



# Red Hat Virtualization 4.1

## 데이터 웨어하우스 가이드

---

Red Hat Virtualization의 데이터 웨어하우스 기능을 사용하는 방법

Red Hat Virtualization Documentation Team



## Red Hat Virtualization의 데이터 웨어하우스 기능을 사용하는 방법

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

## 법적 공지

Copyright © 2016 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.

## 초록

본 문서에서는 Red Hat Virtualization 데이터 웨어하우스와 관련된 절차 및 정보에 대해 설명합니다.

---

## 차례

<b>1장. 데이터 웨어하우스 설치 및 설정</b> .....	<b>2</b>
1.1. 데이터 웨어하우스 설정 개요	2
1.2. 데이터 웨어하우스를 별도의 머신에 설치 및 설정	3
1.3. 데이터 웨어하우스를 별도의 머신으로 마이그레이션	5
1.4. 데이터 웨어하우스 샘플링 규모 변경	10
<b>2장. 기록 데이터베이스 개요</b> .....	<b>12</b>
2.1. 기록 데이터베이스 개요	12
2.2. 설정 기록 추적	12
2.3. 통계 기록	13
2.4. ovirt-engine-dwhd.conf의 데이터 웨어하우스 서비스 애플리케이션 설정	13
2.5. 태그 기록 추적	13
2.6. 기록 데이터베이스에 대한 읽기 전용 액세스 허용	14
2.7. 통계 기록 보기	15
2.8. 설정 기록 보기	23

# 1장. 데이터 웨어하우스 설치 및 설정

## 1.1. 데이터 웨어하우스 설정 개요

Red Hat Virtualization Manager에는 데이터 센터, 클러스터, 호스트 수준에서 다양한 정보를 추출하는데 활용할 수 있는 포괄적인 관리 기록 데이터베이스가 포함되어 있습니다. 데이터 웨어하우스를 설치하면 **ovirt\_engine\_history** 데이터베이스가 생성되며 Manager는 보고 목적으로 해당 데이터베이스에 정보를 기록하도록 설정됩니다.

Red Hat Virtualization에는 데이터 웨어하우스가 필요합니다. 데이터 웨어하우스는 Manager와 같은 머신이나 Manager에 액세스할 수 있는 별도의 머신에 설치 및 설정할 수 있습니다.

### 1. Manager 머신에 데이터 웨어하우스를 설치 및 설정합니다.

이 설정에는 등록된 머신이 하나만 필요하며 설정하기 가장 간단하지만, 호스트 머신에 대한 수요가 늘어납니다. 데이터 웨어하우스 서비스에 대한 액세스 권한이 필요한 사용자는 Manager 머신 자체에 대한 액세스 권한도 필요합니다. 해당 설정에 대한 자세한 내용은 [설치 가이드](#)에 있는 [Red Hat Virtualization Manager 설정](#)을 참조하십시오.

### 2. 데이터 웨어하우스를 별도의 머신에 설치 및 설정합니다.

이 설정에는 등록된 두 개의 머신이 필요합니다. 이 설정을 사용하면 Manager 머신의 로드를 줄이고 머신에서 발생할 수 있는 CPU 및 메모리 공유 충돌을 방지할 수 있습니다. 또한 관리자는 Manager 머신에 대한 액세스 권한을 부여하지 않고도 사용자가 데이터 웨어하우스 머신에 액세스하도록 허용할 수 있습니다. 이 설정에 대한 자세한 내용은 [1.2절. “데이터 웨어하우스를 별도의 머신에 설치 및 설정”](#)을 참조하십시오.

데이터 웨어하우스 배포의 모든 머신에 대한 시스템 시간대를 UTC로 설정하는 것이 좋습니다. 그러면 현지 시간대의 변동(예: 여름철 시간대에서 겨울철 시간대로 변경)에 따라 데이터 수집이 중단되지 않습니다.

**ovirt\_engine\_history** 데이터베이스에서 사용할 공간 및 리소스 추정치를 계산하려면 [RHV Manager 기록 데이터베이스 크기 계산기](#) 도구를 사용합니다. 추정치는 엔티티 수와 기록 레코드를 유지하도록 선택한 기간을 기반으로 계산됩니다.



#### 중요

**engine-setup**을 실행하면 다음 동작이 예상됩니다.

- ※ 데이터 웨어하우스 패키지를 설치하고 **engine-setup**을 실행한 다음 데이터 웨어하우스 설정에 **No**로 응답합니다.

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

- ※ **engine-setup**을 다시 실행합니다. 설치 시 데이터 웨어하우스를 설정하는 옵션이 더 이상 표시되지 않습니다.

**engine-setup**에서 해당 옵션을 다시 표시하도록 하려면 **engine-setup --reconfigure-optional-components**를 실행합니다.

현재 설치된 데이터 웨어하우스 패키지만 설정하고 설치 시 활성화된 리포지터리에 있는 패키지 업데이트를 적용하지 않도록 하려면 **--offline** 옵션을 추가합니다.

## 1.2. 데이터 웨어하우스를 별도의 머신에 설치 및 설정

Red Hat Virtualization Manager가 설치된 머신과 다른 별도의 머신에 데이터 웨어하우스를 설치 및 설정합니다. 데이터 웨어하우스 서비스를 별도의 머신에서 호스트하면 **Manager** 머신의 로드를 줄일 수 있습니다.

### 전제 조건

- ✱ **Manager**가 별도의 머신에 설치 및 설정되어 있어야 합니다.
- ✱ 데이터 웨어하우스 머신을 설치하려면 다음이 있어야 합니다.
  - Red Hat Enterprise Linux 7이 설치된 가상 머신 또는 물리적 머신
  - **Red Hat Enterprise Linux Server** 및 **Red Hat Virtualization** 서브스크립션 풀에 대한 서브스크립션
  - **Manager**의 `/etc/ovirt-engine/engine.conf.d/10-setup-database.conf` 파일에 있는 암호
  - 데이터 웨어하우스 머신에서 **Manager** 데이터베이스 머신의 TCP 포트 5432에 대해 허용된 액세스
- ✱ 원격 데이터 웨어하우스 데이터베이스를 사용하도록 선택하면 데이터 웨어하우스 서비스를 설치하기 전에 데이터베이스를 설정해야 합니다. 데이터베이스 호스트에 대해 다음 정보가 있어야 합니다.
  - 호스트의 정규화된 도메인 이름
  - 데이터베이스 연결에 사용할 수 있는 포트(기본값: 5432)
  - 데이터베이스 이름
  - 데이터베이스 사용자
  - 데이터베이스 암호
  - **postgres.conf** 파일을 편집하여 액세스 권한을 수동으로 부여해야 합니다. `/var/lib/pgsql/data/postgresql.conf` 파일을 편집하고 **listen\_addresses** 행을 다음과 일치하도록 수정합니다.

```
listen_addresses = '*'
```

해당 행이 존재하지 않거나 주석 처리된 경우 수동으로 추가합니다.

데이터베이스가 **Manager** 머신에서 호스트되고 Red Hat Virtualization Manager를 새로 설치하는 동안 설정한 경우에는 액세스 권한이 기본적으로 부여됩니다.

### 절차 1.1. 데이터 웨어하우스를 별도의 머신에 설치 및 설정

1. 시스템에 필요한 서브스크립션을 등록합니다. 자세한 내용은 [설치 가이드](#)에 있는 [필요한 서브스크립션을 등록](#)을 참조하십시오.
2. 현재 설치된 모든 패키지가 최신 상태인지 확인합니다.

```
# yum update
```

3. `ovirt-engine-dwh-setup` 패키지를 설치합니다.

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

4. **engine-setup** 명령을 실행하고 프롬프트에 따라 머신에서 데이터 웨어하우스를 설정합니다.

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

5. **Enter** 키를 눌러 자동으로 감지된 호스트 이름을 수락하거나 대체 호스트 이름을 입력하고 **Enter** 키를 누릅니다.

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

6. **Enter** 키를 눌러 방화벽을 자동으로 설정하거나 **No**를 입력하고 **Enter** 키를 눌러 기존 설정을 유지합니다.

```
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]:
```

사용자가 방화벽을 자동으로 설정하도록 선택하고 현재 활성화 상태인 방화벽 관리자가 없으면 지원되는 옵션 목록에서 방화벽 관리자를 선택하라는 메시지가 표시됩니다. 방화벽 관리자 이름을 입력하고 **Enter** 키를 누릅니다. 목록에 옵션이 하나 밖에 없는 경우에도 이 방법이 적용됩니다.

7. Manager 머신의 정규화된 도메인 이름을 입력하고 **Enter** 키를 누릅니다.

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

8. **Enter** 키를 눌러 설치 시 SSH를 통해 Manager에서 인증서에 서명하도록 허용합니다.

```
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. **Enter** 키를 눌러 기본 SSH 포트를 수락하거나 대체 포트 번호를 입력한 다음 **Enter** 키를 누릅니다.

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

10. Manager 머신의 root 암호를 입력합니다.

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

11. 데이터 웨어하우스 데이터베이스에 대한 다음 질문에 답합니다.

```
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]:
```



**Enter** 키를 눌러 강조 표시된 기본값을 선택하거나 대체 설정을 입력한 다음 **Enter** 키를 누릅니다. **Remote**를 선택하면 원격 데이터베이스 호스트에 대한 상세 정보를 제공하라는 메시지가 표시됩니다. 사전 설정된 원격 데이터베이스 호스트에 대해 다음 값을 입력합니다.

```
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
```

데이터 웨어하우스 데이터베이스를 설정 및 마이그레이션하는 방법에 대한 자세한 내용은 [1.3.1절. “데이터 웨어하우스 데이터베이스를 별도의 머신으로 마이그레이션”](#)을 참조하십시오.

12. **Manager** 데이터베이스 머신의 정규화된 도메인 이름 및 암호를 입력합니다. **Enter** 키를 눌러 각 필드의 기본값을 수락합니다.

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

13. 데이터 웨어하우스에서 수집된 데이터를 유지할 기간을 선택합니다.

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

**Full**을 선택하면 [2.4절. “ovirt-engine-dwhd.conf의 데이터 웨어하우스 서비스 애플리케이션 설정”](#)에 있는 데이터 스토리지 설정 기본값이 사용됩니다 (데이터 웨어하우스가 원격 호스트에 설치된 경우 권장됨).

**Basic**을 선택하면 **DWH\_TABLES\_KEEP\_HOURLY** 값이 **720**으로, **DWH\_TABLES\_KEEP\_DAILY** 값이 **0**으로 감소하여 **Manager** 머신의 로드가 완화된 (Manager와 데이터 웨어하우스를 동일한 머신에 설치한 경우 권장됨).

14. 설치 설정을 확인합니다.

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

15. Red Hat Virtualization Manager에서 **ovirt-engine** 서비스를 다시 시작합니다.

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

## 1.3. 데이터 웨어하우스를 별도의 머신으로 마이그레이션

데이터 웨어하우스 서비스를 Red Hat Virtualization Manager에서 별도의 머신으로 마이그레이션합니다. 데이터 웨어하우스 서비스를 별도의 머신에서 호스트하면 각 개별 머신의 로드가 줄고 각 서비스에서 CPU 및 메모리를 다른 프로세스와 공유함으로써 발생할 수 있는 충돌을 방지할 수 있습니다.

데이터 웨어하우스 서비스를 마이그레이션하고 기존 **ovirt\_engine\_history** 데이터베이스와 연결하거나

선택적으로 데이터 웨어하우스 서비스를 마이그레이션하기 전에 **ovirt\_engine\_history** 데이터베이스를 새 데이터베이스 머신으로 마이그레이션합니다. **ovirt\_engine\_history** 데이터베이스가 Manager에서 호스트되는 경우 데이터 웨어하우스 서비스와 함께 데이터베이스를 마이그레이션하면 Manager 머신에서 리소스 경쟁이 더 줄어들립니다. 데이터베이스를 데이터 웨어하우스 서비스를 마이그레이션할 머신과 동일한 머신이나 Manager 머신 및 새 데이터 웨어하우스 서비스 머신과 별도의 머신에 마이그레이션할 수 있습니다.

### 1.3.1. 데이터 웨어하우스 데이터베이스를 별도의 머신으로 마이그레이션

데이터 웨어하우스 서비스를 마이그레이션하기 전에 선택적으로 **ovirt\_engine\_history** 데이터베이스를 마이그레이션할 수 있습니다. **engine-backup**을 사용하여 데이터베이스 백업을 만들고 새 데이터베이스 머신에서 백업을 복원합니다. **engine-backup**에 대한 자세한 내용을 보려면 **engine-backup --help**를 실행하십시오.

이 절차는 새 데이터베이스 서버에 Red Hat Enterprise Linux 7이 설치되어 있으며 적절한 서브스크립션이 설정되어 있다고 가정합니다. 자세한 내용은 [설치 가이드](#)에 있는 [필요한 서브스크립션을 등록](#)을 참조하십시오.

데이터 웨어하우스 서비스만 마이그레이션하려면 [1.3.2절. “데이터 웨어하우스 서비스를 별도의 머신으로 마이그레이션”](#)을 참조하십시오.

#### 절차 1.2. 데이터 웨어하우스 데이터베이스를 별도의 머신으로 마이그레이션

1. 데이터 웨어하우스 데이터베이스 및 설정 파일 백업을 생성합니다.

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

2. 백업 파일을 Manager에서 새 머신으로 복사합니다.

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

3. 새 머신에 **engine-backup**을 설치합니다.

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

4. 새 머신에서 데이터 웨어하우스 데이터베이스를 복원합니다. **file\_name**은 Manager에서 복사한 백업 파일입니다.

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

### 1.3.2. 데이터 웨어하우스 서비스를 별도의 머신으로 마이그레이션

Red Hat Virtualization Manager에 설치 및 설정된 데이터 웨어하우스 서비스를 전용 호스트 머신으로 마이그레이션합니다. 데이터 웨어하우스 서비스를 별도의 머신에서 호스트하면 Manager 머신의 로드를 줄이는 데 도움이 됩니다. 이 절차는 데이터 웨어하우스 서비스만 마이그레이션합니다. 데이터 웨어하우스 서비스를 마이그레이션하기 전에 데이터 웨어하우스 데이터베이스(**ovirt\_engine\_history** 데이터베이스라고도 함)를 마이그레이션하려면 [1.3.1절. “데이터 웨어하우스 데이터베이스를 별도의 머신으로 마이그레이션”](#)을 참조하십시오.

이 시나리오를 설치하려면 다음 네 가지 주요 단계를 수행해야 합니다.

1. 새 데이터 웨어하우스 머신을 설정합니다.

2. Manager 머신에서 데이터 웨어하우스 서비스를 중지합니다.
3. 새 데이터 웨어하우스 머신을 구성합니다.
4. Manager 머신에서 데이터 웨어하우스 패키지를 제거합니다.

## 전제 조건

다음 전제 조건을 완료해야 합니다.

1. Manager 및 데이터 웨어하우스가 같은 머신에 설치 및 설정되어 있어야 합니다.
2. 새 데이터 웨어하우스 머신을 설치하려면 다음이 필요합니다.
  - ※ Red Hat Enterprise Linux 7이 설치된 가상 머신 또는 물리적 머신
  - ※ Red Hat Enterprise Linux Server 및 Red Hat Virtualization 인타이틀먼트 풀에 대한 서브스크립션
  - ※ Manager의 `/etc/ovirt-engine/engine.conf.d/10-setup-database.conf` 파일에 있는 암호
  - ※ 데이터 웨어하우스 머신에서 Manager 데이터베이스 머신의 TCP 포트 5432에 대해 허용된 액세스
  - ※ Manager의 `/etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/10-setup-database.conf` 파일에 있는 `ovirt_engine_history` 데이터베이스 인증 정보. [1.3.1절. “데이터 웨어하우스 데이터베이스를 별도의 머신으로 마이그레이션”](#)을 통해 `ovirt_engine_history` 데이터베이스를 마이그레이션한 경우 해당 머신에 데이터베이스를 설치하는 동안 정의한 인증 정보를 가져옵니다.

## 절차 1.3. 단계 1: 새 데이터 웨어하우스 머신 설치

1. 메시지가 표시되면 고객 포털 사용자 이름과 암호를 입력하여 시스템을 Content Delivery Network에 등록합니다.

```
# subscription-manager register
```

2. Red Hat Enterprise Linux Server 및 Red Hat Virtualization 서브스크립션 풀을 찾아서 해당 풀 ID를 기록합니다.

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

3. 이전 단계에 있는 풀 ID를 사용하여 시스템에 인타이틀먼트를 연결합니다.

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

4. 기존의 모든 리포지토리를 비활성화합니다.

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

5. 필수 채널을 활성화합니다.

```
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-supplementary-rpms
# subscription-manager repos --enable=rhel-7-server-rhv-4.1-rpms
```

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

6. 현재 설치된 모든 패키지가 최신 상태인지 확인합니다.

```
# yum update
```

7. `ovirt-engine-dwh-setup` 패키지를 설치합니다.

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

#### 절차 1.4. 단계 2: Manager 머신에서 데이터 웨어하우스 서비스 중지

1. 데이터 웨어하우스 서비스를 중지합니다.

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

2. **ovirt\_engine\_history** 데이터베이스나 Manager 데이터베이스 또는 양쪽 모두가 Manager 머신에서 호스트되고 있으며 이전 버전(Red Hat Enterprise Virtualization 3.4 이전)으로 설정된 후 업그레이드된 경우 새 데이터 웨어하우스 머신에서 해당 데이터베이스에 액세스할 수 있도록 허용해야 합니다. `/var/lib/pgsql/data/postgresql.conf` 파일을 편집하여 **listen\_addresses** 행을 다음과 같이 수정합니다.

```
listen_addresses = '*'
```

해당 행이 존재하지 않거나 주석 처리된 경우 수동으로 추가합니다.

하나 또는 두 개의 데이터베이스가 모두 원격 머신에서 호스트되는 경우 각 머신에서 **postgres.conf** 파일을 편집하고 위에 표시된 바와 같이 **listen\_addresses** 행을 추가하여 수동으로 액세스 권한을 부여해야 합니다. 두 데이터베이스가 모두 Manager 머신에서 호스트되고 Red Hat Virtualization Manager를 새로 설치하는 동안 설정된 경우에는 액세스 권한이 기본적으로 부여됩니다.

3. postgresql 서비스를 다시 시작합니다.

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

#### 절차 1.5. 단계 3: 새 데이터 웨어하우스 머신 설정

1. **engine-setup** 명령을 실행하여 머신에서 데이터 웨어하우스 설정을 시작합니다.

```
# engine-setup
```

2. **Enter** 키를 눌러 데이터 웨어하우스를 설정합니다.

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

3. **Enter** 키를 눌러 방화벽을 자동으로 설정하거나 **No**를 입력하고 **Enter** 키를 눌러 기존 설정을 유지합니다.

```
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]:
```

사용자가 방화벽을 자동으로 설정하도록 선택하고 현재 활성 상태인 방화벽 관리자가 없으면 지원되는 옵션 목록에서 방화벽 관리자를 선택하라는 메시지가 표시됩니다. 방화벽 관리자 이름을 입력하고 **Enter** 키를 누릅니다. 목록에 옵션이 하나 밖에 없는 경우에도 이 방법이 적용됩니다.

4. **Enter** 키를 눌러 자동으로 감지된 호스트 이름을 수락하거나 대체 호스트 이름을 입력하고 **Enter** 키를 누릅니다.

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

5. **ovirt\_engine\_history** 데이터베이스 위치에 대한 다음 질문에 답합니다.

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

위에 표시된 대체 옵션을 입력하고 **Enter** 키를 누릅니다.

6. **ovirt\_engine\_history** 데이터베이스 호스트의 정규화된 도메인 이름 및 암호를 입력합니다. **Enter** 키를 눌러 각 필드의 기본값을 수락합니다.

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

7. **Manager** 데이터베이스 머신의 정규화된 도메인 이름 및 암호를 입력합니다. **Enter** 키를 눌러 각 필드의 기본값을 수락합니다.

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

8. **Enter** 키를 눌러 기존 데이터 웨어하우스 데이터베이스의 백업을 생성합니다.

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

데이터베이스 백업에 필요한 시간과 공간은 데이터베이스 크기에 따라 다릅니다. 완료하는 데 몇 시간이 걸릴 수 있습니다. 여기서 데이터베이스를 백업하지 않도록 선택하고 어떤 이유로든 **engine-setup**이 실패하면 데이터베이스 또는 그 안에 있는 데이터를 복원할 수 없습니다. 백업 파일 위치는 설치 스크립트 끝에 표시됩니다.

9. **Manager**에서 기존 데이터 웨어하우스 서비스 연결이 영구적으로 차단되어 있음을 확인합니다.

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

## 10. 설치 설정을 확인합니다.

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

## 절차 1.6. 단계 4: Manager 머신에서 데이터 웨어하우스 패키지 제거

1. 데이터 웨어하우스 패키지를 제거합니다.

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

이 단계에서는 데이터 웨어하우스 서비스가 1시간 후에 자동으로 다시 시작되지 않도록 합니다.

2. 데이터 웨어하우스 파일을 제거합니다.

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

이제 Manager가 호스트되는 머신과 다른 별도의 머신에서 데이터 웨어하우스 서비스가 호스트됩니다.

## 1.4. 데이터 웨어하우스 샘플링 규모 변경

Red Hat Virtualization에는 데이터 웨어하우스가 필요합니다. 데이터 웨어하우스는 Manager와 같은 머신이나 Manager에 액세스할 수 있는 별도의 머신에 설치 및 설정할 수 있습니다. 일부 설정에는 기본 데이터 보존 설정이 필요하지 않을 수 있으므로 **engine-setup**에서는 두 개의 데이터 샘플링 규모, 즉 **Basic** 및 **Full**을 제공합니다.

- ※ **Full**을 선택하면 [2.4절. “ovirt-engine-dwhd.conf의 데이터 웨어하우스 서비스 애플리케이션 설정”](#)에 있는 데이터 보존 설정 기본값이 사용됩니다(데이터 웨어하우스가 원격 호스트에 설치된 경우 권장됨).
- ※ **Basic**을 선택하면 **DWH\_TABLES\_KEEP\_HOURLY** 값이 720으로, **DWH\_TABLES\_KEEP\_DAILY** 값이 0으로 감소하여 Manager 머신의 로드가 완화됩니다(Manager와 데이터 웨어하우스를 동일한 머신에 설치한 경우 권장됨).

샘플링 규모는 설치하는 동안 **engine-setup**을 통해 설정됩니다.

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

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

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

**--reconfigure-dwh-scale** 옵션을 지정하여 다시 **engine-setup**을 실행하여 샘플링 규모를 나중에 변경할 수 있습니다.

## 예 1.1. 데이터 웨어하우스 샘플링 규모 변경

```
# engine-setup --reconfigure-dwh-scale
[...]
Welcome to the RHEV 4.0 setup/upgrade.
Please read the RHEV 4.0 install guide
https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Virtualization/4.1/html/Installation_Guide/index.htm
```

```

1.
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]:

```

필요한 경우 [2.4절. “ovirt-engine-dwhd.conf의 데이터 웨어하우스 서비스 애플리케이션 설정”](#)에 설명된 대로 개별 데이터 보존 설정을 조정할 수 도 있습니다.

## 2장. 기록 데이터베이스 개요

### 2.1. 기록 데이터베이스 개요

Red Hat Virtualization에는 보고 애플리케이션에서 데이터 센터, 클러스터, 호스트 수준의 보고서를 생성할 수 있는 포괄적인 관리 기록 데이터베이스가 포함되어 있습니다. 이 장에서는 기록 데이터베이스에 대한 쿼리 설정에 대해 설명합니다.

Red Hat Virtualization Manager에서는 **PostgreSQL 8.4.x**를 데이터베이스 플랫폼으로 사용하여 가상화 환경의 상태, 설정, 성능에 대한 정보를 저장합니다. 설치 시 Red Hat Virtualization Manager에서는 **engine**이라는 PostgreSQL 데이터베이스를 생성합니다.

**ovirt-engine-dwh** 패키지를 설치하면 **ovirt\_engine\_history**라는 두 번째 데이터베이스가 생성되며 여기에는 **engine** 운영 데이터베이스에서 시간이 지남에 따라 분 단위로 수집한 기록 설정 정보 및 통계 매트릭이 포함됩니다. 데이터베이스 변경 사항을 추적하면 데이터베이스의 개체에 대한 정보가 제공되므로 사용자가 활동을 분석하고 성능을 개선하며 문제를 해결할 수 있습니다.



#### 주의

**ovirt\_engine\_history** 데이터베이스의 데이터 복제는 **ovirt-engine-dwhd** (Red Hat Virtualization Manager Extract Transform Load Service)에 의해 실행됩니다. 이 서비스는 데이터 통합 도구인 Talend Open Studio를 기반으로 하며, 데이터 웨어하우스 패키지를 설치하는 동안 자동으로 시작되도록 설정되어 있습니다. **engine** 데이터베이스에서 데이터를 추출하고 데이터를 기록 데이터베이스 표준으로 변환하여 **ovirt\_engine\_history** 데이터베이스에 로드하는 Java 프로그램입니다.

**ovirt-engine-dwhd** 서비스를 중지해서는 안 됩니다.

**ovirt\_engine\_history** 데이터베이스 스키마는 시간이 지남에 따라 변경됩니다. 데이터베이스에는 버전이 있는 지원 API에 일관된 구조를 제공하는 데이터베이스 뷰가 포함되어 있습니다. 뷰는 데이터베이스 쿼리의 결과 집합으로 구성된 가상 테이블입니다. 데이터베이스는 뷰에 대한 정의를 **SELECT** 문으로 저장합니다. **SELECT** 문의 결과는 뷰에서 반환하는 가상 테이블에 채워집니다. 사용자는 테이블을 참조하는 것과 같은 방식으로 **PL/PGSQL** 문에서 뷰 이름을 참조합니다.

### 2.2. 설정 기록 추적

Red Hat Virtualization 기록 데이터베이스 (**ovirt\_engine\_history**)의 데이터를 사용하여 **engine** 데이터베이스를 추적할 수 있습니다.

ETL 서비스 **ovirt-engine-dwhd**는 다음과 같이 세 가지 유형의 변경 사항을 추적합니다.

- ✦ **engine** 데이터베이스에 새 엔티티 추가 - ETL 서비스에서 변경 사항을 **ovirt\_engine\_history** 데이터베이스에 새 항목으로 복제합니다.
- ✦ 기존 엔티티 업데이트 - ETL 서비스에서 변경 사항을 **ovirt\_engine\_history** 데이터베이스에 새 항목으로 복제합니다.
- ✦ **engine** 데이터베이스에서 엔티티 제거 - **ovirt\_engine\_history** 데이터베이스의 새 항목에서 해당 엔티티가 제거된 것으로 플래그를 지정합니다. 제거된 엔티티는 제거된 것으로만 플래그 지정할 수 있습니다.

**ovirt\_engine\_history** 데이터베이스에 있는 설정 테이블은 **engine** 데이터베이스에 있는 해당 테이블과 여러 가지 면에서 다릅니다. 가장 뚜렷한 차이점은 **설정** 열이 더 적다는 것입니다. 이는 특정 설정 항목이 다른



항목보다 관심도가 낮으며 데이터베이스 크기 문제로 인해 보관되지 않기 때문입니다. 또한 **engine** 데이터베이스의 일부 테이블에 있는 열은 **ovirt\_engine\_history**에서 단일 테이블에 표시되며 데이터를 더 편리하고 이해하기 쉽게 볼 수 있도록 다른 열 이름이 사용됩니다. 모든 설정 테이블에는 다음 항목이 포함됩니다.

- ※ 엔티티의 설정 버전을 나타내는 **history\_id**
- ※ 엔티티가 시스템에 추가된 날짜를 나타내는 **create\_date** 필드
- ※ 엔티티가 변경된 날짜를 나타내는 **update\_date** 필드
- ※ 엔티티가 시스템에서 제거된 날짜를 나타내는 **create\_date** 필드

## 2.3. 통계 기록

ETL 서비스는 분 단위로 데이터를 통계 테이블에 수집합니다. 최소 지난 24시간 동안의 데이터가 분 단위로 저장되지만 삭제 작업을 마지막으로 실행한 시간에 따라 최대 48시간 동안 저장될 수 있습니다. 2시간 이상 초과한 분 단위 데이터는 시간별 데이터로 집계되고 2개월 동안 보관됩니다. 2일 이상 경과한 시간별 데이터는 일별 데이터로 집계되고 5년 동안 보관됩니다.

시간별 데이터 및 일별 데이터는 시간별 및 일별 테이블에서 확인할 수 있습니다.

각 통계 데이터는 샘플, 시간별 및 일별 기록과 같은 개별 집계 수준 테이블에 보관됩니다. 또한 모든 기록 테이블에는 행을 고유하게 식별하는 **history\_id** 열이 있습니다. 테이블은 호스트의 설정 버전을 참조하여 이전 설정과 관련하여 엔티티의 통계에 대한 보고서를 활성화합니다.

## 2.4. ovirt-engine-dwhd.conf의 데이터 웨어하우스 서비스 애플리케이션 설정

다음은 데이터 웨어하우스 서비스의 애플리케이션을 설정하는 옵션 목록입니다. 이러한 옵션은 **/usr/share/ovirt-engine-dwh/services/ovirt-engine-dwhd/ovirt-engine-dwhd.conf** 파일에서 사용할 수 있습니다. **/etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/**의 덮어쓰기 파일에서 기본값에 대한 변경 사항을 설정합니다. 변경 사항을 저장한 후 데이터 웨어하우스 서비스를 다시 시작하십시오.

표 2.1. ovirt-engine-dwhd.conf 애플리케이션 설정 변수

변수 이름	기본값	설명
DWH_DELETE_JOB_HOUR	3	삭제 작업이 실행되는 시간입니다. 값을 0에서 23 사이로 지정합니다. 여기서 0은 자정입니다.
DWH_SAMPLING	60	통계 테이블에 데이터가 수집되는 간격(초)입니다.
DWH_TABLES_KEEP_SAMPLES	24	DWH_SAMPLING의 데이터가 저장되는 시간입니다. 2시간 이상 지난 데이터는 시간별 데이터로 집계됩니다.
DWH_TABLES_KEEP_HOURLY	1440	시간별 데이터가 저장되는 시간입니다. 기본값은 60일입니다. 2일 이상 지난 시간별 데이터는 일별 데이터로 집계됩니다.
DWH_TABLES_KEEP_DAILY	43800	일별 데이터가 저장되는 시간입니다. 기본값은 5년입니다.
DWH_ERROR_EVENT_INTERVAL	300000	Manager의 audit.log로 오류가 푸시되는 최소 간격(밀리초)입니다.

## 2.5. 태그 기록 추적

ETL 서비스에서는 관리 포털에 표시되는 태그 정보를 분 간격으로 수집하여 이 데이터를 태그 기록 테이블에 저장합니다. ETL 서비스는 다음과 같이 다섯 가지 유형의 변경 사항을 추적합니다.

- ❖ 관리 포털에서 태그 생성 - ETL 서비스에서 태그 상세 정보, 태그 트리에서의 위치, 태그 트리에 있는 기타 개체와의 관계를 복사합니다.
- ❖ 관리 포털의 태그 트리에 엔티티 연결 - ETL 서비스에서 추가 항목을 **ovirt\_engine\_history** 데이터베이스에 새 항목으로 복제합니다.
- ❖ 태그 업데이트 - ETL 서비스에서 태그 상세 정보 변경 사항을 **ovirt\_engine\_history** 데이터베이스에 새 항목으로 복제합니다.
- ❖ 관리 포털에서 엔티티 또는 태그 분기 지점 제거 - **ovirt\_engine\_history** 데이터베이스에서 해당 태그 및 관계를 새 항목에서 제거된 것으로 플래그 지정합니다. 제거된 태그 및 관계는 제거되거나 분리된 것으로만 플래그 지정할 수 있습니다.
- ❖ 태그 분기 지점 제거 - 해당 태그 및 관계가 새 항목으로 업데이트됩니다. 이동한 태그 및 관계는 업데이트된 것으로만 플래그 지정할 수 있습니다.

## 2.6. 기록 데이터베이스에 대한 읽기 전용 액세스 허용

편집을 허용하지 않고 기록 데이터베이스에 대한 액세스를 허용하려면 **ovirt\_engine\_history** 데이터베이스에 로그인하여 데이터를 읽을 수 있는 읽기 전용 PostgreSQL 사용자를 만들어야 합니다. 이 절차는 기록 데이터베이스가 설치된 시스템에서 실행해야 합니다.

### 절차 2.1. 기록 데이터베이스에 대한 읽기 전용 액세스 허용

1. 기록 데이터베이스에 대한 읽기 전용 액세스를 부여할 사용자를 만듭니다.

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

2. 새로 만든 사용자에게 기록 데이터베이스에 연결할 수 있는 권한을 부여합니다.

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

3. 새로 만든 사용자에게 **public** 스키마 사용을 허용합니다.

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

4. 새로 만든 사용자에게 부여할 나머지 권한을 생성하여 파일에 저장합니다.

```
# 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. 이전 단계에서 만든 파일을 사용하여 새로 만든 사용자에게 권한을 부여합니다.

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

6. 새로 만든 사용자에게 권한을 부여하는 데 사용한 파일을 제거합니다.

```
# rm grant.sql
```

이제 다음 명령을 사용하여 새로 만든 사용자로 **ovirt\_engine\_history** 데이터베이스에 액세스할 수 있습니다.

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

**ovirt\_engine\_history** 데이터베이스의 테이블 및 보기에 대한 **SELECT** 문은 성공하지만 수정은 실패합니다.

## 2.7. 통계 기록 보기

통계 데이터는 **hourly, daily, samples** 보기에서 확인할 수 있습니다.

통계 보기를 쿼리하려면 **SELECT \* FROM view\_name\_[hourly|daily|samples];**를 실행합니다. 예를 들면 다음과 같습니다.

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

사용 가능한 모든 보기를 나열하려면 다음을 실행합니다.

```
# \dv
```

### 2.7.1. 스토리지 도메인 통계 보기

표 2.2. 시스템의 각 스토리지 도메인에 대한 기록 통계

이름	유형	설명
history_id	bigint	이 테이블 행의 고유 ID입니다.
history_datetime	날짜	이 기록 행의 타임스탬프입니다(집계 수준에 따라 분, 시간, 일로 반올림됨).
storage_domain_id	uuid	시스템에서 해당 스토리지 도메인의 고유 ID입니다.
storage_domain_status	smallint	스토리지 도메인 상태입니다.
seconds_in_status	정수	스토리지 도메인이 집계 기간 동안 상태 열에 표시된 상태에서 경과한 총 시간(초)입니다. 예를 들어, 스토리지 도메인이 1분 중 55초 동안 "활성" 상태였다가 5초 동안 "비활성" 상태가 된 경우 해당 1분에 대해 테이블에 두 개의 행이 보고됩니다. 하나의 행에는 상태 <b>Active</b> 와 <b>seconds_in_status</b> 값인 55가 표시되고 다른 행에는 상태 <b>Inactive</b> 와 <b>seconds_in_status</b> 값인 5가 표시됩니다.

이름	유형	설명
minutes_in_status	숫자(7,2)	스토리지 도메인이 집계 기간 동안 상태 열에 표시된 상태에서 경과한 총 시간(분)입니다. 예를 들어, 스토리지 도메인이 1시간 중 55분 동안 "활성" 상태였다가 5분 동안 "비활성" 상태가 된 경우 해당 1시간에 대해 테이블에 두 개의 행이 보고됩니다. 하나의 행에는 상태 <b>Active</b> 와 minutes_in_status 값인 55가 표시되고 다른 행에는 상태 <b>Inactive</b> 와 minutes_in_status 값인 5가 표시됩니다.
available_disk_size_gb	정수	디스크에서 사용 가능한(사용되지 않은) 총 용량(GB)입니다.
used_disk_size_gb	정수	디스크에서 사용된 총 용량(GB)입니다.
storage_configuration_version	정수	샘플링 시점의 스토리지 도메인 설정 버전입니다.

### 2.7.2. 호스트 통계 보기

표 2.3. 시스템의 각 호스트에 대한 기록 통계

이름	유형	설명
history_id	bigint	이 테이블 행의 고유 ID입니다.
history_datetime	날짜	이 기록 행의 타임스탬프입니다(집계 수준에 따라 분, 시간, 일로 반올림됨).
host_id	uuid	시스템에서 해당 호스트의 고유 ID입니다.
host_status	smallint	<ul style="list-style-type: none"> <li>※ -1 - <b>Unknown Status</b>(ETL의 문제를 나타내는 데에만 사용됩니다. Red Hat 지원팀에 보고하십시오.)</li> <li>※ 1 - <b>Up</b></li> <li>※ 2 - <b>Maintenance</b></li> <li>※ 3 - <b>Problematic</b></li> </ul>
seconds_in_status	정수	호스트가 집계 기간 동안 상태 열에 표시된 상태에서 경과한 총 시간(초)입니다. 예를 들어, 호스트가 1분 중 55초 동안 가동되었다가 5초 동안 중단된 경우 해당 1분에 대해 두 개의 행이 표시됩니다. 하나의 행에는 상태 <b>Up</b> 과 seconds_in_status 값인 55로 표시되고 다른 행에는 상태 <b>Down</b> 과 seconds_in_status 값인 5가 표시됩니다.

이름	유형	설명
minutes_in_status	숫자(7,2)	호스트가 집계 기간 동안 상태 열에 표시된 상태에서 경과한 총 시간(분)입니다. 예를 들어, 호스트가 1시간 중 55분 동안 가동되었다가 5분 동안 중단된 경우 해당 시간에 대해 두 개의 행이 표시됩니다. 하나의 행에는 상태 <b>Up</b> 과 minutes_in_status 값인 55가 표시되고 다른 행에는 상태 <b>Down</b> 과 minutes_in_status 값인 5가 표시됩니다.
memory_usage_percent	smallint	호스트의 메모리 사용률(%)입니다.
max_memory_usage	smallint	집계 기간 동안의 최대 메모리 사용량이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
ksm_shared_memory_mb	bigint	호스트에서 사용하고 있는 커널 공유 메모리 크기(MB)입니다.
max_ksm_shared_memory_mb	bigint	집계 기간 동안의 최대 KSM 메모리 사용량이 메가바이트(MB)로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
cpu_usage_percent	smallint	호스트의 CPU 사용률(%)입니다.
max_cpu_usage	smallint	집계 기간 동안의 최대 CPU 사용량이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
ksm_cpu_percent	smallint	호스트의 CPU KSM 사용률(%)입니다.
max_ksm_cpu_percent	smallint	집계 기간 동안의 최대 KSM 사용량이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
active_vms	smallint	이 집계 기간 동안 활성 상태인 가상 머신의 평균 수입니다.
max_active_vms	smallint	집계 기간 동안 활성 상태인 가상 머신의 최대 수입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
total_vms	smallint	이 집계 동안 호스트에 있는 모든 가상 머신의 평균 수입니다.
max_total_vms	smallint	집계 기간 동안의 최대 총 가상 머신 수입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
total_vms_vcpus	정수	호스트에 할당된 총 vCPU 수입니다.

이름	유형	설명
max_total_vms_vcpus	정수	집계 기간 동안의 최대 총 가상 머신 vCPU 수입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
cpu_load	정수	호스트의 CPU 로드입니다.
max_cpu_load	정수	집계 기간 동안의 최대 CPU 로드입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
system_cpu_usage_percent	smallint	호스트의 CPU 사용률(%)입니다.
max_system_cpu_usage_percent	smallint	집계 기간 동안의 최대 시스템 CPU 사용률이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
user_cpu_usage_percent	smallint	호스트의 사용자 CPU 사용률(%)입니다.
max_user_cpu_usage_percent	smallint	집계 기간 동안의 최대 사용자 CPU 사용률이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
swap_used_mb	정수	호스트의 Swap 크기 사용량(MB)입니다.
max_swap_used_mb	정수	집계 기간 동안 호스트의 최대 사용자 Swap 크기 사용량(MB)이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
host_configuration_version	정수	샘플링 시 호스트 설정 버전입니다.

### 2.7.3. 호스트 인터페이스 통계 보기

표 2.4. 시스템의 각 호스트 네트워크 인터페이스에 대한 기록 통계

이름	유형	설명
history_id	bigint	이 테이블 행의 고유 ID입니다.
history_datetime	날짜	이 기록 보기의 타임스탬프입니다 (집계 수준에 따라 분, 시간, 일로 반올림됨).
host_interface_id	uuid	시스템에서 해당 인터페이스의 고유 ID입니다.
receive_rate_percent	smallint	호스트에서 사용된 수신율(%)입니다.
max_receive_rate_percent	smallint	집계 기간 동안의 최대 수신율이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
transmit_rate_percent	smallint	호스트에서 사용된 전송률(%)입니다.

이름	유형	설명
max_transmit_rate_percent	smallint	집계 기간 동안의 최대 전송률이 백분율로 표시됩니다. 이 값은 시간별 집계 경우 수집된 최대 샘플 값에 해당하고, 일별 집계 경우 시간별 최대 평균 전송률에 해당합니다.
received_total_byte	bigint	호스트에서 수신한 총 바이트 수입니다.
transmitted_total_byte	bigint	호스트에서 전송된 총 바이트 수입니다.
host_interface_configuration_version	정수	샘플링 시점의 호스트 인터페이스 설정 버전입니다.

### 2.7.4. 가상 머신 통계 보기

표 2.5. 시스템의 각 가상 머신에 대한 기록 통계

이름	유형	설명
history_id	bigint	이 테이블 행의 고유 ID입니다.
history_datetime	날짜	이 기록 행의 타임스탬프입니다(집계 수준에 따라 분, 시간, 일 단위로 반올림됨).
vm_id	uuid	시스템에서 해당 가상 머신의 고유 ID입니다.
vm_status	smallint	<ul style="list-style-type: none"> <li>※ -1 - Unknown Status (ETL의 문제를 나타내는 데에만 사용됩니다. Red Hat 지원팀에 보고하십시오.)</li> <li>※ 0 - Down</li> <li>※ 1 - Up</li> <li>※ 2 - Paused</li> <li>※ 3 - Problematic</li> </ul>
seconds_in_status	정수	가상 머신이 집계 기간 동안 상태 열에 표시된 상태에서 경과한 총 시간(초)입니다. 예를 들어, 가상 머신이 1분 중 55초 동안 가동되었다가 5초 동안 중단된 경우 해당 1분에 대해 두 개의 행이 표시됩니다. 하나의 행에는 상태 Up과 seconds_in_status 값인 55가 표시되고 다른 행에는 상태 Down과 seconds_in_status 값인 5가 표시됩니다.
minutes_in_status	숫자(7,2)	가상 머신이 집계 기간 동안 상태 열에 표시된 상태에서 경과한 총 시간(분)입니다. 예를 들어, 가상 머신이 1시간 중 55분 동안 가동되었다가 5분 동안 중단된 경우 이 시간에 대해 두 개의 행이 표시됩니다. 하나의 행에는 상태 Up과 minutes_in_status 값인 55가 표시되고 다른 행에는 상태 Down과 minutes_in_status 값인 5가 표시됩니다.
cpu_usage_percent	smallint	가상 머신에서 사용 중인 CPU의 백분율입니다.



이름	유형	설명
max_cpu_usage	smallint	집계 기간 동안의 최대 CPU 사용량이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
memory_usage_percent	smallint	가상 머신에서 사용된 메모리의 백분율입니다. 메모리 사용량을 기록하려면 가상 머신에 게스트 도구를 설치해야 합니다.
max_memory_usage	smallint	집계 기간 동안의 최대 메모리 사용량이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다. 메모리 사용량을 기록하려면 가상 머신에 게스트 도구를 설치해야 합니다.
user_cpu_usage_percent	smallint	호스트의 사용자 CPU 사용률(%)입니다.
max_user_cpu_usage_percent	smallint	집계 기간 동안의 최대 사용자 CPU 사용량이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
system_cpu_usage_percent	smallint	호스트의 시스템 CPU 사용률(%)입니다.
max_system_cpu_usage_percent	smallint	집계 기간 동안의 최대 시스템 CPU 사용량이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
vm_ip	텍스트	첫 번째 NIC의 IP 주소입니다. 게스트 에이전트가 설치된 경우에만 표시됩니다.
currently_running_on_host	uuid	가상 머신이 실행되고 있는 호스트의 고유 ID입니다.
current_user_id	uuid	게스트 에이전트가 설치된 경우 가상 머신 콘솔에 로그인한 사용자의 고유 ID입니다.
disks_usage	텍스트	디스크 설명입니다. 파일 시스템 유형, 마운트 지점, 전체 크기, 사용된 크기가 포함됩니다.
vm_configuration_version	정수	샘플링 시점의 가상 머신 설정 버전입니다.
current_host_configuration_version	정수	샘플링 시점의 호스트 설정 버전입니다.
memory_buffered_kb	bigint	가상 머신의 버퍼링된 메모리 크기(KB)입니다.
memory_cached_kb	bigint	가상 머신의 캐시된 메모리 크기(KB)입니다.
max_memory_buffered_kb	bigint	집계 기간 동안 버퍼링된 최대 메모리 (KB)입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.



이름	유형	설명
max_memory_cached_kb	bigint	집계 기간 동안 캐시된 최대 메모리(KB)입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.

### 2.7.5. 가상 머신 인터페이스 통계 보기

표 2.6. 시스템의 가상 머신 네트워크 인터페이스에 대한 기록 통계

이름	유형	설명
history_id	정수	이 테이블 행의 고유 ID입니다.
history_datetime	날짜	이 기록 행의 타임스탬프입니다(집계 수준에 따라 분, 시간, 일 단위로 반올림됨).
vm_interface_id	uuid	시스템에서 해당 인터페이스의 고유 ID입니다.
receive_rate_percent	smallint	호스트에서 사용된 수신율(%)입니다.
max_receive_rate_percent	smallint	집계 기간 동안의 최대 수신율이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
transmit_rate_percent	smallint	호스트에서 사용된 전송률(%)입니다.
max_transmit_rate_percent	smallint	집계 기간 동안의 최대 전송률이 백분율로 표시됩니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 전송률에 해당합니다.
received_total_byte	bigint	가상 머신에서 수신한 총 바이트 수입니다.
transmitted_total_byte	bigint	가상 머신에서 전송한 총 바이트 수입니다.
vm_interface_configuration_version	정수	샘플링 시점의 가상 머신 인터페이스 설정 버전입니다.

### 2.7.6. 가상 디스크 통계 보기

표 2.7. 시스템의 가상 디스크에 대한 기록 통계

이름	유형	설명
history_id	bigint	이 테이블 행의 고유 ID입니다.
history_datetime	날짜	이 기록 행의 타임스탬프입니다(집계 수준에 따라 분, 시간, 일 단위로 반올림됨).
vm_disk_id	uuid	시스템에서 해당 디스크의 고유 ID입니다.
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>

이름	유형	설명
seconds_in_status	정수	가상 디스크가 집계 기간 동안 상태 열에 표시된 상태에서 경과한 총 시간(초)입니다. 예를 들어, 가상 디스크가 1분 중 55초 동안 잠겨 있다가 5초 동안 OK 상태가 된 경우 해당 1분에 대해 두 개의 행이 표시됩니다. 하나의 행에는 상태 <b>Locked</b> 와 seconds_in_status 값인 55가 표시되고 다른 행에는 상태 OK와 seconds_in_status 값인 5가 표시됩니다.
minutes_in_status	숫자(7,2)	가상 디스크가 집계 기간 동안 상태 열에 표시된 상태에서 경과한 총 시간(분)입니다. 예를 들어, 가상 디스크가 1시간 중 55분 동안 잠겨 있다가 5분 동안 OK 상태가 된 경우 이 1시간에 대해 두 개의 행이 표시됩니다. 하나의 행에는 상태 <b>Locked</b> 와 minutes_in_status 값인 55가 표시되고 다른 행에는 상태 OK와 minutes_in_status 값인 5가 표시됩니다.
vm_disk_actual_size_mb	정수	디스크에 할당된 실제 크기입니다.
read_rate_bytes_per_second	정수	디스크 읽기 속도(초당 바이트 수)입니다.
max_read_rate_bytes_per_second	정수	집계 기간 동안의 최대 읽기 속도입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
read_latency_seconds	숫자(18,9)	측정된 가상 디스크 읽기 대기 시간(초)입니다.
max_read_latency_seconds	숫자(18,9)	집계 기간 동안 측정된 최대 읽기 대기 시간(초)입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
write_rate_bytes_per_second	정수	디스크 쓰기 속도(초당 바이트 수)입니다.
max_write_rate_bytes_per_second	정수	집계 기간 동안의 최대 쓰기 속도입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
write_latency_seconds	숫자(18,9)	측정된 가상 디스크 쓰기 대기 시간(초)입니다.
max_write_latency_seconds	숫자(18,9)	집계 기간 동안 측정된 최대 쓰기 대기 시간(초)입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.
flush_latency_seconds	숫자(18,9)	측정된 가상 디스크 플러시 대기 시간(초)입니다.
max_flush_latency_seconds	숫자(18,9)	집계 기간 동안 측정된 최대 플러시 대기 시간(초)입니다. 이 값은 시간별 집계의 경우 수집된 최대 샘플 값에 해당하고, 일별 집계의 경우 시간별 최대 평균 값에 해당합니다.

이름	유형	설명
vm_disk_configuration_version	정수	샘플링 시점의 가상 디스크 설정 버전입니다.

## 2.8. 설정 기록 보기

설정 보기를 쿼리하려면 **SELECT \* FROM view\_name;**을 실행합니다. 예를 들면 다음과 같습니다.

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

사용 가능한 모든 보기를 나열하려면 다음을 실행합니다.

```
# \dv
```



### 참고

**delete\_date**는 최신 보기에 나타나지 않습니다. 최신 보기에서는 정의에 따라 삭제되지 않고 존재하는 엔티티의 최신 설정을 제공하기 때문입니다.

### 2.8.1. 데이터 센터 설정

다음 표에서는 시스템에 있는 데이터 센터의 설정 기록 매개 변수를 보여줍니다.

표 2.8. v4\_0\_configuration\_history\_datacenters

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
datacenter_id	uuid	시스템에서 해당 데이터 센터의 고유 ID입니다.
datacenter_name	다양한 문자(40)	데이터 센터 이름으로 편집 대화 상자에 표시됩니다.
datacenter_description	다양한 문자(4000)	데이터 센터에 대한 설명으로 편집 대화 상자에 표시됩니다.
is_local_storage	부울	데이터 센터에서 로컬 스토리지를 사용하는지를 나타내는 플래그입니다.
create_date	시간대가 포함된 타임스탬프	이 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	시스템에서 이 엔티티가 변경된 날짜입니다.
delete_date	시간대가 포함된 타임스탬프	시스템에서 이 엔티티가 삭제된 날짜입니다.

### 2.8.2. 데이터 센터 스토리지 도메인 맵

다음 표에서는 시스템에 있는 스토리지 도메인과 데이터 센터의 관계를 보여줍니다.

표 2.9. v4\_0\_map\_history\_datacenters\_storage\_domains

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
storage_domain_id	uuid	시스템에서 이 스토리지 도메인의 고유 ID입니다.
datacenter_id	uuid	시스템에서 해당 데이터 센터의 고유 ID입니다.
attach_date	시간대가 포함된 타임스탬프	스토리지 도메인이 데이터 센터에 연결된 날짜입니다.
detach_date	시간대가 포함된 타임스탬프	스토리지 도메인이 데이터 센터에서 분리된 날짜입니다.

### 2.8.3. 스토리지 도메인 설정

다음 표에서는 시스템에 있는 스토리지 도메인의 설정 기록 매개 변수를 보여줍니다.

표 2.10. v4\_0\_configuration\_history\_storage\_domains

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
storage_domain_id	uuid	시스템에서 이 스토리지 도메인의 고유 ID입니다.
storage_domain_name	다양한 문자(250)	스토리지 도메인 이름입니다.
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>
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	시간대가 포함된 타임스탬프	해당 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 변경된 날짜입니다.
delete_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 삭제된 날짜입니다.

### 2.8.4. 클러스터 설정

다음 표에서는 시스템에 있는 클러스터의 설정 기록 매개 변수를 보여줍니다.

표 2.11. v4\_0\_configuration\_history\_clusters

이름	유형	설명
----	----	----

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
cluster_id	uuid	이 클러스터가 속하는 데이터 센터의 고유 ID입니다.
cluster_name	다양한 문자(40)	클러스터 이름으로, 편집 대화 상자에 표시됩니다.
cluster_description	다양한 문자(4000)	편집 대화 상자에 정의됩니다.
datacenter_id	uuid	이 클러스터가 속하는 데이터 센터의 고유 ID입니다.
cpu_name	다양한 문자(255)	편집 대화 상자에 표시됩니다.
compatibility_version	다양한 문자(40)	편집 대화 상자에 표시됩니다.
datacenter_configuration_version	정수	생성 또는 업데이트 시점의 데이터 센터 설정 버전입니다.
create_date	시간대가 포함된 타임스탬프	이 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	시스템에서 이 엔티티가 변경된 날짜입니다.
delete_date	시간대가 포함된 타임스탬프	시스템에서 이 엔티티가 삭제된 날짜입니다.

### 2.8.5. 호스트 설정

다음 표에서는 시스템에서 해당 호스트의 설정 기록 매개 변수를 보여줍니다.

표 2.12. v4\_0\_configuration\_history\_hosts

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
host_id	uuid	시스템에서 해당 호스트의 고유 ID입니다.
host_unique_id	다양한 문자(128)	이 필드는 호스트의 실제 UUID와 MAC 주소 중 하나가 조합된 것으로 시스템에 이미 등록된 호스트를 감지하는 데 사용됩니다.
host_name	다양한 문자(255)	호스트 이름으로 편집 대화 상자에 있는 이름과 동일합니다.
cluster_id	uuid	이 호스트가 속하는 클러스터의 고유 ID입니다.
host_type	smallint	<ul style="list-style-type: none"> <li>※ 0 - RHEL Host</li> <li>※ 2 - RHEV Hypervisor Node</li> </ul>
fqdn_or_ip	다양한 문자(255)	통신에 사용할 Red Hat Virtualization Manager에 대한 호스트의 DNS 이름 또는 IP 주소로 편집 대화 상자에 표시됩니다.
memory_size_mb	정수	호스트의 실제 메모리 용량(MB)입니다.
swap_size_mb	정수	호스트 Swap 파티션 크기입니다.
cpu_model	다양한 문자(255)	호스트의 CPU 모델입니다.
number_of_cores	smallint	호스트의 총 CPU 코어 수입니다.

이름	유형	설명
number_of_sockets	smallint	총 CPU 소켓 수입니다.
cpu_speed_mh	숫자(18,0)	호스트의 CPU 속도(MHz)입니다.
host_os	다양한 문자(255)	호스트의 운영 체제 버전입니다.
kernel_version	다양한 문자(255)	호스트의 커널 버전입니다.
kvm_version	다양한 문자(255)	호스트의 KVM 버전입니다.
vdsd_version	다양한 문자	호스트의 VDSM 버전입니다.
vdsd_port	정수	편집 대화 상자에 표시됩니다.
threads_per_core	smallint	코어당 총 스레드 수입니다.
hardware_manufacturer	다양한 문자(255)	호스트의 하드웨어 제조 업체입니다.
hardware_product_name	다양한 문자(255)	호스트의 하드웨어 제품 이름입니다.
hardware_version	다양한 문자(255)	호스트의 하드웨어 버전입니다.
hardware_serial_number	다양한 문자(255)	호스트의 하드웨어 일련 번호입니다.
cluster_configuration_version	정수	생성 또는 업데이트 시점의 클러스터 설정 버전입니다.
create_date	시간대가 포함된 타임스탬프	이 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	시스템에서 이 엔티티가 변경된 날짜입니다.
delete_date	시간대가 포함된 타임스탬프	시스템에서 이 엔티티가 삭제된 날짜입니다.

### 2.8.6. 호스트 인터페이스 설정

다음 표에서는 시스템에 있는 호스트 인터페이스의 설정 기록 매개 변수를 보여줍니다.

표 2.13. v4\_0\_configuration\_history\_hosts\_interfaces

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
host_interface_id	uuid	시스템에서 해당 인터페이스의 고유 ID입니다.
host_interface_name	다양한 문자(50)	호스트에서 보고되는 인터페이스 이름입니다.
host_id	uuid	이 인터페이스가 속하는 호스트의 고유 ID입니다.
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	정수	인터페이스 속도(초당 비트 수)입니다.
mac_address	다양한 문자(59)	인터페이스 MAC 주소입니다.
logical_network_name	다양한 문자(50)	인터페이스와 연결된 논리 네트워크입니다.
ip_address	다양한 문자(20)	편집 대화 상자에 표시됩니다.
gateway	다양한 문자(20)	편집 대화 상자에 표시됩니다.
bond	부울	이 인터페이스가 본딩된 인터페이스인지를 나타내는 플래그입니다.

이름	유형	설명
bond_name	다양한 문자(50)	이 인터페이스가 속하는 본딩의 이름입니다(본딩의 일부인 경우).
vlan_id	정수	편집 대화 상자에 표시됩니다.
host_configuration_version	정수	생성 또는 업데이트 시점의 호스트 설정 버전입니다.
create_date	시간대가 포함된 타임스탬프	이 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	시스템에서 이 엔티티가 변경된 날짜입니다.
delete_date	시간대가 포함된 타임스탬프	시스템에서 이 엔티티가 삭제된 날짜입니다.

### 2.8.7. 가상 머신 설정

다음 표에서는 시스템에 있는 가상 머신의 설정 기록 매개 변수를 보여줍니다.

표 2.14. v4\_0\_configuration\_history\_vms

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
vm_id	uuid	시스템에서 가상 머신의 고유 ID입니다.
vm_name	다양한 문자(255)	가상 머신의 이름입니다.
vm_description	다양한 문자(4000)	편집 대화 상자에 표시됩니다.
vm_type	smallint	<ul style="list-style-type: none"> <li>✱ 0 - Desktop</li> <li>✱ 1 - Server</li> </ul>
cluster_id	uuid	이 가상 머신이 속하는 클러스터의 고유 ID입니다.
template_id	uuid	이 가상 머신이 파생되는 템플릿의 고유 ID입니다. 이 필드는 나중에 템플릿이 이 버전의 기록 데이터베이스와 동기화되지 않을 때 사용하기 위한 것입니다.
template_name	다양한 문자(40)	이 가상 머신이 파생되는 템플릿의 이름입니다.
cpu_per_socket	smallint	소켓당 가상 CPU 수입니다.
number_of_sockets	smallint	총 가상 CPU 소켓 수입니다.
memory_size_mb	정수	가상 머신에 할당된 총 메모리(MB)입니다.

이름	유형	설명
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> <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	시스템에 있는 기본 호스트의 ID로, 편집 대화 상자에 표시됩니다.
high_availability	부울	편집 대화 상자에 표시됩니다.
initialized	부울	이 가상 머신이 Sysprep 초기화를 위해 한 번 이상 시작되었는지를 나타내는 플래그입니다.
stateless	부울	편집 대화 상자에 표시됩니다.
fail_back	부울	편집 대화 상자에 표시됩니다.
usb_policy	smallint	편집 대화 상자에 표시됩니다.
time_zone	다양한 문자(40)	편집 대화 상자에 표시됩니다.
vm_pool_id	uuid	이 가상 머신이 속하는 풀의 ID입니다.
vm_pool_name	다양한 문자(255)	가상 머신 풀의 이름입니다.



이름	유형	설명
created_by_user_id	uuid	이 가상 머신을 만든 사용자의 ID입니다.
cluster_configuration_version	정수	생성 또는 업데이트 시점의 클러스터 설정 버전입니다.
default_host_configuration_version	정수	생성 또는 업데이트 시점의 호스트 설정 버전입니다.
create_date	시간대가 포함된 타임스탬프	해당 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 변경된 날짜입니다.
delete_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 삭제된 날짜입니다.

### 2.8.8. 가상 머신 인터페이스 설정

다음 표에서는 시스템에 있는 가상 인터페이스의 설정 기록 매개 변수를 보여줍니다.

표 2.15. v4\_0\_configuration\_history\_vms\_interfaces

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
vm_interface_id	uuid	시스템에서 해당 인터페이스의 고유 ID입니다.
vm_interface_name	다양한 문자(50)	편집 대화 상자에 표시됩니다.
vm_interface_type	smallint	가상 인터페이스의 유형입니다. <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	정수	집계 중인 인터페이스의 평균 속도(초당 비트 수)입니다.
mac_address	다양한 문자(20)	편집 대화 상자에 표시됩니다.
logical_network_name	다양한 문자(50)	편집 대화 상자에 표시됩니다.
vm_configuration_version	정수	생성 또는 업데이트 시점의 가상 머신 설정 버전입니다.
create_date	시간대가 포함된 타임스탬프	해당 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 변경된 날짜입니다.
delete_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 삭제된 날짜입니다.

### 2.8.9. 가상 머신 장치 설정

다음 표에서는 디스크 및 가상 인터페이스를 포함하여 가상 머신 및 해당 연결 장치 간의 관계를 보여줍니다.

표 2.16. v4\_0\_configuration\_history\_vms\_devices

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
vm_id	uuid	시스템에서 해당 가상 머신의 고유 ID입니다.
device_id	uuid	시스템에서 해당 장치의 고유 ID입니다.
type	다양한 문자(30)	가상 머신 장치의 유형입니다. "disk" 또는 "interface"가 될 수 있습니다.
address	다양한 문자(255)	장치의 실제 주소입니다.
is_managed	부울	해당 장치를 <b>Manager</b> 로 관리하는지를 나타내는 플래그입니다.
is_plugged	부울	해당 장치가 가상 머신에 연결되어 있는지를 나타내는 플래그입니다.
is_readonly	부울	해당 장치가 읽기 전용인지를 나타내는 플래그입니다.
vm_configuration_version	정수	샘플을 가져온 시점의 가상 머신 설정 버전입니다.
device_configuration_version	정수	샘플을 가져온 시점의 장치 설정 버전입니다.
create_date	시간대가 포함된 타임스탬프	해당 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	해당 엔티티가 시스템에 추가된 날짜입니다.
delete_date	시간대가 포함된 타임스탬프	해당 엔티티가 시스템에 추가된 날짜입니다.

### 2.8.10. 가상 디스크 설정

다음 표에서는 시스템에 있는 가상 디스크의 설정 기록 매개 변수를 보여줍니다.

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

이름	유형	설명
history_id	정수	기록 데이터베이스의 설정 버전 ID입니다.
vm_disk_id	uuid	시스템에서 이 디스크의 고유 ID입니다.
vm_disk_name	text	가상 디스크 이름으로, 편집 대화 상자에 표시됩니다.
vm_disk_description	다양한 문자(500)	편집 대화 상자에 표시됩니다.
image_id	uuid	시스템에서 해당 이미지의 고유 ID입니다.
storage_domain_id	uuid	디스크 이미지가 속하는 스토리지 도메인의 ID입니다.
vm_disk_size_mb	정수	정의된 디스크 크기(MB)입니다.

이름	유형	설명
vm_disk_type	smallint	편집 대화 상자에 표시됩니다. 현재는 시스템 및 데이터만 사용됩니다. <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	편집 대화 상자에 표시됩니다. <ul style="list-style-type: none"> <li>3 - Unassigned</li> <li>4 - COW</li> <li>5 - RAW</li> </ul>
is_shared	부울	해당 가상 머신의 디스크를 공유하는지를 나타내는 플래그입니다.
create_date	시간대가 포함된 타임스탬프	해당 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 변경된 날짜입니다.
delete_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 삭제된 날짜입니다.

### 2.8.11. 사용자 상세 정보 기록

다음 표에서는 시스템에 있는 사용자의 설정 기록 매개 변수를 보여줍니다.

표 2.18. v4\_0\_users\_details\_history

이름	유형	설명
user_id	uuid	시스템의 사용자 고유 ID로 Manager에서 생성합니다.
first_name	다양한 문자(255)	사용자의 이름입니다.
last_name	다양한 문자(255)	사용자의 성입니다.
domain	다양한 문자(255)	인증 확장 프로그램의 이름입니다.
username	다양한 문자(255)	계정 이름입니다.
department	다양한 문자(255)	사용자가 속한 조직의 부서입니다.
user_role_title	다양한 문자(255)	조직 내 사용자의 직책 또는 역할입니다.
email	다양한 문자(255)	조직에 포함된 사용자의 이메일입니다.
external_id	텍스트	외부 시스템의 사용자 고유 ID입니다.
active	부울	사용자가 활성 상태인지 아닌지를 나타내는 플래그입니다. 이 플래그는 시간별로 확인됩니다. 사용자를 인증 확장 프로그램에서 확인할 수 있는 경우 활성 상태로 유지됩니다. 사용자가 로그인에 성공하면 활성 상태가 됩니다.
create_date	시간대가 포함된 타임스탬프	해당 엔티티가 시스템에 추가된 날짜입니다.
update_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 변경된 날짜입니다.

이름	유형	설명
delete_date	시간대가 포함된 타임스탬프	시스템에서 해당 엔티티가 삭제된 날짜입니다.