



## Red Hat Satellite 6.15

### 가상 머신 서브스크립션 구성

virt-who를 사용하여 호스트 기반 서브스크립션 관리



## Red Hat Satellite 6.15 가상 머신 서브스크립션 구성

---

virt-who를 사용하여 호스트 기반 서브스크립션 관리

Red Hat Satellite Documentation Team

[satellite-doc-list@redhat.com](mailto:satellite-doc-list@redhat.com)

## 법적 공지

Copyright © 2024 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

이 가이드에서는 Red Hat Satellite의 서브스크립션 및 활성화 키 준비, virt-who 구성, 하이퍼바이저에서 서브스크립션을 상속하도록 가상 시스템 등록에 대한 정보를 제공합니다.

## 차례

RED HAT 문서에 관한 피드백 제공 .....	3
1장. 소개 .....	4
1.1. 호스트 기반 서브스크립션 .....	4
1.2. 구성 개요 .....	6
1.3. 각 가상화 플랫폼에 대한 VIRT-WHO 구성 .....	6
2장. 가상 머신의 활성화 키 생성 .....	9
3장. 하이퍼바이저에 호스트 기반 서브스크립션 연결 .....	11
4장. VIRT-WHO 구성 생성 .....	12
5장. VIRT-WHO 구성 배포 .....	15
5.1. 하이퍼바이저에 VIRT-WHO 구성 배포 .....	15
5.2. SATELLITE SERVER에 VIRT-WHO 구성 배포 .....	16
5.3. 별도의 RED HAT ENTERPRISE LINUX 서버에 VIRT-WHO 구성 배포 .....	16
6장. 호스트 기반 서브스크립션을 사용하도록 가상 머신 등록 .....	18
부록 A. VIRT-WHO 문제 해결 .....	19
A.1. VIRT-WHO 구성 수정 .....	19
A.2. 기존 VIRT-WHO 구성 제거 .....	19
A.3. VIRT-WHO 문제 해결 방법 .....	19
A.4. VIRT-WHO 문제 해결 시나리오 .....	21



## RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견에 감사드립니다. 어떻게 개선할 수 있는지 알려주십시오.

Red Hat Jira에서 **문제 생성** 양식을 사용하여 피드백을 제공합니다. Jira 문제는 Red Hat Satellite Jira 프로젝트에서 생성되어 진행 상황을 추적할 수 있습니다.

### 사전 요구 사항

- [Red Hat 계정](#)을 등록했는지 확인합니다.

### 프로세스

1. 다음 링크를 클릭합니다. [문제 만들기](#). Jira가 로그인 오류를 표시하는 경우 양식으로 리디렉션된 후 로그인하고 계속 진행합니다.
2. **요약** 및 **설명** 필드를 작성합니다. **설명** 필드에 문서 URL, 장 또는 섹션 번호, 문제에 대한 자세한 설명을 포함합니다. 양식의 다른 필드를 수정하지 마십시오.
3. **생성**을 클릭합니다.

# 1장. 소개

다음 가상화 플랫폼에서 Red Hat Enterprise Linux 가상 머신에 호스트 기반 서브스크립션을 사용할 수 있습니다.

- Red Hat Virtualization
- Red Hat Enterprise Linux Virtualization(KVM)
- Red Hat OpenStack Platform
- VMware vSphere
- Microsoft Hyper-V
- Nutanix AHV

서브스크립션 모델은 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다. 대신 [Simple Content Access](#) 를 사용하는 것이 좋습니다.

## 1.1. 호스트 기반 서브스크립션

가상 머신은 물리적 서브스크립션에서 인타이틀먼트를 사용하는 대신 호스트 기반 서브스크립션을 사용할 수 있습니다. 호스트 기반 서브스크립션이 하이퍼바이저에 연결되어 있으며 가상 머신에 서브스크립션을 제공할 수 있습니다. 많은 호스트 기반 서브스크립션은 무제한 가상 머신에 대한 인타이틀먼트를 제공합니다.

가상 머신이 하이퍼바이저에서 서브스크립션을 상속할 수 있도록 하려면 `virt-who`를 설치하고 구성해야 합니다. `virt-who`는 가상화 플랫폼에 쿼리하고 하이퍼바이저 및 가상 머신 정보를 Red Hat Satellite에 보고합니다.

서브스크립션이 연결되어 있지 않은 활성화 키를 사용하여 가상 머신을 등록하고 `true` 로 설정된 자동 연결 및 호스트 기반 서브스크립션을 사용할 수 있는 경우 다음 동작 중 하나가 발생합니다.

- 가상 머신이 `virt-who`에 의해 보고되고 호스트 기반 서브스크립션이 하이퍼바이저에 연결된 경우 가상 머신은 하이퍼바이저의 서브스크립션을 상속합니다.
- 가상 시스템이 `virt-who`에 의해 보고되고 하이퍼바이저가 Satellite에 등록되어 있지만 호스트 기반 서브스크립션이 연결되어 있지 않은 경우 호스트 기반 서브스크립션이 하이퍼바이저에 연결되어 가상 머신에서 상속됩니다.
- `virt-who`에서 가상 머신 또는 해당 하이퍼바이저를 보고하지 않은 경우 Satellite는 가상 시스템에 최대 7일 동안 유효한 임시 서브스크립션을 부여합니다. `virt-who`가 업데이트된 정보를 보고한 후 Satellite에서 가상 시스템이 실행 중인 하이퍼바이저를 확인하고 가상 머신에 영구 서브스크립션을 연결할 수 있습니다.

자동 연결이 활성화되었지만 `virt-who`가 실행 중이 아니거나 호스트 기반 서브스크립션을 사용할 수 없는 경우 Satellite는 대신 물리적 서브스크립션을 가상 머신에 연결합니다. 이 경우 의도한 것보다 많은 인타이틀먼트를 사용할 수 있습니다.

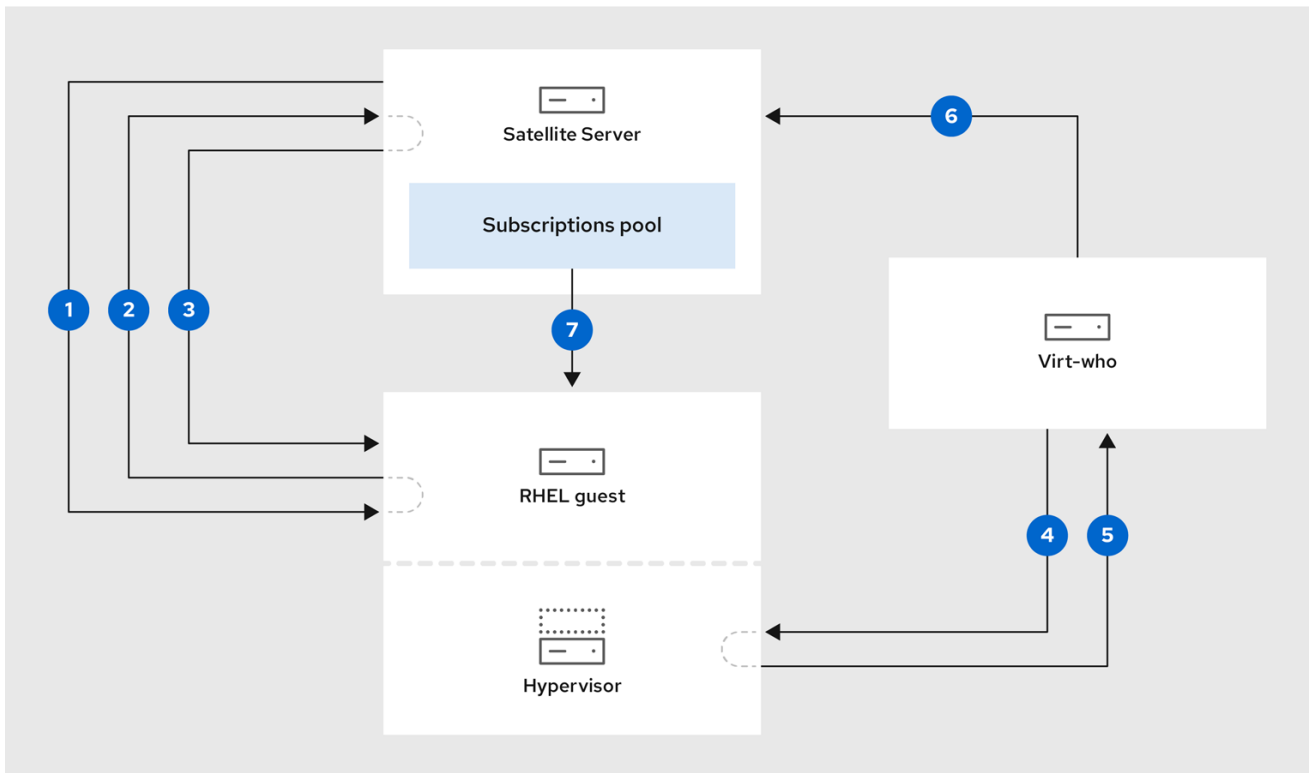
자동 연결이 활성화되지 않은 경우 가상 머신에서 호스트 기반 서브스크립션을 사용할 수 없습니다.

서브스크립션에 `virt-who`가 필요한지 확인하려면 Satellite 웹 UI에서 [콘텐츠 > 서브스크립션](#)으로 이동합니다. **Requires Virt-Who** 열에 눈금이 있는 경우 해당 서브스크립션을 사용하도록 `virt-who`를 구성해야 합니다.



## 가상 머신 서브스크립션 프로세스

이 다이어그램은 가상 머신이 아직 virt-who에 의해 보고되지 않은 경우 서브스크립션 워크플로를 보여줍니다.



278\_Satellite\_0922

- 1 Satellite는 가상 시스템을 프로비저닝합니다.
- 2 가상 시스템은 Satellite Server에서 서브스크립션을 요청합니다.
- 3 Satellite Server는 가상 시스템에 가상 시스템이 속한 하이퍼바이저를 결정하는 동안 최대 7일 동안 유효한 임시 서브스크립션을 가상 시스템에 부여합니다.
- 4 virt-who는 하이퍼바이저 또는 가상화 관리자에 연결하여 해당 가상 머신에 대한 정보를 요청합니다.
- 5 하이퍼바이저 또는 가상화 관리자는 각 UUID를 포함하여 가상 머신 목록을 virt-who로 반환합니다.
- 6 virt-who는 가상 시스템 목록과 하이퍼바이저를 Satellite Server에 보고합니다.
- 7 사용 가능한 권한이 충분한 경우 Satellite Server는 가상 시스템에 영구 서브스크립션을 연결합니다.

### 추가 리소스

Red Hat 서브스크립션 모델에 대한 자세한 내용은 [Red Hat 서브스크립션 관리 워크플로 소개](#)를 참조하십시오.

## 1.2. 구성 개요

가상 머신이 하이퍼바이저에서 서브스크립션을 상속할 수 있도록 하려면 다음 단계를 완료합니다.

### 사전 요구 사항

- 호스트 기반 서브스크립션이 포함된 서브스크립션 매니페스트를 Satellite Server로 가져옵니다. 자세한 내용은 [콘텐츠 관리에서 Red Hat 서브스크립션 매니페스트를 Satellite 서버로 가져오기](#)를 참조하십시오.
- 호스트 기반 서브스크립션에 사용할 모든 하이퍼바이저를 포괄할 수 있는 충분한 권한이 있는지 확인합니다.
- Microsoft Hyper-V를 사용하는 경우 하이퍼바이저에서 원격 관리를 활성화합니다.
- Nutanix AHV를 사용하는 경우 [Red Hat 지식 베이스에서 RHSM과 작동하도록 virt-who for Nutanix AHV를 구성하는 방법을 참조하십시오](#).
- 각 하이퍼바이저 또는 가상화 관리자에서 읽기 전용 액세스 권한과 만료되지 않은 암호를 사용하여 사용자를 생성합니다. virt-who는 이 계정을 사용하여 Satellite Server에 보고할 가상 시스템 목록을 검색합니다.
  - Red Hat 제품 및 Microsoft Hyper-V의 경우 Red Hat Enterprise Linux 가상 머신을 실행하는 각 하이퍼바이저에 virt-who 사용자를 만듭니다.
  - VMware vSphere의 경우 vCenter Server에 virt-who 사용자를 생성합니다. virt-who 사용자는 vCenter Data Center의 모든 오브젝트에 대한 읽기 전용 액세스 권한이 필요합니다.

### 절차 개요

1. [1.3절. "각 가상화 플랫폼에 대한 virt-who 구성"](#). 이 섹션의 표를 사용하여 가상화 플랫폼에 대해 virt-who를 구성하고 배포하는 방법을 계획하십시오.
2. [2장. 가상 머신의 활성화 키 생성](#). 자동 연결이 활성화되고 서브스크립션이 연결되지 않은 활성화 키를 생성합니다.
3. [3장. 하이퍼바이저에 호스트 기반 서브스크립션 연결](#). 사용하려는 모든 하이퍼바이저에 호스트 기반 서브스크립션을 연결합니다.
4. [4장. virt-who 구성 생성](#). 각 하이퍼바이저 또는 가상화 관리자에 대한 virt-who 구성을 생성합니다.
5. [5장. virt-who 구성 배포](#). Satellite에서 생성한 스크립트를 사용하여 virt-who 구성을 배포합니다.
6. [6장. 호스트 기반 서브스크립션을 사용하도록 가상 머신 등록](#). 자동 연결 활성화 키를 사용하여 가상 머신을 등록합니다.

## 1.3. 각 가상화 플랫폼에 대한 VIRT-WHO 구성

virt-who는 가상화 유형 및 하이퍼바이저 또는 가상화 관리자와 같은 세부 정보를 쿼리하는 파일을 사용하여 구성합니다. 지원되는 구성은 각 가상화 플랫폼에 따라 다릅니다.

### 일반적인 virt-who 구성 파일

이 예에서는 Satellite 웹 UI 또는 Hammer CLI를 사용하여 생성된 일반적인 virt-who 구성 파일을 보여줍니다.

```
[virt-who-config-1]
type=libvirt
hypervisor_id=hostname
owner=Default_Organization
env=Library
server=hypervisor1.example.com
username=virt_who_user
encrypted_password=$cr_password
rasm_hostname=satellite.example.com
rasm_username=virt_who_reporter_1
rasm_encrypted_password=$user_password
rasm_prefix=/rasm
```

유형 및 서버 값은 가상화 플랫폼에 따라 다릅니다. 다음 표에서는 자세히 설명합니다.

사용자 이름은 virt-who를 구성하기 전에 생성해야 하는 하이퍼바이저 또는 가상화 관리자의 읽기 전용 사용자를 나타냅니다. **rasm-username** 은 virt-who 보고만 Satellite Server에 대한 권한이 있는 자동으로 생성된 사용자를 나타냅니다.

### 각 가상화 플랫폼에 필요한 구성

이 표를 사용하여 virt-who 구성을 계획합니다.

지원되는 가상화 플랫폼	구성 파일에 지정된 유형	구성 파일에 지정된 서버	구성 파일이 배포된 서버
Red Hat Virtualization RHEL Virtualization(KVM) Red Hat OpenStack Platform	libvirt	하이퍼바이저(각 하이퍼 바이저마다 하나의 파일)	각 하이퍼바이저
VMware vSphere	esx	vCenter Server	Satellite Server, Capsule Server 또는 전용 RHEL 서버
Microsoft Hyper-V	hyperv	Hyper-V 하이퍼바이저 (하이퍼바이저당 하나의 파일)	Satellite Server, Capsule Server 또는 전용 RHEL 서버

### virt-who 구성 파일의 예

몇 가지 일반적인 하이퍼바이저 유형의 virt-who 구성 파일의 예가 표시됩니다.

OpenStack virt-who 구성 예

```
[root@compute-node]# cat /etc/virt-who.d/virt-who-config-1.conf
This configuration file is managed via the virt-who configure plugin
manual edits will be deleted.
[virt-who-config-1]
type=libvirt
hypervisor_id=hostname
owner=ORG
```

```

env=Library
server=qemu:///system <====
username=virt-who-user
encrypted_password=xxxxxxxxxx
rhsm_hostname=satellite.example.com
rhsm_username=virt_who_reporter_1
rhsm_encrypted_password=yyyyyyyyyyy
rhsm_prefix=/rhsm

```

KVM virt-who 구성 예

```

type=libvirt
hypervisor_id=hostname
owner=gss
env=Library
server=qemu+ssh://root@libvirt.example.com/system
username=root
encrypted_password=33di3ksskd
rhsm_hostname=satellite.example.com
rhsm_username=virt_who_reporter_2
rhsm_encrypted_password=23233dj3j3k
rhsm_prefix=/rhsm

```

VMware virt-who 구성 예

```

type=esx
hypervisor_id=hostname
owner=gss
env=Library
server=vcenter.example.com
username=username_vcenter@example.com
encrypted_password=33di3ksskd
rhsm_hostname=satellite.example.com
rhsm_username=virt_who_reporter_2
rhsm_encrypted_password=23233dj3j3k
rhsm_prefix=/rhsm

```



#### 중요

Cryo **stat** 및 **xen** 하이퍼바이저 유형은 지원되지 않습니다.

**kubervirt** 하이퍼바이저 유형은 기술 프리뷰로만 제공됩니다.

## 2장. 가상 머신의 활성화 키 생성

자동 연결이 활성화되고 서브스크립션이 연결되지 않은 활성화 키를 생성하려면 다음 절차를 사용하십시오. 이 활성화 키를 사용하여 가상 머신을 등록할 때 해당 하이퍼바이저에 호스트 기반 서브스크립션이 연결된 경우 가상 머신은 하이퍼바이저에서 서브스크립션을 상속합니다.



### 참고

이 절차는 Satellite에서 SCA(Simple Content Access)가 비활성화된 경우에만 유효합니다. SCA가 활성화된 경우 서브스크립션을 활성화 키에 연결할 필요가 없습니다. SCA는 새로 생성된 조직에 대해 기본적으로 활성화되어 있습니다. SCA에 대한 자세한 내용은 [Simple Content Access](#) 를 참조하십시오.

활성화 키에 대한 자세한 내용은 [콘텐츠 관리의 활성화 키 관리](#)를 참조하십시오.

### 프로세스

1. Satellite 웹 UI에서 [콘텐츠 > 활성화 키](#)로 이동하여 **Create Activation Key** 를 클릭합니다.
2. **이름** 필드에 활성화 키의 이름을 입력합니다.
3. **Unlimited Hosts** 확인란이 선택되어 있는지 확인합니다. 또는 제한을 설정하려면 **Unlimited hosts** 확인란을 지우고 **Limit** 필드에 활성화 키로 등록할 수 있는 최대 가상 머신 수를 입력합니다.
4. **설명** 필드에 활성화 키에 대한 설명을 입력합니다.
5. **환경** 목록에서 사용할 환경을 선택합니다.
6. **콘텐츠 뷰** 목록에서 사용할 콘텐츠 뷰를 선택합니다.
7. **저장**을 클릭합니다.
8. 활성화 키 창에 새 **활성화 키**가 표시되면 키 이름을 클릭합니다.
9. **서브스크립션 탭**을 클릭합니다.
10. **Auto-Attach** 가 **Yes** 로 설정되어 있는지 확인합니다. 그렇지 않은 경우 편집 아이콘을 클릭하고 확인란을 선택한 다음 **저장**을 클릭합니다.  
이 활성화 키에 서브스크립션을 추가하지 마십시오.

### CLI 사용자의 경우

1. 활성화 키를 생성합니다.

```
# hammer activation-key create \
--name "VM_Activation_Key" \
--unlimited-hosts \
--description "For VMs to inherit a VDC Subscription" \
--lifecycle-environment "Environment_Name" \
--content-view "CV_Name" \
--organization "My_Organization"
```

2. 활성화 키에서 자동 연결을 활성화합니다.

```
# hammer activation-key update --name "VM_Activation_Key" \  
--organization "My_Organization" --auto-attach true
```

## 3장. 하이퍼바이저에 호스트 기반 서브스크립션 연결

가상 데이터센터용 **Red Hat Enterprise Linux**와 같은 호스트 기반 서브스크립션을 **Red Hat Satellite**에 이미 등록된 하이퍼바이저에 연결하려면 다음 절차를 사용하십시오.



### 참고

이 절차는 Satellite에서 SCA(Simple Content Access)가 비활성화된 경우에만 유효합니다. SCA가 활성화된 경우 서브스크립션을 수동으로 연결할 필요가 없습니다. SCA는 새로 생성된 조직에 대해 기본적으로 활성화되어 있습니다. SCA에 대한 자세한 내용은 [Simple Content Access](#) 를 참조하십시오.

새 하이퍼바이저를 등록하려면 호스트 활성화 키에 호스트 기반 서브스크립션이 포함되어 있는지 확인한 다음 호스트 관리에서 [Satellite에 호스트 등록을 참조하십시오](#). virt-who를 구성하기 전에 하이퍼바이저를 등록해야 합니다.

### 사전 요구 사항

- 호스트 기반 서브스크립션이 포함된 서브스크립션 매니페스트를 Satellite Server로 가져옵니다.
- 호스트 기반 서브스크립션에 사용할 모든 하이퍼바이저를 포괄할 수 있는 충분한 권한이 있는지 확인합니다.

### 프로세스

1. Satellite 웹 UI에서 **호스트 > 콘텐츠 호스트로 이동합니다**.
2. **콘텐츠 호스트** 목록에서 서브스크립션을 연결할 각 하이퍼바이저의 이름 옆에 있는 확인란을 선택합니다.
3. **Select Action** 목록에서 **서브스크립션 관리**를 선택합니다.
4. Content Host Bulk Subscriptions 창에서 호스트 기반 서브스크립션을 선택한 다음 **Add Selected** 를 클릭합니다.

### CLI 사용자의 경우

1. Satellite Server에서 사용 가능한 서브스크립션을 나열하여 호스트 기반 서브스크립션의 ID를 찾습니다.

```
# hammer subscription list \
--organization-id organization_id
```

2. 호스트 기반 서브스크립션을 하이퍼바이저에 연결합니다.

```
# hammer host subscription attach \
--host host_name \
--subscription-id subscription_id
```

사용하려는 각 하이퍼바이저에 대해 다음 단계를 반복합니다.

## 4장. VIRT-WHO 구성 생성

virt-who 구성을 생성하려면 다음 절차를 사용하십시오. Red Hat 제품 및 Microsoft Hyper-V에는 Red Hat Enterprise Linux 가상 머신을 실행하는 각 하이퍼바이저에 대해 하나의 구성이 필요합니다. VMware vSphere에는 각 vCenter Server에 대해 하나의 구성이 필요합니다.

이 절차의 대부분의 값은 `/etc/virt-who.d/conf_name.conf` 파일에 추가되며 지정된 하이퍼바이저 또는 가상화 관리자에게만 적용됩니다. `interval`, `Enable debugging output`, `HTTP Proxy`, `Ignore Proxy` 는 `/etc/sysconfig/virt-who` 파일에 저장된 전역 값입니다. 글로벌 구성 값은 동일한 서버의 모든 virt-who 구성에 적용되며 해당 서버에 새 virt-who 구성이 배포될 때마다 덮어씁니다.

모든 virt-who 구성은 `virt_who_reporter_[id]` 사용자를 생성하고 virt **-who Reporter** 역할을 할당하여 virt-who reporting to Satellite Server에 대한 최소한의 권한을 제공합니다. 이 사용자는 수동으로 구성하거나 Satellite Server에 로그인하는 데 사용할 수 없습니다.

### 사전 요구 사항

- Red Hat 하이퍼바이저(Red Hat Enterprise Linux 또는 Red Hat Virtualization Host)에 대한 virt-who 구성을 생성하는 경우 하이퍼바이저를 Red Hat Satellite에 등록합니다.
- 하이퍼바이저 또는 가상화 관리자에 읽기 전용 virt-who 사용자를 생성합니다.
  - Red Hat 제품 및 Microsoft Hyper-V의 경우 Red Hat Enterprise Linux 가상 머신을 실행하는 각 하이퍼바이저에 virt-who 사용자를 만듭니다.
  - VMware vSphere의 경우 vCenter Server에 virt-who 사용자를 생성합니다. virt-who 사용자는 vCenter Data Center의 모든 오브젝트에 대한 읽기 전용 액세스 권한이 필요합니다.

### 프로세스

1. Satellite 웹 UI에서 **Infrastructure > Virt-who** 구성으로 이동합니다.
2. 구성 **생성**을 클릭합니다. 새 Virt-who Config 창에서 도움말 아이콘을 클릭하여 각 필드에 대한 자세한 정보를 확인할 수 있습니다.
3. 구성의 이름을 입력합니다.
4. Hypervisor 유형 목록에서 가상화 플랫폼을 선택합니다.
  - Red Hat Enterprise Linux Virtualization(KVM), Red Hat Virtualization 또는 Red Hat OpenStack Platform: libvirt
  - VMware vSphere: esx
  - Microsoft Hyper-V: hyperv
  - Cryostat 및 xen 하이퍼바이저 유형은 지원되지 않으며 kubevirt 하이퍼바이저 유형은 기술 프리뷰로만 제공됩니다.
5. Hypervisor Server 필드에 이 구성이 적용되는 서버의 FQDN 또는 IP 주소를 입력합니다. VMware vSphere의 경우 vCenter 서버의 FQDN 또는 IP 주소를 사용합니다. 다른 모든 제품의 경우 하이퍼바이저의 FQDN 또는 IP 주소를 사용합니다.
6. Hypervisor Username 필드에 하이퍼바이저 또는 가상화 관리자에서 생성한 virt-who 사용자의 이름을 입력합니다.



7. Hypervisor Password 필드에 virt-who 사용자의 암호를 입력합니다. 이 암호는 구성을 배포할 때 암호화됩니다. libvirt 유형을 사용할 때는 이 필드가 필요하지 않습니다.
8. Interval 목록에서 virt-who에서 신규 또는 업데이트된 가상 머신 정보를 요청하는 빈도를 선택합니다. 가상 머신에 최대 7일 동안 임시 서브스크립션이 부여되므로 빈번한 쿼리가 필요하지 않으므로 환경 크기에 맞는 간격을 선택할 수 있습니다. 24시간마다 대부분의 환경에 적합합니다.
9. Satellite 서버 FQDN 을 입력합니다.
10. Hypervisor ID 목록에서 호스트 이름 또는 UUID로 Satellite Server의 하이퍼바이저를 확인할지 여부를 선택합니다.
11. 필터링 옵션 선택:
  - 기본값은 무제한입니다. 이 구성에서 적용되는 모든 하이퍼바이저는 virt-who에서 쿼리합니다. 쿼리할 하이퍼바이저를 제한할 필요가 없는 경우 이 옵션을 사용합니다.
  - 화이트리스트 또는 블랙리스트 하이퍼바이저를 사용할 수 있습니다. 예를 들어 일부 하이퍼바이저가 Microsoft Windows Server 가상 머신만 실행하는 경우 해당 하이퍼바이저를 virt-who에서 보고할 필요가 없습니다.
12. 화이트리스트 또는 블랙리스트: 필터 호스트 또는 제외 호스트 필드에 선택한 하이퍼바이저 ID 에 따라 쉼표로 구분된 하이퍼바이저 목록을 입력합니다. 예를 들어 하이퍼바이저가 호스트 이름으로 식별되는 경우 호스트 이름으로 포함되거나 제외되어야 합니다. 하이퍼바이저 이름에 특수 문자가 포함된 경우 따옴표로 묶습니다. 와일드카드 및 정규식이 지원됩니다. 정규식을 사용할 때는 백슬래시를 이스케이프해야 합니다.
13. esx 유형 및 화이트리스트 또는 블랙리스트: Filter host parents 또는 Exclude host parent 필드를 선택한 경우 쉼표로 구분된 클러스터 목록을 입력합니다. 허용 목록 클러스터의 하이퍼바이저는 virt-who에 의해 보고됩니다. 블랙리스트에 지정된 클러스터의 하이퍼바이저는 virt-who에 의해 보고되지 않습니다. 클러스터 이름에 특수 문자가 포함된 경우 따옴표로 묶습니다. 와일드카드 및 정규식이 지원됩니다. 정규식을 사용할 때는 백슬래시를 이스케이프해야 합니다.
14. 선택 사항: 문제 해결을 위해 디버깅 출력이 필요한 경우 디버깅 출력 사용확인란을 선택합니다.
15. 선택 사항: virt-who가 배포된 서버와 하이퍼바이저 또는 가상화 관리자 간의 통신에 사용할 HTTP 프록시를 입력합니다. 예: `http://proxy.example.com:3128`. 프록시를 사용하지 않으려면 이 필드를 비워 둡니다. 프록시 무시 필드에 \*를 입력하는 것과 동일한 결과가 발생합니다.
16. 선택 사항: 프록시 무시 필드에 호스트 이름, IP 주소 또는 도메인의 쉼표로 구분된 목록을 입력하여 기존 프록시 설정을 바이패스합니다.
17. Submit 을 클릭합니다.

#### CLI 사용자의 경우

- Satellite Server에서 **hammer virt-who-config create** 명령을 입력합니다. 옵션에 대한 자세한 내용은 **hammer virt-who-config create --help** 를 입력합니다. 이 예에서는 Red Hat Enterprise Linux 하이퍼바이저에 대한 virt-who 구성을 생성합니다.

```
# hammer virt-who-config create \
--name rhel.example.com \
--organization "Example Company" \
--interval 720 \ 1
--filtering-mode none \ 2
```

```

--hypervisor-id hostname \ 3
--hypervisor-type libvirt \ 4
--hypervisor-server rhel.example.com \ 5
--hypervisor-username virt-who \ 6
--proxy 'http://proxy.example.com:3128' \ 7
--satellite-url satellite.example.com

```

- 1 가상화 플랫폼을 쿼리하는 빈도 및 분 내에 **virt-who**를 선택합니다. 가상 머신에 최대 7일 동안 임시 서브스크립션이 부여되므로 빈번한 쿼리가 필요하지 않으므로 환경 크기에 맞는 간격을 선택할 수 있습니다. 하루에 한 번 (1440)은 대부분의 환경에 적합합니다.
- 2 하이퍼바이저를 필터링할지 여부를 선택합니다. 쿼리할 하이퍼바이저를 제한할 필요가 없는 경우 **none**을 사용합니다. 또는 하이퍼바이저를 허용 목록에 추가하거나 블랙리스트로 지정할 수 있습니다. 예를 들어 일부 하이퍼바이저가 **Microsoft Windows Server** 가상 머신만 실행하는 경우 해당 하이퍼바이저를 **virt-who**에서 보고할 필요가 없습니다.
- 3 이 구성이 적용되는 하이퍼바이저(또는 하이퍼바이저)를 식별하는 방법을 선택합니다. 호스트 이름을 사용하여 **Satellite** 웹 UI에서 의미 있는 호스트 이름을 제공합니다. 또는 하이퍼바이저의 이름이 변경된 경우 **uuid**를 사용하여 중복을 방지할 수 있습니다.**hwuuid**는 이 구성이 개별 하이퍼바이저 대신 가상화 관리자에게 적용되는 경우에만 사용할 수 있으며, **virt-who**가 서브스크립션 관리자의 중복 항목을 유발하므로 **virt-who**가 실행을 시작한 후 다른 옵션으로 변경하지 않아야 합니다.
- 4 가상화 플랫폼에 지원되는 유형을 지정합니다.
  - Red Hat Enterprise Linux Virtualization(KVM), Red Hat Virtualization 또는 Red Hat OpenStack Platform: **libvirt**
  - VMware vSphere: **esx**
  - Microsoft Hyper-V: **hyperv**
  - Cryostat 및 **xen** 하이퍼바이저 유형은 지원되지 않으며 **kubevirt** 하이퍼바이저 유형은 기술 프리뷰로만 제공됩니다.
- 5 이 구성이 적용되는 서버의 **FQDN** 또는 **IP** 주소를 지정합니다. VMware vSphere의 경우 vCenter 서버의 **FQDN** 또는 **IP** 주소를 사용합니다. 다른 모든 제품의 경우 하이퍼바이저의 **FQDN** 또는 **IP** 주소를 사용합니다.
- 6 하이퍼바이저 또는 가상화 관리자에서 생성한 읽기 전용 **virt-who** 사용자를 지정합니다. **libvirt** 유형을 사용할 때 암호를 지정할 필요가 없습니다. 다른 유형의 경우 **--hypervisor-password** 옵션을 사용하여 **virt-who** 사용자 암호를 지정해야 합니다.
- 7 선택 사항: **virt-who**가 배포된 서버와 하이퍼바이저 또는 가상화 관리자 간의 통신에 사용할 **HTTP** 프로кси를 입력합니다.

## 5장. VIRT-WHO 구성 배포

virt-who 구성을 생성한 후 Satellite는 배포 프로세스를 자동화하는 스크립트를 제공합니다. 이 스크립트는 virt-who를 설치하고 개별 및 글로벌 virt-who 구성 파일을 생성합니다.

Red Hat 제품의 경우 파일에 지정된 하이퍼바이저에 각 구성 파일을 배포해야 합니다. 다른 제품의 경우 Satellite Server, Capsule Server 또는 virt-who 실행 전용 별도의 Red Hat Enterprise Linux 서버에 구성 파일을 배포해야 합니다.

- 하이퍼바이저 또는 Capsule Server에 파일을 배포하려면 [5.1절. "하이퍼바이저에 virt-who 구성 배포"](#) 을 참조하십시오.
- Satellite Server에 파일을 배포하려면 [5.2절. "Satellite Server에 virt-who 구성 배포"](#) 을 참조하십시오.
- 별도의 Red Hat Enterprise Linux 서버에 파일을 배포하려면 [5.3절. "별도의 Red Hat Enterprise Linux 서버에 virt-who 구성 배포"](#) 을 참조하십시오.

### 5.1. 하이퍼바이저에 VIRT-WHO 구성 배포

파일에서 지정한 Red Hat 하이퍼바이저에 virt-who 구성을 배포하려면 다음 절차를 사용하십시오. 전역 값은 이 하이퍼바이저에만 적용됩니다.

이 절차를 사용하여 Capsule Server에 vCenter 또는 Hyper-V virt-who 구성을 배포할 수도 있습니다. 글로벌 구성 값은 동일한 Capsule Server의 모든 virt-who 구성에 적용되며 새 virt-who 구성이 배포될 때마다 덮어씁니다.

#### 사전 요구 사항

- 하이퍼바이저를 Red Hat Satellite에 등록합니다.
- RHVH(Red Hat Virtualization Host)를 사용하는 경우 최소 virt-who 버전을 사용할 수 있도록 최신 버전으로 업데이트합니다. virt-who는 기본적으로 RHVH에서 사용할 수 있지만 `rhel-7-server-rhvh-4-rpms` 리포지토리에서 개별적으로 업데이트할 수 없습니다.
- 하이퍼바이저에 읽기 전용 virt-who 사용자를 생성합니다.
- 가상화 플랫폼에 대한 virt-who 구성을 생성합니다.

#### 프로세스

1. Satellite 웹 UI에서 Infrastructure > Virt-who 구성으로 이동합니다.
2. virt-who 구성의 이름을 클릭합니다.
3. Deploy 탭을 클릭합니다.
4. 구성 스크립트에서 스크립트 다운로드를 클릭합니다.
5. 스크립트를 하이퍼바이저에 복사합니다.

```
# scp deploy_virt_who_config_1.sh root@hypervisor.example.com:
```

6. 배포 스크립트를 실행 가능하게 설정하고 실행합니다.

```
# chmod +x deploy_virt_who_config_1.sh
# sh deploy_virt_who_config_1.sh
```

7. 배포가 완료되면 스크립트를 삭제합니다.

```
# rm deploy_virt_who_config_1
```

## 5.2. SATELLITE SERVER에 VIRT-WHO 구성 배포

Satellite Server에 vCenter 또는 Hyper-V virt-who 구성을 배포하려면 다음 절차를 사용하십시오.

글로벌 구성 값은 Satellite Server의 모든 virt-who 구성에 적용되며 새 virt-who 구성이 배포될 때마다 덮어씁니다.

사전 요구 사항

- 하이퍼바이저 또는 가상화 관리자에 읽기 전용 virt-who 사용자를 생성합니다.
- Hyper-V virt-who 구성을 배포하는 경우 Hyper-V 하이퍼바이저에서 원격 관리를 활성화합니다.
- 가상화 플랫폼에 대한 virt-who 구성을 생성합니다.

프로세스

1. Satellite 웹 UI에서 Infrastructure >Virt-who 구성으로 이동합니다.
2. virt-who 구성의 이름을 클릭합니다.
3. Hammer 명령에서 클립보드에 복사를 클릭합니다.
4. Satellite Server에서 Hammer 명령을 터미널에 붙여넣습니다.

## 5.3. 별도의 RED HAT ENTERPRISE LINUX 서버에 VIRT-WHO 구성 배포

전용 Red Hat Enterprise Linux 7 서버에 vCenter 또는 Hyper-V virt-who 구성을 배포하려면 다음 절차를 사용하십시오. 서버는 물리적 또는 가상일 수 있습니다.

글로벌 구성 값은 이 서버의 모든 virt-who 구성에 적용되며 새 virt-who 구성이 배포될 때마다 덮어씁니다.

사전 요구 사항

- 하이퍼바이저 또는 가상화 관리자에 읽기 전용 virt-who 사용자를 생성합니다.
- Hyper-V virt-who 구성을 배포하는 경우 Hyper-V 하이퍼바이저에서 원격 관리를 활성화합니다.
- 가상화 플랫폼에 대한 virt-who 구성을 생성합니다.

프로세스

1. Red Hat Enterprise Linux 서버에서 Satellite 서버의 CA 인증서를 설치합니다.

```
# rpm -ivh http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

2. Red Hat Enterprise Linux 서버를 Satellite Server에 등록합니다.

```
# subscription-manager register \
--org=organization_label \
--auto-attach
```

3. virt-who와 Satellite Server 간의 통신을 위해 네트워크 포트를 엽니다.

```
# firewall-cmd --add-port="443/tcp"
# firewall-cmd --add-port="443/tcp" --permanent
```

4. virt-who와 각 하이퍼바이저 또는 가상화 관리자 간의 통신을 위해 네트워크 포트를 엽니다.

- VMware vCenter: TCP 포트 443
- Microsoft Hyper-V: TCP 포트 5985

5. Satellite 웹 UI에서 Infrastructure >Virt-who 구성으로 이동합니다.

6. virt-who 구성 파일의 이름을 클릭합니다.

7. Deploy 탭을 클릭합니다.

8. 구성 스크립트에서 스크립트 다운로드를 클릭합니다.

9. 스크립트를 Red Hat Enterprise Linux 서버에 복사합니다.

```
# scp deploy_virt_who_config_1.sh root@rhel.example.com:
```

10. 배포 스크립트를 실행 가능하게 설정하고 실행합니다.

```
# chmod +x deploy_virt_who_config_1.sh
# sh deploy_virt_who_config_1.sh
```

11. 배포가 완료되면 스크립트를 삭제합니다.

```
# rm deploy_virt_who_config_1
```

## 6장. 호스트 기반 서브스크립션을 사용하도록 가상 머신 등록

자동 연결 활성화 키를 사용하여 하이퍼바이저에서 서브스크립션을 상속하도록 가상 머신을 등록합니다.



### 참고

이 절차는 Satellite에서 SCA(Simple Content Access)가 비활성화된 경우에만 유효합니다. SCA가 활성화된 경우 서브스크립션을 활성화 키에 연결할 필요가 없습니다. 대신 호스트 관리의 호스트 등록 절차에 따라 가상 시스템을 등록합니다. SCA는 새로 생성된 조직에 대해 기본적으로 활성화되어 있습니다. SCA에 대한 자세한 내용은 [Simple Content Access](#)를 참조하십시오.

활성화 키에 대한 자세한 내용은 [콘텐츠 관리의 활성화 키 관리](#)를 참조하십시오.

### 사전 요구 사항

- 자동 연결이 활성화되고 서브스크립션이 연결되지 않은 활성화 키를 생성합니다.
- 가상 머신의 하이퍼바이저에 호스트 기반 서브스크립션을 연결합니다.
- 가상 머신의 하이퍼바이저를 쿼리하는 virt-who 구성을 생성하고 배포합니다.

### 프로세스

1. Satellite Server에 등록하도록 가상 머신을 구성합니다.

```
# yum install http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

2. 자동 연결 활성화 키를 사용하여 가상 머신을 등록합니다.

```
# subscription-manager register --org=organization_label \
--activationkey=activation_key_name
```

가상 시스템은 임시 서브스크립션을 수신하는 동안 Satellite Server는 가상 시스템이 실행 중인 하이퍼바이저에 대한 정보를 제공할 때까지 기다립니다. virt-who가 이 정보를 제공한 후 Satellite Server는 영구 서브스크립션을 가상 머신에 연결합니다.

가상 머신이 virt-who에 보고하고 호스트 기반 서브스크립션이 연결된 다른 하이퍼바이저로 마이그레이션 되는 경우 가상 머신은 해당 하이퍼바이저의 서브스크립션을 상속합니다. 하이퍼바이저에 호스트 기반 서브스크립션이 연결되어 있지 않은 경우 Satellite Server는 호스트 기반 서브스크립션을 하이퍼바이저에 자동으로 연결하고 가상 시스템은 해당 서브스크립션을 상속합니다.

가상 머신이 virt-who에 보고하지 않는 다른 하이퍼바이저로 마이그레이션하거나 하이퍼바이저에 연결할 수 있는 서브스크립션이 충분하지 않은 경우 가상 머신은 서브스크립션을 상속할 수 없습니다. 이를 방지하기 위해 가상 머신에서 모든 하이퍼바이저를 마이그레이션하여 호스트 기반 서브스크립션을 연결하고 virt-who에 보고할 수 있는지 확인하거나 가상 시스템의 특정 하이퍼바이저로 마이그레이션을 제한합니다.

## 부록 A. VIRT-WHO 문제 해결

### A.1. VIRT-WHO 구성 수정

Satellite 웹 UI 또는 Hammer CLI를 사용하여 기존 virt-who 구성을 수정할 수 있습니다. 예를 들어 virt-who가 실행되는 빈도를 변경해야 하는 경우 virt-who 구성을 업데이트하고 다시 배포해야 합니다.

#### 프로세스

1. Satellite 웹 UI에서 Infrastructure >Virt-who 구성으로 이동합니다.
2. 수정할 virt-who 구성을 찾은 다음 Actions 열에서 편집을 클릭합니다.
3. 변경할 필드를 편집합니다.
4. Submit 을 클릭합니다.
5. 수정된 virt-who 구성을 재배포합니다.

#### CLI 사용자의 경우

1. Satellite Server에서 수정할 구성 이름 및 변경할 옵션의 새 값을 지정하여 **hammer virt-who-config update** 명령을 입력합니다. 구성 이름을 변경하려면 **--new-name** 옵션을 사용해야 합니다.

```
# hammer virt-who-config update \
--name current_name \
--new-name new_name \
--interval 1440
```

2. 수정된 virt-who 구성을 재배포합니다.

### A.2. 기존 VIRT-WHO 구성 제거

기존 virt-who 구성을 제거하려면 먼저 Satellite 웹 UI에서 구성 항목을 제거한 다음 구성이 배포된 호스트의 파일 시스템에서 구성 파일을 제거해야 합니다.

#### 프로세스

1. Satellite 웹 UI에서 Infrastructure >Virt-who 구성으로 이동합니다.
2. 제거할 구성의 작업 목록에서 삭제 를 선택합니다.
3. virt-who 구성을 제거하려는 호스트에서 구성 파일을 제거합니다.

```
# rm /etc/virt-who.d/conf_name.conf
```

### A.3. VIRT-WHO 문제 해결 방법

#### virt-who 상태 확인

Satellite 웹 UI에서 virt-who의 상태를 확인하려면 Infrastructure >Virt-who 구성으로 이동하여 각 virt-who 인스턴스의 Status 열을 확인합니다. OK 상태는 virt-who가 Satellite 서버에 성공적으로 연결하고 각 하이퍼바이저에서 관리하는 가상 시스템을 보고 있음을 나타냅니다.

CLI를 사용하여 모든 virt-who 인스턴스의 상태를 나열하려면 Satellite Server에서 다음 명령을 입력합니다.

```
# hammer virt-who-config list
```

명령의 출력에는 각 virt-who 인스턴스가 Satellite Server에 보고되는 날짜와 시간이 포함됩니다.

### 디버그 로깅

virt-who가 기본적으로 모든 활동을 로깅하는 `/var/log/rhsm/rhsm.log` 파일을 확인합니다.

보다 자세한 로깅을 활성화하려면 virt-who 구성을 수정합니다.

- Satellite 웹 UI에서 디버깅 출력 활성화 확인란을 선택합니다.
- Hammer CLI에서 `--debug true` 옵션을 추가합니다.

변경 사항을 적용하려면 구성을 재배포합니다.

기본 문제가 해결되면 virt-who 구성을 수정하여 디버깅을 비활성화한 다음 구성을 다시 배포합니다.

### 설정 옵션 테스트

필요에 따라 변경을 수행하고 결과를 테스트하십시오. virt-who는 구성 파일, 인증 정보 및 가상화 플랫폼 연결을 테스트하는 데 도움이 되는 두 가지 옵션을 제공합니다.

- `virt-who --one-shot` 명령은 구성 파일을 읽고 가상 시스템 목록을 검색한 다음 Satellite 서버로 보낸 다음 즉시 종료합니다.
- `virt-who --print` 명령은 구성 파일을 읽고 가상 시스템 목록을 출력하지만 Satellite 서버로 보내지는 않습니다.

예상되는 출력은 JSON 형식의 하이퍼바이저와 해당 가상 머신 목록입니다. 다음은 VMware vSphere 인스턴스에서 추출한 것입니다. 모든 하이퍼바이저의 출력은 동일한 구조를 따릅니다.

```
{
  "guestId": "422f24ed-71f1-8ddf-de53-86da7900df12",
  "state": 5,
  "attributes": {
    "active": 0,
    "virtWhoType": "esx",
    "hypervisorType": "vmware"
  }
},
```

### 여러 virt-who 구성 파일을 사용할 때 문제 식별

한 서버에 여러 개의 virt-who 구성 파일이 있는 경우 각 파일이 이동한 후 테스트하면서 한 번에 하나의 파일을 다른 디렉터리로 이동합니다. 문제가 더 이상 발생하지 않으면 원인은 가장 최근에 이동된 파일과 연결됩니다. 문제를 해결한 후 virt-who 구성 파일을 원래 위치로 반환합니다.

또는 `--config` 옵션을 사용하여 해당 위치를 지정하여 개별 파일을 이동한 후 테스트할 수 있습니다. 예를 들면 다음과 같습니다.

```
# virt-who --debug --one-shot --config /tmp/conf_name.conf
```



## 중복 하이퍼바이저 확인

중복 하이퍼바이저는 서브스크립션 및 인타이틀먼트 오류가 발생할 수 있습니다. 다음 명령을 입력하여 중복 하이퍼바이저를 확인합니다.

```
# systemctl stop virt-who
# virt-who -op >/tmp/virt-who.json
# systemctl start virt-who
# cat /tmp/virt-who.json | json_reformat | grep name | sort | uniq -c | sort -nr | head -n10
3  "name": "localhost"
1  "name": "rhel1.example.com"
1  "name": "rhel2.example.com"
1  "name": "rhel3.example.com"
1  "name": "rhel4.example.com"
1  "name": "rhvh1.example.com"
1  "name": "rhvh2.example.com"
1  "name": "rhvh3.example.com"
1  "name": "rhvh4.example.com"
1  "name": "rhvh5.example.com"
```

이 예에서 세 개의 하이퍼바이저는 동일한 FQDN(**localhost**)을 가지며 고유한 FQDN을 사용하도록 수정해야 합니다.

## 중복 가상 머신 식별

다음 명령을 입력하여 중복 가상 머신을 확인합니다.

```
# systemctl stop virt-who
# virt-who -op >/tmp/virt-who.json
# systemctl start virt-who
# cat /tmp/virt-who.json | json_reformat | grep "guestId" | sort | uniq -c | sort -nr | head -n10
```

## 하이퍼바이저 수 확인

다음 명령을 입력하여 현재 보고하는 하이퍼바이저 **virt-who** 수를 확인합니다.

```
# systemctl stop virt-who
# virt-who -op >/tmp/virt-who.json
# systemctl start virt-who
# cat /tmp/virt-who.json | json_reformat | grep name | sort | uniq -c | wc -l
```

## 가상 머신 수 확인

다음 명령을 입력하여 **virt-who** 현재 보고하는 가상 머신의 수를 확인합니다.

```
# systemctl stop virt-who
# virt-who -op >/tmp/virt-who.json
# systemctl start virt-who
# cat /tmp/virt-who.json | json_reformat | grep "guestId" | sort | uniq -c | wc -l
```

## A.4. VIRT-WHO 문제 해결 시나리오

**virt-who** 가상화 플랫폼에 연결하지 못했습니다

**virt-who** 가 하이퍼바이저 또는 가상화 관리자에 연결하지 못하는 경우 Red Hat Subscription Manager 로

그 파일 `/var/log/rhsm/rhsm.log` 를 확인하십시오. **No route to host** 라는 메시지가 표시되면 하이퍼바이저가 잘못된 포트에서 수신 대기 중일 수 있습니다. 이 경우 `virt-who` 구성을 수정하고 올바른 포트 번호를 Hypervisor Server 값에 추가합니다.

`virt-who` 구성을 수정한 후 다시 배포해야 합니다.

**virt-who 로컬 네트워크의 HTTP 프록시를 통해 가상화 플랫폼에 연결하지 못했습니다**

`virt-who` 가 HTTP 프록시를 통해 하이퍼바이저 또는 가상화 관리자에 연결할 수 없는 경우 로컬 트래픽이 통과할 수 있도록 프록시를 구성하거나 프록시를 사용하지 않도록 `virt-who` 구성을 수정합니다.

`virt-who` 구성을 수정한 후 다시 배포해야 합니다.

**virt-who failed to report the host-guest mapping to Red Hat Satellite Server**

`virt-who`는 다음과 같은 상황에서 Red Hat Satellite 서버에 대한 host-guest 매핑을 다시 보고하지 않습니다.

1. `virt-who`는 Red Hat Satellite 서버에 구성 및 배포됩니다.
2. Red Hat Satellite 서버의 `rhsm.conf` 파일은 프록시 서버를 사용하여 `subscription.rhsm.redhat.com` 및 `cdn.redhat.com`과 통신하도록 구성됩니다.
3. `/etc/sysconfig/virt-who` 의 `no_proxy=*` 설정은 `subscription-manager`에 의해 무시되고 `virt-who`는 프록시 서버를 통해 Satellite 서버에 다시 연결하려고 하지만 실패합니다.

이 경우 `/etc/rhsm/rhsm.conf` 파일에 다음 매개 변수를 추가합니다.

```
no_proxy = satellite.example.com
```