库服务有助于减少管理器计算机上的负载。

请注意，这个过程仅迁移数据仓库服务。

要在迁移数据仓库服务前迁移数据仓库数据库(**ovirt_engine_history**)，请参阅将 [数据仓库数据库迁移到独立机器](#)。



注意

红帽只支持安装数据仓库数据库、数据仓库服务和 Grafana，它们都与彼此相同，尽管您可以在独立的机器上分别安装这些组件。

先决条件

- 您必须在同一台机器上安装并配置了管理器和数据仓库。
- 要设置新的数据仓库机器，您必须有以下内容：
 - Manager 的 `/etc/ovirt-engine/engine.conf.d/10-setup-database.conf` 文件中的密码。
 - 允许从数据仓库计算机访问 Manager 数据库计算机的 TCP 端口 5432。
 - 数据仓库数据库中的 `/etc/ovirt-engine-dwh/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/10-setup-database.conf` 文件中的用户名和密码。
如果您使用 [将数据仓库数据库迁移到一个独立的机器中所述的步骤迁移 ovirt_engine_history](#) 数据库，备份会包括您在该机器上的数据库设置中定义的这些凭证。

安装此场景需要四个步骤：

1. 设置新的数据仓库机器
2. 在管理器机器上停止数据仓库服务
3. 配置新的数据仓库
4. 在 Manager 计算机上禁用数据仓库软件包

2.2.1. 设置新的数据仓库机器

启用 Red Hat Virtualization 软件仓库并在 Red Hat Enterprise Linux 8 机器上安装数据仓库设置软件包：

1. 启用所需的软件仓库：
 - a. 使用 Content Delivery Network 注册您的系统，在提示时输入您的客户门户网站用户名和密码：

```
# subscription-manager register
```

- b. 查找 **Red Hat Virtualization Manager** 订阅池并记录池 ID：

```
# subscription-manager list --available
```

- c. 使用池 ID 将订阅附加到系统：

```
# subscription-manager attach --pool=pool_id
```

- d. 配置存储库：

```
# subscription-manager repos \
  --disable='*' \
  --enable=rhel-8-for-x86_64-baseos-eus-rpms \
  --enable=rhel-8-for-x86_64-appstream-eus-rpms \
  --enable=rhv-4.4-manager-for-rhel-8-x86_64-rpms \
  --enable=fast-datapath-for-rhel-8-x86_64-rpms \
  --enable=jb-eap-7.4-for-rhel-8-x86_64-rpms
```

```
# subscription-manager release --set=8.6
```

2. 启用 **pki-deps** 模块。

```
# dnf module -y enable pki-deps
```

3. 确保当前安装的所有软件包都为最新版本：

```
# dnf upgrade --nobest
```

4. 安装 **ovirt-engine-dwh-setup** 软件包：

```
# dnf install ovirt-engine-dwh-setup
```

2.2.2. 在 Manager 机器上停止数据仓库服务

流程

1. 停止数据仓库服务：

```
# systemctl stop ovirt-engine-dwhd.service
```

2. 如果数据库托管在远程计算机上，您必须通过编辑 `postgres.conf` 文件来手动授予访问权限。编辑 `/var/lib/pgsql/data/postgresql.conf` 文件并修改 `listen_addresses` 行，使其与以下内容匹配：

```
listen_addresses = '*'
```

如果该行不存在或已被注释掉，请手动添加。

如果数据库托管在 Manager 机器上，且是在 Red Hat Virtualization Manager 完全设置过程中配置的，则默认授予访问权限。

3. 重启 `postgresql` 服务：

```
# systemctl restart postgresql
```

2.2.3. 配置新的数据仓库（Data Warehouse 机器）

本节中显示的选项或设置顺序可能因您的环境而异。

1. 如果您要同时将 **ovirt_engine_history** 数据库和数据仓库服务迁移到同一个计算机上，请运行以下命令，否则继续下一步。

```
# sed -i '/^ENGINE_DB_/d' \
    /etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/10-setup-database.conf

# sed -i \
    -e 's;^\(OVESETUP_ENGINE_CORE/enable=bool\):True;1:False;' \
    -e '/^OVESETUP_CONFIG/fqdn/d' \
    /etc/ovirt-engine-setup.conf.d/20-setup-ovirt-post.conf
```

2. 删除 `apache/grafana` PKI 文件，以便 **engine-setup** 使用正确值重新生成这些文件：

```
# rm -f \
```

```

/etc/pki/ovirt-engine/certs/apache.cer \
/etc/pki/ovirt-engine/certs/apache-grafana.cer \
/etc/pki/ovirt-engine/keys/apache.key.nopass \
/etc/pki/ovirt-engine/keys/apache-grafana.key.nopass \
/etc/pki/ovirt-engine/apache-ca.pem \
/etc/pki/ovirt-engine/apache-grafana-ca.pem

```

3. 运行 **engine-setup** 命令，开始在机器上配置数据仓库：

```
# engine-setup
```

4. 按 **Enter** 接受自动检测到的主机名，或者输入替代主机名并按 **Enter**：

```
Host fully qualified DNS name of this server [autodetected host name]:
```

5. 按 **Enter** 键自动配置防火墙，或者键入 **No** 并按 **Enter** 来维护现有设置：

```

Setup can automatically configure the firewall on this system.
Note: automatic configuration of the firewall may overwrite current settings.
Do you want Setup to configure the firewall? (Yes, No) [Yes]:

```

如果您选择自动配置防火墙，且没有防火墙管理器处于活动状态，系统会提示您从支持的选项列表中选择您所选的防火墙管理器。输入防火墙管理器的名称，然后按 **Enter**。即使只列出了一个选项，也是如此。

6. 为 Manager 输入完全限定域名和密码。按 **Enter** 键接受其他字段中的默认值：

```

Host fully qualified DNS name of the engine server []: engine-fqdn
Setup needs to do some actions on the remote engine server. Either automatically, using ssh
as root to access it, or you will be prompted to manually perform each such action.
Please choose one of the following:
1 - Access remote engine server using ssh as root
2 - Perform each action manually, use files to copy content around
(1, 2) [1]:
ssh port on remote engine server [22]:
root password on remote engine server engine-fqdn: password

```

7. 输入 Manager 数据库计算机的 FQDN 和密码。按 **Enter** 键接受其他字段中的默认值：

```

Engine database host []: manager-db-fqdn
Engine database port [5432]:
Engine database secured connection (Yes, No) [No]:
Engine database name [engine]:
Engine database user [engine]:
Engine database password: password

```

8. 确认安装设置：

```
Please confirm installation settings (OK, Cancel) [OK]:
```

数据仓库服务现在在远程计算机上配置。继续在管理器计算机上禁用数据仓库服务。



注意

如果要将数据仓库抽样范围改为远程服务器上的推荐规模，请参阅[更改数据仓库采样扩展](#)。

2.2.4. 在 Manager 机器上禁用数据仓库服务

先决条件

- Manager 机器上的 Grafana 服务被禁用：

```
# systemctl disable --now grafana-server.service
```

流程

1. 在 Manager 机器中重启 Manager：

```
# service ovirt-engine restart
```

2. 运行以下命令修改文件 `/etc/ovirt-engine-setup.conf.d/20-setup-ovirt-post.conf`，并将选项设置为 **False**：

```
# sed -i \
-e 's;^\(OVESETUP_DWH_CORE/enable=bool\):True;\1:False;' \
-e 's;^\(OVESETUP_DWH_CONFIG/remoteEngineConfigured=bool\):True;\1:False;' \
/etc/ovirt-engine-setup.conf.d/20-setup-ovirt-post.conf

# sed -i \
-e 's;^\(OVESETUP_GRAFANA_CORE/enable=bool\):True;\1:False;' \
/etc/ovirt-engine-setup.conf.d/20-setup-ovirt-post.conf
```

3. 禁用数据仓库服务：

```
# systemctl disable ovirt-engine-dwhd.service
```

4. 删除数据仓库文件：

```
# rm -f /etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/*.conf /var/lib/ovirt-engine-dwh/backups/*
```

数据仓库服务现在托管在管理器之外的独立机器上。

2.3. 更改数据仓库采样规模

Red Hat Virtualization 需要数据仓库。它可以在与 Manager 相同的机器上安装和配置，也可以在可访问 Manager 的单独机器上安装和配置。所有设置都不需要默认数据保留设置，因此 **engine-setup** 提供两个数据抽样扩展：**Basic** 和 **Full**。

- **Full** 使用默认值用于数据保留设置（在 [Application Settings for the Data Warehouse service in ovirt-engine-dwhd.conf](#) 中列出），在远程主机上安装 Data Warehouse 时这是推荐的设置。

- **Basic** 会将 `DWH_TABLES_KEEP_HOURLY` 的值减小到 **720**, `DWH_TABLES_KEEP_DAILY` 的值减小到 **0**, 从而减轻了 Manager 集群的负载。在同一机器上安装 Manager 和数据仓库时, 请使用 **Basic**。

抽样扩展会在安装过程中由 `engine-setup` 配置 :

```
==== MISC CONFIGURATION ====
```

```
Please choose Data Warehouse sampling scale:
```

```
(1) Basic
```

```
(2) Full
```

```
(1, 2)[1]:
```

您可以使用 `--reconfigure-dwh-scale` 选项再次运行 `engine-setup`, 从而更改抽样扩展。

更改数据仓库采样规模

```
# engine-setup --reconfigure-dwh-scale
```

```
[...]
```

```
Setup can automatically configure the firewall on this system.
```

```
Note: automatic configuration of the firewall may overwrite current settings.
```

```
Do you want Setup to configure the firewall? (Yes, No) [Yes]:
```

```
[...]
```

```
Perform full vacuum on the oVirt engine history
```

```
database ovirt_engine_history@localhost?
```

```
This operation may take a while depending on this setup health and the  
configuration of the db vacuum process.
```

```
See https://www.postgresql.org/docs/12/static/sql-vacuum.html
```

```
(Yes, No) [No]:
```

```
[...]
```

```
Setup can backup the existing database. The time and space required for the database backup  
depend on its size. This process takes time, and in some cases (for instance, when the size is few  
GBs) may take several hours to complete.
```

```
If you choose to not back up the database, and Setup later fails for some reason, it will not be able to  
restore the database and all DWH data will be lost.
```

```
Would you like to backup the existing database before upgrading it? (Yes, No) [Yes]:
```

```
[...]
```

```
Please choose Data Warehouse sampling scale:
```

```
(1) Basic
```

```
(2) Full
```

```
(1, 2)[1]: 2
```

```
[...]
```

```
During execution engine service will be stopped (OK, Cancel) [OK]:
```

```
[...]
```

```
Please confirm installation settings (OK, Cancel) [OK]:
```

您还可以根据需要调整单个数据保留设置, 如[数据仓库服务的应用程序设置](#)中所述。

第 3 章 关于历史记录数据库

3.1. 历史记录数据库概述

Red Hat Virtualization 包括一个全面的管理历史数据库，可用于报告应用程序在数据中心、集群和主机级别生成报告。本章提供了用来设置对历史记录数据库的查询的信息。

Red Hat Virtualization Manager 使用 **PostgreSQL 12.x** 作为数据库平台，用于存储有关虚拟化环境状态、配置和性能的信息。在安装时，Red Hat Virtualization Manager 会创建一个名为 **engine** 的 PostgreSQL 数据库。

安装 **ovirt-engine-dwh** 软件包会创建一个名为 **ovirt_engine_history** 的第二个数据库，它每分钟包含了从 **engine** 操作数据库每分钟收集的历史配置信息和统计指标。跟踪对数据库的更改将提供有关数据库中对象的信息，让用户能够分析活动、提高性能和解决困难。



警告

ovirt_engine_history 数据库中数据的复制由 Red Hat Virtualization Manager Extract Transform Load Service, **ovirt-engine-dwhd** 来执行。该服务基于 Talend Open Studio，是一种数据集成工具。此服务配置为在数据仓库软件包设置过程中自动启动。这是一个 Java 程序，负责从 **engine** 数据库提取数据，将数据转换为历史记录数据库标准，并将它加载到 **ovirt_engine_history** 数据库中。

不得停止 **ovirt-engine-dwhd** 服务。

ovirt_engine_history 数据库架构随着时间而变化。数据库包括一组数据库视图，以提供受支持的版本化的 API，具有一致的结构。视图是包含数据库查询结果集的虚拟表。数据库以 **SELECT** 语句形式存储视图的定义。**SELECT** 语句的结果会填充视图返回的虚拟表。用户以与引用表相同的方式引用 **PL/PGSQL** 语句中的视图名称。

3.2. 跟踪配置历史

Red Hat Virtualization History 数据库（称为 **ovirt_engine_history**）中的数据可用于跟踪引擎数据库。

ETL 服务 **ovirt-engine-dwhd** 跟踪三种类型的更改：

- 将新的实体添加到引擎数据库中 - ETL Service 会将更改复制到 **ovirt_engine_history** 数据库作为新条目。
- 现有实体已更新 - ETL 服务将更改复制到 **ovirt_engine_history** 数据库作为新条目。
- 实体从引擎数据库中移除 - **ovirt_engine_history** 数据库中的新条目会标记为已删除的对应实体。删除的实体仅标记为已删除。

ovirt_engine_history 数据库中的配置表与引擎数据库中对应的表有多种方式的不同。最明显的区别是它们包含较少的配置列。这是因为，某些配置项目与其它配置项目相比并不重要，因此处于对数据库大小的考虑不会保存这些项。另外，引擎数据库中的几个表列会出现在 **ovirt_engine_history** 的单个表中，具有不同的列名称，使查看数据更加方便和理解。所有配置表都包含：

- 用于指示实体配置版本的 **history_id**；

- 用于指示何时在系统中添加实体的 `create_date` 字段；
- 用于指示实体何时更改的 `update_date` 字段；以及
- `delete_date` 字段表示从系统中删除实体的日期。

3.3. 记录工资历史

ETL 服务每分钟将数据收集数据到统计表中。在过去 24 小时内，数据存储一次都会存储，但至少可以存储为 48 小时，具体取决于上次删除作业的时间。分钟以上的数据超过两小时，将汇总为每小时数据，并且存储两个月。超过 2 天的每小时数据聚合为每日数据，并且存储 5 年。

每小时数据和每日数据可在每小时和每日表中找到。

每个统计 datum 保存在其相应的聚合级别表中：示例、每小时和每日历史记录。所有历史记录表也包含一个 `history_id` 列，用于唯一标识行。表引用主机的配置版本，以便启用与过去配置相关的实体统计信息的报告。

3.4. OVIRT-ENGINE-DWHD.CONF 中的数据仓库服务的应用程序设置

下表列出了为数据仓库服务配置应用设置的选项。这些选项在 `/usr/share/ovirt-engine-dwh/services/ovirt-engine-dwhd/ovirt-engine-dwhd.conf` 文件中提供。在 `/etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/` 下配置覆盖文件中的默认值。在保存更改后重启数据仓库服务。

表 3.1. `ovirt-engine-dwhd.conf` 应用程序设置变量

变量名称	默认值	备注
<code>DWH_DELETE_JOB_HOUR</code>	3	运行删除作业的时间。指定一个 0 到 23 之间的值，其中 0 为午夜。
<code>DWH_SAMPLING</code>	60	数据收集到统计表的时间间隔（以秒为单位）。
<code>DWH_TABLES_KEEP_SAMPLES</code>	24	存储 <code>DWH_SAMPLING</code> 数据的小时数。超过 2 小时的数据会被汇总到每小时数据。
<code>DWH_TABLES_KEEP_HOURLY</code>	1440	每小时数据存储的小时数。默认值为 60 天。超过 2 天的每小时数据将聚合到每日数据。
<code>DWH_TABLES_KEEP_DAILY</code>	43800	每日数据存储的小时数。默认值为 5 年。
<code>DWH_ERROR_EVENT_INTERVAL</code>	300000	最小间隔，以毫秒为单位，将错误推送到 Manager 的 <code>audit.log</code> 中。

3.5. 跟踪标签历史记录

ETL Service 每分钟会收集管理门户中显示的标签信息，并将这些数据存储在标签历史表中。ETL 服务跟踪五种更改：

- 在管理门户中创建了标签 - ETL Service 复制标签详细信息，与标签树中的其他对象定位。
- 实体连接到管理门户中的标签树 - ETL Service 复制添加的 **ovirt_engine_history** 数据库作为新条目。
- 标签被更新 - ETL Service 将标签详细信息的更改复制到 **ovirt_engine_history** 数据库作为新条目。
- 从管理门户中移除了实体或标签分支 - **ovirt_engine_history** 数据库会将对应的标签和关系标记为在新条目中被删除。删除了的标签和关系仅标记为删除或分离。
- 标签分支被移动 - 对应的标签和关系会作为新条目进行更新。移动的标签和关系仅标记为更新。

3.6. 允许只读访问 HISTORY 数据库

要允许在不允许编辑的情况下访问历史记录数据库，您必须创建一个只读 PostgreSQL 用户，该用户可登录并从 **ovirt_engine_history** 数据库读取。这个过程必须在安装历史记录数据库的系统中执行。

允许只读访问 History 数据库

1. 以 root 用户身份登录 postgres 用户 shell :

```
# su - postgres
```

2. 创建用户以对历史记录数据库具有只读访问权限 :

```
# psql -U postgres -c "CREATE ROLE username WITH LOGIN ENCRYPTED PASSWORD 'password;' -d ovirt_engine_history
```

3. 授予新创建的用户权限以连接到历史记录数据库 :

```
# psql -U postgres -c "GRANT CONNECT ON DATABASE ovirt_engine_history TO username;"
```

4. 授予新创建的用户使用 **public** 模式 :

```
# psql -U postgres -c "GRANT USAGE ON SCHEMA public TO username;"  
ovirt_engine_history
```

5. 生成授予新建用户的剩余权限，并将其保存到文件中 :

```
# psql -U postgres -c "SELECT 'GRANT SELECT ON ' || relname || ' TO username;' FROM  
pg_class JOIN pg_namespace ON pg_namespace.oid = pg_class.relnamespace WHERE  
nspname = 'public' AND relkind IN ('r', 'v');" --pset=tuples_only=on ovirt_engine_history >  
grant.sql
```

6. 使用您在上一步中创建的文件为新创建的用户授予权限 :

```
# psql -U postgres -f grant.sql ovirt_engine_history
```

7. 删除您用来为新创建的用户授予权限的文件 :

```
# rm grant.sql
```

- 按 **Ctrl+d** 推出 postgres 用户 shell
- 为新创建的用户将以下行添加到 `/var/lib/pgsql/data/pg_hba.conf` 中（在以 **local all all** 开始的行之前）：

```
# TYPE DATABASE          USER          ADDRESS          METHOD
host  ovirt_engine_history  username      0.0.0.0/0        md5
host  ovirt_engine_history  username      ::0/0            md5
local all                all           peer
```

- 重新载入 PostgreSQL 服务：

```
# systemctl reload postgresql
```

- 测试只读用户的访问权限：

```
# su - postgres -c 'psql -U username ovirt_engine_history -h localhost'
Password for user username:
psql (9.2.23)
Type "help" for help.

ovirt_engine_history=>
```

- 要退出 `ovirt_engine_history` 数据库，请输入 `\q`。

对于 `ovirt_engine_history` 数据库中的表和视图，只读用户的 **SELECT** 语句会成功，而修改会失败。

3.7. 统计历史视图

统计数据数据 **每小时**、**每天**，以及**示例**视图。

要查询统计视图，请运行 **SELECT * FROM *view_name*_[hourly|daily|samples];**。例如：

```
# SELECT * FROM v4_4_statistics_hosts_resources_usage_daily;
```

要列出所有可用视图，请运行：

```
# \dv
```

3.7.1. 启用调试模式

您可以启用调试模式，在 `/var/log/ovirt-engine-dwh/ovirt-engine-dwhd.log` 文件中记录日志抽样、每小时和每日作业时间。这对于检查 ETL 进程非常有用。默认情况下禁用调试模式。

- 登录 Manager 机器并创建配置文件（例如，`/etc/ovirt-engine-dwh/ovirt-engine-dwhd/logging.conf`）。
- 在配置文件中添加以下行：

```
DWH_AGGREGATION_DEBUG=true
```

- 重启 `ovirt-engine-dwhd` 服务：

```
# systemctl restart ovirt-engine-dwhd.service
```

```
To disable debug mode, delete the configuration file and restart the service.
// removed note
```

3.7.2. Storage Domain Statistics Views

表 3.2. 系统中每个存储域的历史统计

名称	类型	描述	索引
history_id	bigint	表中此行的唯一 ID。	否
history_datetime	date	此历史记录行的时间戳 (根据聚合级别, 舍入到分钟、小时、天)。	是
storage_domain_id	uuid	系统中存储域的唯一 ID。	是
storage_domain_status	smallint	存储域状态。	否
seconds_in_status	整数	存储域处于状态的秒数, 如聚合期间的状态列中所示。例如, 如果存储域在一分钟内“活跃”了 55 秒, “不活跃” 5 秒, 则在同一分钟的表中报告两行。一行是 Active (活跃) 状态 (seconds_in_status 为 55), 另一个行是 Inactive (不活跃) 状态 (seconds_in_status 为 5)。	否
minutes_in_status	numeric(7,2)	存储域处于状态的分钟总数, 如聚合期间的状态列中所示。例如, 如果存储域在一小时内 55 分钟“活跃”, 5 分钟“不活跃”, 则在同一小时的表中将报告两行。一行为 Active 状态 (minutes_in_status 为 55), 另一行是 Inactive 状态 (minutes_in_status 为 5)。	否

名称	类型	描述	索引
available_disk_size_gb	整数	磁盘上可用（未使用的）容量，以 GB 为单位表示。	否
used_disk_size_gb	整数	磁盘上使用的总容量，以 GB 为单位表示。	否
storage_configuration_version	整数	示例时存储域配置版本。这与 v4_4_configuration_history_storage_domains 视图中的 history_id 值相同，它可用于加入它们。	是

3.7.3. 主机统计信息视图

表 3.3. 系统中每个主机的历史统计

名称	类型	描述	索引
history_id	bigint	表中此行的唯一 ID。	否
history_datetime	date	此历史记录行的时间戳（根据聚合级别，舍入到分钟、小时、天）。	是
host_id	uuid	系统中主机的唯一 ID。	是
host_status	smallint	<ul style="list-style-type: none"> ● -1 - 未知状态（仅用于指明 ETL 存在问题）。请通知红帽支持。） ● 1 - 在线 ● 2 - 维护 ● 3 - 有问题 	否

名称	类型	描述	索引
seconds_in_status	整数	主机处于聚合期间 status 列中显示的秒数。例如，如果主机在一分钟内有 55 秒在线，5 秒下线，则这分钟包括两个行。一个状态将为 Up seconds_in_status 为 55，另一个状态为 Down ，seconds_in_status 为 5。	否
minutes_in_status	numeric(7,2)	主机处于聚合期间 status 列中显示的分钟数。例如，如果主机在一小时内为 55 分钟，并且停机 5 分钟，则两个行将显示为本小时。一个状态将为 Up minutes_in_status 为 55，另一个状态为 Down ，minutes_in_status 为 5。	否
memory_usage_percent	smallint	主机上已用内存的百分比。	否
max_memory_usage	smallint	聚合期间的最大内存用量，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
ksm_shared_memory_mb	bigint	内核共享内存大小，以 MB(MB)表示主机使用。	否
max_ksm_shared_memory_mb	bigint	聚合周期内的最大 KSM 内存用量以 MB(MB)表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
cpu_usage_percent	smallint	主机上使用的 CPU 百分比。	否

名称	类型	描述	索引
max_cpu_usage	smallint	聚合周期的最大 CPU 使用量，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
kvm_cpu_percent	smallint	主机上的 CPU 百分比 kvm 使用。	否
max_kvm_cpu_percent	smallint	聚合周期的最大 KVM 使用量，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
active_vms	smallint	这个聚合的活动虚拟机的平均数量。	否
max_active_vms	smallint	聚合期间内虚拟机的最大数量。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
total_vms	smallint	此聚合主机上所有虚拟机的平均数量。	否
max_total_vms	smallint	聚合期间虚拟机的最大数量。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
total_vms_vcpus	整数	分配给主机的 vCPU 总数。	否
max_total_vms_vcpus	整数	聚合期间的最大虚拟机 vCPU 数量。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
cpu_load	整数	主机的 CPU 负载。	否

名称	类型	描述	索引
max_cpu_load	整数	聚合期间的最大 CPU 负载。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
system_cpu_usage_percent	smallint	主机上使用的 CPU 百分比。	否
max_system_cpu_usage_percent	smallint	聚合周期的最大系统 CPU 使用量，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
user_cpu_usage_percent	smallint	在主机上使用了用户 CPU 百分比。	否
max_user_cpu_usage_percent	smallint	聚合周期的最大用户 CPU 使用量，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
swap_used_mb	整数	以 MB(MB)为单位使用主机的交换大小。	否
max_swap_used_mb	整数	以 MB(MB)表示的聚合周期内的最大用户交换大小使用主机，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
host_configuration_version	整数	示例时主机配置版本。示例时主机配置版本。这与 v4_4_configuration_history_hosts 视图中的 history_id 值相同，它可用于加入它们。	是

3.7.4. 主机接口统计信息视图

表 3.4. 系统中每个主机网络接口的历史统计

名称	类型	描述	索引
history_id	bigint	表中此行的唯一 ID。	否
history_datetime	date	此历史记录视图的时间戳（根据聚合级别，舍入为分钟、小时、天）。	是
host_interface_id	uuid	系统中接口的唯一标识符。	是
receive_rate_percent	smallint	在主机上使用的接收率百分比。	否
max_receive_rate_percent	smallint	聚合周期的最大接收率，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
transmit_rate_percent	smallint	在主机上使用传输率百分比。	否
max_transmit_rate_percent	smallint	聚合期间的最大传输率，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
received_total_byte	bigint	主机接收的字节数。	否
transmitted_total_byte	bigint	从主机传输的字节数。	否
host_interface_configuration_version	整数	示例时主机接口配置版本。这与 v4_4_configuration_history_hosts_interfaces 视图中的 history_id 值相同，它可用于加入它们。	是

3.7.5. 虚拟机统计视图

表 3.5. 系统中每个虚拟机的历史统计

名称	类型	描述	索引
history_id	bigint	表中此行的唯一 ID。	否

名称	类型	描述	索引
history_datetime	date	此历史记录行的时间戳 (根据聚合级别, 舍入到分钟、小时、天)。	是
vm_id	uuid	系统中虚拟机的唯一 ID。	是
vm_status	smallint	<ul style="list-style-type: none"> ● -1 - 未知状态 (仅用于指明 ETL 的问题。请通知红帽支持。) ● 0 - 下线 ● 1 - 在线 ● 2 - 暂停 ● 3 - 有问题 	否
seconds_in_status	整数	虚拟机处于聚合期间 status 列中显示的秒数。例如, 如果虚拟机在一分钟内 55 秒在线, 5 秒下线, 则这分钟会有两个行。一个状态将为 Up 和 seconds_in_status, 另一个状态为 Down, seconds_in_status 为 5。	否
minutes_in_status	numeric(7,2)	虚拟机处于聚合期间状态中显示的分钟总数。例如, 如果虚拟机在一小时内为 55 分钟上线, 5 分钟下线, 则这个小时会有两个行。一个状态将为 Up 和 minutes_in_status, 另一个状态为 Down, minutes_in_status 为 5。	否
cpu_usage_percent	smallint	虚拟机使用的 CPU 的百分比。	否

名称	类型	描述	索引
max_cpu_usage	smallint	聚合周期的最大 CPU 使用量，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
memory_usage_percent	smallint	虚拟机中使用的内存百分比。必须在虚拟机上安装客户机工具，以便记录内存用量。	否
max_memory_usage	smallint	聚合期间的最大内存用量，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。必须在虚拟机上安装客户机工具，以便记录内存用量。	否
user_cpu_usage_percent	smallint	在主机上使用了用户 CPU 百分比。	否
max_user_cpu_usage_percent	smallint	聚合周期的最大用户 CPU 使用量，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
system_cpu_usage_percent	smallint	在主机上使用的系统 CPU 百分比。	否
max_system_cpu_usage_percent	smallint	聚合周期的最大系统 CPU 使用量，以百分比表示。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
vm_ip	text	第一个 NIC 的 IP 地址。仅在安装了客户机代理时才显示。	否
currently_running_on_host	uuid	虚拟机在其上运行的主机的唯一 ID。	否

名称	类型	描述	索引
current_user_id	uuid	如果安装了客户机代理，登录到虚拟机控制台的用户唯一 ID。	否
disks_usage	text	磁盘描述。文件系统类型、挂载点、总大小和使用的空间。	否
vm_configuration_version	整数	示例时虚拟机配置版本。这与 v4_4_configuration_history_vms 视图中的 history_id 值相同。	是
current_host_configuration_version	整数	示例时主机配置版本。这与 v4_4_configuration_history_hosts 视图中的 history_id 值相同，它可用于加入它们。	是
memory_buffered_kb	bigint	虚拟机上缓冲的内存量 (KB)。	否
memory_cached_kb	bigint	虚拟机上缓存的内存量 (KB)。	否
max_memory_buffered_kb	bigint	聚合周期的最大缓冲内存 (以 KB 为单位)。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
max_memory_cached_kb	bigint	聚合期间的最大缓存内存 (以 KB 为单位)。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否

3.7.6. 虚拟机接口统计信息视图

表 3.6. 系统中虚拟机网络接口的历史统计

名称	类型	描述	索引
history_id	整数	表中此行的唯一 ID。	否
history_datetime	date	此历史记录行的时间戳 (根据聚合级别, 舍入到分钟、小时、天)。	是
vm_interface_id	uuid	系统中接口的唯一 ID。	是
receive_rate_percent	smallint	在主机上使用的接收率百分比。	否
max_receive_rate_percent	smallint	聚合周期的最大接收率, 以百分比表示。对于每小时聚合, 这是收集最多的样本值。对于每日聚合, 它是每小时的平均值最大值。	否
transmit_rate_percent	smallint	在主机上使用传输率百分比。	否
max_transmit_rate_percent	smallint	聚合期间的最大传输率, 以百分比表示。对于每小时聚合, 这是收集最多的样本值。对于每日聚合, 它是每小时平均率上限。	否
received_total_byte	bigint	虚拟机接收的字节数。	否
transmitted_total_byte	bigint	从虚拟机传输的字节数。	否
vm_interface_configuration_version	整数	抽样时虚拟机接口配置版本。这与 v4_4_configuration_history_vms_interfaces 视图中的 history_id 值相同, 它可用于加入它们。	是

3.7.7. 虚拟磁盘统计视图

表 3.7. 系统中虚拟磁盘的历史统计

名称	类型	描述	索引
history_id	bigint	表中此行的唯一 ID。	否

名称	类型	描述	索引
history_datetime	date	此历史记录行的时间戳（根据聚合级别，舍入到分钟、小时、天）。	是
vm_disk_id	uuid	系统中磁盘的唯一 ID。	是
vm_disk_status	smallint	<ul style="list-style-type: none"> ● 0 - 未分配 ● 1 - OK ● 2 - 锁定 ● 3 - 无效 ● 4 - 非法 	否
seconds_in_status	整数	虚拟磁盘处于聚合期间 status 列中显示的秒数。例如，如果虚拟机在一分钟内 55 被锁定，5 秒正常，则这分钟会有两个行。一个状态将为 Locked seconds_in_status 为 55，另一个状态为 OK，seconds_in_status 为 5。	否
minutes_in_status	numeric(7,2)	虚拟磁盘处于聚合期间状态中显示的分钟总数。例如，如果虚拟磁盘在一小时内 55 分钟被锁定，5 分钟正常，则这小时会有两个行。一个状态将为 Locked minutes_in_status 为 55，另一个状态为 OK，minutes_in_status 为 5。	否
vm_disk_actual_size_mb	整数	分配给磁盘的实际大小。	否
read_rate_bytes_per_second	整数	磁盘读取率（以字节/秒为单位）。	否

名称	类型	描述	索引
max_read_rate_bytes_per_second	整数	聚合期间的最大读取率。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
read_ops_total_count	numeric(20,0)	从 vm start 开始，将 I/O 操作读取到磁盘。	否
read_latency_seconds	numeric(18,9)	虚拟磁盘读取延迟（以秒为单位）。	否
write_rate_bytes_per_second	整数	对磁盘的写入率（以字节/秒为单位）。	否
max_read_latency_seconds	numeric(18,9)	聚合周期的最大读取延迟，单位为秒。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
max_write_rate_bytes_per_second	整数	聚合期间的最大写入率。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
write_ops_total_count	numeric(20,0)	从 vm 启动后向磁盘写入 I/O 操作。	否
write_latency_seconds	numeric(18,9)	虚拟磁盘写入延迟（以秒为单位）。	否
max_write_latency_seconds	numeric(18,9)	聚合期间的最大写入延迟（以秒为单位）。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
flush_latency_seconds	numeric(18,9)	虚拟磁盘 flush 延迟（以秒为单位）。	否

名称	类型	描述	索引
max_flush_latency_seconds	numeric(18,9)	聚合周期的最大 flush 延迟（以秒为单位）。对于每小时聚合，这是收集最多的样本值。对于每日聚合，它是每小时的平均值最大值。	否
vm_disk_configuration_version	整数	抽样时的虚拟磁盘配置版本。这与 v4_4_configuration_history_vms_disks 视图中的 history_id 值相同，它可用于加入它们。	是

3.8. 配置历史视图

要查询配置视图，请运行 **SELECT * FROM view_name;**。例如：

```
# SELECT * FROM v4_4_configuration_history_datacenters;
```

要列出所有可用视图，请运行：

```
# \dv
```



注意

delete_date 不会显示在最新的视图中，因为这些视图提供了按照定义未被删除的最新实体配置。

3.8.1. 数据中心配置

下表显示了系统中数据中心的配置历史记录参数。

表 3.8. v4_4_configuration_history_datacenters

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。这与 v4_4_configuration_history_clusters 视图中的 datacenter_configuration_version 的值相同，它可用于加入它们。	否

名称	类型	描述	索引
datacenter_id	uuid	系统中数据中心的唯一 ID。	是
datacenter_name	character varying(40)	数据中心的名称，如编辑对话框中显示。	否
datacenter_description	character varying(4000)	数据中心的描述，如编辑对话框中显示。	否
is_local_storage	布尔值	指明数据中心是否使用本地存储的标志。	否
create_date	使用时区的时间戳	这个实体被添加到系统的日期。	否
update_date	使用时区的时间戳	这个实体已在系统中更改的日期。	否
delete_date	使用时区的时间戳	这个实体已从系统中删除的日期。	否

3.8.2. 数据中心存储域映射

下表显示了系统中存储域和数据中心之间的关系。

表 3.9. v4_4_map_history_datacenters_storage_domains

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。这与 v4_4_configuration_history_clusters 视图中的 datacenter_configuration_version 的值相同，它可用于加入它们。	否
storage_domain_id	uuid	系统中这个存储域的唯一 ID。	是
datacenter_id	uuid	系统中数据中心的唯一 ID。	否
attach_date	使用时区的时间戳	存储域附加到数据中心的日期。	否

名称	类型	描述	索引
detach_date	使用时区的时间戳	存储域从数据中心分离的日期。	否

3.8.3. 存储域配置

下表显示了系统中存储域的配置历史记录参数。

表 3.10. v4_4_configuration_history_storage_domains

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。这与存储域统计视图中的 storage_configuration_version 值相同，并可用于加入它们。	否
storage_domain_id	uuid	系统中这个存储域的唯一 ID。	是
storage_domain_name	character varying(250)	存储域名称。	否
storage_domain_type	smallint	* 0 - 数据 (Master) * 1 - 数据 * 2 - ISO * 3 - 导出	否
storage_type	smallint	* 0 - 未知 * 1 - NFS * 2 - FCP * 3 - iSCSI * 4 - 本地 * 6 - 所有	否
create_date	使用时区的时间戳	这个实体被添加到系统的日期。	否
update_date	使用时区的时间戳	这个实体已在系统中更改的日期。	否

名称	类型	描述	索引
delete_date	使用时区的时间戳	这个实体已从系统中删除的日期。	否

3.8.4. 集群配置

下表显示了系统中集群的配置历史记录参数。

表 3.11. v4_4_configuration_history_clusters

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。这与 v4_4_configuration_history_hosts 和 v4_4_configuration_history_vms 视图中的 cluster_configuration_version 的值相同，它可用于加入它们。	否
cluster_id	uuid	此集群所在的数据中心的唯一标识符。	是
cluster_name	character varying(40)	集群名称，如编辑对话框中显示。	否
cluster_description	character varying(4000)	按照编辑对话框中所定义。	否
datacenter_id	uuid	此集群所在的数据中心的唯一标识符。	是
cpu_name	character varying(255)	如编辑对话框中所示。	否
count_threads_as_cores	布尔值	如果主机可以运行虚拟机，超过主机中的内核数大于主机中的内核数，则 Count Threads As Cores 列会显示 true 。	否
compatibility_version	character varying(40)	如编辑对话框中所示。	否

名称	类型	描述	索引
datacenter_configuration_version	整数	在创建或更新时，数据中心配置版本。在创建或更新时，数据中心配置版本。这与 v4_4_configuration_history_datacenters 视图中的 history_id 值相同，它可用于加入它们。	否
create_date	使用时区的时间戳	这个实体被添加到系统的日期。	否
update_date	使用时区的时间戳	这个实体已在系统中更改的日期。	否
delete_date	使用时区的时间戳	这个实体已从系统中删除的日期。	否

3.8.5. 主机配置

下表显示了系统中主机的配置历史记录参数。

表 3.12. v4_4_configuration_history_hosts

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。这与主机统计视图中的 host_configuration_version 的值相同，并可用于加入它们。	否
host_id	uuid	系统中主机的唯一 ID。	是
host_unique_id	character varying(128)	此字段是主机的物理 UUID 及其 MAC 地址之一的组合，用于检测系统中已注册的主机。	否
host_name	character varying(255)	主机的名称（与编辑对话框中相同）。	否
cluster_id	uuid	此主机所属集群的唯一 ID。	是

名称	类型	描述	索引
host_type	smallint	* 0 - RHEL 主机 * 2 - RHEV Hypervisor 节点	否
fqdn_or_ip	character varying(255)	Red Hat Virtualization Manager 的 DNS 名称或 IP 地址与（如编辑对话框中显示）进行通信。	否
memory_size_mb	整数	主机的物理内存容量，以 MB(MB)表示。	否
swap_size_mb	整数	主机 swap 分区大小。	否
cpu_model	character varying(255)	主机的 CPU 模型。	否
number_of_cores	smallint	主机上的 CPU 内核数。	否
number_of_sockets	smallint	CPU 插槽总数。	否
cpu_speed_mh	numeric(18,0)	主机的 CPU 速度，以 megahertz(MHz)表示。	否
host_os	character varying(255)	主机的操作系统版本。	否
kernel_version	character varying(255)	主机的内核版本。	否
kvm_version	character varying(255)	主机的 KVM 版本。	否
vdsm_version	character varying	主机的 VDSM 版本。	否
vdsm_port	整数	如编辑对话框中所示。	否
threads_per_core	smallint	每个内核的线程总数。	否
hardware_manufacturer	character varying(255)	主机的硬件制造商。	否
hardware_product_name	character varying(255)	主机硬件的产品名称。	否
hardware_version	character varying(255)	主机硬件的版本。	否
hardware_serial_number	character varying(255)	主机硬件的序列号。	否

名称	类型	描述	索引
cluster_configuration_version	整数	创建或更新时的集群配置版本。这与 v4_4_configuration_history_clusters 视图中的 history_id 值相同，它可用于加入它们。	否
create_date	使用时区的时间戳	这个实体被添加到系统的日期。	否
update_date	使用时区的时间戳	这个实体已在系统中更改的日期。	否
delete_date	使用时区的时间戳	这个实体已从系统中删除的日期。	否

3.8.6. 主机接口配置

下表显示了系统中主机接口的配置历史记录参数。

表 3.13. v4_4_configuration_history_hosts_interfaces

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。这与主机接口统计视图中的 host_interface_configuration_version 的值相同，并可用于加入它们。	否
host_interface_id	uuid	系统中这个接口的唯一 ID。	是
host_interface_name	character varying(50)	主机报告的接口名称。	否
host_id	uuid	此接口所属的主机的唯一 ID。	是
host_interface_type	smallint	* 0 - rt18139_pv * 1 - rt18139 * 2 - e1000 * 3 - PV	否

名称	类型	描述	索引
host_interface_speed_bps	整数	接口速度（以位/每秒表示）。	否
mac_address	character varying(59)	接口 MAC 地址。	否
logical_network_name	character varying(50)	与接口关联的逻辑网络。	否
ip_address	character varying(20)	如编辑对话框中所示。	否
gateway	character varying(20)	如编辑对话框中所示。	否
bond	布尔值	用于指示该接口是否为绑定接口的标志。	否
bond_name	character varying(50)	这个接口其中一部分的绑定的名称（如果属于绑定的一部分）。	否
vlan_id	整数	如编辑对话框中所示。	否
host_configuration_version	整数	在创建或更新时主机配置版本。这与 v4_4_configuration_history_hosts 视图中的 history_id 值相同，它可用于加入它们。	否
create_date	使用时区的时间戳	这个实体被添加到系统的日期。	否
update_date	使用时区的时间戳	这个实体已在系统中更改的日期。	否
delete_date	使用时区的时间戳	这个实体已从系统中删除的日期。	否

3.8.7. 虚拟机配置

下表显示了系统中虚拟机的配置历史记录参数。

表 3.14. v4_4_configuration_history_vms

名称	类型	描述	索引
----	----	----	----

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。这与虚拟机统计视图中的 vm_configuration_version 的值相同，并可用于加入它们。	否
vm_id	uuid	系统中此虚拟机的唯一 ID。	是
vm_name	character varying(255)	虚拟机的名称。	否
vm_description	character varying(4000)	如编辑对话框中所示。	否
vm_type	smallint	* 0 - Desktop * 1 - server	否
cluster_id	uuid	此虚拟机所属集群的唯一 ID。	是
template_id	uuid	此虚拟机派生的模板的唯一 ID。模板没有与 Red Hat Virtualization 版本中的历史数据库同步。	否
template_name	character varying(40)	此虚拟机派生的模板的名称。	否
cpu_per_socket	smallint	每个插槽的虚拟 CPU。	否
number_of_sockets	smallint	虚拟 CPU 插槽总数。	否
memory_size_mb	整数	分配给虚拟机的总内存，以兆字节(MB)表示。	否
operating_system	smallint	* 0 - 其他操作系统 * 1 - Windows XP * 3 - Windows 2003 * 4 - Windows 2008 * 5 - Linux	否

名称	类型	* 7 - Red Hat Enterprise Linux 5.x 描述	索引
		* 8 - Red Hat Enterprise Linux 4.x	
		* 9 - Red Hat Enterprise Linux 3.x	
		* 10 - Windows 2003 x64	
		* 11 - Windows 7	
		* 12 - Windows 7 x64	
		* 13 - Red Hat Enterprise Linux 5.x x64	
		* 14 - Red Hat Enterprise Linux 4.x x64	
		* 15 - Red Hat Enterprise Linux 3.x x64	
		* 16 - Windows 2008 x64	
		* 17 - Windows 2008 R2 x64	
		* 18 - Red Hat Enterprise Linux 6.x	
		* 19 - Red Hat Enterprise Linux 6.x x64	
		* 20 - Windows 8	
		* 21 - Windows 8 x64	
		* 23 - Windows 2012 x64	
		* 1001 - Other	
		* 1002 - Linux	
		* 1003 - Red Hat Enterprise Linux 6.x	
		* 1004 - SUSE Linux Enterprise Server 11	
		* 1193 - SUSE Linux Enterprise Server 11	
		* 1252 - Ubuntu Precise Pangolin LTS	

名称	类型	* 1253 - Ubuntu Quantal Quetzal 描述	索引
		* 1254 - Ubuntu Raring Ringtails * 1255 - Ubuntu Saucy Salamander	
default_host	uuid	如编辑对话框中所示，系统中默认主机的 ID。	否
high_availability	布尔值	如编辑对话框中所示。	否
初始化	布尔值	用于指示此虚拟机至少为 Sysprep 初始化目的启动一次的标志。	否
无状态	布尔值	如编辑对话框中所示。	否
fail_back	布尔值	如编辑对话框中所示。	否
usb_policy	smallint	如编辑对话框中所示。	否
time_zone	character varying(40)	如编辑对话框中所示。	否
vm_pool_id	uuid	此虚拟机所属的池的 ID。	否
vm_pool_name	character varying(255)	虚拟机的名称。	否
created_by_user_id	uuid	创建此虚拟机的用户的 ID。	否
cluster_configuration_version	整数	创建或更新时的集群配置版本。这与 v4_4_configuration_history_clusters 视图中的 history_id 值相同，它可用于加入它们。	否
default_host_configuration_version	整数	在创建或更新时主机配置版本。这与 v4_4_configuration_history_hosts 视图中的 history_id 值相同，它可用于加入它们。	否
create_date	使用时区的时间戳	这个实体被添加到系统的日期。	否

名称	类型	描述	索引
update_date	使用时区的时间戳	这个实体已在系统中更改的日期。	否
delete_date	使用时区的时间戳	这个实体已从系统中删除的日期。	否

3.8.8. 虚拟机接口配置

下表显示了系统中虚拟接口的配置历史记录参数。

表 3.15. v4_4_configuration_history_vms_interfaces

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。这与虚拟机接口统计视图中的 vm_interface_configuration_version 的值相同，并可用于加入它们。	否
vm_id	uuid	系统中虚拟机的唯一 ID。	是
vm_interface_id	uuid	系统中这个接口的唯一 ID。	是
vm_interface_name	character varying(50)	如编辑对话框中所示。	否
vm_interface_type	smallint	虚拟接口的类型。 * 0 - rt18139_pv * 1 - rt18139 * 2 - e1000 * 3 - PV	否
vm_interface_speed_bps	整数	在聚合期间接口平均速度（以每秒位数为单位）。	否
mac_address	character varying(20)	如编辑对话框中所示。	否
logical_network_name	character varying(50)	如编辑对话框中所示。	否

名称	类型	描述	索引
vm_configuration_version	整数	创建或更新时的虚拟机配置版本。这与 v4_4_configuration_history_vms 视图中的 history_id 值相同，它可用于加入它们。	否
create_date	使用时区的时间戳	这个实体被添加到系统的日期。	否
update_date	使用时区的时间戳	这个实体已在系统中更改的日期。	否
delete_date	使用时区的时间戳	这个实体已从系统中删除的日期。	否

3.8.9. 虚拟机设备配置

下表显示了虚拟机及其关联的设备之间的关系，包括磁盘和虚拟接口。

表 3.16. v4_4_configuration_history_vms_devices

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。	否
vm_id	uuid	系统中虚拟机的唯一 ID。	是
device_id	uuid	系统中设备的唯一 ID。	否
type	character varying(30)	虚拟机设备的类型。这可以是 "disk" 或 "interface"。	是
address	character varying(255)	设备的物理地址。	否
is_managed	布尔值	表示该设备是否由 Manager 管理的标记。	否
is_plugged	布尔值	表示该设备是否插入到虚拟机的标记。	否

名称	类型	描述	索引
is_readonly	布尔值	表示该设备是否只读的标记。	否
vm_configuration_version	整数	生成示例时的虚拟机配置版本。	否
device_configuration_version	整数	抽样发生时的设备配置版本。 - 如果 type 字段的值被设置为 interface , 则此字段与 v4_configuration_history_vms_interfaces 视图中的 history_id 字段相连接。 - 如果 type 字段的值被设置为 disk , 则会与 v4_4_configuration_history_vms_disks 视图中的 history_id 字段相连接。	否
create_date	使用时区的时间戳	这个实体被添加到系统的日期。	否
update_date	使用时区的时间戳	这个实体被添加到系统的日期。	否
delete_date	使用时区的时间戳	这个实体被添加到系统的日期。	否

3.8.10. 虚拟磁盘配置

下表显示了系统中虚拟磁盘的配置历史记录参数。

表 3.17. v4_4_configuration_history_vms_disks

名称	类型	描述	索引
history_id	整数	历史记录数据库中配置版本的 ID。这与虚拟磁盘统计视图中的 vm_disk_configuration_version 的值相同, 并可用于加入它们。	否
vm_disk_id	uuid	系统中这个磁盘的唯一 ID。	是

名称	类型	描述	索引
vm_disk_name	text	虚拟磁盘的名称，如编辑对话框中显示。	否
vm_disk_description	character varying(500)	如编辑对话框中所示。	否
image_id	uuid	系统中镜像的唯一 ID。	否
storage_domain_id	uuid	此磁盘镜像所属存储域的 ID。	是
vm_disk_size_mb	整数	以 MB(MB)为单位定义的磁盘大小。	否
vm_disk_type	smallint	如编辑对话框中所示。当前只使用系统和数据。 * 0 - 未分配 * 1 - 系统 * 2 - 数据 * 3 - 共享 * 4 - 交换 * 5 - 临时	否
vm_disk_format	smallint	如编辑对话框中所示。 * 3 - 未签名 * 4 - COW * 5 - Raw	否
is_shared	布尔值	表示虚拟机磁盘是否已共享的标志。	否
create_date	使用时区的时间戳	这个实体被添加到系统的日期。	否
update_date	使用时区的时间戳	这个实体已在系统中更改的日期。	否
delete_date	使用时区的时间戳	这个实体已从系统中删除的日期。	否

3.8.11. 用户详情历史

下表显示了系统中用户的配置历史记录参数。

表 3.18. v4_4_users_details_history

名称	类型	描述
user_id	uuid	系统中用户的唯一 ID，如 Manager 生成。
first_name	character varying(255)	用户的名字。
last_name	character varying(255)	用户的姓氏。
domain	character varying(255)	授权扩展的名称。
username	character varying(255)	帐户名称。
department	character varying(255)	用户所属的组织部门。
user_role_title	character varying(255)	组织中用户的标题或角色。
email	character varying(255)	组织中用户的电子邮件。
external_id	text	来自外部系统的用户的唯一标识符。
active	布尔值	用于指示用户是否活跃的标记。这会每小时检查。如果用户可以在授权扩展名中找到，则它将保持活动状态。用户在成功登录后变为活动状态。
create_date	使用时区的时间戳	这个实体被添加到系统的日期。
update_date	使用时区的时间戳	这个实体已在系统中更改的日期。
delete_date	使用时区的时间戳	这个实体已从系统中删除的日期。

附录 A. 法律通知

Copyright © 2022 Red Hat, Inc.

Licensed under the ([Creative Commons Attribution–ShareAlike 4.0 International License](#)).从(oVirt Project)的文档衍生而来。如果您发布本文档或对其进行改编，您必须提供原始版本的 URL。

修改后的版本必须删除所有红帽商标。

Red Hat、Red Hat Enterprise Linux、Red Hat 商标、Shadowman 商标、JBoss、OpenShift、Fedora、Infinity 商标以及 RHCE 都是在美国及其他国家的注册商标。

Linux® 是 Linus Torvalds 在美国和其他国家/地区的注册商标。

Java® 是 Oracle 和/或其附属公司的注册商标。

XFS® 是 Silicon Graphics International Corp. 或其子公司在美国和/或其他国家的商标。

MySQL® 是 MySQL AB 在美国、欧盟和其他国家/地区的注册商标。

Node.js® 是 Joyent 的官方商标。Red Hat Software Collections 与官方 Joyent Node.js 开源或商业项目没有正式关联或被正式认可。

The OpenStack® Word Mark 和 OpenStack 标识是 OpenStack Foundation 在美国及其他国家的注册商标/服务标记或商标/服务标记，可根据 OpenStack Foundation 授权使用。我们不附属于 OpenStack Foundation 或 OpenStack 社区。

所有其他商标均由其各自所有者所有。