



Red Hat Virtualization 4.1 Data Warehouse 指南

如何使用 Red Hat Virtualization 的 Data Warehouse 功能

Red Hat Virtualization Documentation Team

如何使用 Red Hat Virtualization 的 Data Warehouse 功能

Red Hat Virtualization Documentation Team
Red Hat Customer Content Services
rhev-docs@redhat.com

法律通告

Copyright © 2016 Red Hat.

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

本书包含与 Red Hat Virtualization Data Warehouse 相关的信息和操作步骤。

目录

第 1 章 安装和配置 Data Warehouse 2

- 1.1. Data Warehouse 配置概述 2
- 1.2. 在独立的机器上安装和配置 Data Warehouse 2
- 1.3. 将 Data Warehouse 迁移到独立的机器上 5
- 1.4. 更改 Data Warehouse 采样规模 9

第 2 章 关于历史数据库 12

- 2.1. 历史数据库介绍 12
- 2.2. 跟踪配置历史 12
- 2.3. 记录统计历史 13
- 2.4. ovirt-engine-dwhd.conf 中的 Data Warehouse 服务的应用设置 13
- 2.5. 跟踪标签 (Tag) 历史 13
- 2.6. 对历史数据库的只读访问 14
- 2.7. 用于统计数据的 History 视图 15
- 2.8. 配置历史数据视图 21

第 1 章 安装和配置 Data Warehouse

1.1. Data Warehouse 配置概述

Red Hat Virtualization Manager 包含了全面的管理历史数据库，可供任何应用程序用于提取数据中心、集群和主机级别的一系列信息。安装 Data Warehouse 将创建 **ovirt_engine_history** 数据库，Manager 将配置为把报表所需的信息记录到该数据库中。

Red Hat Virtualization 中必须要有 Data Warehouse。可以在 Manager 所在的同一机器上安装和配置，或者安装到有权访问 Manager 的其他机器上：

1. 在安装 Manager 所在机器上安装和配置 Data Warehouse。

此配置仅需要注册一台机器，配置也最简单；不过，它会增加对主机的需求。需要访问 Data Warehouse 服务的用户也将需要具备 Manager 机器本身的访问权限。如需了解与此配置相关的更多信息，请参阅 [安装指南中的配置 Red Hat Virtualization Manager](#)。

2. 在独立的机器上安装和配置 Data Warehouse。

此配置需要注册两台机器。它可降低 Manager 机器上的负载，并可避免该机器上可能出现的 CPU 和内存共享冲突。管理员也可以允许用户访问 Data Warehouse 机器，而不必授予用户 Manager 机器的访问权限。如需了解与此配置相关的更多信息，请参阅 [第 1.2 节“在独立的机器上安装和配置 Data Warehouse”](#)。

建议您将 Data Warehouse 部署中所有机器的系统时区设置为 UTC。这可以确保数据收集不会因为本地时区变动（例如从夏令时变为冬令时）而中断。

若要估算 **ovirt_engine_history** 数据库要使用的空间和资源，请使用 [RHV Manager History Database Size Calculator](#) 工具。其估算依据是实体的数量和您选择的历史记录保留时长。



重要

engine-setup 命令有以下行为：

- ✧ 安装 Data Warehouse 软件包，运行 **engine-setup**，对是否配置 Data Warehouse 回答 No：

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

- ✧ 再次运行 **engine-setup** 命令；安装程序不再显示配置 Data Warehouse 的选项。

若要强制 **engine-setup** 再次显示该选项，请运行 **engine-setup --reconfigure-optional-components/**

若要仅配置当前安装的 Data Warehouse 软件包，并且防止安装程序应用那些在已启用存储库中找到的软件包更新，请加上 **--offline** 选项。

1.2. 在独立的机器上安装和配置 Data Warehouse

在安装 Red Hat Virtualization Manager 的机器以外的机器上安装和配置 Data Warehouse。在独立的机器上运行 Data Warehouse 服务有助于降低 Manager 机器的负载。

先决条件

- ✧ 您已经在另外一台机器上安装并配置了 Manager。
- ✧ 要设置 Data Warehouse，您需要：
 - 安装了 Red Hat Enterprise Linux 7 的虚拟机或物理机。
 - **Red Hat Enterprise Linux Server** 订阅和 **Red Hat Virtualization** 订阅池。
 - Manager 的 `/etc/ovirt-engine/engine.conf.d/10-setup-database.conf` 文件中的密码。
 - 允许从 Data Warehouse 所在机器上访问 Manager 数据库所在机器的 TCP 端口 5432。
- ✧ 如果选择使用远程 Data Warehouse 数据库，您必须先设置数据库，然后再安装 Data Warehouse 服务。您必须具有以下有关数据库主机的信息：
 - 主机的全局域名（FQDN）
 - 访问数据库的端口号（默认是 5432）
 - 数据库名
 - 数据库用户
 - 数据库密码
 - 您必须通过编辑 `postgres.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，使它与如下相符：

```
listen_addresses = '*'
```

如果这行没有存在或被注释掉，您需要手工添加它。

如果数据库运行于 Manager 机器上，并在全新安装 Red Hat Virtualization Manager 过程中进行了配置，则已默认授予访问权限。

过程 1.1. 在独立的机器上安装和配置 Data Warehouse

1. 为您的系统订阅所需的权利。如需了解更多信息，请参阅 [安装指南中的订阅所需的权利](#)。
2. 确认当前安装的软件包已被升级到最新版本：

```
# yum update
```

3. 安装 `ovirt-engine-dwh-setup` 软件包：

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

4. 运行 `engine-setup` 命令，再按照机器上的提示配置 Data Warehouse：

```
# engine-setup
Configure Data Warehouse on this host (Yes, No) [Yes]:
```

5. 按 **Enter** 接受自动检测到的主机名，或者输入其他主机名并按 **Enter**：

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

6. 点 **Enter** 自动配置防火墙，或输入 **No** 并点 **Enter** 来保持存在的设置：

```
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]:
```

如果您选择了自动配置防火墙，但还没有激活的防火墙管理器，您将会被提示从一个列表中选择需要使用的防火墙管理器。输入防火墙管理器的名称后按 **Enter**。

7. 输入 Manager 所在机器的完全限定域名并点 **Enter**：

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

8. 按 **Enter** 以允许安装程序通过 SSH 在 Manager 中为证书签名：

```
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]:
```

9. 点 **Enter** 接受默认的 SSH 端口，或输入另外一个端口号后点 **Enter**：

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

10. 输入 Manager 机器的 root 密码：

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

11. 回答以下与 Data Warehouse 数据库相关的问题：

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

按 **Enter** 以选择突出显示的默认值，或输入您的其他选择并按 **Enter**。如果选择 **Remote**，您要根据提示提供远程数据库主机相关的详细信息。为预配置的远程数据库主机输入以下值：

```
DWH database host []: dwh-db-fqdn
DWH database port [5432]:
DWH database secured connection (Yes, No) [No]:
DWH database name [ovirt_engine_history]:
DWH database user [ovirt_engine_history]:
DWH database password: password
```


请参阅[第 1.3.1 节 “把 Data Warehouse 数据库迁移到一个独立的机器上”](#)，了解有关如何配置和迁移 Data Warehouse 数据库的更多信息。

12. 输入 Manager 数据库所在机器的完全限定域名和密码。点 **Enter** 接受每个项的默认值：

```
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
```

13. 选择 Data Warehouse 保留所收集数据的时长：

```
Please choose Data Warehouse sampling scale:
(1) Basic
(2) Full
(1, 2)[1]:
```

Full 使用在[第 2.4 节 “ovirt-engine-dwhd.conf 中的 Data Warehouse 服务的应用设置”](#)中列出的数据存储设置默认值（建议在 Data Warehouse 安装于远程主机的情况下使用）。

Basic 将 `DWH_TABLES_KEEP_HOURLY` 的值降低至 **720**，将 `DWH_TABLES_KEEP_DAILY` 的值降低至 **0**，减轻 Manager 机器上的负载（建议在同一主机同时安装 Manager 和 Data Warehouse 的情况下使用）。

14. 确认您的安装设置：

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

15. 在 Red Hat Virtualization Manager 上，重新启动 **ovirt-engine** 服务：

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

1.3. 将 Data Warehouse 迁移到独立的机器上

将 Data Warehouse 服务从 Red Hat Virtualization Manager 迁移到独立的机器上。在独立的机器上运行 Data Warehouse 服务可以降低每一机器的负载，并可避免各项服务因为与其他进程共享 CPU 和内存而出现潜在冲突。

迁移 Data Warehouse 服务，并把它和存在的 **ovirt_engine_history** 数据库进行连接；或在迁移 Data Warehouse 服务前，把 **ovirt_engine_history** 数据库迁移到一个新的数据库机器上。如果 **ovirt_engine_history** 数据库原来存在于 Manager 所在的机器上，迁移 Data Warehouse 服务的同时迁移这个数据库，可以进一步减轻 Manager 所在机器的负载。您可以把数据库迁移到 Data Warehouse 服务所在的机器上，也可以把它迁移到一个完全独立的机器上。

1.3.1. 把 Data Warehouse 数据库迁移到一个独立的机器上

在迁移 Data Warehouse 服务前，可以选择先迁移 **ovirt_engine_history** 数据库。使用 **engine-backup** 创建数据库备份，并将它恢复到新数据库机器上。如需了解与 **engine-backup** 相关的更多信息，可运行 **engine-backup --help**。

此步骤假定新的数据库服务器已安装了 Red Hat Enterprise Linux 7，并且配置有适当的订阅。请参阅[安装指南中的\[订阅所需的权利\]\(#\)](#)。

若要仅迁移 Data Warehouse 服务，请参阅[第 1.3.2 节 “把 Data Warehouse 服务迁移到一个独立的机器上”](#)。

过程 1.2. 把 Data Warehouse 数据库迁移到一个独立的机器上

1. 创建 Data Warehouse 数据库和配置文件的备份：

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

2. 将备份文件从 Manager 复制到新机器上：

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

3. 在新机器上安装 **engine-backup**：

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

4. 在新机器上恢复 Data Warehouse 数据库。file_name 是从 Manager 复制的备份文件。

```
# engine-backup --mode=restore --scope=files --scope=dwhdb --
file=file_name --log=log_file_name --provision-dwh-db --no-restore-
permissions
```

1.3.2. 把 Data Warehouse 服务迁移到一个独立的机器上

把 Red Hat Enterprise Virtualization Manager 上安装和配置的 Data Warehouse 服务迁移到一台专用的主机计算机上。在独立的机器上托管 Data Warehouse 服务有助于减少 Manager 计算机上的负载。请注意，此步骤仅迁移 Data Warehouse 服务；如需在迁移 Data Warehouse 服务前迁移 Data Warehouse 数据库（也称为 **ovirt_engine_history** 数据库），请参阅[第 1.3.1 节 “把 Data Warehouse 数据库迁移到一个独立的机器上”](#)。

这个操作会涉及到 4 个关键步骤：

1. 设置新的 Data Warehouse 机器。
2. 在 Manager 所在的机器上停止 Data Warehouse 服务。
3. 配置新的 Data Warehouse 机器。
4. 从 Manager 所在的机器上删除 Data Warehouse 软件包。

先决条件

请确认您已经满足了以下前提条件。

1. 您已经在同一台机器上安装并配置了 Manager 和 Data Warehouse。
2. 为了设置新的 Data Warehouse 机器，您需要：
 - ✦ 安装了 Red Hat Enterprise Linux 7 的虚拟机或物理机。
 - ✦ 订阅了 **Red Hat Enterprise Linux Server** 和 **Red Hat Enterprise Virtualization** 权利池。

- ✧ Manager 的 `/etc/ovirt-engine/engine.conf.d/10-setup-database.conf` 文件中的密码。
- ✧ 允许从 Data Warehouse 所在机器上访问 Manager 数据库所在机器的 TCP 端口 5432。
- ✧ Manager 所在系统上的 `/etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/10-setup-database.conf` 文件中的 `ovirt_engine_history` 数据库凭证。如果您根据 [第 1.3.1 节“把 Data Warehouse 数据库迁移到一个独立的机器上”](#) 的介绍迁移了 `ovirt_engine_history` 数据库，则需要获得在设置数据库时设定的用户凭证。

过程 1.3. 第 1 步：设置新的 Data Warehouse 机器

1. 在 Content Delivery Network 中注册您的系统，在出现提示时输入您的 Customer Portal 用户名和密码：

```
# subscription-manager register
```

2. 找到 **Red Hat Enterprise Linux Server** 和 **Red Hat Enterprise Virtualization** 订阅池，并记录下它们的 ID。

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

3. 使用前一步获得的池 ID 为系统附加权利：

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

4. 禁用所有存在的软件仓库：

```
# subscription-manager repos --disable=*
```

5. 启用所需的频道：

```
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-supplementary-rpms
# subscription-manager repos --enable=rhel-7-server-rhv-4.1-rpms
# subscription-manager repos --enable=rhel-7-server-rhv-4-tools-rpms
# subscription-manager repos --enable=jb-eap-7-for-rhel-7-server-rpms
```

6. 确认当前安装的软件包已被升级到最新版本：

```
# yum update
```

7. 安装 `ovirt-engine-dwh-setup` 软件包：

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

过程 1.4. 第 2 步：在 Manager 的机器上停止 Data Warehouse 服务

1. 停止 Data Warehouse 服务：

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

- 如果 **ovirt_engine_history** 数据库、Manager 数据库或这两个数据库都运行在 Manager 所在的机器上，而且是由一个较早版本的系统（Red Hat Enterprise Virtualization 3.4 或更早版本）所配置，然后进行了升级，您则需要保证新的 Data Warehouse 机器可以访问它们。打开 **/var/lib/pgsql/data/postgresql.conf** 文件并把 **listen_addresses** 行的内容修改为：

```
listen_addresses = '*'
```

如果这行没有存在或被注释掉，您需要手工添加它。

如果其中有数据库位于远程机器上，您必须手动授予访问权限（编辑每台机器上的 **postgres.conf** 文件，根据上面的介绍添加 **listen_addresses** 行）。如果两个数据库都位于 Manager 机器上，并且在全新安装 Red Hat Enterprise Virtualization Manager 的过程中进行了配置，则已默认授予了相关的访问权限。

- 重启 postgresql 服务：

```
# systemctl restart postgresql.service
```

过程 1.5. 第 3 步：配置新的 Data Warehouse 机器

- 运行 **engine-setup** 命令在机器上开始配置 Data Warehouse：

```
# engine-setup
```

- 点 **Enter** 配置 Data Warehouse：

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

- 点 **Enter** 自动配置防火墙，或输入 **No** 并点 **Enter** 来保持存在的设置：

```
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]:
```

如果您选择了自动配置防火墙，但还没有激活的防火墙管理器，您将会被提示从一个列表中选择需要使用的防火墙管理器。输入防火墙管理器的名称后按 **Enter**。

- 点 **Enter** 接受自动检测到的主机名，或输入一个其它主机名后点 **Enter**：

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

- 输入 **ovirt_engine_history** 数据库的以下位置信息：

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

输入您的选择后点 **Enter**。

- 输入 **ovirt_engine_history** 数据库主机的完全限定域名。点 **Enter** 接受每个项的默认值：

```
DWH database host []: dwh-db-fqdn
DWH database port [5432]:
DWH database secured connection (Yes, No) [No]:
DWH database name [ovirt_engine_history]:
```

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

7. 输入 Manager 数据库所在机器的完全限定域名和密码。点 **Enter** 接受每个项的默认值：

```
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
```

8. 点 **Enter** 为已存在的 Data Warehouse 数据库创建一个备份：

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

创建备份所需的时间以及存储空间取决于数据库的大小，它可能会需要几个小时来完成。备份文件的位置出现在设置脚本的后面。如果您在这一步没有选择备份数据库，当因为某些原因 **engine-setup** 运行失败，您将无法恢复数据库或其中的任何数据。

9. 确认您将永久地把 Data Warehouse 服务与 Manager 断开：

```
Do you want to permanently disconnect this DWH from the engine? (Yes,
No) [No]:
```

10. 确认您的安装设置：

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

过程 1.6. 第 4 步：在 Manager 的机器上删除 Data Warehouse 软件包

1. 删除 Data Warehouse 软件包：

```
# yum remove ovirt-engine-dwh
```

这一步操作会防止 Data Warehouse 服务在一个小时后自动重启。

2. 删除 Data Warehouse 文件：

```
# rm -rf /etc/ovirt-engine-dwh /var/lib/ovirt-engine-dwh
```

现在，Data Warehouse 服务运行于一个和 Manager 所在机器不同的机器上。

1.4. 更改 Data Warehouse 采样规模

Red Hat Virtualization 中必须要有 Data Warehouse。它可以安装并配置到与 Manager 相同的机器上，或者有权访问 Manager 的独立机器上。部分设置可能不需要默认的数据保留设置，因此 **engine-setup** 提供了两种采样规模：**Basic** 和 **Full**。

- ✱ **Full** 使用[第 2.4 节 “ovirt-engine-dwhd.conf 中的 Data Warehouse 服务的应用设置”](#)中列出的数据保留设置默认值（建议在 Data Warehouse 安装于远程主机的情况下使用）。

- ✱ **Basic** 将 `DWH_TABLES_KEEP_HOURLY` 的值降低至 **720**，将 `DWH_TABLES_KEEP_DAILY` 的值降低至 **0**，减轻 Manager 机器上的负载（建议在同一主机同时安装 Manager 和 Data Warehouse 的情况下使用）。

采样规模可以在安装过程中通过 **engine-setup** 配置：

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

Please choose Data Warehouse sampling scale:
(1) Basic
(2) Full
(1, 2)[1]:
```

以后若要更改采样规模，您也可以再次运行 **engine-setup** 并使用 **--reconfigure-dwh-scale** 选项。

例 1.1. 更改 Data Warehouse 采样规模

```
# engine-setup --reconfigure-dwh-scale
[...]
Welcome to the RHEV 4.0 setup/upgrade.
Please read the RHEV 4.0 install guide
https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Virtualization/4.1/html/Installation_Guide/index.html.
Please refer to the RHEV Upgrade Helper application
https://access.redhat.com/labs/rhevupgradehelper/ which will guide you in the upgrading process.
Would you like to proceed? (Yes, No) [Yes]:
[...]
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]:
[...]
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]:
```

必要时，您还可以调整个别数据保留设置，具体如[第 2.4 节 “ovirt-engine-dwhd.conf 中的 Data Warehouse 服务的应用设置”](#)中所述。

第 2 章 关于历史数据库

2.1. 历史数据库介绍

Red Hat Enterprise Virtualization 包括了一个全面的管理历史数据库，它可由报表应用程序使用，用于生成数据中心、集群和主机级别上的数据报表。本章介绍如何查询历史数据库。

Red Hat Enterprise Virtualization Manager 使用 **PostgreSQL 8.4.x**，作为存储虚拟环境的状态、配置信息和性能数据的数据库平台。在安装时，Red Hat Enterprise Virtualization Manager 会创建一个名为 **engine** 的 PostgreSQL 数据库。

安装 **ovirt-engine-dwh** 软件包时会创建另外一个名为 **ovirt_engine_history** 的数据库，它包括了配置的历史信息，以及每一分钟从 **engine** 运行数据库所收集的统计数据。跟踪此数据库的更改可以提供与数据库中对象相关的信息，帮助用户分析活动、增强性能和攻克难题。



警告

复制 **ovirt_engine_history** 数据库中的数据是通过 Red Hat Enterprise Virtualization Manager 的 Extract Transform Load 服务 (**ovirt-engine-dwhd**) 进行的。这个服务基于名为 Talend Open Studio 的一款数据集成工具。此服务配置为在数据仓库软件包安装过程中自动启动。它是一个 Java 程序，负责从 **engine** 数据库提取数据，把这些数据转换为符合历史数据库的标准，并加载到 **ovirt_engine_history** 数据库中。

ovirt-engine-dwhd 服务不能被停止。

ovirt_engine_history 数据库的 schema 会随着时间有一些改变，而数据库会通过一组数据库视图 (view) 来对所支持的 API 提供一个统一的结构。视图 (view) 就是一个由数据库查询结果所组成的虚拟的表。数据库会把一个视图的定义以一个 **SELECT** 命令的形式保存，这个 **SELECT** 命令的结果数据被用来产生视图的虚拟表数据。在 **PL/PGSQL** 命令中指定视图的方法和指定普通表的方法一样。

2.2. 跟踪配置历史

可利用来自 Red Hat Virtualization 历史数据库 (名为 **ovirt_engine_history**) 的数据，跟踪 **engine** 数据库。

ETL 服务 (**ovirt-engine-dwhd**) 会跟踪三类的变化：

- ✧ 一个新的项被添加到引擎数据库 - ETL 服务会把这个变化作为一条新的记录添加到 **ovirt_engine_history** 数据库。
- ✧ 一个已经存在的项被更新 - ETL 服务会把这个变化作为一条新的记录添加到 **ovirt_engine_history** 数据库。
- ✧ 一个实体已从 **engine** 数据库中删除 - **ovirt_engine_history** 数据库中的新项将对应的实体标记为已删除。删除的实体仅标记为已删除。

ovirt_engine_history 数据库中的配置表和 **engine** 数据库的配置表有一些不同。其中最直观的不同是它有较少的配置列。这是因为一些特定的配置项对于报表来说没有什么实际的意义，不包括它们可以减小数据库的大小。另外，**engine** 数据库中的几个表中的一些列在 **ovirt_engine_history** 数据库中被保存为一个单一的表，并有不同的列名。所有的配置表包括：

- ✧ **history_id** 记录项的配置版本；

- » **create_date** 记录项被添加到系统的时间；
- » **update_date** 记录项被更新的时间；
- » **delete_date** 记录项被从系统中删除的时间。

2.3. 记录统计历史

ETL 服务会把每分钟收集的数据保存在统计表中。这些基于每分钟的数据最少会被保持 24 小时，最多可以保存 48 小时（取决于最后一次运行删除任务的时间）。超过 2 个小时的基于每分钟的数据会被积累为每小时的数据，并保持两个月。超过 2 天的每小时的数据会被积累为每天的数据，并保持 5 年。

每小时的数据和每天的数据可以在 hourly 和 daily 的表中获得。

每一个统计数据会保存在它们相应的累计级别的表中：samples、hourly 和 daily 历史表。所有的历史表也都包括一个 history_id 列来区分每一个数据行。这些表也包括一个主机的配置版本来记录和以前配置的关系。

2.4. ovirt-engine-dwhd.conf 中的 Data Warehouse 服务的应用设置

以下列出了用来配置 Data Warehouse 服务的应用设置选项，这些选项可以在 `/usr/share/ovirt-engine-dwh/services/ovirt-engine-dwhd/ovirt-engine-dwhd.conf` 文件中使用。而对默认设置值的改变包括在一个覆盖文件中（`/etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/`）。在对设置进行修改后，需要重启 Data Warehouse 服务。

表 2.1. ovirt-engine-dwhd.conf 应用设置变量

变量名	默认值	备注
DWH_DELETE_JOB_HOUR	3	删除命令运行的时间的小时值。它需要是一个 0 到 23 间的值，其中 0 代表午夜。
DWH_SAMPLING	60	为生成统计数据表采集数据的间隔时间（以秒为单位）。
DWH_TABLES_KEEP_SAMPLES	24	DWH_SAMPLING 数据被保存的小时数。当数据存在的时间超过两小时后，会被归纳到每小时的数据中。
DWH_TABLES_KEEP_HOURLY	1440	每小时的数据需要被保存的小时数。它的默认值是 60 天。当数据存在的时间超过两天后，会被归纳到每天的数据中。
DWH_TABLES_KEEP_DAILY	43800	数据需要被保存的小时数。它的默认值是 5 年。
DWH_ERROR_EVENT_INTERVAL	300000	错误被记录到 Manager 的 audit.log 文件中的最小间隔时间（以毫秒为单位）。

2.5. 跟踪标签 (Tag) 历史

ETL 服务会每分钟收集管理门户显示的标签信息，并把这些数据保存在标签历史表中。ETL 服务会跟踪 5 类改变：

- » 一个标签在管理门户中被创建 - ETL 服务复制标签的详情、标签在标签树中的位置以及和标签树中的其它项间的关系。

- ✱ 一个项在管理门户中被添加到标签树中 - ETL 服务会把它作为一个新的记录保存在 **ovirt_engine_history** 数据库中。
- ✱ 一个标签被更新 - ETL 服务会把标签改变的信息作为一个新的记录保存在 **ovirt_engine_history** 数据库中。
- ✱ 从管理门户中删除了一个实体或标签分支 - **ovirt_engine_history** 数据库在新条目中将对应的标签和关系标记为已删除。删除的标签和关系仅标记为已删除或已分离。
- ✱ 移动了一个标签分支 - 对应的标签和关系更新为新条目。移动的标签和关系仅标记为已更新。

2.6. 对历史数据库的只读访问

您可以创建一个只读的 PostgreSQL 用户，这个用户将只能对 **ovirt_engine_history** 数据库的数据进行读操作，而没有权限进行写操作。以下步骤必须在历史数据库所在的系统上进行。

过程 2.1. 对历史数据库的只读访问

1. 创建对历史数据库只有只读权限的用户：

```
# psql -U postgres -c "CREATE ROLE [user name] WITH LOGIN ENCRYPTED
PASSWORD '[password]';" -d ovirt_engine_history
```

2. 赋予新创建的用户连接到历史数据库的权限：

```
# psql -U postgres -c "GRANT CONNECT ON DATABASE ovirt_engine_history
TO [user name];"
```

3. 赋予新创建的用户使用 **public** schema 的权限：

```
# psql -U postgres -c "GRANT USAGE ON SCHEMA public TO [user name];"
ovirt_engine_history
```

4. 创建新建用户需要的其它权限，并把它们保存到一个文件：

```
# psql -U postgres -c "SELECT 'GRANT SELECT ON ' || relname || ' TO
[user name];' 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
```

5. 使用您在上一步创建的文件来为新创建的用户赋予权限：

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

6. 删除被用来为新创建的用户赋予权限的文件：

```
# rm grant.sql
```

您可以使用以下命令来以新创建的用户身份访问 **ovirt_engine_history** 数据库：

```
# psql -U [user name] ovirt_engine_history
```

针对表和视图 (view) 的 **SELECT** 命令会成功运行，而修改的命令会失败。

2.7. 用于统计数据的 History 视图

统计数据以 **hourly**、**daily** 和 **samples** 等视图提供。

若要查询统计数据视图，请运行 **SELECT * FROM view_name_[hourly|daily|samples];**。例如：

```
# SELECT * FROM v4_0_statistics_hosts_resources_usage_daily;
```

若要列出所有的可用视图，请运行：

```
# \dv
```

2.7.1. 存储域统计数据视图

表 2.2. 系统中的每个存储域的历史统计数据

名称	类型	描述
history_id	bigint	数据表中的本行数据的 ID。
history_datetime	date	历史数据行的时间戳。
storage_domain_id	uuid	系统中的存储域的唯一 ID。
storage_domain_status	smallint	存储域的状态。
seconds_in_status	整数	在整合时间段内，存储域处于状态列中所示状态的总时间（以秒数为单位）。例如，某存储域在某一分钟内有 55 秒处于活跃状态、5 秒处于不活跃状态，则这一分钟在表格内报告为两行。其中一行的状态是“Active”，seconds_in_status 为 55；另一行的状态值“Inactive”，seconds_in_status 为 5。
minutes_in_status	numeric(7,2)	在整合的时间内，存储域处于状态栏中所显示的状态的总时间（以分钟为单位）。例如，在一个小时内，一个存储域有 55 分钟处于活跃状态，5 分钟处于不活跃状态，这一个小时将会有两行。其中一行的状态值是“Active”，minutes_in_status 为 55；另外一行的状态值是“Inactive”，minutes_in_status 为 5。
available_disk_size_gb	整数	磁盘中可用的空间（以 GB 为单位）。
used_disk_size_gb	整数	磁盘中已使用的空间总量（以 GB 为单位）。
storage_configuration_version	整数	在采集数据时存储域的配置版本。

2.7.2. 主机统计数据视图

表 2.3. 系统中的每个主机的历史统计数据

名称	类型	描述
history_id	bigint	数据表中的本行数据的 ID。

名称	类型	描述
history_datetime	date	历史数据行的时间戳
host_id	uuid	系统中主机的唯一 ID。
host_status	smallint	<ul style="list-style-type: none"> ✎ -1 - Unknown Status (仅用于表示 ETL 存在问题。请与红帽支持部门联系) ✎ 1 - Up ✎ 2 - Maintenance ✎ 3 - Problematic
seconds_in_status	整数	主机在整合时间段内处于状态列中所示状态的总秒数。例如，如果某主机在某一分钟内有 55 秒处于运行状态、5 秒处于关闭状态，则这一分钟将显示为两行。一行显示状态 Up ，其 seconds_in_status 为 55，另一行显示状态 Down ，其 seconds_in_status 为 5。
minutes_in_status	numeric(7,2)	在整合的时间内，主机处于状态栏中所显示的状态的总时间（以分钟为单位）。例如，在一个小时内，一个主机有 55 分钟处于 up 状态，5 分钟处于维护模式，这一个小时将会有两行。其中一行的状态为 Up ，minutes_in_status 为 55；另外一行的状态是 Down ，minutes_in_status 为 5。
memory_usage_percent	smallint	主机上内存使用的百分比。
max_memory_usage	smallint	在一个整合的时间段内，最大的内存使用百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
ksm_shared_memory_mb	bigint	主机正在使用的内核共享内存大小（以 MB 表示）。
max_ksm_shared_memory_mb	bigint	在一个整合的时间段内，最大的 KSM 内存使用大小（MB）。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
cpu_usage_percent	smallint	主机上使用的 CPU 百分比。
max_cpu_usage	smallint	在一个整合的时间段内，最大的 CPU 使用百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
ksm_cpu_percent	smallint	ksm 在主机上使用的 CPU 百分比。
max_ksm_cpu_percent	smallint	在一个整合的时间段内，最大的 KSM 使用百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
active_vms	smallint	在指定整合时间段中活跃虚拟机的平均数量。
max_active_vms	smallint	在一个整合的时间段内，最大的活跃虚拟机的数量。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。

名称	类型	描述
total_vms	smallint	在指定整合时间段中，主机上的所有虚拟机的平均数量。
max_total_vms	smallint	在一个整合的时间段内，所有虚拟机的最大数量。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
total_vms_vcpus	整数	分配给主机的 vCPU 总数。
max_total_vms_vcpus	整数	在整合时间段内，虚拟机 vCPU 总数的最大值。对于每小时整合时间段，它是收集到的最大采样值；对于每天的整合时间段，它是每个小时平均值的最大值。
cpu_load	整数	主机的 CPU 负载。
max_cpu_load	整数	在一个整合的时间段内，最大的 CPU 负载。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
system_cpu_usage_percent	smallint	主机上使用的 CPU 百分比。
max_system_cpu_usage_percent	smallint	在一个整合的时间段内，最大的系统 CPU 使用的百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
user_cpu_usage_percent	smallint	主机上使用的用户 CPU 百分比。
max_user_cpu_usage_percent	smallint	在一个整合的时间段内，最大的用户 CPU 使用的百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
swap_used_mb	整数	使用的主机交换分区的大小 (MB)。
max_swap_used_mb	整数	在一个整合的时间段内，使用的最大主机交换分区的百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
host_configuration_version	整数	采样时的主机配置版本。

2.7.3. 主机接口统计数据视图

表 2.4. 系统中的每个主机网络接口的历史统计数据

名称	类型	描述
history_id	bigint	数据表中的本行数据的 ID。
history_datetime	date	历史数据行的时间戳。
host_interface_id	uuid	系统中的接口的唯一 ID。
receive_rate_percent	smallint	主机上使用的接收速率的百分比。
max_receive_rate_percent	smallint	在一个整合的时间段内，最大的接收速率百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
transmit_rate_percent	smallint	主机上使用的传输速率的百分比。

名称	类型	描述
max_transmit_rate_percent	smallint	在一个整合的时间段内，最大的传输速率百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
received_total_byte	bigint	主机收到的总字节数。
transmitted_total_byte	bigint	主机传出的总字节数。
host_interface_configuration_version	整数	采样时的主机接口配置版本。

2.7.4. 虚拟机统计数据视图

表 2.5. 系统中每一虚拟机的历史统计数据

名称	类型	描述
history_id	bigint	数据表中的本行数据的 ID。
history_datetime	date	历史数据行的时间戳。
vm_id	uuid	虚拟机在系统中唯一的 ID。
vm_status	smallint	<ul style="list-style-type: none"> » -1 - 未知状态（仅用于指出 ETL 存在问题。请通知红帽支持部门。） » 0 - Down » 1 - Up » 2 - Paused » 3 - Problematic
seconds_in_status	整数	在整合时间段内，虚拟机处于状态列中所示状态的总时间，以秒为单位表示。例如，如果虚拟机在某一分钟内有 55 秒正常运行、5 秒停机，则这一分钟将显示为两行。一行中状态为 Up，Seconds_in_status 为 55；另一行中状态为 Down，seconds_in_status 为 5。
minutes_in_status	numeric(7,2)	在整合的时间内，虚拟机处于状态栏中所显示的状态的总时间（以分钟为单位）。例如，在一个小时内，一个虚拟机有 55 分钟处于 up 状态，5 分钟处于维护模式，这一个小时将会有两行。其中一行的状态为 Up ，minutes_in_status 为 55；另外一行的状态是 Down ，minutes_in_status 为 5。
cpu_usage_percent	smallint	虚拟机使用的 CPU 百分比。
max_cpu_usage	smallint	在一个整合的时间段内，最大的 CPU 使用百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
memory_usage_percent	smallint	虚拟机使用内存的百分比。为了获得这个数据，需要在虚拟机上安装 guest 工具程序。

名称	类型	描述
max_memory_usage	smallint	在一个整合的时间段内，最大的内存使用百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。为了获得这个数据，需要在虚拟机上安装 guest 工具程序。
user_cpu_usage_percent	smallint	主机上使用的用户 CPU 百分比。
max_user_cpu_usage_percent	smallint	在一个整合的时间段内，最大的用户 CPU 使用的百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
system_cpu_usage_percent	smallint	主机上使用的系统 CPU 百分比。
max_system_cpu_usage_percent	smallint	在一个整合的时间段内，最大的系统 CPU 使用的百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
vm_ip	text	第一个 NIC 的 IP 地址。它只会在安装了 guest 代理的情况下被显示。
currently_running_on_host	uuid	正在运行这个虚拟机的主机的唯一 ID。
current_user_id	uuid	登录到虚拟机控制台的用户的唯一 ID（若已安装了客户机代理）。
disks_usage	text	磁盘说明。文件系统类型、挂载点、总大小和已用大小。
vm_configuration_version	整数	在采样数据被采集时虚拟机配置的版本。
current_host_configuration_version	整数	采样时主机配置版本。
memory_buffered_kb	bigint	虚拟机上缓冲内存的数量，以 KB 为单位表示。
memory_cached_kb	bigint	虚拟机上被缓存的内存数量，以 KB 为单位表示。
max_memory_buffered_kb	bigint	整合时间段内被缓冲的内存数量的最大值，以 KB 为单位表示。对于每小时整合，这是收集的采样值的最大值。对于每日整合，这是每小时平均值的最大值。
max_memory_cached_kb	bigint	整合时间段内被缓存的内存数量的最大值，以 KB 为单位表示。对于每小时整合，这是收集的采样值的最大值。对于每日整合，这是每小时平均值的最大值。

2.7.5. 虚拟机接口统计数据视图

表 2.6. 系统中的虚拟机网络接口历史统计数据

名称	类型	描述
history_id	整数	数据表中的本行数据的 ID。
history_datetime	date	历史数据行的时间戳
vm_interface_id	uuid	接口在系统中的唯一 ID。
receive_rate_percent	smallint	主机上使用的接收速率的百分比。
max_receive_rate_percent	smallint	在一个整合的时间段内，最大的接收速率百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。

名称	类型	描述
transmit_rate_percent	smallint	主机上使用的传输速率的百分比。
max_transmit_rate_percent	smallint	在一个整合的时间段内，最大的传输速率百分比。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均速率的最大值。
received_total_byte	bigint	虚拟机收到的总字节数。
transmitted_total_byte	bigint	虚拟机传出的总字节数。
vm_interface_configuration_version	整数	在采样数据被采集时虚拟机接口配置的版本。

2.7.6. 虚拟机磁盘统计数据视图

表 2.7. 系统中的虚拟磁盘历史统计数据

名称	类型	描述
history_id	bigint	数据表中的本行数据的 ID。
history_datetime	date	历史数据行的时间戳。
vm_disk_id	uuid	磁盘在系统中的唯一的 ID。
vm_disk_status	smallint	<ul style="list-style-type: none"> » 0 - Unassigned » 1 - OK » 2 - Locked » 3 - Invalid » 4 - Illegal
seconds_in_status	整数	在整合时间段内，虚拟磁盘处于状态列中所示状态的总时间，以秒为单位表示。例如，如果虚拟磁盘在某一分钟内有 55 秒处于锁定状态、5 秒处于正常状态，则这一分钟将显示为两行。一行中状态为 Locked ，seconds_in_status 为 55；另一行中状态为 OK ，seconds_in_status 为 5。
minutes_in_status	numeric(7,2)	在整合时间段内，虚拟磁盘处于状态列中所示状态的总时间，以分钟为单位表示。例如，如果虚拟磁盘在某一小时内有 55 分钟处于锁定状态、5 分钟处于正常状态，则这一小时将显示为两行。一行中状态为 Locked ，minutes_in_status 为 55；另一行中状态为 OK ，minutes_in_status 为 5。
vm_disk_actual_size_mb	整数	为磁盘分配的实际空间的大小。
read_rate_bytes_per_second	整数	磁盘的读速率（字节每秒）
max_read_rate_bytes_per_second	整数	在一个整合的时间段内，最大的读速率。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
read_latency_seconds	numeric(18,9)	虚拟磁盘的读延迟时间（单位为秒）。

名称	类型	描述
max_read_latency_seconds	numeric(18,9)	在一个整合的时间段内，最大的读延迟时间（以秒为单位）。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
write_rate_bytes_per_second	整数	磁盘的写速率（字节每秒）
max_write_rate_bytes_per_second	整数	在一个整合的时间段内，最大的写速率。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
write_latency_seconds	numeric(18,9)	虚拟磁盘的写延迟时间（单位为秒）。
max_write_latency_seconds	numeric(18,9)	在一个整合的时间段内，最大的写延迟时间（以秒为单位）。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
flush_latency_seconds	numeric(18,9)	虚拟磁盘的清除延迟时间（单位为秒）。
max_flush_latency_seconds	numeric(18,9)	在一个整合的时间段内，最大的清除延迟时间（以秒为单位）。对于每小时的时间段，它是收集到的最大采样值；对于每天的时间段，它是每个小时的平均值的最大值。
vm_disk_configuration_version	整数	采样时的虚拟磁盘配置版本。


2.8. 配置历史数据视图

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

```
# SELECT * FROM v4_0_configuration_history_datacenters;
```

若要列出所有的可用视图，请运行：

```
# \dv
```



注意

delete_date 不会出现在最新的视图中，这是因为这些视图只提供最新的有效项（还没有被删除的项）的信息。

2.8.1. 数据中心配置

下表显示了系统中的数据中心的配置历史参数。

表 2.8. v4_0_configuration_history_datacenters

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
datacenter_id	UUID：	数据中心在系统中的唯一的 ID。

名称	类型	描述
datacenter_name	character varying(40)	数据中心的名称（与编辑对话框中的相同）。
datacenter_description	character varying(4000)	数据中心的描述信息（与编辑对话框中的相同）。
is_local_storage	boolean	此标记用于指出数据中心是否使用本地存储。
create_date	带有时区的时间戳	被添加到系统中的时间。
update_date	带有时区的时间戳	在系统中被修改的时间。
delete_date	带有时区的时间戳	被从系统中删除的时间。

2.8.2. 数据中心存储域映射

下表显示了系统中的存储域和数据中心间的关系。

表 2.9. v4_0_map_history_datacenters_storage_domains

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
storage_domain_id	uuid	系统中的存储域的唯一 ID。
datacenter_id	uuid	数据中心在系统中的唯一的 ID。
attach_date	带有时区的时间戳	存储域被附加到数据中心的时间。
detach_date	带有时区的时间戳	存储域被从数据中心中分离的时间。

2.8.3. 存储域配置

下表显示了系统中的存储域配置的历史参数。

表 2.10. v4_0_configuration_history_storage_domains

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
storage_domain_id	uuid	系统中的存储域的唯一 ID。
storage_domain_name	character varying(250)	存储域的名称。
storage_domain_type	smallint	<ul style="list-style-type: none"> ➤ 0 - Data (Master) ➤ 1 - Data ➤ 2 - ISO ➤ 3 - Export
storage_type	smallint	<ul style="list-style-type: none"> ➤ 0 - Unknown ➤ 1 - NFS ➤ 2 - FCP ➤ 3 - iSCSI ➤ 4 - Local ➤ 6 - All
create_date	带有时区的时间戳	被添加到系统中的时间。
update_date	带有时区的时间戳	在系统中被修改的时间。
delete_date	带有时区的时间戳	被从系统中删除的时间。

2.8.4. 集群配置

下表显示了系统中的集群配置的历史参数。

表 2.11. v4_0_configuration_history_clusters

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
cluster_id	uuid	集群所在数据中心的唯一标识符。
cluster_name	character varying(40)	集群的名称（与编辑对话框中的相同）。
cluster_description	character varying(4000)	与编辑对话框中的相同
datacenter_id	uuid	集群所在数据中心的唯一标识符。
cpu_name	character varying(255)	与编辑对话框中的相同。
compatibility_version	character varying(40)	与编辑对话框中的相同。
datacenter_configuration_version	整数	在创建或更新时数据中心的配置版本。
create_date	带有时区的时间戳	被添加到系统中的时间。
update_date	带有时区的时间戳	在系统中被修改的时间。
delete_date	带有时区的时间戳	被从系统中删除的时间。

2.8.5. 主机配置

下表显示了系统中的主机配置的历史参数。

表 2.12. v4_0_configuration_history_hosts

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
host_id	uuid	系统中主机的唯一 ID。
host_unique_id	character varying(128)	此字段是主机物理 UUID 和它的一个 MAC 地址的结合，用于检测系统中已注册的主机。
host_name	character varying(255)	主机的名称（与编辑对话框中的相同）。
cluster_id	uuid	主机所属集群的唯一 ID。
host_type	smallint	<ul style="list-style-type: none"> ✧ 0 - RHEL Host ✧ 2 - RHEV Hypervisor Node
fqdn_or_ip	character varying(255)	供 Red Hat Enterprise Virtualization Manager 与之通信的主机 DNS 名称或 IP 地址，如编辑对话框中所示。
memory_size_mb	整数	主机物理内存的大小（MB）。
swap_size_mb	整数	主机交换分区的大小。
cpu_model	character varying(255)	主机的 CPU 型号。
number_of_cores	smallint	主机中的 CPU 内核总量。
number_of_sockets	smallint	CPU 插槽总量。
cpu_speed_mh	numeric(18,0)	主机 CPU 的速度（MHz）。

名称	类型	描述
host_os	character varying(255)	主机操作系统版本。
kernel_version	character varying(255)	主机的内核版本。
kvm_version	character varying(255)	主机的 KVM 版本。
vdsd_version	character varying	主机的 VDSM 版本。
vdsd_port	整数	与编辑对话框中的相同。
threads_per_core	smallint	每个内核的线程总数。
hardware_manufacturer	character varying(255)	主机的硬件生厂商。
hardware_product_name	character varying(255)	主机的硬件产品名称。
hardware_version	character varying(255)	主机的硬件版本。
hardware_serial_number	character varying(255)	主机的硬件序列号。
cluster_configuration_version	整数	在创建或更新时集群的配置版本。
create_date	带有时区的时间戳	被添加到系统中的时间。
update_date	带有时区的时间戳	在系统中被修改的时间。
delete_date	带有时区的时间戳	被从系统中删除的时间。

2.8.6. 主机接口配置

下表显示了系统中的主机接口配置的历史参数。

表 2.13. v4_0_configuration_history_hosts_interfaces

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
host_interface_id	uuid	接口在系统中的唯一的 ID。
host_interface_name	character varying(50)	主机报告的接口名。
host_id	uuid	接口所属主机的唯一 ID。
host_interface_type	smallint	<ul style="list-style-type: none"> ✧ 0 - rt18139_pv ✧ 1 - rt18139 ✧ 2 - e1000 ✧ 3 - pv
host_interface_speed_bps	整数	接口的速率（位每秒）。
mac_address	character varying(59)	接口的 MAC 地址。
logical_network_name	character varying(50)	与接口相关联的逻辑网络。
ip_address	character varying(20)	与编辑对话框中的相同。
网关	character varying(20)	与编辑对话框中的相同。
绑定	boolean	标识这个接口是否是一个绑定接口。

名称	类型	描述
bond_name	character varying(50)	接口所在的绑定名称（如果这个接口是绑定的一部分）。
vlan_id	整数	与编辑对话框中的相同。
host_configuration_version	整数	在创建或更新时主机的配置版本。
create_date	带有时区的时间戳	被添加到系统中的时间。
update_date	带有时区的时间戳	在系统中被修改的时间。
delete_date	带有时区的时间戳	被从系统中删除的时间。

2.8.7. 虚拟机配置

下表显示了系统中的虚拟机配置的历史参数。

表 2.14. v4_0_configuration_history_vms

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
vm_id	uuid	虚拟机在系统中的唯一 ID。
vm_name	character varying(255)	虚拟机的名称。
vm_description	character varying(4000)	与编辑对话框中的相同。
vm_type	smallint	<ul style="list-style-type: none"> ✱ 0 - Desktop ✱ 1 - Server
cluster_id	uuid	此虚拟机所属的集群的唯一 ID。
template_id	uuid	虚拟机所源于的模板的唯一 ID。此字段供将来使用，因为此版本中模板尚未同步到历史数据库。
template_name	character varying(40)	此虚拟机所源于的模板的名称。
cpu_per_socket	smallint	每个插槽的虚拟 CPU 的数量。
number_of_sockets	smallint	虚拟 CPU 插槽的总量。
memory_size_mb	整数	分配到虚拟机的内存总量，以 MB 为单位。

名称	类型	描述
operating_system	smallint	<ul style="list-style-type: none"> ➤ 0 - Other OS ➤ 1 - Windows XP ➤ 3 - Windows 2003 ➤ 4 - Windows 2008 ➤ 5 - Linux ➤ 7 - Red Hat Enterprise Linux 5.x ➤ 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 ➤ 1253 - Ubuntu Quantal Quetzal ➤ 1254 - Ubuntu Raring Ringtails ➤ 1255 - Ubuntu Saucy Salamander
default_host	uuid	系统中默认主机的 ID（与编辑对话框中的相同）。
high_availability	boolean	与编辑对话框中的相同。
initialized	boolean	此标记用于指出此虚拟机是否为 Sysprep 初始化目的至少启动过一次。
无状态运行	boolean	与编辑对话框中的相同。
fail_back	boolean	与编辑对话框中的相同。
usb_policy	smallint	与编辑对话框中的相同。
time_zone	character varying(40)	与编辑对话框中的相同。
vm_pool_id	uuid	此虚拟机所属的池的 ID。
vm_pool_name	character varying(255)	虚拟机所需的虚拟机池的名称。

名称	类型	描述
created_by_user_id	uuid	创建此虚拟机的用户的 ID。
cluster_configuration_version	整数	在创建或更新时集群的配置版本。
default_host_configuration_version	整数	在创建或更新时主机的配置版本。
create_date	带有时区的时间戳	被添加到系统中的时间。
update_date	带有时区的时间戳	在系统中被修改的时间。
delete_date	带有时区的时间戳	被从系统中删除的时间。

2.8.8. 虚拟机接口配置

下表显示了系统中的虚拟机接口配置的历史参数。

表 2.15. v4_0_configuration_history_vms_interfaces

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
vm_interface_id	uuid	接口在系统中的唯一的 ID。
vm_interface_name	character varying(50)	与编辑对话框中的相同。
vm_interface_type	smallint	虚拟接口的类型。 <ul style="list-style-type: none"> ✧ 0 - rt18139_pv ✧ 1 - rt18139 ✧ 2 - e1000 ✧ 3 - pv
vm_interface_speed_bps	整数	在整合时间段内接口的平均速率（位每秒）。
mac_address	character varying(20)	与编辑对话框中的相同。
logical_network_name	character varying(50)	与编辑对话框中的相同。
vm_configuration_version	整数	在创建或更新时虚拟机配置的版本。
create_date	带有时区的时间戳	被添加到系统中的时间。
update_date	带有时区的时间戳	在系统中被修改的时间。
delete_date	带有时区的时间戳	被从系统中删除的时间。

2.8.9. 虚拟机设备配置

下表列出了虚拟机和它们的相关设备（如磁盘、虚拟接口）间的关系。

表 2.16. v4_0_configuration_history_vms_devices

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
vm_id	uuid	虚拟机在系统中唯一的 ID。
device_id	uuid	设备在系统中的唯一 ID。
类型	character varying(30)	虚拟机设备的类型。它可以是 "disk" 或 "interface"。
地址	character varying(255)	设备的物理地址。

名称	类型	描述
is_managed	boolean	此标记用于指定设备是否由 Manager 管理。
is_plugged	boolean	指定设备是否被插入到虚拟机的标识。
is_readonly	boolean	指定设备是否是只读的标识。
vm_configuration_version	整数	在采样数据被采集时虚拟机配置的版本。
device_configuration_version	整数	在采样数据被采集时设备配置的版本。
create_date	带有时区的时间戳	被添加到系统中的时间。
update_date	带有时区的时间戳	被添加到系统中的时间。
delete_date	带有时区的时间戳	被添加到系统中的时间。

2.8.10. 虚拟磁盘配置

下表显示了系统中的虚拟磁盘配置的历史参数。

表 2.17. v4_0_configuration_history_vms_disks

名称	类型	描述
history_id	整数	history 数据库中的配置版本的 ID。
vm_disk_id	uuid	磁盘在系统中的唯一的 ID。
vm_disk_name	text	虚拟磁盘的名称，如编辑对话框中所示。
vm_disk_description	character varying(500)	与编辑对话框中的相同。
image_id	uuid	镜像在系统中的唯一的 ID。
storage_domain_id	uuid	磁盘镜像所属存储域的唯一 ID。
vm_disk_size_mb	整数	定义的磁盘大小（MB）。
vm_disk_type	smallint	如编辑对话框中所示。当前只使用“系统”和“数据”。 <ul style="list-style-type: none"> ➤ 0 - Unassigned ➤ 1 - System ➤ 2 - Data ➤ 3 - Shared ➤ 4 - Swap ➤ 5 - Temp
vm_disk_format	smallint	与编辑对话框中的相同。 <ul style="list-style-type: none"> ➤ 3 - Unassigned ➤ 4 - COW ➤ 5 - RAW
is_shared	boolean	标识这个虚拟机磁盘是否被共享。
create_date	带有时区的时间戳	被添加到系统中的时间。
update_date	带有时区的时间戳	在系统中被修改的时间。
delete_date	带有时区的时间戳	被从系统中删除的时间。

2.8.11. 用户详情历史

下表显示了系统中的用户的配置历史参数。

表 2.18. v4_0_users_details_history

名称	类型	描述
user_id	uuid	用户在系统中的唯一 ID，由 Manager 生成。
first_name	character varying(255)	用户的名
last_name	character varying(255)	用户的姓。
domain	character varying(255)	用户身份验证扩展的名称。
username	character varying(255)	帐户名称。
department	character varying(255)	用户所在的部门。
user_role_title	character varying(255)	用户在所处机构中的职位。
email	character varying(255)	用户的电子邮件地址。
external_id	text	外部系统中的用户的唯一 ID。
active	boolean	此标记用于指出用户是否处于活跃状态。每小时检查一次。如果用户可以在授权扩展中找到，它将保持活跃状态。成功登录后，用户便处于活跃状态。
create_date	timestamp with time zone	被添加到系统中的时间。
update_date	timestamp with time zone	在系统中被修改的时间。
delete_date	timestamp with time zone	被从系统中删除的时间。