



# **Red Hat Virtualization 4.0**

## **Data Warehouse Guide**

How to Use Data Warehouse Capabilities of Red Hat Virtualization



# Red Hat Virtualization 4.0 Data Warehouse Guide

---

How to Use Data Warehouse Capabilities of Red Hat Virtualization

Red Hat Virtualization Documentation Team  
Red Hat Customer Content Services  
[rhev-docs@redhat.com](mailto:rhev-docs@redhat.com)

## Legal Notice

Copyright © 2018 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). 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.

## Abstract

This book contains information and procedures relevant to Red Hat Virtualization Data Warehouse.

---

## Table of Contents

<b>CHAPTER 1. INSTALLING AND CONFIGURING DATA WAREHOUSE</b> .....	<b>3</b>
1.1. OVERVIEW OF CONFIGURING DATA WAREHOUSE	3
1.2. INSTALLING AND CONFIGURING DATA WAREHOUSE ON A SEPARATE MACHINE	4
1.3. MIGRATING DATA WAREHOUSE TO A SEPARATE MACHINE	7
1.4. CHANGING THE DATA WAREHOUSE SAMPLING SCALE	12
<b>CHAPTER 2. ABOUT THE HISTORY DATABASE</b> .....	<b>15</b>
2.1. HISTORY DATABASE OVERVIEW	15
2.2. TRACKING CONFIGURATION HISTORY	15
2.3. RECORDING STATISTICAL HISTORY	16
2.4. APPLICATION SETTINGS FOR THE DATA WAREHOUSE SERVICE IN OVIRT-ENGINE-DWHD.CONF	16
2.5. TRACKING TAG HISTORY	17
2.6. ALLOWING READ-ONLY ACCESS TO THE HISTORY DATABASE	17
2.7. STATISTICS HISTORY VIEWS	19
2.8. CONFIGURATION HISTORY VIEWS	31



# CHAPTER 1. INSTALLING AND CONFIGURING DATA WAREHOUSE

## 1.1. OVERVIEW OF CONFIGURING DATA WAREHOUSE

The Red Hat Virtualization Manager includes a comprehensive management history database, which can be utilized by any application to extract a range of information at the data center, cluster, and host levels. Installing Data Warehouse creates the **ovirt\_engine\_history** database, to which the Manager is configured to log information for reporting purposes.

Data Warehouse is required in Red Hat Virtualization. It can be installed and configured on the same machine as the Manager, or on a separate machine with access to the Manager:

### 1. Install and configure Data Warehouse on the Manager machine.

This configuration requires only a single registered machine, and is the simplest to configure; however, it increases the demand on the host machine. Users who require access to the Data Warehouse service will require access to the Manager machine itself. See [Configuring the Red Hat Virtualization Manager](#) in the *Installation Guide* for more information on this configuration.

### 2. Install and configure Data Warehouse a separate machine.

This configuration requires two registered machines. It reduces the load on the Manager machine and avoids potential CPU and memory-sharing conflicts on that machine. Administrators can also allow user access to the Data Warehouse machine, without the need to grant access to the Manager machine. See [Section 1.2, “Installing and Configuring Data Warehouse on a Separate Machine”](#) for more information on this configuration.

It is recommended that you set the system time zone for all machines in your Data Warehouse deployment to UTC. This ensures that data collection is not interrupted by variations in your local time zone: for example, a change from summer time to winter time.

To calculate an estimate of the space and resources the **ovirt\_engine\_history** database will use, use the [RHV Manager History Database Size Calculator](#) tool. The estimate is based on the number of entities and the length of time you have chosen to retain the history records.



## IMPORTANT

The following behavior is expected in **engine-setup**:

- Install the Data Warehouse package, run **engine-setup**, and answer **No** to configuring Data Warehouse:

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

- Run **engine-setup** again; setup no longer presents the option to configure Data Warehouse.

To force **engine-setup** to present the option again, run **engine-setup --reconfigure-optional-components**.

To configure only the currently installed Data Warehouse packages, and prevent setup from applying package updates found in enabled repositories, add the **--offline** option .

## 1.2. INSTALLING AND CONFIGURING DATA WAREHOUSE ON A SEPARATE MACHINE

Install and configure Data Warehouse on a separate machine from that on which the Red Hat Virtualization Manager is installed. Hosting the Data Warehouse service on a separate machine helps to reduce the load on the Manager machine.

### Prerequisites

- You must have installed and configured the Manager on a separate machine.
- To set up the Data Warehouse machine, you must have the following:
  - A virtual or physical machine with Red Hat Enterprise Linux 7 installed.
  - A subscription to the **Red Hat Enterprise Linux Server** and **Red Hat Virtualization** subscription pools.
  - The password from the Manager's **/etc/ovirt-engine/engine.conf.d/10-setup-database.conf** file.
  - Allowed access from the Data Warehouse machine to the Manager database machine's TCP port 5432.
- If you choose to use a remote Data Warehouse database, you must set up the database before installing the Data Warehouse service. You must have the following information about the database host:
  - The fully qualified domain name of the host
  - The port through which the database can be reached (5432 by default)
  - The database name
  - The database user



- The database password
- You must manually grant access by editing the `postgres.conf` file. Edit the `/var/lib/pgsql/data/postgresql.conf` file and modify the `listen_addresses` line so that it matches the following:

```
listen_addresses = '*'
```

If the line does not exist or has been commented out, add it manually.

If the database is hosted on the Manager machine and was configured during a clean setup of the Red Hat Virtualization Manager, access is granted by default.

### Procedure 1.1. Installing and Configuring Data Warehouse on a Separate Machine

1. Subscribe your system to the required entitlements. See [Subscribing to the Required Entitlements](#) in the *Installation Guide* for more information.
2. Ensure that all packages currently installed are up to date:

```
# yum update
```

3. Install the `ovirt-engine-dwh-setup` package:

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

4. Run the `engine-setup` command and follow the prompts to configure Data Warehouse on the machine:

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

5. Press **Enter** to accept the automatically-detected host name, or enter an alternative host name and press **Enter**:

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

6. Press **Enter** to automatically configure the firewall, or type **No** and press **Enter** to maintain existing settings:

```
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]:
```

If you choose to automatically configure the firewall, and no firewall managers are active, you are prompted to select your chosen firewall manager from a list of supported options. Type the name of the firewall manager and press **Enter**. This applies even in cases where only one option is listed.

7. Enter the fully qualified domain name of the Manager machine, and then press **Enter**:

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

8. Press **Enter** to allow setup to sign the certificate on the Manager viaSSH:

```
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. Press **Enter** to accept the defaultSSH port, or enter an alternative port number and then press **Enter**:

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

10. Enter the root password for the Manager machine:

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

11. Answer the following questions about the Data Warehouse database:

```
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]:
```

Press **Enter** to choose the highlighted defaults, or type your alternative preference and then press **Enter**. If you select **Remote**, you are prompted to provide details about the remote database host. Input the following values for the preconfigured remote database host:

```
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
```

See [Section 1.3.1, “Migrating the Data Warehouse Database to a Separate Machine”](#) for more information on how to configure and migrate the Data Warehouse database.

12. Enter the fully qualified domain name and password for the Manager database machine. Press **Enter** to accept the default values in each other field:

```
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
```

- Choose how long Data Warehouse will retain collected data:

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

**Full** uses the default values for the data storage settings listed in [Section 2.4, “Application Settings for the Data Warehouse service in `ovirt-engine-dwhd.conf`”](#) (recommended when Data Warehouse is installed on a remote host).

**Basic** reduces the values of `DWH_TABLES_KEEP_HOURLY` to **720** and `DWH_TABLES_KEEP_DAILY` to **0**, easing the load on the Manager machine (recommended when the Manager and Data Warehouse are installed on the same machine).

- Confirm your installation settings:

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

- On the Red Hat Virtualization Manager, restart the **ovirt-engine** service:

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

- Optionally, set up SSL to secure database connections using the instructions at <http://www.postgresql.org/docs/9.2/static/ssl-tcp.html#SSL-FILE-USAGE>.

## 1.3. MIGRATING DATA WAREHOUSE TO A SEPARATE MACHINE

Migrate the Data Warehouse service from the Red Hat Virtualization Manager to a separate machine. Hosting the Data Warehouse service on a separate machine reduces the load on each individual machine, and allows each service to avoid potential conflicts caused by sharing CPU and memory with other processes.

Migrate the Data Warehouse service and connect it with the existing **ovirt\_engine\_history** database, or optionally migrate the **ovirt\_engine\_history** database to a new database machine before migrating the Data Warehouse service. If the **ovirt\_engine\_history** database is hosted on the Manager, migrating the database in addition to the Data Warehouse service further reduces the competition for resources on the Manager machine. You can migrate the database to the same machine onto which you will migrate the Data Warehouse service, or to a machine that is separate from both the Manager machine and the new Data Warehouse service machine.

### 1.3.1. Migrating the Data Warehouse Database to a Separate Machine

Optionally migrate the **ovirt\_engine\_history** database before you migrate the Data Warehouse service. Use **engine-backup** to create a database backup and restore it on the

new database machine. For more information on **engine-backup**, run **engine-backup --help**.

Subscribe your system to the required entitlements. See [Subscribing to the Required Entitlements](#) in the *Installation Guide*.

To migrate the Data Warehouse service only, see [Section 1.3.2, “Migrating the Data Warehouse Service to a Separate Machine”](#).

### Procedure 1.2. Migrating the Data Warehouse Database to a Separate Machine

1. Create a backup of the Data Warehouse database and configuration files:

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

2. Copy the backup file from the Manager to the new machine:

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

3. Install **engine-backup** on the new machine:

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

4. Restore the Data Warehouse database on the new machine. *file\_name* is the backup file copied from the Manager.

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

The Data Warehouse database is now hosted on a separate machine from that on which the Manager is hosted. Proceed to [Section 1.3.2, “Migrating the Data Warehouse Service to a Separate Machine”](#) to complete the migration.

### 1.3.2. Migrating the Data Warehouse Service to a Separate Machine

Migrate a Data Warehouse service that was installed and configured on the Red Hat Virtualization Manager to a dedicated host machine. Hosting the Data Warehouse service on a separate machine helps to reduce the load on the Manager machine. Note that this procedure migrates the Data Warehouse service only; to migrate the Data Warehouse database (also known as the **ovirt\_engine\_history** database) prior to migrating the Data Warehouse service, see [Section 1.3.1, “Migrating the Data Warehouse Database to a Separate Machine”](#).

#### Prerequisites

Ensure that you have completed the following prerequisites:

1. You must have installed and configured the Manager and Data Warehouse on the same machine.
2. To set up the new Data Warehouse machine, you must have the following:

- A virtual or physical machine with Red Hat Enterprise Linux 7 installed.
- A subscription to the **Red Hat Enterprise Linux Server** and **Red Hat Virtualization** entitlement pools.
- The password from the Manager's `/etc/ovirt-engine/engine.conf.d/10-setup-database.conf` file.
- Allowed access from the Data Warehouse machine to the Manager database machine's TCP port 5432.
- The `ovirt_engine_history` database credentials from the Manager's `/etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/10-setup-database.conf` file. If you migrated the `ovirt_engine_history` database using [Section 1.3.1, “Migrating the Data Warehouse Database to a Separate Machine”](#); retrieve the credentials you defined during the database setup on that machine.

*Installing this scenario requires four steps:*

1. Set up the new Data Warehouse machine: Procedure 1.3. Setting up the New Data Warehouse Machine
2. Stop the Data Warehouse service on the Manager machine: Procedure 1.4. Stopping the Data Warehouse Service on the Manager Machine
3. Configure the new Data Warehouse machine: Procedure 1.5. Configuring the New Data Warehouse Machine
4. Disable the Data Warehouse service on the Manager machine: Procedure 1.6. Disabling the Data Warehouse Service on the Manager Machine

### Procedure 1.3. Setting up the New Data Warehouse Machine

1. Subscribe your system to the required entitlements. See [Subscribing to the Required Entitlements](#) in the *Installation Guide* for more information.
2. Ensure that all packages currently installed are up to date:

```
# yum update
```

3. Install the `ovirt-engine-dwh-setup` package:

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

### Procedure 1.4. Stopping the Data Warehouse Service on the Manager Machine

1. Stop the Data Warehouse service:

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

2. If the database is hosted on a remote machine, you must manually grant access by editing the `postgres.conf` file. Edit the `/var/lib/pgsql/data/postgresql.conf` file and modify the `listen_addresses` line so that it matches the following:

```
listen_addresses = '*'
```

■

If the line does not exist or has been commented out, add it manually.

If the database is hosted on the Manager machine and was configured during a clean setup of the Red Hat Virtualization Manager, access is granted by default.

See [Section 1.3.1, “Migrating the Data Warehouse Database to a Separate Machine”](#) for more information on how to configure and migrate the Data Warehouse database.

- Restart the postgresql service:

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

### Procedure 1.5. Configuring the New Data Warehouse Machine

The questions shown in this step only appear if you are migrating the **ovirt\_engine\_history** database to one machine and Data Warehouse service to a different machine. The order of the questions may differ depending on your environment.

- If you are migrating both the **ovirt\_engine\_history** database and the Data Warehouse service to the *same* machine, run the following, otherwise proceed to the next step.

```
# 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
```

- Run the **engine-setup** command to begin configuration of Data Warehouse on the machine:

```
# engine-setup
```

- Press **Enter** to configure Data Warehouse:

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

- Press **Enter** to accept the automatically detected hostname, or enter an alternative hostname and press **Enter**:

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

- Press **Enter** to automatically configure the firewall, or type **No** and press **Enter** to maintain existing settings:

```
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]:

If you choose to automatically configure the firewall, and no firewall managers are active, you are prompted to select your chosen firewall manager from a list of supported options. Type the name of the firewall manager and press **Enter**. This applies even in cases where only one option is listed.

6. Enter the fully qualified domain name and password for the Manager. Press **Enter** to accept the default values in each other field:

```
Host fully qualified DNS name of the engine server []: engine-fqdn
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]:
ssh port on remote engine server [22]:
root password on remote engine server engine-fqdn: password
```

7. Answer the following question about the location of the **ovirt\_engine\_history** database:

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

Type the alternative option as shown above then press **Enter**.

8. Enter the fully qualified domain name and password for your **ovirt\_engine\_history** host. Press **Enter** to accept the default values in each other field:

```
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
```

See [Section 1.3.1, “Migrating the Data Warehouse Database to a Separate Machine”](#) for more information on how to configure and migrate the Data Warehouse database.

9. Enter the fully qualified domain name and password for the Manager database machine. Press **Enter** to accept the default values in each other field:

```
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
```

- Choose how long Data Warehouse will retain collected data:

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

- ```
(1) Basic
(2) Full
(1, 2)[1]:
```

**Full** uses the default values for the data storage settings listed in [Section 2.4, “Application Settings for the Data Warehouse service in ovirt-engine-dwhd.conf”](#) (recommended when Data Warehouse is installed on a remote host).



#### NOTE

If you migrate from **Basic** to **Full**, initially only the existing basic data will be available.

**Basic** reduces the values of `DWH_TABLES_KEEP_HOURLY` to **720** and `DWH_TABLES_KEEP_DAILY` to **0**, easing the load on the Manager machine but with a less detailed history.

- Confirm that you want to permanently disconnect the existing Data Warehouse service from the Manager:

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

- Confirm your installation settings:

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

### Procedure 1.6. Disabling the Data Warehouse Service on the Manager Machine

- On the Manager machine, restart the Manager:

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

This step is required for the Dashboard to function.

- Disable the Data Warehouse service:

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

- Remove the Data Warehouse files:

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

The Data Warehouse service is now hosted on a separate machine from that on which the Manager is hosted.

## 1.4. CHANGING THE DATA WAREHOUSE SAMPLING SCALE



Data Warehouse is required in Red Hat Virtualization. It can be installed and configured on the same machine as the Manager, or on a separate machine with access to the Manager. The default data retention settings may not be required for all setups, so **engine-setup** offers two data sampling scales: **Basic** and **Full**.

- **Full** uses the default values for the data retention settings listed in [Section 2.4, “Application Settings for the Data Warehouse service in ovirt-engine-dwhd.conf”](#) (recommended when Data Warehouse is installed on a remote host).
- **Basic** reduces the values of **DWH\_TABLES\_KEEP\_HOURLY** to **720** and **DWH\_TABLES\_KEEP\_DAILY** to **0**, easing the load on the Manager machine (recommended when the Manager and Data Warehouse are installed on the same machine).

The sampling scale is configured by **engine-setup** during installation:

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

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

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

You can change the sampling scale later by running **engine-setup** again with the **--reconfigure-dwh-scale** option.

### Example 1.1. Changing the Data Warehouse Sampling Scale

```
# 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/4.0/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]:
```

You can also adjust individual data retention settings if necessary, as documented in [Section 2.4, “Application Settings for the Data Warehouse service in ovirt-engine-dwhd.conf”](#).

## CHAPTER 2. ABOUT THE HISTORY DATABASE

### 2.1. HISTORY DATABASE OVERVIEW

Red Hat Virtualization includes a comprehensive management history database, which can be used by reporting applications to generate reports at data center, cluster and host levels. This chapter provides information to enable you to set up queries against the history database.

Red Hat Virtualization Manager uses **PostgreSQL 9.2.x** as a database platform to store information about the state of the virtualization environment, its configuration and performance. At install time, Red Hat Virtualization Manager creates a PostgreSQL database called **engine**.

Installing the **ovirt-engine-dwh** package creates a second database called **ovirt\_engine\_history**, which contains historical configuration information and statistical metrics collected every minute over time from the **engine** operational database. Tracking the changes to the database provides information on the objects in the database, enabling the user to analyze activity, enhance performance, and resolve difficulties.



#### WARNING

The replication of data in the **ovirt\_engine\_history** database is performed by the Red Hat Virtualization Manager Extract Transform Load Service, **ovirt-engine-dwhd**. The service is based on Talend Open Studio, a data integration tool. This service is configured to start automatically during the data warehouse package setup. It is a Java program responsible for extracting data from the **engine** database, transforming the data to the history database standard and loading it to the **ovirt\_engine\_history** database.

The **ovirt-engine-dwhd** service must not be stopped.

The **ovirt\_engine\_history** database schema changes over time. The database includes a set of database views to provide a supported, versioned API with a consistent structure. A view is a virtual table composed of the result set of a database query. The database stores the definition of a view as a **SELECT** statement. The result of the **SELECT** statement populates the virtual table that the view returns. A user references the view name in **PL/PGSQL** statements the same way a table is referenced.

### 2.2. TRACKING CONFIGURATION HISTORY

Data from the Red Hat Virtualization History Database (called **ovirt\_engine\_history**) can be used to track the **engine** database.

The ETL service, **ovirt-engine-dwhd**, tracks three types of changes:

- A new entity is added to the **engine** database - the ETL Service replicates the change to the **ovirt\_engine\_history** database as a new entry.

- An existing entity is updated - the ETL Service replicates the change to the **ovirt\_engine\_history** database as a new entry.
- An entity is removed from the **engine** database - A new entry in the **ovirt\_engine\_history** database flags the corresponding entity as removed. Removed entities are only flagged as removed.

The configuration tables in the **ovirt\_engine\_history** database differ from the corresponding tables in the **engine** database in several ways. The most apparent difference is they contain fewer **configuration** columns. This is because certain configuration items are less interesting to report than others and are not kept due to database size considerations. Also, columns from a few tables in the **engine** database appear in a single table in **ovirt\_engine\_history** and have different column names to make viewing data more convenient and comprehensible. All configuration tables contain:

- a **history\_id** to indicate the configuration version of the entity;
- a **create\_date** field to indicate when the entity was added to the system;
- an **update\_date** field to indicate when the entity was changed; and
- a **delete\_date** field to indicate the date the entity was removed from the system.

## 2.3. RECORDING STATISTICAL HISTORY

The ETL service collects data into the statistical tables every minute. Data is stored for every minute of the past 24 hours, at a minimum, but can be stored for as long as 48 hours depending on the last time a deletion job was run. Minute-by-minute data more than two hours old is aggregated into hourly data and stored for two months. Hourly data more than two days old is aggregated into daily data and stored for five years.

Hourly data and daily data can be found in the hourly and daily tables.

Each statistical datum is kept in its respective aggregation level table: samples, hourly, and daily history. All history tables also contain a **history\_id** column to uniquely identify rows. Tables reference the configuration version of a host in order to enable reports on statistics of an entity in relation to its past configuration.

## 2.4. APPLICATION SETTINGS FOR THE DATA WAREHOUSE SERVICE IN OVIRT-ENGINE-DWHD.CONF

The following is a list of options for configuring application settings for the Data Warehouse service. These options are available in the **/usr/share/ovirt-engine-dwh/services/ovirt-engine-dwhd/ovirt-engine-dwhd.conf** file. Configure any changes to the default values in an override file under **/etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/**. Restart the Data Warehouse service after saving the changes.

**Table 2.1. ovirt-engine-dwhd.conf application settings variables**

Variable name	Default Value	Remarks
<b>DWH_DELETE_JOB_HOUR</b>	<b>3</b>	The time at which a deletion job is run. Specify a value between <b>0</b> and <b>23</b> , where <b>0</b> is midnight.

Variable name	Default Value	Remarks
<b>DWH_SAMPLING</b>	<b>60</b>	The interval, in seconds, at which data is collected into statistical tables.
<b>DWH_TABLES_KEEP_SAMPLES</b>	<b>24</b>	The number of hours that data from <b>DWH_SAMPLING</b> is stored. Data more than two hours old is aggregated into hourly data.
<b>DWH_TABLES_KEEP_HOURLY</b>	<b>1440</b>	The number of hours that hourly data is stored. The default is 60 days. Hourly data more than two days old is aggregated into daily data.
<b>DWH_TABLES_KEEP_DAILY</b>	<b>43800</b>	The number of hours that daily data is stored. The default is five years.
<b>DWH_ERROR_EVENT_INTERVAL</b>	<b>300000</b>	The minimum interval, in milliseconds, at which errors are pushed to the Manager's <b>audit.log</b> .

## 2.5. TRACKING TAG HISTORY

The ETL Service collects tag information as displayed in the Administration Portal every minute and stores this data in the tags historical tables. The ETL Service tracks five types of changes:

- A tag is created in the Administration Portal - the ETL Service copies the tag details, position in the tag tree and relation to other objects in the tag tree.
- A entity is attached to the tag tree in the Administration Portal - the ETL Service replicates the addition to the **ovirt\_engine\_history** database as a new entry.
- A tag is updated - the ETL Service replicates the change of tag details to the **ovirt\_engine\_history** database as a new entry.
- An entity or tag branch is removed from the Administration Portal - the **ovirt\_engine\_history** database flags the corresponding tag and relations as removed in new entries. Removed tags and relations are only flagged as removed or detached.
- A tag branch is moved - the corresponding tag and relations are updated as new entries. Moved tags and relations are only flagged as updated.

## 2.6. ALLOWING READ-ONLY ACCESS TO THE HISTORY DATABASE

To allow access to the history database without allowing edits, you must create a read-only PostgreSQL user that can log in to and read from the **ovirt\_engine\_history** database. This procedure must be executed on the system on which the history database is installed.

### Procedure 2.1. Allowing Read-Only Access to the History Database

1. Create the user to be granted read-only access to the history database:

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

2. Grant the newly created user permission to connect to the history database:

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

3. Grant the newly created user usage of the **public** schema:

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

4. Generate the rest of the permissions that will be granted to the newly created user and save them to a file:

```
# 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. Use the file you created in the previous step to grant permissions to the newly created user:

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

6. Remove the file you used to grant permissions to the newly created user:

```
# rm grant.sql
```

7. Add the following lines for the newly created user to the **/var/lib/pgsql/data/ph\_hba.conf** file:

```
# TYPE      DATABASE          USER              ADDRESS
METHOD
host       ovirt_engine_history  [user name]      0.0.0.0/0
md5
host       ovirt_engine_history  [user name]      ::0/0
md5
```

8. Reload the PostgreSQL service:

```
# systemctl reload postgresql
```

9. You can test the read-only user's access permissions:

```
# psql -U [user name] ovirt_engine_history -h localhost
Password for user [user name]:
psql (9.2.23)
Type "help" for help.

ovirt_engine_history=>
```

10. To exit the **ovirt\_engine\_history** database, enter **\q**.

The read-only user's **SELECT** statements against tables and views in the **ovirt\_engine\_history** database succeed, while modifications fail.

## 2.7. STATISTICS HISTORY VIEWS

Statistics data is available in **hourly**, **daily**, and **samples** views.

To query a statistics view, run **SELECT \* FROM view\_name\_[hourly|daily|samples];**. For example:

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

To list all available views, run:

```
# \dv
```

### 2.7.1. Enabling Debug Mode

You can enable debug mode to record log sampling, hourly, and daily job times in the **/var/log/ovirt-engine-dwh/ovirt-engine-dwhd.log** file. This is useful for checking the ETL process. Debug mode is disabled by default.

1. Log in to the Manager machine and create a configuration file (for example, **/etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/logging.conf**).
2. Add the following line to the configuration file:

```
# DWH_AGGREGATION_DEBUG=true
```

3. Restart the **ovirt-engine-dwhd** service:

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

To disable debug mode, delete the configuration file and restart the service.

### 2.7.2. Storage Domain Statistics Views

**Table 2.2. Historical Statistics for Each Storage Domain in the System**

Name	Type	Description
history_id	bigint	The unique ID of this row in the table.
history_datetime	date	The timestamp of this history row (rounded to minute, hour, day as per the aggregation level).
storage_domain_id	uuid	Unique ID of the storage domain in the system.
storage_domain_status	smallint	The storage domain status.
seconds_in_status	integer	The total number of seconds that the storage domain was in the status shown state as shown in the status column for the aggregation period. For example, if a storage domain was "Active" for 55 seconds and "Inactive" for 5 seconds within a minute, two rows will be reported in the table for the same minute. One row will have a status of Active with seconds_in_status of 55, the other will have a status of Inactive and seconds_in_status of 5.
minutes_in_status	numeric(7,2)	The total number of minutes that the storage domain was in the status shown state as shown in the status column for the aggregation period. For example, if a storage domain was "Active" for 55 minutes and "Inactive" for 5 minutes within an hour, two rows will be reported in the table for the same hour. One row will have a status of Active with minutes_in_status of 55, the other will have a status of Inactive and minutes_in_status of 5.
available_disk_size_gb	integer	The total available (unused) capacity on the disk, expressed in gigabytes (GB).
used_disk_size_gb	integer	The total used capacity on the disk, expressed in gigabytes (GB).
storage_configuration_version	integer	The storage domain configuration version at the time of sample.



### 2.7.3. Host Statistics Views

**Table 2.3. Historical Statistics for Each Host in the System**

Name	Type	Description
history_id	bigint	The unique ID of this row in the table.
history_datetime	date	The timestamp of this history row (rounded to minute, hour, day as per the aggregation level).
host_id	uuid	Unique ID of the host in the system.
host_status	smallint	<ul style="list-style-type: none"> <li>• -1 - <b>Unknown Status</b> (used only to indicate a problem with the ETL. Please notify Red Hat Support)</li> <li>• 1 - <b>Up</b></li> <li>• 2 - <b>Maintenance</b></li> <li>• 3 - <b>Problematic</b></li> </ul>
seconds_in_status	integer	The total number of seconds that the host was in the status shown in the status column for the aggregation period. For example, if a host was up for 55 seconds and down for 5 seconds during a minute, two rows will show for this minute. One will have a status of <b>Up</b> and seconds_in_status of 55, the other will have a status of <b>Down</b> and a seconds_in_status of 5.
minutes_in_status	numeric(7,2)	The total number of minutes that the host was in the status shown in the status column for the aggregation period. For example, if a host was up for 55 minutes and down for 5 minutes during an hour, two rows will show for this hour. One will have a status of <b>Up</b> and minutes_in_status of 55, the other will have a status of <b>Down</b> and a minutes_in_status of 5.
memory_usage_percent	smallint	Percentage of used memory on the host.

Name	Type	Description
max_memory_usage	smallint	The maximum memory usage for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
ksm_shared_memory_mb	bigint	The Kernel Shared Memory size, in megabytes (MB), that the host is using.
max_ksm_shared_memory_mb	bigint	The maximum KSM memory usage for the aggregation period expressed in megabytes (MB). For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
cpu_usage_percent	smallint	Used CPU percentage on the host.
max_cpu_usage	smallint	The maximum CPU usage for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
ksm_cpu_percent	smallint	CPU percentage ksm on the host is using.
max_ksm_cpu_percent	smallint	The maximum KSM usage for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
active_vms	smallint	The average number of active virtual machines for this aggregation.

Name	Type	Description
max_active_vms	smallint	The maximum active number of virtual machines for the aggregation period. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
total_vms	smallint	The average number of all virtual machines on the host for this aggregation.
max_total_vms	smallint	The maximum total number of virtual machines for the aggregation period. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
total_vms_vcpus	integer	Total number of vCPUs allocated to the host.
max_total_vms_vcpus	integer	The maximum total virtual machine vCPU number for the aggregation period. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
cpu_load	integer	The CPU load of the host.
max_cpu_load	integer	The maximum CPU load for the aggregation period. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
system_cpu_usage_percent	smallint	Used CPU percentage on the host.
max_system_cpu_usage_percent	smallint	The maximum system CPU usage for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
user_cpu_usage_percent	smallint	Used user CPU percentage on the host.

Name	Type	Description
max_user_cpu_usage_percent	smallint	The maximum user CPU usage for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
swap_used_mb	integer	Used swap size usage of the host in megabytes (MB).
max_swap_used_mb	integer	The maximum user swap size usage of the host for the aggregation period in megabytes (MB), expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
host_configuration_version	integer	The host configuration version at the time of sample.

#### 2.7.4. Host Interface Statistics Views

**Table 2.4. Historical Statistics for Each Host Network Interface in the System**

Name	Type	Description
history_id	bigint	The unique ID of this row in the table.
history_datetime	date	The timestamp of this history view (rounded to minute, hour, day as per the aggregation level).
host_interface_id	uuid	Unique identifier of the interface in the system.
receive_rate_percent	smallint	Used receive rate percentage on the host.

Name	Type	Description
max_receive_rate_percent	smallint	The maximum receive rate for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
transmit_rate_percent	smallint	Used transmit rate percentage on the host.
max_transmit_rate_percent	smallint	The maximum transmit rate for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
received_total_byte	bigint	The total number of bytes received by the host.
transmitted_total_byte	bigint	The total number of bytes transmitted from the host.
host_interface_configuration_version	integer	The host interface configuration version at the time of sample.

### 2.7.5. Virtual Machine Statistics Views

**Table 2.5. Historical Statistics for Each Virtual Machine in the System**

Name	Type	Description
history_id	bigint	The unique ID of this row in the table.
history_datetime	date	The timestamp of this history row (rounded to minute, hour, day as per the aggregation level).
vm_id	uuid	Unique ID of the virtual machine in the system.

Name	Type	Description
vm_status	smallint	<ul style="list-style-type: none"> <li>• -1 - Unknown Status (used only to indicate problems with the ETL. Please notify Red Hat Support)</li> <li>• 0 - Down</li> <li>• 1 - Up</li> <li>• 2 - Paused</li> <li>• 3 - Problematic</li> </ul>
seconds_in_status	integer	<p>The total number of seconds that the virtual machine was in the status shown in the status column for the aggregation period. For example, if a virtual machine was up for 55 seconds and down for 5 seconds during a minute, two rows will show for this minute. One will have a status of Up and seconds_in_status, the other will have a status of Down and a seconds_in_status of 5.</p>
minutes_in_status	numeric(7,2)	<p>The total number of minutes that the virtual machine was in the status shown in the status column for the aggregation period. For example, if a virtual machine was up for 55 minutes and down for 5 minutes during an hour, two rows will show for this hour. One will have a status of Up and minutes_in_status, the other will have a status of Down and a minutes_in_status of 5.</p>
cpu_usage_percent	smallint	<p>The percentage of the CPU in use by the virtual machine.</p>
max_cpu_usage	smallint	<p>The maximum CPU usage for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.</p>

Name	Type	Description
memory_usage_percent	smallint	Percentage of used memory in the virtual machine. The guest tools must be installed on the virtual machine for memory usage to be recorded.
max_memory_usage	smallint	The maximum memory usage for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value. The guest tools must be installed on the virtual machine for memory usage to be recorded.
user_cpu_usage_percent	smallint	Used user CPU percentage on the host.
max_user_cpu_usage_percent	smallint	The maximum user CPU usage for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregation, it is the maximum hourly average value.
system_cpu_usage_percent	smallint	Used system CPU percentage on the host.
max_system_cpu_usage_percent	smallint	The maximum system CPU usage for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
vm_ip	text	The IP address of the first NIC. Only shown if the guest agent is installed.
currently_running_on_host	uuid	The unique ID of the host the virtual machine is running on.
current_user_id	uuid	The unique ID of the user logged into the virtual machine console, if the guest agent is installed.

Name	Type	Description
disks_usage	text	The disk description. File systems type, mount point, total size, and used size.
vm_configuration_version	integer	The virtual machine configuration version at the time of sample.
current_host_configuration_version	integer	The host configuration version at the time of sample.
memory_buffered_kb	bigint	The amount of buffered memory on the virtual machine, in kilobytes (KB).
memory_cached_kb	bigint	The amount of cached memory on the virtual machine, in kilobytes (KB).
max_memory_buffered_kb	bigint	The maximum buffered memory for the aggregation period, in kilobytes (KB). For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
max_memory_cached_kb	bigint	The maximum cached memory for the aggregation period, in kilobytes (KB). For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.

## 2.7.6. Virtual Machine Interface Statistics Views

**Table 2.6. Historical Statistics for the Virtual Machine Network Interfaces in the System**

Name	Type	Description
history_id	integer	The unique ID of this row in the table.
history_datetime	date	The timestamp of this history row (rounded to minute, hour, day as per the aggregation level).
vm_interface_id	uuid	Unique ID of the interface in the system.



Name	Type	Description
receive_rate_percent	smallint	Used receive rate percentage on the host.
max_receive_rate_percent	smallint	The maximum receive rate for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
transmit_rate_percent	smallint	Used transmit rate percentage on the host.
max_transmit_rate_percent	smallint	The maximum transmit rate for the aggregation period, expressed as a percentage. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average rate.
received_total_byte	bigint	The total number of bytes received by the virtual machine.
transmitted_total_byte	bigint	The total number of bytes transmitted from the virtual machine.
vm_interface_configuration_version	integer	The virtual machine interface configuration version at the time of sample.

### 2.7.7. Virtual Disk Statistics Views

**Table 2.7. Historical Statistics for the Virtual Disks in the System**

Name	Type	Description
history_id	bigint	The unique ID of this row in the table.
history_datetime	date	The timestamp of this history row (rounded to minute, hour, day as per the aggregation level).
vm_disk_id	uuid	Unique ID of the disk in the system.

Name	Type	Description
vm_disk_status	smallint	<ul style="list-style-type: none"> <li>• 0 - Unassigned</li> <li>• 1 - OK</li> <li>• 2 - Locked</li> <li>• 3 - Invalid</li> <li>• 4 - Illegal</li> </ul>
seconds_in_status	integer	The total number of seconds that the virtual disk was in the status shown in the status column for the aggregation period. For example, if a virtual disk was locked for 55 seconds and OK for 5 seconds during a minute, two rows will show for this minute. One will have a status of <b>Locked</b> and seconds_in_status of 55, the other will have a status of OK and a seconds_in_status of 5.
minutes_in_status	numeric(7,2)	The total number of minutes that the virtual disk was in the status shown in the status column for the aggregation period. For example, if a virtual disk was locked for 55 minutes and OK for 5 minutes during an hour, two rows will show for this hour. One will have a status of <b>Locked</b> and minutes_in_status of 55, the other will have a status of OK and a minutes_in_status of 5.
vm_disk_actual_size_mb	integer	The actual size allocated to the disk.
read_rate_bytes_per_second	integer	Read rate to disk in bytes per second.
max_read_rate_bytes_per_second	integer	The maximum read rate for the aggregation period. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
read_latency_seconds	numeric(18,9)	The virtual disk read latency measured in seconds.

Name	Type	Description
max_read_latency_seconds	numeric(18,9)	The maximum read latency for the aggregation period, measured in seconds. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
write_rate_bytes_per_second	integer	Write rate to disk in bytes per second.
max_write_rate_bytes_per_second	integer	The maximum write rate for the aggregation period. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
write_latency_seconds	numeric(18,9)	The virtual disk write latency measured in seconds.
max_write_latency_seconds	numeric(18,9)	The maximum write latency for the aggregation period, measured in seconds. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
flush_latency_seconds	numeric(18,9)	The virtual disk flush latency measured in seconds.
max_flush_latency_seconds	numeric(18,9)	The maximum flush latency for the aggregation period, measured in seconds. For hourly aggregations, this is the maximum collected sample value. For daily aggregations, it is the maximum hourly average value.
vm_disk_configuration_version	integer	The virtual disk configuration version at the time of sample.

## 2.8. CONFIGURATION HISTORY VIEWS

To query a configuration view, run **SELECT \* FROM *view\_name*;** For example:

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

To list all available views, run:

```
# \dv
```



#### NOTE

**delete\_date** does not appear in latest views because these views provide the latest configuration of living entities, which, by definition, have not been deleted.

### 2.8.1. Data Center Configuration

The following table shows the configuration history parameters of the data centers in the system.

**Table 2.8. v4\_0\_configuration\_history\_datacenters**

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
datacenter_id	uuid	The unique ID of the data center in the system.
datacenter_name	character varying(40)	Name of the data center, as displayed in the edit dialog.
datacenter_description	character varying(4000)	Description of the data center, as displayed in the edit dialog.
is_local_storage	boolean	A flag to indicate whether the data center uses local storage.
create_date	timestamp with time zone	The date this entity was added to the system.
update_date	timestamp with time zone	The date this entity was changed in the system.
delete_date	timestamp with time zone	The date this entity was deleted from the system.

### 2.8.2. Datacenter Storage Domain Map

The following table shows the relationships between storage domains and data centers in the system.

**Table 2.9. v4\_0\_map\_history\_datacenters\_storage\_domains**

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
storage_domain_id	uuid	The unique ID of this storage domain in the system.
datacenter_id	uuid	The unique ID of the data center in the system.
attach_date	timestamp with time zone	The date the storage domain was attached to the data center.
detach_date	timestamp with time zone	The date the storage domain was detached from the data center.

### 2.8.3. Storage Domain Configuration

The following table shows the configuration history parameters of the storage domains in the system.

**Table 2.10. v4\_0\_configuration\_history\_storage\_domains**

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
storage_domain_id	uuid	The unique ID of this storage domain in the system.
storage_domain_name	character varying(250)	Storage domain name.
storage_domain_type	smallint	<ul style="list-style-type: none"> <li>• 0 - Data (Master)</li> <li>• 1 - Data</li> <li>• 2 - ISO</li> <li>• 3 - Export</li> </ul>

Name	Type	Description
storage_type	smallint	<ul style="list-style-type: none"> <li>• 0 - Unknown</li> <li>• 1 - NFS</li> <li>• 2 - FCP</li> <li>• 3 - iSCSI</li> <li>• 4 - Local</li> <li>• 6 - All</li> </ul>
create_date	timestamp with time zone	The date this entity was added to the system.
update_date	timestamp with time zone	The date this entity was changed in the system.
delete_date	timestamp with time zone	The date this entity was deleted from the system.

#### 2.8.4. Cluster Configuration

The following table shows the configuration history parameters of the clusters in the system.

**Table 2.11. v4\_0\_configuration\_history\_clusters**

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
cluster_id	uuid	The unique identifier of the datacenter this cluster resides in.
cluster_name	character varying(40)	Name of the cluster, as displayed in the edit dialog.
cluster_description	character varying(4000)	As defined in the edit dialog.
datacenter_id	uuid	The unique identifier of the datacenter this cluster resides in.
cpu_name	character varying(255)	As displayed in the edit dialog.

Name	Type	Description
compatibility_version	character varying(40)	As displayed in the edit dialog.
datacenter_configuration_version	integer	The data center configuration version at the time of creation or update.
create_date	timestamp with time zone	The date this entity was added to the system.
update_date	timestamp with time zone	The date this entity was changed in the system.
delete_date	timestamp with time zone	The date this entity was deleted from the system.

### 2.8.5. Host Configuration

The following table shows the configuration history parameters of the hosts in the system.

**Table 2.12. v4\_0\_configuration\_history\_hosts**

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
host_id	uuid	The unique ID of the host in the system.
host_unique_id	character varying(128)	This field is a combination of the host's physical UUID and one of its MAC addresses, and is used to detect hosts already registered in the system.
host_name	character varying(255)	Name of the host (same as in the edit dialog).
cluster_id	uuid	The unique ID of the cluster that this host belongs to.
host_type	smallint	<ul style="list-style-type: none"> <li>• 0 - RHEL Host</li> <li>• 2 - RHEV Hypervisor Node</li> </ul>

Name	Type	Description
fqdn_or_ip	character varying(255)	The host's DNS name or its IP address for Red Hat Virtualization Manager to communicate with (as displayed in the edit dialog).
memory_size_mb	integer	The host's physical memory capacity, expressed in megabytes (MB).
swap_size_mb	integer	The host swap partition size.
cpu_model	character varying(255)	The host's CPU model.
number_of_cores	smallint	Total number of CPU cores in the host.
number_of_sockets	smallint	Total number of CPU sockets.
cpu_speed_mh	numeric(18,0)	The host's CPU speed, expressed in megahertz (MHz).
host_os	character varying(255)	The host's operating system version.
kernel_version	character varying(255)	The host's kernel version.
kvm_version	character varying(255)	The host's KVM version.
vdsm_version	character varying	The host's VDSM version.
vdsm_port	integer	As displayed in the edit dialog.
threads_per_core	smallint	Total number of threads per core.
hardware_manufacturer	character varying(255)	The host's hardware manufacturer.
hardware_product_name	character varying(255)	The product name of the host's hardware.
hardware_version	character varying(255)	The version of the host's hardware.



Name	Type	Description
hardware_serial_number	character varying(255)	The serial number of the host's hardware.
cluster_configuration_version	integer	The cluster configuration version at the time of creation or update.
create_date	timestamp with time zone	The date this entity was added to the system.
update_date	timestamp with time zone	The date this entity was changed in the system.
delete_date	timestamp with time zone	The date this entity was deleted from the system.

### 2.8.6. Host Interface Configuration

The following table shows the configuration history parameters of the host interfaces in the system.

**Table 2.13. v4\_0\_configuration\_history\_hosts\_interfaces**

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
host_interface_id	uuid	The unique ID of this interface in the system.
host_interface_name	character varying(50)	The interface name as reported by the host.
host_id	uuid	Unique ID of the host this interface belongs to.
host_interface_type	smallint	<ul style="list-style-type: none"> <li>• 0 - rt18139_pv</li> <li>• 1 - rt18139</li> <li>• 2 - e1000</li> <li>• 3 - pv</li> </ul>
host_interface_speed_bps	integer	The interface speed in bits per second.

Name	Type	Description
mac_address	character varying(59)	The interface MAC address.
logical_network_name	character varying(50)	The logical network associated with the interface.
ip_address	character varying(20)	As displayed in the edit dialog.
gateway	character varying(20)	As displayed in the edit dialog.
bond	boolean	A flag to indicate if this interface is a bonded interface.
bond_name	character varying(50)	The name of the bond this interface is part of (if it is part of a bond).
vlan_id	integer	As displayed in the edit dialog.
host_configuration_version	integer	The host configuration version at the time of creation or update.
create_date	timestamp with time zone	The date this entity was added to the system.
update_date	timestamp with time zone	The date this entity was changed in the system.
delete_date	timestamp with time zone	The date this entity was deleted from the system.

### 2.8.7. Virtual Machine Configuration

The following table shows the configuration history parameters of the virtual machines in the system.

**Table 2.14. v4\_0\_configuration\_history\_vms**

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
vm_id	uuid	The unique ID of this virtual machine in the system.

Name	Type	Description
vm_name	character varying(255)	The name of the virtual machine.
vm_description	character varying(4000)	As displayed in the edit dialog.
vm_type	smallint	<ul style="list-style-type: none"> <li>• 0 - Desktop</li> <li>• 1 - Server</li> </ul>
cluster_id	uuid	The unique ID of the cluster this virtual machine belongs to.
template_id	uuid	The unique ID of the template this virtual machine is derived from. The field is for future use, as the templates are not synchronized to the history database in this version.
template_name	character varying(40)	Name of the template from which this virtual machine is derived.
cpu_per_socket	smallint	Virtual CPUs per socket.
number_of_sockets	smallint	Total number of virtual CPU sockets.
memory_size_mb	integer	Total memory allocated to the virtual machine, expressed in megabytes (MB).
operating_system	smallint	<ul style="list-style-type: none"> <li>• 0 - Other OS</li> <li>• 1 - Windows XP</li> <li>• 3 - Windows 2003</li> <li>• 4 - Windows 2008</li> <li>• 5 - Linux</li> <li>• 7 - Red Hat Enterprise Linux 5.x</li> <li>• 8 - Red Hat Enterprise Linux 4.x</li> <li>• 9 - Red Hat Enterprise Linux 3.x</li> <li>• 10 - Windows 2003 x64</li> </ul>

Name	Type	Description
		<ul style="list-style-type: none"> <li>• 11 - Windows 7</li> <li>• 12 - Windows 7 x64</li> <li>• 13 - Red Hat Enterprise Linux 5.x x64</li> <li>• 14 - Red Hat Enterprise Linux 4.x x64</li> <li>• 15 - Red Hat Enterprise Linux 3.x x64</li> <li>• 16 - Windows 2008 x64</li> <li>• 17 - Windows 2008 R2 x64</li> <li>• 18 - Red Hat Enterprise Linux 6.x</li> <li>• 19 - Red Hat Enterprise Linux 6.x x64</li> <li>• 20 - Windows 8</li> <li>• 21 - Windows 8 x64</li> <li>• 23 - Windows 2012 x64</li> <li>• 1001 - Other</li> <li>• 1002 - Linux</li> <li>• 1003 - Red Hat Enterprise Linux 6.x</li> <li>• 1004 - SUSE Linux Enterprise Server 11</li> <li>• 1193 - SUSE Linux Enterprise Server 11</li> <li>• 1252 - Ubuntu Precise Pangolin LTS</li> <li>• 1253 - Ubuntu Quantal Quetzal</li> <li>• 1254 - Ubuntu Raring Ringtails</li> <li>• 1255 - Ubuntu Saucy Salamander</li> </ul>
default_host	uuid	As displayed in the edit dialog, the ID of the default host in the system.
high_availability	boolean	As displayed in the edit dialog.

Name	Type	Description
initialized	boolean	A flag to indicate if this virtual machine was started at least once for Sysprep initialization purposes.
stateless	boolean	As displayed in the edit dialog.
fail_back	boolean	As displayed in the edit dialog.
usb_policy	smallint	As displayed in the edit dialog.
time_zone	character varying(40)	As displayed in the edit dialog.
vm_pool_id	uuid	The ID of the pool to which this virtual machine belongs.
vm_pool_name	character varying(255)	The name of the virtual machine's pool.
created_by_user_id	uuid	The ID of the user that created this virtual machine.
cluster_configuration_version	integer	The cluster configuration version at the time of creation or update.
default_host_configuration_version	integer	The host configuration version at the time of creation or update.
create_date	timestamp with time zone	The date this entity was added to the system.
update_date	timestamp with time zone	The date this entity was changed in the system.
delete_date	timestamp with time zone	The date this entity was deleted from the system.

### 2.8.8. Virtual Machine Interface Configuration

The following table shows the configuration history parameters of the virtual interfaces in the system.

**Table 2.15. v4\_0\_configuration\_history\_vms\_interfaces**

Name	Type	Description
------	------	-------------

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
vm_interface_id	uuid	The unique ID of this interface in the system.
vm_interface_name	character varying(50)	As displayed in the edit dialog.
vm_interface_type	smallint	The type of the virtual interface. <ul style="list-style-type: none"> <li>• 0 - rt18139_pv</li> <li>• 1 - rt18139</li> <li>• 2 - e1000</li> <li>• 3 - pv</li> </ul>
vm_interface_speed_bps	integer	The average speed of the interface during the aggregation in bits per second.
mac_address	character varying(20)	As displayed in the edit dialog.
logical_network_name	character varying(50)	As displayed in the edit dialog.
vm_configuration_version	integer	The virtual machine configuration version at the time of creation or update.
create_date	timestamp with time zone	The date this entity was added to the system.
update_date	timestamp with time zone	The date this entity was changed in the system.
delete_date	timestamp with time zone	The date this entity was deleted from the system.

### 2.8.9. Virtual Machine Device Configuration

The following table shows the relationships between virtual machines and their associated devices, including disks and virtual interfaces.

**Table 2.16. v4\_0\_configuration\_history\_vms\_devices**

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
vm_id	uuid	The unique ID of the virtual machine in the system.
device_id	uuid	The unique ID of the device in the system.
type	character varying(30)	The type of virtual machine device. This can be "disk" or "interface".
address	character varying(255)	The device's physical address.
is_managed	boolean	Flag that indicates if the device is managed by the Manager.
is_plugged	boolean	Flag that indicates if the device is plugged into the virtual machine.
is_readonly	boolean	Flag that indicates if the device is read only.
vm_configuration_version	integer	The virtual machine configuration version at the time the sample was taken.
device_configuration_version	integer	The device configuration version at the time the sample was taken.
create_date	timestamp with time zone	The date this entity was added to the system.
update_date	timestamp with time zone	The date this entity was added to the system.
delete_date	timestamp with time zone	The date this entity was added to the system.

### 2.8.10. Virtual Disk Configuration

The following table shows the configuration history parameters of the virtual disks in the system.

**Table 2.17. v4\_0\_configuration\_history\_vms\_disks**

Name	Type	Description
history_id	integer	The ID of the configuration version in the history database.
vm_disk_id	uuid	The unique ID of this disk in the system.
vm_disk_name	text	The name of the virtual disk, as displayed in the edit dialog.
vm_disk_description	character varying(500)	As displayed in the edit dialog.
image_id	uuid	The unique ID of the image in the system.
storage_domain_id	uuid	The ID of the storage domain this disk image belongs to.
vm_disk_size_mb	integer	The defined size of the disk in megabytes (MB).
vm_disk_type	smallint	As displayed in the edit dialog. Only System and Data are currently used. <ul style="list-style-type: none"> <li>• 0 - Unassigned</li> <li>• 1 - System</li> <li>• 2 - Data</li> <li>• 3 - Shared</li> <li>• 4 - Swap</li> <li>• 5 - Temp</li> </ul>
vm_disk_format	smallint	As displayed in the edit dialog. <ul style="list-style-type: none"> <li>• 3 - Unassigned</li> <li>• 4 - COW</li> <li>• 5 - RAW</li> </ul>
is_shared	boolean	Flag that indicates if the virtual machine's disk is shared.
create_date	timestamp with time zone	The date this entity was added to the system.



Name	Type	Description
update_date	timestamp with time zone	The date this entity was changed in the system.
delete_date	timestamp with time zone	The date this entity was deleted from the system.

### 2.8.11. User Details History

The following table shows the configuration history parameters of the users in the system.

**Table 2.18. v4\_0\_users\_details\_history**

Name	Type	Description
user_id	uuid	The unique ID of the user in the system, as generated by the Manager.
first_name	character varying(255)	The user's first name.
last_name	character varying(255)	The user's last name.
domain	character varying(255)	The name of the authorization extension.
username	character varying(255)	The account name.
department	character varying(255)	The organizational department the user belongs to.
user_role_title	character varying(255)	The title or role of the user within the organization.
email	character varying(255)	The email of the user in the organization.
external_id	text	The unique identifier of the user from the external system.

Name	Type	Description
active	boolean	A flag to indicate if the user is active or not. This is checked hourly. If the user can be found in the authorization extension then it will remain active. A user becomes active on successful login.
create_date	timestamp with time zone	The date this entity was added to the system.
update_date	timestamp with time zone	The date this entity was changed in the system.
delete_date	timestamp with time zone	The date this entity was deleted from the system.