



Red Hat Virtualization 4.4

Data Warehouse Guide

How to use the Data Warehouse capabilities of Red Hat Virtualization

Red Hat Virtualization 4.4 Data Warehouse Guide

How to use the Data Warehouse capabilities of Red Hat Virtualization

Red Hat Virtualization Documentation Team

Red Hat Customer Content Services

rhev-docs@redhat.com

Legal Notice

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document provides information and procedures relevant to Red Hat Virtualization Data Warehouse.

Table of Contents

| | |
|--|-----------|
| PREFACE | 3 |
| MAKING OPEN SOURCE MORE INCLUSIVE | 4 |
| CHAPTER 1. INSTALLING AND CONFIGURING DATA WAREHOUSE | 5 |
| 1.1. OVERVIEW OF CONFIGURING DATA WAREHOUSE | 5 |
| 1.2. INSTALLING AND CONFIGURING DATA WAREHOUSE ON A SEPARATE MACHINE | 5 |
| 1.3. MIGRATING DATA WAREHOUSE TO A SEPARATE MACHINE | 10 |
| 1.4. CHANGING THE DATA WAREHOUSE SAMPLING SCALE | 17 |
| CHAPTER 2. ABOUT THE HISTORY DATABASE | 19 |
| 2.1. HISTORY DATABASE OVERVIEW | 19 |
| 2.2. TRACKING CONFIGURATION HISTORY | 19 |
| 2.3. RECORDING STATISTICAL HISTORY | 20 |
| 2.4. APPLICATION SETTINGS FOR THE DATA WAREHOUSE SERVICE IN OVIRT-ENGINE-DWHD.CONF | 20 |
| 2.5. TRACKING TAG HISTORY | 21 |
| 2.6. ALLOWING READ-ONLY ACCESS TO THE HISTORY DATABASE | 21 |
| 2.7. STATISTICS HISTORY VIEWS | 23 |
| 2.8. CONFIGURATION HISTORY VIEWS | 41 |

PREFACE

The Red Hat Virtualization Manager includes a data warehouse that collects monitoring data about hosts, virtual machines, and storage. Data Warehouse, which includes a database and a service, must be installed and configured along with the Manager setup, either on the same machine or on a separate server.

The Red Hat Virtualization installation creates two databases:

- The Manager database (**engine**) is the primary data store used by the Red Hat Virtualization Manager. Information about the virtualization environment like its state, configuration, and performance are stored in this database.
- The Data Warehouse database (**ovirt_engine_history**) contains configuration information and statistical data which is collated over time from the Manager database. The configuration data in the Manager database is examined every minute, and changes are replicated to the Data Warehouse database. Tracking the changes to the database provides information on the objects in the database. This enables you to analyze and enhance the performance of your Red Hat Virtualization environment and resolve difficulties.

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.

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. Because of the enormity of this endeavor, these changes are being updated gradually and where possible. For more details, see [our CTO Chris Wright's message](#).

CHAPTER 1. INSTALLING AND CONFIGURING DATA WAREHOUSE

1.1. OVERVIEW OF CONFIGURING DATA WAREHOUSE

You can install and configure Data Warehouse on the same machine as the Manager, or on a separate machine with access to the Manager:

Install and configure Data Warehouse on the Manager machine

This configuration requires only a single registered machine, and is the simplest to configure, but it increases the demand on the Manager machine. Users who require access to the Data Warehouse service require access to the Manager machine itself. See [Configuring the Red Hat Virtualization Manager](#) in *Installing Red Hat Virtualization as a standalone Manager with local databases* .

Install and configure Data Warehouse on 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

This section describes installing and configuring the Data Warehouse service on a separate machine from the Red Hat Virtualization Manager. Installing Data Warehouse on a separate machine helps to reduce the load on the Manager machine.

**NOTE**

Red Hat only supports installing the Data Warehouse database, the Data Warehouse service and Grafana all on the same machine as each other, even though you can install each of these components on separate machines from each other.

Prerequisites

- The Red Hat Virtualization Manager is installed on a separate machine.
- A physical server or virtual machine running Red Hat Enterprise Linux 8.
- The Manager database password.
- Access from the Data Warehouse machine to the Manager database machine's TCP port 5432.
- If you choose to install the Data Warehouse database separately from the Data Warehouse service, you must set up the database first. To install a remote Data Warehouse database manually, see [Preparing a Remote PostgreSQL Database](#). You must have the following information about the database machine:
 - The FQDN
 - 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 **postgresql.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.

Enabling the Red Hat Virtualization Manager Repositories

You need to log in and register the Data Warehouse machine with Red Hat Subscription Manager, attach the **Red Hat Virtualization Manager** subscription, and enable the Manager repositories.

Procedure

1. Register your system with the Content Delivery Network, entering your Customer Portal user name and password when prompted:

```
# subscription-manager register
```

**NOTE**

If you are using an IPv6 network, use an IPv6 transition mechanism to access the Content Delivery Network and subscription manager.

- Find the **Red Hat Virtualization Manager** subscription pool and record the pool ID:

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

- Use the pool ID to attach the subscription to the system:

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



NOTE

To view currently attached subscriptions:

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

To list all enabled repositories:

```
# dnf repolist
```

- Configure the repositories:

```
# subscription-manager repos \
  --disable='*' \
  --enable=rhel-8-for-x86_64-baseos-rpms \
  --enable=rhel-8-for-x86_64-appstream-rpms \
  --enable=rhv-4.4-manager-for-rhel-8-x86_64-rpms
```

- Enable the **pki-deps** module.

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

- Enable version 12 of the **postgresql** module.

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

- Synchronize installed packages to update them to the latest available versions.

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

Additional resources

For information on modules and module streams, see the following sections in *Installing, managing, and removing user-space components*

- [Module streams](#)
- [Selecting a stream before installation of packages](#)
- [Resetting module streams](#)
- [Switching to a later stream](#)

Installing Data Warehouse on a Separate Machine

Procedure

1. Log in to the machine where you want to install the database.
2. Ensure that all packages are up to date:

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

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

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

4. Run the **engine-setup** command to begin the installation:

```
# engine-setup
```

5. Answer **Yes** to install Data Warehouse on this machine:

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

6. Answer **Yes** to install Grafana on this machine:

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

7. 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]:
```

8. 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.

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

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

10. Press **Enter** to allow setup to sign the certificate on the Manager via SSH:

```
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]:
```

- Press **Enter** to accept the default SSH port, or enter an alternative port number and then press **Enter**:

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

- Enter the root password for the Manager machine:

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

- Specify whether to host the Data Warehouse database on this machine (Local), or on another machine (Remote):



NOTE

Red Hat only supports installing the Data Warehouse database, the Data Warehouse service and Grafana all on the same machine as each other, even though you can install each of these components on separate machines from each other.

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

- If you select **Local**, the **engine-setup** script can configure your database automatically (including adding a user and a database), or it can connect to a preconfigured local database:

```
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]:
```

- If you select **Automatic** by pressing **Enter**, no further action is required here.
- If you select **Manual**, input the following values for the manually-configured local database:

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

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

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

```
DWH database password:
```

- Enter the fully qualified domain name and password for the Manager database machine. If you are installing the Data Warehouse database on the same machine where the Manager database is installed, use the same FQDN. 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 [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. Use **Basic** when the Manager and Data Warehouse are installed on the same machine.

16. Confirm your installation settings:

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

17. After the Data Warehouse configuration is complete, on the Red Hat Virtualization Manager, restart the **ovirt-engine** service:

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

18. Optionally, set up SSL to secure database connections using the instructions at link: <https://www.postgresql.org/docs/12/ssl-tcp.html#SSL-FILE-USAGE>.

1.3. MIGRATING DATA WAREHOUSE TO A SEPARATE MACHINE

This section describes how to migrate the Data Warehouse database and service from the Red Hat Virtualization Manager machine to a separate machine. Hosting the Data Warehouse service on a separate machine reduces the load on each individual machine, and avoids potential conflicts caused by sharing CPU and memory resources with other processes.



NOTE

Red Hat only supports installing the Data Warehouse database, the Data Warehouse service and Grafana all on the same machine as each other, even though you can install each of these components on separate machines from each other.

You have the following migration options:

- You can migrate the Data Warehouse service away from the Manager machine and connect it with the existing Data Warehouse database (**ovirt_engine_history**).
- You can migrate the Data Warehouse database away from the Manager machine and then migrate the Data Warehouse service.

1.3.1. Migrating the Data Warehouse Database to a Separate Machine

Migrate the Data Warehouse database (**ovirt_engine_history**) 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**.

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



NOTE

Red Hat only supports installing the Data Warehouse database, the Data Warehouse service and Grafana all on the same machine as each other, even though you can install each of these components on separate machines from each other.

The new database server must have Red Hat Enterprise Linux 8 installed. Enable the required repositories on the new database server.

Enabling the Red Hat Virtualization Manager Repositories

You need to log in and register the Data Warehouse machine with Red Hat Subscription Manager, attach the **Red Hat Virtualization Manager** subscription, and enable the Manager repositories.

Procedure

1. Register your system with the Content Delivery Network, entering your Customer Portal user name and password when prompted:

```
# subscription-manager register
```



NOTE

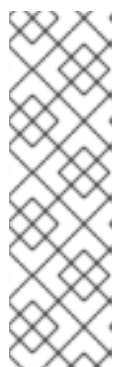
If you are using an IPv6 network, use an IPv6 transition mechanism to access the Content Delivery Network and subscription manager.

2. Find the **Red Hat Virtualization Manager** subscription pool and record the pool ID:

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

3. Use the pool ID to attach the subscription to the system:

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



NOTE

To view currently attached subscriptions:

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

To list all enabled repositories:

```
# dnf repolist
```

4. Configure the repositories:

```
# subscription-manager repos \
  --disable='*' \
  --enable=rhel-8-for-x86_64-baseos-rpms \
```

```
--enable=rhel-8-for-x86_64-appstream-rpms \  
--enable=rhv-4.4-manager-for-rhel-8-x86_64-rpms
```

5. Enable version 12 of the **postgresql** module.

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

6. Synchronize installed packages to update them to the latest available versions.

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

Additional resources

For information on modules and module streams, see the following sections in *Installing, managing, and removing user-space components*

- [Module streams](#)
- [Selecting a stream before installation of packages](#)
- [Resetting module streams](#)
- [Switching to a later stream](#)

Migrating the Data Warehouse Database to a Separate Machine

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

```
# 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:

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

4. Install the PostgreSQL server package:

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

5. Initialize the PostgreSQL database, start the **postgresql** service, and ensure that this service starts on boot:

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

6. 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 --restore-permissions
```

The Data Warehouse database is now hosted on a separate machine from that on which the Manager is hosted. After successfully restoring the Data Warehouse database, a prompt instructs you to run the **engine-setup** command. Before running this command, migrate the Data Warehouse service.

1.3.2. Migrating the Data Warehouse Service to a Separate Machine

You can migrate the Data Warehouse service installed and configured on the Red Hat Virtualization Manager to a separate machine. Hosting the Data Warehouse service on a separate machine helps to reduce the load on the Manager machine.

Notice that this procedure migrates the Data Warehouse service only.

To migrate the Data Warehouse database (**ovirt_engine_history**) prior to migrating the Data Warehouse service, see [Section 1.3.1, “Migrating the Data Warehouse Database to a Separate Machine”](#).



NOTE

Red Hat only supports installing the Data Warehouse database, the Data Warehouse service and Grafana all on the same machine as each other, even though you can install each of these components on separate machines from each other.

Prerequisites

- You must have installed and configured the Manager and Data Warehouse on the same machine.
- To set up the new Data Warehouse machine, you must have the following:
 - 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 username and password for the Data Warehouse database 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”](#), the backup includes these credentials, which you defined during the database setup on that machine.

Installing this scenario requires four steps:

1. [Setting up the New Data Warehouse Machine](#)
2. [Stopping the Data Warehouse service on the Manager machine](#)
3. [Configuring the new Data Warehouse machine](#)
4. [Disabling the Data Warehouse package on the Manager machine](#)

1.3.2.1. Setting up the New Data Warehouse Machine

Enable the Red Hat Virtualization repositories and install the Data Warehouse setup package on a Red Hat Enterprise Linux 8 machine:

1. Enable the required repositories:

- a. Register your system with the Content Delivery Network, entering your Customer Portal user name and password when prompted:

```
# subscription-manager register
```

- b. Find the **Red Hat Virtualization Manager** subscription pool and record the pool ID:

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

- c. Use the pool ID to attach the subscription to the system:

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

- d. Configure the repositories:

```
# subscription-manager repos \  
--disable='*' \  
--enable=rhel-8-for-x86_64-baseos-rpms \  
--enable=rhel-8-for-x86_64-appstream-rpms \  
--enable=rhv-4.4-manager-for-rhel-8-x86_64-rpms \  
--enable=fast-datapath-for-rhel-8-x86_64-rpms \  
--enable=jb-eap-7.3-for-rhel-8-x86_64-rpms
```

2. Ensure that all packages currently installed are up to date:

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

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

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

1.3.2.2. 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
```

1.3.2.3. Configuring the New Data Warehouse Machine

The order of the options or settings shown in this section 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 accept the automatically detected host name, or enter an alternative host name 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.

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

```
(1, 2) [1]:
ssh port on remote engine server [22]:
root password on remote engine server engine-fqdn: password
```

- Enter the FQDN and password for the Manager database machine. Press **Enter** to accept the default values in each other field:

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

- Confirm your installation settings:

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

The Data Warehouse service is now configured on the remote machine. Proceed to disable the Data Warehouse service on the Manager machine.



NOTE

If you want to change the Data Warehouse sampling scale to the recommended scale on a remote server, see [Section 1.4, “Changing the Data Warehouse Sampling Scale”](#).

1.3.2.4. Disabling the Data Warehouse Service on the Manager Machine

- On the Manager machine, restart the Manager:

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

- Run the following command to modify the file `/etc/ovirt-engine-setup.conf.d/20-setup-ovirt-post.conf` and set the options to **False**:

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

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

- 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 the Manager.

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. Use **Basic** 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.

Changing the Data Warehouse Sampling Scale

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

```
[...]
```

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

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

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

```
[...]
```

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

```
database ovirt_engine_history@localhost?
```

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

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

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

```
[...]
```

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

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

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

```
[...]
```

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

```
(1) Basic
```

```
(2) Full
```

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

[...]

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

[...]

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

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 12.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 |
|----------------------------|---------------|---|
| DWH_DELETE_JOB_HOUR | 3 | The time at which a deletion job is run. Specify a value between 0 and 23 , where 0 is midnight. |
| DWH_SAMPLING | 60 | The interval, in seconds, at which data is collected into statistical tables. |

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

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.

Allowing Read-Only Access to the History Database

1. Log in as root to the postgres user shell:

■

```
# su - postgres
```

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

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

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

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

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

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

5. 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 username;' FROM pg_class JOIN pg_namespace ON pg_namespace.oid = pg_class.relnamespace WHERE nspname = 'public' AND relkind IN ('r', 'v');" --pset=tuples_only=on ovirt_engine_history > grant.sql
```

6. 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
```

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

```
# rm grant.sql
```

8. Exit the postgres user shell by pressing **Ctrl+d**.

9. Add the following lines for the newly created user to `/var/lib/pgsql/data/pg_hba.conf` preceding the line beginning **local all all**:

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

10. Reload the PostgreSQL service:

```
# systemctl reload postgresql
```

11. To test the read-only user's access permissions:

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

```
Type "help" for help.
```

```
ovirt_engine_history=>
```

- 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_4_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.

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

```
DWH_AGGREGATION_DEBUG=true
```

- 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 | Indexed |
|------------|--------|---|---------|
| history_id | bigint | The unique ID of this row in the table. | No |

| Name | Type | Description | Indexed |
|-----------------------|----------|--|---------|
| history_datetime | date | The timestamp of this history row (rounded to minute, hour, day as per the aggregation level). | Yes |
| storage_domain_id | uuid | Unique ID of the storage domain in the system. | Yes |
| storage_domain_status | smallint | The storage domain status. | No |
| 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. | No |

| Name | Type | Description | Indexed |
|-------------------------------|--------------|---|---------|
| 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. | No |
| available_disk_size_gb | integer | The total available (unused) capacity on the disk, expressed in gigabytes (GB). | No |
| used_disk_size_gb | integer | The total used capacity on the disk, expressed in gigabytes (GB). | No |
| storage_configuration_version | integer | The storage domain configuration version at the time of sample. This is identical to the value of history_id in the v4_4_configuration_history_storage_domains view and it can be used to join them. | Yes |

2.7.3. Host Statistics Views

Table 2.3. Historical Statistics for Each Host in the System

| Name | Type | Description | Indexed |
|------------|--------|---|---------|
| history_id | bigint | The unique ID of this row in the table. | No |

| Name | Type | Description | Indexed |
|-------------------|----------|--|---------|
| history_datetime | date | The timestamp of this history row (rounded to minute, hour, day as per the aggregation level). | Yes |
| host_id | uuid | Unique ID of the host in the system. | Yes |
| host_status | smallint | <ul style="list-style-type: none"> ● -1 - Unknown Status (used only to indicate a problem with the ETL. Please notify Red Hat Support.) ● 1 - Up ● 2 - Maintenance ● 3 - Problematic | No |
| 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 Up and seconds_in_status of 55, the other will have a status of Down and a seconds_in_status of 5. | No |

| Name | Type | Description | Indexed |
|--------------------------|--------------|---|---------|
| 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 Up and minutes_in_status of 55, the other will have a status of Down and a minutes_in_status of 5. | No |
| memory_usage_percent | smallint | Percentage of used memory on the host. | No |
| 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. | No |
| ksm_shared_memory_mb | bigint | The Kernel Shared Memory size, in megabytes (MB), that the host is using. | No |
| 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. | No |
| cpu_usage_percent | smallint | Used CPU percentage on the host. | No |

| Name | Type | Description | Indexed |
|---------------------|----------|---|---------|
| 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. | No |
| ksm_cpu_percent | smallint | CPU percentage ksm on the host is using. | No |
| 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. | No |
| active_vms | smallint | The average number of active virtual machines for this aggregation. | No |
| 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. | No |
| total_vms | smallint | The average number of all virtual machines on the host for this aggregation. | No |

| Name | Type | Description | Indexed |
|--------------------------|----------|--|---------|
| 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. | No |
| total_vms_vcpus | integer | Total number of vCPUs allocated to the host. | No |
| 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. | No |
| cpu_load | integer | The CPU load of the host. | No |
| 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. | No |
| system_cpu_usage_percent | smallint | Used CPU percentage on the host. | No |

| Name | Type | Description | Indexed |
|------------------------------|----------|--|---------|
| 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. | No |
| user_cpu_usage_percent | smallint | Used user CPU percentage on the host. | No |
| 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. | No |
| swap_used_mb | integer | Used swap size usage of the host in megabytes (MB). | No |
| 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. | No |

| Name | Type | Description | Indexed |
|----------------------------|---------|--|---------|
| host_configuration_version | integer | The host configuration version at the time of sample. The host configuration version at the time of sample. This is identical to the value of history_id in the v4_4_configuration_history_hosts view and it can be used to join them. | Yes |

2.7.4. Host Interface Statistics Views

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

| Name | Type | Description | Indexed |
|--------------------------|----------|--|---------|
| history_id | bigint | The unique ID of this row in the table. | No |
| history_datetime | date | The timestamp of this history view (rounded to minute, hour, day as per the aggregation level). | Yes |
| host_interface_id | uuid | Unique identifier of the interface in the system. | Yes |
| receive_rate_percent | smallint | Used receive rate percentage on the host. | No |
| 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. | No |
| transmit_rate_percent | smallint | Used transmit rate percentage on the host. | No |

| Name | Type | Description | Indexed |
|--------------------------------------|----------|---|---------|
| 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. | No |
| received_total_byte | bigint | The total number of bytes received by the host. | No |
| transmitted_total_byte | bigint | The total number of bytes transmitted from the host. | No |
| host_interface_configuration_version | integer | The host interface configuration version at the time of sample. This is identical to the value of history_id in the v4_4_configuration_history_hosts_interfaces view and it can be used to join them. | Yes |

2.7.5. Virtual Machine Statistics Views

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

| Name | Type | Description | Indexed |
|------------------|--------|--|---------|
| history_id | bigint | The unique ID of this row in the table. | No |
| history_datetime | date | The timestamp of this history row (rounded to minute, hour, day as per the aggregation level). | Yes |
| vm_id | uuid | Unique ID of the virtual machine in the system. | Yes |

| Name | Type | Description | Indexed |
|-------------------|--------------|---|---------|
| vm_status | smallint | <ul style="list-style-type: none"> ● -1 - Unknown Status (used only to indicate problems with the ETL. Please notify Red Hat Support.) ● 0 - Down ● 1 - Up ● 2 - Paused ● 3 - Problematic | No |
| 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> | No |
| 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> | No |

| Name | Type | Description | Indexed |
|------------------------|----------|--|---------|
| cpu_usage_percent | smallint | The percentage of the CPU in use by the virtual machine. | No |
| 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. | No |
| 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. | No |
| 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. | No |
| user_cpu_usage_percent | smallint | Used user CPU percentage on the host. | No |

| Name | Type | Description | Indexed |
|------------------------------|----------|--|---------|
| 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. | No |
| system_cpu_usage_percent | smallint | Used system CPU percentage on the host. | No |
| 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. | No |
| vm_ip | text | The IP address of the first NIC. Only shown if the guest agent is installed. | No |
| currently_running_on_host | uuid | The unique ID of the host the virtual machine is running on. | No |
| current_user_id | uuid | The unique ID of the user logged into the virtual machine console, if the guest agent is installed. | No |
| disks_usage | text | The disk description. File systems type, mount point, total size, and used size. | No |

| Name | Type | Description | Indexed |
|------------------------------------|---------|---|---------|
| vm_configuration_version | integer | The virtual machine configuration version at the time of sample. This is identical to the value of history_id in the v4_4_configuration_history_vms view. | Yes |
| current_host_configuration_version | integer | The host configuration version at the time of sample. This is identical to the value of history_id in the v4_4_configuration_history_hosts view and it can be used to join them. | Yes |
| memory_buffered_kb | bigint | The amount of buffered memory on the virtual machine, in kilobytes (KB). | No |
| memory_cached_kb | bigint | The amount of cached memory on the virtual machine, in kilobytes (KB). | No |
| 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. | No |

| Name | Type | Description | Indexed |
|----------------------|--------|---|---------|
| 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. | No |

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 | Indexed |
|--------------------------|----------|--|---------|
| history_id | integer | The unique ID of this row in the table. | No |
| history_datetime | date | The timestamp of this history row (rounded to minute, hour, day as per the aggregation level). | Yes |
| vm_interface_id | uuid | Unique ID of the interface in the system. | Yes |
| receive_rate_percent | smallint | Used receive rate percentage on the host. | No |
| 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. | No |
| transmit_rate_percent | smallint | Used transmit rate percentage on the host. | No |

| Name | Type | Description | Indexed |
|------------------------------------|----------|--|---------|
| 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. | No |
| received_total_byte | bigint | The total number of bytes received by the virtual machine. | No |
| transmitted_total_byte | bigint | The total number of bytes transmitted from the virtual machine. | No |
| vm_interface_configuration_version | integer | The virtual machine interface configuration version at the time of sample. This is identical to the value of history_id in the v4_4_configuration_history_vms_interfaces view and it can be used to join them. | Yes |

2.7.7. Virtual Disk Statistics Views

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

| Name | Type | Description | Indexed |
|------------------|--------|--|---------|
| history_id | bigint | The unique ID of this row in the table. | No |
| history_datetime | date | The timestamp of this history row (rounded to minute, hour, day as per the aggregation level). | Yes |
| vm_disk_id | uuid | Unique ID of the disk in the system. | Yes |

| Name | Type | Description | Indexed |
|------------------------|--------------|---|---------|
| vm_disk_status | smallint | <ul style="list-style-type: none"> ● 0 - Unassigned ● 1 - OK ● 2 - Locked ● 3 - Invalid ● 4 - Illegal | No |
| 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 Locked and seconds_in_status of 55, the other will have a status of OK and a seconds_in_status of 5. | No |
| 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 Locked and minutes_in_status of 55, the other will have a status of OK and a minutes_in_status of 5. | No |
| vm_disk_actual_size_mb | integer | The actual size allocated to the disk. | No |

| Name | Type | Description | Indexed |
|---------------------------------|---------------|--|---------|
| read_rate_bytes_per_second | integer | Read rate to disk in bytes per second. | No |
| 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. | No |
| read_latency_seconds | numeric(18,9) | The virtual disk read latency measured in seconds. | No |
| write_rate_bytes_per_second | integer | Write rate to disk in bytes per second. | No |
| 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. | No |
| 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. | No |
| write_latency_seconds | numeric(18,9) | The virtual disk write latency measured in seconds. | No |

| Name | Type | Description | Indexed |
|-------------------------------|---------------|---|---------|
| 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. | No |
| flush_latency_seconds | numeric(18,9) | The virtual disk flush latency measured in seconds. | No |
| 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. | No |
| vm_disk_configuration_version | integer | The virtual disk configuration version at the time of sample. This is identical to the value of history_id in the v4_4_configuration_history_vms_disks view and it can be used to join them. | Yes |

2.8. CONFIGURATION HISTORY VIEWS

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

```
# SELECT * FROM v4_4_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_4_configuration_history_datacenters

| Name | Type | Description | Indexed |
|------------------------|--------------------------|--|---------|
| history_id | integer | The ID of the configuration version in the history database. This is identical to the value of datacenter_configuration_version in the v4_4_configuration_history_clusters view and it can be used to join them. | No |
| datacenter_id | uuid | The unique ID of the data center in the system. | Yes |
| datacenter_name | character varying(40) | Name of the data center, as displayed in the edit dialog. | No |
| datacenter_description | character varying(4000) | Description of the data center, as displayed in the edit dialog. | No |
| is_local_storage | boolean | A flag to indicate whether the data center uses local storage. | No |
| create_date | timestamp with time zone | The date this entity was added to the system. | No |
| update_date | timestamp with time zone | The date this entity was changed in the system. | No |
| delete_date | timestamp with time zone | The date this entity was deleted from the system. | No |

2.8.2. Data Center Storage Domain Map

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

Table 2.9. v4_4_map_history_datacenters_storage_domains

| Name | Type | Description | Indexed |
|-------------------|--------------------------|--|---------|
| history_id | integer | The ID of the configuration version in the history database. This is identical to the value of datacenter_configuration_version in the v4_4_configuration_history_clusters view and it can be used to join them. | No |
| storage_domain_id | uuid | The unique ID of this storage domain in the system. | Yes |
| datacenter_id | uuid | The unique ID of the data center in the system. | No |
| attach_date | timestamp with time zone | The date the storage domain was attached to the data center. | No |
| detach_date | timestamp with time zone | The date the storage domain was detached from the data center. | No |

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_4_configuration_history_storage_domains

| Name | Type | Description | Indexed |
|------|------|-------------|---------|
|------|------|-------------|---------|

| Name | Type | Description | Indexed |
|---------------------|--------------------------|---|---------|
| history_id | integer | The ID of the configuration version in the history database. This is identical to the value of storage_configuration_version in the storage domain statistics views and it can be used to join them. | No |
| storage_domain_id | uuid | The unique ID of this storage domain in the system. | Yes |
| storage_domain_name | character varying(250) | Storage domain name. | No |
| storage_domain_type | smallint | * 0 - Data (Master) * 1 - Data * 2 - ISO * 3 - Export | No |
| storage_type | smallint | * 0 - Unknown * 1 - NFS * 2 - FCP * 3 - iSCSI * 4 - Local * 6 - All | No |
| create_date | timestamp with time zone | The date this entity was added to the system. | No |
| update_date | timestamp with time zone | The date this entity was changed in the system. | No |
| delete_date | timestamp with time zone | The date this entity was deleted from the system. | No |

2.8.4. Cluster Configuration

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

Table 2.11. v4_4_configuration_history_clusters

| Name | Type | Description | Indexed |
|----------------------------------|-------------------------|--|---------|
| history_id | integer | The ID of the configuration version in the history database. This is identical to the value of cluster_configuration_version in the v4_4_configuration_history_hosts and v4_4_configuration_history_vms views and it can be used to join them. | No |
| cluster_id | uuid | The unique identifier of the datacenter this cluster resides in. | Yes |
| cluster_name | character varying(40) | Name of the cluster, as displayed in the edit dialog. | No |
| cluster_description | character varying(4000) | As defined in the edit dialog. | No |
| datacenter_id | uuid | The unique identifier of the datacenter this cluster resides in. | Yes |
| cpu_name | character varying(255) | As displayed in the edit dialog. | No |
| compatibility_version | character varying(40) | As displayed in the edit dialog. | No |
| datacenter_configuration_version | integer | The data center configuration version at the time of creation or update. The data center configuration version at the time of creation or update. This is identical to the value of history_id in the v4_4_configuration_history_datacenters view and it can be used to join them. | No |

| Name | Type | Description | Indexed |
|-------------|--------------------------|---|---------|
| create_date | timestamp with time zone | The date this entity was added to the system. | No |
| update_date | timestamp with time zone | The date this entity was changed in the system. | No |
| delete_date | timestamp with time zone | The date this entity was deleted from the system. | No |

2.8.5. Host Configuration

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

Table 2.12. v4_4_configuration_history_hosts

| Name | Type | Description | Indexed |
|----------------|------------------------|--|---------|
| history_id | integer | The ID of the configuration version in the history database. This is identical to the value of host_configuration_version in the host statistics views and it can be used to join them. | No |
| host_id | uuid | The unique ID of the host in the system. | Yes |
| 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. | No |
| host_name | character varying(255) | Name of the host (same as in the edit dialog). | No |
| cluster_id | uuid | The unique ID of the cluster that this host belongs to. | Yes |

| Name | Type | Description | Indexed |
|-------------------|------------------------|---|---------|
| host_type | smallint | * 0 - RHEL Host * 2 - RHEV Hypervisor Node | No |
| 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). | No |
| memory_size_mb | integer | The host's physical memory capacity, expressed in megabytes (MB). | No |
| swap_size_mb | integer | The host swap partition size. | No |
| cpu_model | character varying(255) | The host's CPU model. | No |
| number_of_cores | smallint | Total number of CPU cores in the host. | No |
| number_of_sockets | smallint | Total number of CPU sockets. | No |
| cpu_speed_mh | numeric(18,0) | The host's CPU speed, expressed in megahertz (MHz). | No |
| host_os | character varying(255) | The host's operating system version. | No |
| kernel_version | character varying(255) | The host's kernel version. | No |
| kvm_version | character varying(255) | The host's KVM version. | No |
| vdsmd_version | character varying | The host's VDSM version. | No |
| vdsmd_port | integer | As displayed in the edit dialog. | No |

| Name | Type | Description | Indexed |
|-------------------------------|--------------------------|--|---------|
| threads_per_core | smallint | Total number of threads per core. | No |
| hardware_manufacturer | character varying(255) | The host's hardware manufacturer. | No |
| hardware_product_name | character varying(255) | The product name of the host's hardware. | No |
| hardware_version | character varying(255) | The version of the host's hardware. | No |
| hardware_serial_number | character varying(255) | The serial number of the host's hardware. | No |
| cluster_configuration_version | integer | The cluster configuration version at the time of creation or update. This is identical to the value of history_id in the v4_4_configuration_history_clusters view and it can be used to join them. | No |
| create_date | timestamp with time zone | The date this entity was added to the system. | No |
| update_date | timestamp with time zone | The date this entity was changed in the system. | No |
| delete_date | timestamp with time zone | The date this entity was deleted from the system. | No |

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_4_configuration_history_hosts_interfaces

| Name | Type | Description | Indexed |
|------|------|-------------|---------|
|------|------|-------------|---------|

| Name | Type | Description | Indexed |
|--------------------------|-----------------------|--|---------|
| history_id | integer | The ID of the configuration version in the history database. This is identical to the value of host_interface_configuration_version in the host interface statistics views and it can be used to join them. | No |
| host_interface_id | uuid | The unique ID of this interface in the system. | Yes |
| host_interface_name | character varying(50) | The interface name as reported by the host. | No |
| host_id | uuid | Unique ID of the host this interface belongs to. | Yes |
| host_interface_type | smallint | * 0 - rt18139_pv * 1 - rt18139 * 2 - e1000 * 3 - pv | No |
| host_interface_speed_bps | integer | The interface speed in bits per second. | No |
| mac_address | character varying(59) | The interface MAC address. | No |
| logical_network_name | character varying(50) | The logical network associated with the interface. | No |
| ip_address | character varying(20) | As displayed in the edit dialog. | No |
| gateway | character varying(20) | As displayed in the edit dialog. | No |
| bond | boolean | A flag to indicate if this interface is a bonded interface. | No |

| Name | Type | Description | Indexed |
|----------------------------|--------------------------|--|---------|
| bond_name | character varying(50) | The name of the bond this interface is part of (if it is part of a bond). | No |
| vlan_id | integer | As displayed in the edit dialog. | No |
| host_configuration_version | integer | The host configuration version at the time of creation or update. This is identical to the value of history_id in the v4_4_configuration_history_hosts view and it can be used to join them. | No |
| create_date | timestamp with time zone | The date this entity was added to the system. | No |
| update_date | timestamp with time zone | The date this entity was changed in the system. | No |
| delete_date | timestamp with time zone | The date this entity was deleted from the system. | No |

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_4_configuration_history_vms

| Name | Type | Description | Indexed |
|------------|---------|---|---------|
| history_id | integer | The ID of the configuration version in the history database. This is identical to the value of vm_configuration_version in the virtual machine statistics views and it can be used to join them. | No |

| Name | Type | Description | Indexed |
|-------------------|-------------------------|---|---------|
| vm_id | uuid | The unique ID of this virtual machine in the system. | Yes |
| vm_name | character varying(255) | The name of the virtual machine. | No |
| vm_description | character varying(4000) | As displayed in the edit dialog. | No |
| vm_type | smallint | * 0 - Desktop * 1 - Server | No |
| cluster_id | uuid | The unique ID of the cluster this virtual machine belongs to. | Yes |
| template_id | uuid | The unique ID of the template this virtual machine is derived from. Templates are not synchronized to the history database in this version of Red Hat Virtualization. | No |
| template_name | character varying(40) | Name of the template from which this virtual machine is derived. | No |
| cpu_per_socket | smallint | Virtual CPUs per socket. | No |
| number_of_sockets | smallint | Total number of virtual CPU sockets. | No |
| memory_size_mb | integer | Total memory allocated to the virtual machine, expressed in megabytes (MB). | No |
| operating_system | smallint | * 0 - Other OS * 1 - Windows XP * 3 - Windows 2003 * 4 - Windows 2008 | No |

| Name | Type | * 5 - Linux Description | Indexed |
|------|------|--|---------|
| | | <p>* 7 - Red Hat Enterprise Linux 5.x</p> <p>* 8 - Red Hat Enterprise Linux 4.x</p> <p>* 9 - Red Hat Enterprise Linux 3.x</p> <p>* 10 - Windows 2003 x64</p> <p>* 11 - Windows 7</p> <p>* 12 - Windows 7 x64</p> <p>* 13 - Red Hat Enterprise Linux 5.x x64</p> <p>* 14 - Red Hat Enterprise Linux 4.x x64</p> <p>* 15 - Red Hat Enterprise Linux 3.x x64</p> <p>* 16 - Windows 2008 x64</p> <p>* 17 - Windows 2008 R2 x64</p> <p>* 18 - Red Hat Enterprise Linux 6.x</p> <p>* 19 - Red Hat Enterprise Linux 6.x x64</p> <p>* 20 - Windows 8</p> <p>* 21 - Windows 8 x64</p> <p>* 23 - Windows 2012 x64</p> <p>* 1001 - Other</p> <p>* 1002 - Linux</p> <p>* 1003 - Red Hat Enterprise Linux 6.x</p> <p>* 1004 - SUSE Linux Enterprise Server 11</p> <p>* 1193 - SUSE Linux Enterprise Server 11</p> | |

| Name | Type | Description | Indexed |
|--------------------|------------------------|---|---------|
| | | * 1252 - Ubuntu Precise Pangolin LTS | |
| | | * 1253 - Ubuntu Quantal Quetzal | |
| | | * 1254 - Ubuntu Raring Ringtails | |
| | | * 1255 - Ubuntu Saucy Salamander | |
| default_host | uuid | As displayed in the edit dialog, the ID of the default host in the system. | No |
| high_availability | boolean | As displayed in the edit dialog. | No |
| initialized | boolean | A flag to indicate if this virtual machine was started at least once for Sysprep initialization purposes. | No |
| stateless | boolean | As displayed in the edit dialog. | No |
| fail_back | boolean | As displayed in the edit dialog. | No |
| usb_policy | smallint | As displayed in the edit dialog. | No |
| time_zone | character varying(40) | As displayed in the edit dialog. | No |
| vm_pool_id | uuid | The ID of the pool to which this virtual machine belongs. | No |
| vm_pool_name | character varying(255) | The name of the virtual machine's pool. | No |
| created_by_user_id | uuid | The ID of the user that created this virtual machine. | No |

| Name | Type | Description | Indexed |
|------------------------------------|--------------------------|--|---------|
| cluster_configuration_version | integer | The cluster configuration version at the time of creation or update. This is identical to the value of history_id in the v4_4_configuration_history_clusters view and it can be used to join them. | No |
| default_host_configuration_version | integer | The host configuration version at the time of creation or update. This is identical to the value of history_id in the v4_4_configuration_history_hosts view and it can be used to join them. | No |
| create_date | timestamp with time zone | The date this entity was added to the system. | No |
| update_date | timestamp with time zone | The date this entity was changed in the system. | No |
| delete_date | timestamp with time zone | The date this entity was deleted from the system. | No |

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_4_configuration_history_vms_interfaces

| Name | Type | Description | Indexed |
|------------|---------|--|---------|
| history_id | integer | The ID of the configuration version in the history database. This is identical to the value of vm_interface_configuration_version in the virtual machine interface statistics view and it can be used to join them. | No |

| Name | Type | Description | Indexed |
|--------------------------|--------------------------|---|---------|
| vm_id | uuid | Unique ID of the virtual machine in the system. | Yes |
| vm_interface_id | uuid | The unique ID of this interface in the system. | Yes |
| vm_interface_name | character varying(50) | As displayed in the edit dialog. | No |
| vm_interface_type | smallint | The type of the virtual interface. * 0 - rt18139_pv * 1 - rt18139 * 2 - e1000 * 3 - pv | No |
| vm_interface_speed_bps | integer | The average speed of the interface during the aggregation in bits per second. | No |
| mac_address | character varying(20) | As displayed in the edit dialog. | No |
| logical_network_name | character varying(50) | As displayed in the edit dialog. | No |
| vm_configuration_version | integer | The virtual machine configuration version at the time of creation or update. This is identical to the value of history_id in the v4_4_configuration_history_vms view and it can be used to join them. | No |
| create_date | timestamp with time zone | The date this entity was added to the system. | No |
| update_date | timestamp with time zone | The date this entity was changed in the system. | No |

| Name | Type | Description | Indexed |
|-------------|--------------------------|---|---------|
| delete_date | timestamp with time zone | The date this entity was deleted from the system. | No |

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_4_configuration_history_vms_devices

| Name | Type | Description | Indexed |
|--------------------------|------------------------|---|---------|
| history_id | integer | The ID of the configuration version in the history database. | No |
| vm_id | uuid | The unique ID of the virtual machine in the system. | Yes |
| device_id | uuid | The unique ID of the device in the system. | No |
| type | character varying(30) | The type of virtual machine device. This can be "disk" or "interface". | Yes |
| address | character varying(255) | The device's physical address. | No |
| is_managed | boolean | Flag that indicates if the device is managed by the Manager. | No |
| is_plugged | boolean | Flag that indicates if the device is plugged into the virtual machine. | No |
| is_readonly | boolean | Flag that indicates if the device is read only. | No |
| vm_configuration_version | integer | The virtual machine configuration version at the time the sample was taken. | No |

| Name | Type | Description | Indexed |
|------------------------------|--------------------------|--|---------|
| device_configuration_version | integer | The device configuration version at the time the sample was taken. - If the value of the type field is set to interface , this field is joined with the history_id field in the v4_4_configuration_history_vms_interfaces view. - If the value of the type field is set to disk, this field is joined with the history_id field in the v4_4_configuration_history_vms_disks view. | No |
| create_date | timestamp with time zone | The date this entity was added to the system. | No |
| update_date | timestamp with time zone | The date this entity was added to the system. | No |
| delete_date | timestamp with time zone | The date this entity was added to the system. | No |

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_4_configuration_history_vms_disks

| Name | Type | Description | Indexed |
|------------|---------|--|---------|
| history_id | integer | The ID of the configuration version in the history database. This is identical to the value of vm_disk_configuration_version in the virtual disks statistics views and it can be used to join them. | No |

| Name | Type | Description | Indexed |
|---------------------|------------------------|---|---------|
| vm_disk_id | uuid | The unique ID of this disk in the system. | Yes |
| vm_disk_name | text | The name of the virtual disk, as displayed in the edit dialog. | No |
| vm_disk_description | character varying(500) | As displayed in the edit dialog. | No |
| image_id | uuid | The unique ID of the image in the system. | No |
| storage_domain_id | uuid | The ID of the storage domain this disk image belongs to. | Yes |
| vm_disk_size_mb | integer | The defined size of the disk in megabytes (MB). | No |
| vm_disk_type | smallint | As displayed in the edit dialog. Only System and Data are currently used. * 0 - Unassigned * 1 - System * 2 - Data * 3 - Shared * 4 - Swap * 5 - Temp | No |
| vm_disk_format | smallint | As displayed in the edit dialog. * 3 - Unassigned * 4 - COW * 5 - Raw | No |
| is_shared | boolean | Flag that indicates if the virtual machine's disk is shared. | No |

| Name | Type | Description | Indexed |
|-------------|--------------------------|---|---------|
| create_date | timestamp with time zone | The date this entity was added to the system. | No |
| update_date | timestamp with time zone | The date this entity was changed in the system. | No |
| delete_date | timestamp with time zone | The date this entity was deleted from the system. | No |

2.8.11. User Details History

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

Table 2.18. v4_4_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. |