



# Red Hat Subscription Management 2023

在红帽订阅管理中配置虚拟机订阅

使用 virt-who 管理基于主机的订阅



# Red Hat Subscription Management 2023 在红帽订阅管理中配置虚拟机订阅

---

使用 virt-who 管理基于主机的订阅

Red Hat Subscription Management Documentation Team  
rhsm-docs@redhat.com

## 法律通告

Copyright © 2023 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.

## 摘要

本指南提供有关在红帽订阅管理中准备订阅、配置 virt-who 和注册虚拟机的信息，以便它们从 hypervisor 中继承订阅。

---

# 目录

<b>第 1 章 简介</b> .....	<b>3</b>
1.1. 基于主机的订阅	3
1.2. 配置概述	4
1.3. 每个虚拟化平台的 VIRT-WHO 配置	5
1.4. VIRT-WHO 常规配置	6
<b>第 2 章 将基于主机的订阅附加到虚拟机监控程序</b> .....	<b>9</b>
<b>第 3 章 准备 VIRT-WHO 主机</b> .....	<b>11</b>
<b>第 4 章 配置 VIRT-WHO</b> .....	<b>13</b>
4.1. 在 RED HAT HYPERVISOR 上安装和配置 VIRT-WHO	13
4.2. 配置 VIRT-WHO 以连接到 VMWARE VCENTER	15
4.3. 配置 VIRT-WHO 以连接到 MICROSOFT HYPER-V	17
4.4. 配置 VIRT-WHO 以连接到 OPENSIFT VIRTUALIZATION	19
<b>第 5 章 注册虚拟机以使用基于主机的订阅</b> .....	<b>23</b>
<b>第 6 章 VIRT-WHO 故障排除方法</b> .....	<b>25</b>
<b>第 7 章 VIRT-WHO 故障排除场景</b> .....	<b>30</b>



# 第 1 章 简介

您可以在以下虚拟化平台中为 Red Hat Enterprise Linux 虚拟机使用基于主机的订阅：

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

## 1.1. 基于主机的订阅

虚拟机可以使用基于主机的订阅，而不必使用物理订阅的权利。基于主机的订阅附加到虚拟机监控程序中，并授权它为其虚拟机提供订阅。许多基于主机的订阅都为无限虚拟机提供权利。

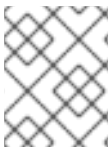
要允许虚拟机从 hypervisor 中继承订阅，您必须安装并配置 virt-who。virt-who 查询虚拟化平台，并将 hypervisor 和虚拟机信息报告给红帽订阅管理。

当虚拟机注册启用了自动附加功能，且有足够的基于主机的订阅可用时，会出现以下行为之一：

- 如果 virt-who 报告了虚拟机，并且基于主机的订阅已附加到 hypervisor，虚拟机会从 hypervisor 中继承订阅。
- 如果 virt-who 报告了虚拟机，并且管理程序已注册到 Subscription Management，但没有附加基于主机的订阅，则基于主机的订阅将附加到 hypervisor 中，并由虚拟机继承。
- 如果 virt-who 没有报告虚拟机或其管理程序，则 Subscription Management 为虚拟机授予临时订阅，最多有效 7 天。在 virt-who 报告更新的信息后，订阅管理可以决定虚拟机正在运行的管理程序，并将永久订阅附加到虚拟机。

如果启用了自动附加，但 virt-who 没有运行，或者没有可用的基于主机的订阅，则订阅管理会将物理订阅附加到虚拟机，这可能会消耗比预期更多的权利。

如果没有启用自动附加，虚拟机无法使用基于主机的订阅。



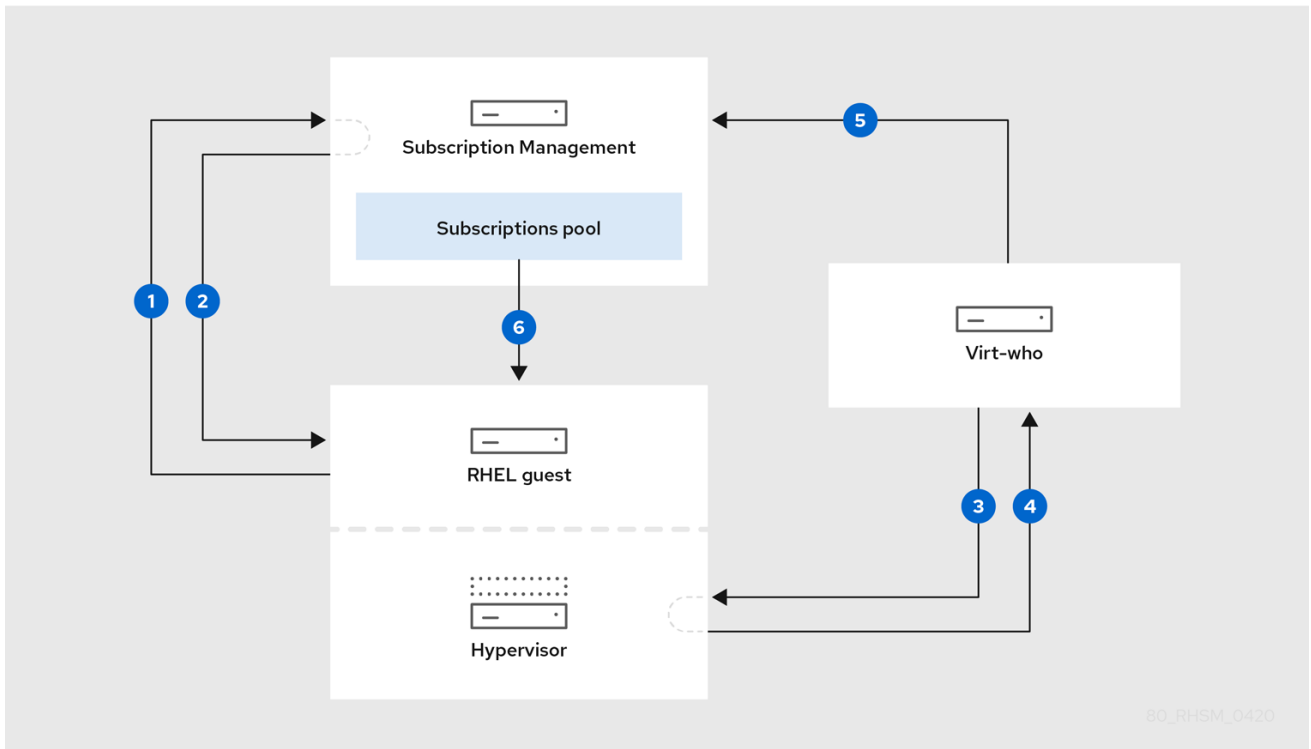
### 注意

系统目的 [附加组件](#) 对 Red Hat Enterprise Linux 8.0、8.1 和 8.2 中的自动附加功能没有影响。

要查看订阅是否需要 virt-who，登录到客户门户网站 <https://access.redhat.com>，进入 **Subscriptions > Subscription Utilization**，然后选择订阅。如果 SKU 详情中显示 "Virt-Who: Required"，您必须配置 virt-who 才能使用该订阅。

### 虚拟机订阅过程

当虚拟机尚未由 virt-who 报告时，此图显示了订阅工作流：



80\_RHSM\_0420

- 1 虚拟机从 Subscription Management 请求订阅。
- 2 订阅管理为虚拟机授予一个临时订阅，在 7 天内有效，同时决定虚拟机所属的虚拟机监控程序。
- 3 virt-who 连接到虚拟机监控程序或虚拟化管理器，并请求有关其虚拟机的信息。
- 4 管理程序或虚拟化管理器将其虚拟机列表返回到 virt-who，包括每个 UUID。
- 5 virt-who 将虚拟机列表及其 hypervisor 报告为 Subscription Management。
- 6 如果有足够的权利，订阅管理会将永久订阅附加到虚拟机。

### 其他资源

有关红帽订阅模型的更多信息，请参阅 [红帽订阅管理工作流简介](#)。

## 1.2. 配置概述

要允许虚拟机从 hypervisor 中继承订阅，请完成以下步骤：

### 先决条件

- 确保您有足够的权利来基于主机的订阅，以覆盖您计划使用的所有管理程序。
- 对于 Microsoft Hyper-V，在运行 Red Hat Enterprise Linux 虚拟机的每个 hypervisor 上创建一个只读 virt-who 用户。



- 对于 VMware vSphere，在 vCenter 服务器上创建带有非过期密码的只读 virt-who 用户。virt-who 用户需要至少对 vCenter Data Center 中所有对象的只读访问权限。
- 对于 OpenShift Virtualization，创建一个服务帐户，并使用 OpenShift 集群 master 中的 admin 角色授予它，virt-who 需要服务帐户令牌来连接 OpenShift 集群。

## 流程概述

1. [第 1.3 节 “每个虚拟化平台的 virt-who 配置”](#)。使用本节中的表来计划如何为您的虚拟化平台安装和配置 virt-who。
2. [第 2 章 将基于主机的订阅附加到虚拟机监控程序](#)。将基于主机的订阅附加到您计划使用的所有 hypervisor。
3. [第 4 章 配置 virt-who](#)。创建单独的配置文件，将 virt-who 连接到您的虚拟机监控程序。
4. [第 5 章 注册虚拟机以使用基于主机的订阅](#)。注册虚拟机并启用自动附加。

## 1.3. 每个虚拟化平台的 VIRT-WHO 配置

virt-who 使用指定虚拟化类型和虚拟机监控程序或虚拟化管理等详情的文件进行配置。每个虚拟化平台的支持的配置都有所不同。

- 单个配置文件存储在 `/etc/virt-who.d/` 目录中。您必须为每个虚拟机监控程序或虚拟化管理器创建单独的配置文件。

### virt-who 配置文件示例

本例展示了 Microsoft Hyper-V hypervisor 的独立 virt-who 配置文件：

```
[hypervisor1]
type=hyperv
server=hypervisor1.example.com
username=virt_who_user
encrypted_password=bd257f93d@482B76e6390cc54aec1a4d
hypervisor_id=hostname
owner=1234567
```

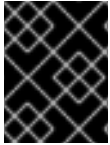
**type** 和 **server** 值取决于虚拟化平台。下表提供了更详细的信息。

**用户名** 指的是 Microsoft Hyper-V 或 VMware vCenter 上的只读用户，您必须在配置 virt-who 前创建它。virt-who 使用此帐户检索虚拟机列表。您不需要红帽虚拟机监控程序的专用 virt-who 用户。

### 每个虚拟化平台所需的配置

使用此表规划 virt-who 配置：

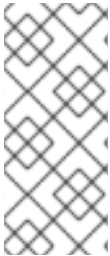
支持的虚拟化平台	配置文件中指定的类型	配置文件中指定的服务器	安装 virt-who 的服务器
Red Hat Virtualization RHEL Virtualization (KVM) Red Hat OpenStack Platform	libvirt	不是必需的	每个 hypervisor
VMware vSphere	esx	vCenter Server	专用的 RHEL 服务器
Microsoft Hyper-V	hyperv	虚拟机监控程序 (Hypervisor)	专用的 RHEL 服务器
OpenShift Virtualization	kubevirt	OpenShift Cluster Master	专用的 RHEL 服务器



### 重要

不支持 **xen** 和 xen 管理程序类型。

## 1.4. VIRT-WHO 常规配置



### 注意

下一个主发行版本中不支持 `/etc/sysconfig/virt-who`，全局配置文件将被 `/etc/virt-who.conf`。（例如：`VIRTWHO_DEBUG`，`VIRTWHO_ONE_SHOT`，`VIRTWHO_INTERVAL`，`HTTPS_PROXY`，`NO_PROXY`）。

在安装 `virt-who` 时，会自动创建常规配置文件（位于 `/etc/virt-who.conf`）。如果需要，您可以使用默认值或编辑此文件。它有三个特殊部分：`[global]`、`[defaults]` 和 `[system_environment]`。

`global` 部分中的设置会影响应用程序的整体操作。

示例：全局部分

```
[global]
interval=3600 1
debug=True 2
```

1

检查连接的 **hypervisor** 的频率（秒）。也会影响报告映射的频率。由于虚拟机被授予最多 7 天的临时订阅，因此不需要频繁查询；您可以选择适合环境大小的间隔。

2

启用调试输出

默认值中的设置可默认应用于 `"/etc/virt-who.d/.conf"` 中发现的配置。如果您在这部分启用选项，则不需要再次在 `"/etc/virt-who.d/.conf"` 中设置它们。

示例：Defaults 部分

```
[defaults]
owner=1234567 1
hypervisor_id=hostname 2
```

1

管理程序所属的组织。您可以通过在虚拟机监控程序上运行 `subscription-manager orgs` 来找到机构。

2

如何识别虚拟机监控程序，其中之一：`uuid`、`hostname`、`hwuuid`

`system_environment` 中的设置被写入系统环境，并在进程执行期间可用，无论 `virt-who` 作为服务还是从命令行启动。

示例：`system_environment` 部分

```
[system_environment]
http_proxy= https://proxy.example.com:443 1
no_proxy=* 2
```

1

使用 HTTP 代理进行 virt-who 通信

2

如果您不想将 HTTP 代理用于来自此服务器的任何 virt-who 通信，您可以将 no\_proxy 设置为 192.168.1.0/24



注意

virt-who-0.30.x-1.el8 (RHEL 8.4)支持部分 [system\_environment]。如果您使用旧的 virt-who 版本，请将 'HTTP\_PROXY' 设置 'NO\_PROXY' by '/etc/sysconfig/virt-who'。

## 第 2 章 将基于主机的订阅附加到虚拟机监控程序

使用这个流程将基于主机的订阅（如 Red Hat Enterprise Linux for Virtual Platforms）附加到已经注册的虚拟机监控程序。

要注册新管理程序，[请参阅使用和配置 Red Hat Subscription Manager](#)。在配置 virt-who 以查询它之前，您必须注册虚拟机监控程序。

### 先决条件

- 确保您有足够的权利来基于主机的订阅，以覆盖您计划使用的所有管理程序。

### Web UI 流程

1. 登录到客户门户网站 <https://access.redhat.com>。
2. 导航到 **Subscriptions > Systems** 并点击 hypervisor 的名称。
3. 点击 **Subscriptions** 选项卡。
4. 点 **Attach Subscriptions**。
5. 选择基于主机的订阅，然后单击 **Attach Subscriptions**。

为每个 hypervisor 重复这些步骤。

### CLI 过程

1. 在虚拟机监控程序上，识别并记录基于主机的订阅池 ID：

```
# subscription-manager list --all --available --matches 'Host-based Subscription Name'
```

2. 将基于主机的订阅附加到 hypervisor：

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

3.

验证基于主机的订阅是否已附加：

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

为每个 hypervisor 重复这些步骤。

## 第 3 章 准备 VIRT-WHO 主机

使用这个流程将 Red Hat Enterprise Linux 7 服务器配置为为 VMware vCenter 和 Microsoft Hyper-V 运行 virt-who 服务。服务器可以是物理或者虚拟的。

对于 Red Hat hypervisor，您不需要单独的 virt-who 主机。

### 流程

1. 安装 Red Hat Enterprise Linux 7 服务器。只需要一个 CLI 环境。如需更多信息，请参阅 [Red Hat Enterprise Linux 7 安装指南](#)。

2. 注册服务器：

```
# subscription-manager register --auto-attach
```

3. 为 virt-who 和订阅服务之间的通信打开网络端口：

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

4. 为 virt-who 和每个 hypervisor 或虚拟化管理器之间的通信打开网络端口：

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

5. 安装 virt-who：

```
# yum install virt-who
```

6. 可选：编辑 `/etc/virt-who.conf` 文件以更改或添加全局设置。这些设置适用于来自此服务器的所有 virt-who 连接。

-

更改 `VIRTWHO_INTERVAL` 的值，以指定以分钟为单位查询虚拟化平台的频率。由于虚拟机被授予最多 7 天的临时订阅，因此不需要频繁查询；您可以选择适合环境大小的间隔。一天(1440)适合大多数环境。

- 如果要将在 HTTP 代理用于 `virt-who` 通信，请添加指定代理的行：

```
http_proxy=https://proxy.example.com:443
```

- 如果您不想将 HTTP 代理用于来自此服务器的任何 `virt-who` 通信，请添加以下行：

```
NO_PROXY=*
```

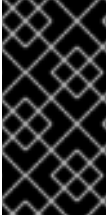
7.

启动并启用 `virt-who` 服务：

```
# systemctl enable --now virt-who
```



## 第 4 章 配置 VIRT-WHO



### 重要

使用环境变量以及使用 `sysconfig` 文件配置 `virt-who` 已被弃用。在下一个主发行版本中会忽略它们的使用。

每个虚拟化平台支持的 `virt-who` 配置都有所不同：

- 要为红帽产品配置 `virt-who`，请参阅 [第 4.1 节“在 Red Hat hypervisor 上安装和配置 `virt-who`”](#)。
- 要为 VMware vCenter 配置 `virt-who`，请参阅 [第 4.2 节“配置 `virt-who` 以连接到 VMware vCenter”](#)。
- 要为 Microsoft Hyper-V 配置 `virt-who`，请参阅 [第 4.3 节“配置 `virt-who` 以连接到 Microsoft Hyper-V”](#)。
- 要为 OpenShift Virtualization 配置 `virt-who`，请参阅 [第 4.4 节“配置 `virt-who` 以连接到 OpenShift Virtualization”](#)。

### 4.1. 在 RED HAT HYPERVISOR 上安装和配置 VIRT-WHO

使用这个流程在 Red Hat Enterprise Linux Virtualization (KVM)、Red Hat Virtualization 或 Red Hat OpenStack Platform 的每个 hypervisor 上安装和配置 `virt-who`。

#### 先决条件

- 将 hypervisor 注册到红帽订阅管理。
- 如果您使用 Red Hat Virtualization Host (RHVH)，请将其更新至最新版本，以便最小 `virt-who` 版本可用。默认情况下，`virt-who` 在 RHVH 上可用，但不能从 `rhel-7-server-rhvh-4-rpms` 存储库单独更新。

#### 流程

1. 在 **hypervisor** 上安装 **virt-who** :

```
# yum install virt-who
```

2. 可选：编辑 `/etc/virt-who.conf` 文件以更改或添加全局设置。由于 **virt-who** 在本地安装，因此这些设置仅适用于这个 **hypervisor**。

- 更改 **VIRTWHO\_INTERVAL** 的值，以指定在分钟内查询 **hypervisor** 的频率。由于虚拟机被授予最多 7 天的临时订阅，因此不需要频繁查询；您可以选择适合环境大小的间隔。一天(1440)适合大多数环境。

- 如果要将在 HTTP 代理用于 **virt-who** 通信，请添加指定代理的行：

```
http_proxy=https://proxy.example.com:443
```

- 如果您不想将 HTTP 代理用于来自此服务器的任何 **virt-who** 通信，请添加以下行：

```
NO_PROXY=*
```



#### 注意

**NO\_PROXY=** 可以使用，但只能在 `/etc/sysconfig/virt-who` 中。

**NO\_PROXY** 不是 `/etc/virt-who.conf` 中的有效配置。

3. 将模板配置文件复制到新的独立配置文件中：

```
# cp /etc/virt-who.d/template.conf /etc/virt-who.d/local.conf
```

4. 编辑您刚才创建的配置文件，将示例值改为特定于您的配置的文件：

```
[local] ①
type=libvirt ②
owner=1234567 ③
hypervisor_id=hostname ④
```

1

名称不需要是唯一的，因为此配置文件是唯一由此 `virt-who` 实例管理的文件。

2

指定这个 `virt-who` 连接与 Red Hat hypervisor 的连接。

3

管理程序所属的组织。您可以通过在虚拟机监控程序上运行 `subscription-manager orgs` 来找到机构。

4

指定如何识别 hypervisor。使用主机名为 Subscription Management 提供有意义的主机名。或者，如果虚拟机监控程序重命名，您可以使用 `uuid` 来避免重复。不要将 `hwuuid` 用于单独的 hypervisor。

5.

启动并启用 `virt-who` 服务：

```
# systemctl enable --now virt-who
```

为每个 hypervisor 重复这些步骤。

## 4.2. 配置 VIRT-WHO 以连接到 VMWARE VCENTER

使用这个流程配置 `virt-who` 以连接到 VMware vCenter 服务器。

### 先决条件

- 在 vCenter 服务器上创建一个只读 `virt-who` 用户。`virt-who` 用户需要至少对 vCenter Data Center 中所有对象的只读访问权限。
- 在 Red Hat Enterprise Linux 服务器中准备 `virt-who` 主机。

### 流程

1. 在 **virt-who** 主机上，使用 **virt-who-password** 工具加密 **virt-who** 用户的密码：

```
# virt-who-password
```

出现提示时，输入 **virt-who** 用户的密码，然后记下密码的加密形式。

2. 将模板配置文件复制到新的独立配置文件中：

```
# cp /etc/virt-who.d/template.conf /etc/virt-who.d/vcenter1.conf
```

要在故障排除时轻松识别配置文件，请使用 **VMware vCenter** 主机名作为新文件的名称。在本例中，主机名是 **vcenter1**。

3. 编辑您刚才创建的配置文件，使用特定于您的配置更改示例值：

```
[vcenter1] 1  
type=esx 2  
server=vcenter1.example.com 3  
username=virt_who_user 4  
encrypted_password=bd257f93d@482B76e6390cc54aec1a4d 5  
owner=1234567 6  
hypervisor_id=hostname 7  
filter_hosts=esx1.example.com, esx2.example.com 8
```

1

名称对于每个单独的配置文件必须是唯一的。使用 **vCenter Server** 主机名可轻松识别每个 **hypervisor** 的配置文件。

2

指定此 **virt-who** 连接与 **VMware vCenter** 服务器。

3

**vCenter** 服务器的 **FQDN**。

4

**vCenter** 服务器上的 **virt-who** 用户的名称。

5

6

虚拟机监控程序所属的组织。您可以通过在虚拟机监控程序上运行 `subscription-manager orgs` 来找到机构。

7

指定如何识别 `hypervisor`。使用主机名为 `Subscription Management` 提供有意义的主机名。或者，您可以使用 `uuid` 或 `hwuuid` 来避免在虚拟机监控程序重命名时重复。

8

如果某些管理程序从不运行 Red Hat Enterprise Linux 虚拟机，则不需要由 `virt-who` 报告这些 `hypervisor`。您可以使用以下选项之一过滤 `hypervisor`：支持通配符和正则表达式。如果名称包含特殊字符，则用引号括起来。

- `filter_hosts` 或 `exclude_hosts`：根据指定的 `hypervisor_id` 提供以逗号分隔的虚拟机监控程序列表。例如，如果虚拟机监控程序通过其主机名来标识，则必须包含或排除它们的主机名。
- `filter_host_parents` 或 `exclude_host_parents`：提供以逗号分隔的集群列表。过滤集群中的虚拟机监控程序由 `virt-who` 报告。`virt-who` 不会报告排除集群中的虚拟机监控程序。

4.

重启 `virt-who` 服务：

```
# systemctl restart virt-who
```

为每个 vCenter 服务器重复这些步骤。

### 4.3. 配置 VIRT-WHO 以连接到 MICROSOFT HYPER-V

使用这个流程配置 `virt-who` 以连接到 Microsoft Hyper-V hypervisor。

先决条件

- Red Hat Enterprise Linux 9 或更高版本。
- 在 Red Hat Enterprise Linux 服务器中准备 virt-who 主机。
- 为 hypervisor 启用基本身份验证模式。
- 在 hypervisor 上启用远程管理。
- 在虚拟机监控程序上创建只读 virt-who 用户。

## 流程

1. 在 virt-who 主机上，使用 virt-who-password 工具加密虚拟机监控程序的 virt-who 用户的密码：

```
# virt-who-password
```

出现提示时，输入 virt-who 用户的密码，然后记下密码的加密形式。

2. 将模板配置文件复制到新的独立配置文件中：

```
# cp /etc/virt-who.d/template.conf /etc/virt-who.d/hyperv1.conf
```

要在故障排除时轻松识别配置文件，请使用 hypervisor 的主机名作为新文件的名称。在本例中，主机名为 *hyperv1*。

3. 编辑您刚才创建的配置文件，使用特定于您的配置更改示例值：

```
[hyperv1] ①  
type=hyperv ②  
server=hyperv1.example.com ③  
username=virt_who_user ④  
encrypted_password=bd257f93d@482B76e6390cc54aec1a4d ⑤  
owner=1234567 ⑥  
hypervisor_id=hostname ⑦
```

1

名称对于每个单独的配置文件必须是唯一的。使用虚拟机监控程序的主机名，方便识别每个 hypervisor 的配置文件。

2

指定这个 virt-who 连接与 Microsoft Hyper-V hypervisor。

3

Hyper-V hypervisor 的 FQDN。

4

hypervisor 上的 virt-who 用户的名称。

5

virt-who 用户的加密密码。

6

此管理程序所属的组织。您可以通过在虚拟机监控程序上运行 subscription-manager orgs 来找到机构。

7

指定如何识别 hypervisor。使用主机名为 Subscription Management 提供有意义的主机名。或者，如果虚拟机监控程序重命名，您可以使用 uuid 来避免重复。不要将 hwwuid 用于单独的 hypervisor。

4.

重启 virt-who 服务：

```
# systemctl restart virt-who
```

为每个 hypervisor 重复这些步骤。

#### 4.4. 配置 VIRT-WHO 以连接到 OPENSIFT VIRTUALIZATION

支持的平台

**virt-who 支持的 OpenShift Virtualization 状态：**

- **virt-who-0.28.x-1.el7 (RHEL 7.9)**
- **virt-who-0.29.x-1.el8 (RHEL 8.3)**

**流程**

1. 在您要订阅的集群中，创建一个项目和名为 **virt-who** 的服务帐户：

```
$ oc new-project virt-who  
$ oc create serviceaccount virt-who
```

2. 创建集群角色来列出节点和虚拟机实例。

```
$ oc create clusterrole lsnodes --verb=list --resource=nodes  
$ oc create clusterrole lsvmis --verb=list --resource=vmis
```

3. 创建集群角色绑定。

```
$ oc adm policy add-cluster-role-to-user lsnodes system:serviceaccount:virt-who:virt-who  
$ oc adm policy add-cluster-role-to-user lsvmis system:serviceaccount:virt-who:virt-who
```

4. 验证 **virt-who** 系统帐户是否有列出所有正在运行的虚拟机的权限：

```
$ oc get vmis -A --as=system:serviceaccount:virt-who:virt-who
```

5. 在主机上安装 **virt-who**，可以是在 **OpenShift Virtualization** 本身中运行的虚拟机：

```
[virtwho-host]$ yum install virt-who
```

6. 在订阅的主机上查找您的所有者号：

```
$ subscription-manager orgs
```

- 7.



将模板配置文件复制到新的单独的配置文件中。要在故障排除时轻松识别配置文件，请使用集群 API 的主机名。在本例中，主机名为 `openshift-cluster-1`。

```
[virtwho-host]# cp /etc/virt-who.d/template.conf /etc/virt-who.d/openshift-cluster-1.conf
[cnv]
type=kubevirt
kubeconfig=/root/.kube/config
hypervisor_id=hostname
owner=<owner_number>
```

8. 获取 `virt-who` 服务帐户的令牌：

```
# oc serviceaccounts get-token virt-who
```

9. 如果 `/usr/bin/oc` 不可用，请安装 `/usr/bin/oc`，并使用令牌登录并创建有效的 `kubeconfig` 文件。您必须通过包含 `url` 来指定集群 `api`。例如：

```
[virtwho-host]# oc login https://api.testcluster-1.example.org:6443 --token=<token>
```

- a. 要在 `kubeconfig` 文件中使用 **OpenShift Virtualization** 证书(CA)证书，将其从集群中提取，并将它保存到运行 `virt-who` 作为控制器守护进程的系统中的文件：

```
oc get secret -n openshift-kube-apiserver-operator loadbalancer-serving-signer -o
jsonpath='{.data.tls.crt}' | base64 -d > $cluster-ca.pem
```

- b. 更改 `kubeconfig` 文件，使其包含提取的 CA 证书。例如：

```
[virtwho-host]$ cat /root/.kube/config
apiVersion: v1
clusters:
- cluster:
  server: https://api.testcluster.example.org:6443
  certificate-authority: /root/testcluster-ca.pem
  name: api-testcluster-example-org:6443
contexts:
- context:
  cluster: api-test-cluster-example-org:6443
  namespace: default
```

10. 在启动该服务前，您可以手动测试配置：

```
[virtwho-host]# virt-who --print
```



## 注意

如果安装了 `jq` 程序，您可以使用它更轻松地阅读输出：`VirtualMachine virt-who --print | jq`

1.

启用 `virt-who` 服务：

```
[virtwho-host]# systemctl enable virt-who
```

2.

重启 `virt-who` 服务以使用新配置。

```
[virtwho-host]# systemctl restart virt-who
```

`virt-who` 日志位于 `/var/log/rhsm/rhsm.log` 中。在这个文件中，您可以查看配置或连接错误。

## 第 5 章 注册虚拟机以使用基于主机的订阅

使用自动附加注册虚拟机，以便它们从 **hypervisor** 中继承订阅。

### 先决条件

- 将基于主机的订阅附加到虚拟机的 **hypervisor**。
- 配置 **virt-who** 以查询虚拟机的虚拟机监控程序。
- 确保虚拟机可以迁移到已附加基于主机的订阅并将其报告给 **virt-who**，或将虚拟机迁移到特定的虚拟机监控程序。

### Web UI 流程

1. 登录到客户门户网站 <https://access.redhat.com>。
2. 导航到 **Subscriptions > Systems** 并点击虚拟机的名称。
3. 点击 **Subscriptions** 选项卡。
4. 单击 **Run Auto-Attach**。

为每个虚拟机重复这些步骤。

### CLI 过程

1. 使用 **auto-attach** 选项注册虚拟机：

```
# subscription-manager register --auto-attach
```

2. 出现提示时，输入您的用户名和密码。

为每个虚拟机重复这些步骤。

如果 `virt-who` 已报告了虚拟机，虚拟机会从其 `hypervisor` 中继承订阅。

如果 `virt` 没有报告虚拟机，则虚拟机会收到临时订阅，而 `Subscription Management` 会等待 `virt-who` 提供有关虚拟机在其上运行的 `hypervisor` 的信息。在 `virt-who` 提供此信息后，虚拟机会从其 `hypervisor` 中继承订阅。

## 第 6 章 VIRT-WHO 故障排除方法

### 验证 virt-who 状态

验证 virt-who 服务的状态：

```
# systemctl status virt-who.service
```

### 调试日志记录

检查 `/var/log/rhsm/rhsm.log` 文件，其中 virt-who 默认记录其所有活动。

如需更详细的日志记录，请在 `/etc/virt-who.conf` 文件中启用 `debug` 选项：

```
[global]
debug=True
```

重启 virt-who 服务以使更改生效。

解决底层问题后，修改 `/etc/virt-who.conf` 文件以禁用调试，然后重启 virt-who 服务。

### 测试配置选项

进行更改并测试结果，根据需要重复。virt-who 提供三个选项，可帮助测试到虚拟化平台的配置文件、凭证和连接：

- **virt-who --one-shot** 命令读取配置文件，检索虚拟机列表并将其发送到订阅管理系统，如 Red Hat Satellite 或 Red Hat Subscription Management，然后立即退出。
- **virt-who --print** 命令读取配置文件并打印虚拟机列表，但不会将其发送到订阅管理系统。
- 从 RHEL 9 Beta 开始，**virt-who --status** 命令读取配置文件，并输出源和目标系统的连接状态摘要。
  - **virt-who --status** 命令和 **--json** 选项为每个配置提供额外的连接数据，采用 JSON 格式。

`virt-who --one-shot` 和 `virt-who --print` 命令的预期输出是虚拟机监控程序及其虚拟机列表，采用 JSON 格式。以下是从 VMware vSphere 实例中提取的。所有 `hypervisor` 的输出遵循相同的结构。

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

`virt-who --status` 命令的预期输出是 `virt-who` 中每个配置的连接状态的纯文本摘要。

```
+-----+
|           |
| Configuration Status |
|           |
+-----+
| Configuration Name: esx_config1 |
| Source Status: success          |
| Destination Status: success     |
|
| Configuration Name: hyperv-55   |
| Source Status: failure          |
| Destination Status: failure     |
```

`virt-who --status` 命令的预期输出以及 `--json` 选项提供有关每个配置的额外信息，包括其最后一次成功运行，采用 JSON 格式。此输出还包括每个配置成功或失败状态的详细信息。

- 当状态报告指示配置成功时，JSON 输出包括 `virt-who` 在最后一次成功运行周期期间报告的虚拟机监控程序和客户机数量。
- 当状态报告指示配置失败时，JSON 输出会包含关联的错误消息。

```
"configurations": [
  {
    "name": "esx-conf1",
    "source": {
      "connection": "https://esx_system.example.com",
      "status": "success",
      "last_successful_retrieve": "2020-02-28 07:25:25 UTC",
      "hypervisors": 20,
      "guests": 37
    },
    "destination": {
```

```

    "connection":"candlepin.example.com",
    "status":"success",
    "last_successful_send":"2020-02-28 07:25:27 UTC",
    "last_successful_send_job_status":"FINISHED"
  }
},
{
  "name":"hyperv-55",
  "source":{
    "connection":"windows10-3.company.com",
    "status":"failure",
    "message":"Unable to connect to server: invalid credentials",
    "last_successful_retrieve":null
  },
  "destination":{
    "connection":"candlepin.company.com",
    "status":"failure",
    "message":"ConnectionRefusedError: [Errno 111] Connection refused",
    "last_successful_send":null,
    "last_successful_send_job_status":null
  }
}
]
}

```

`virt-who --status` 命令也可以与 `--debug` 和 `--config` 选项一起使用，以提供有关配置文件的附加信息。

在使用多个 `virt-who` 配置文件时识别问题

如果您在一个服务器上有多多个 `virt-who` 配置文件，请在每次移动后测试时将一个文件移动到不同的目录中。如果不再出现这个问题，原因将与最近移动的文件相关联。解决问题后，将 `virt-who` 配置文件返回到其原始位置。

或者，您可以在移动后测试单个文件，使用 `--config` 选项指定其位置。例如：

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

从 RHEL 9 Beta 版开始，您可以使用 `--debug` 和 `--config` 选项输入 `virt-who --status` 选项来识别配置文件，而不从目录中删除任何其他文件。例如：

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

您还可以使用 `--json` 选项输入命令，以 JSON 格式查看有关每个配置的更多详细信息。例如：

```
#virt-who --debug --status --json --config /tmp/conf_name.conf
```

## 识别重复的虚拟机监控程序

重复的 **hypervisor** 可能会导致订阅和权利错误。输入以下命令检查重复的虚拟机监控程序：

```
# 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"
```

在本例中，三个 **hypervisor** 具有相同的 FQDN (**localhost**)，必须更正才能使用唯一的 FQDN。

## 识别重复的虚拟机

输入以下命令检查重复的虚拟机：

```
# 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
```

## 检查虚拟机监控程序数量

输入以下命令检查当前报告的 **hypervisor virt-who** 数量：

```
# 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
```

从 RHEL 9 Beta 版开始，输入以下命令检查 **virt-who** 在最后一次成功运行周期期间报告的虚拟机监控程序数量：

```
# virt-who --status --json
```



---

## 检查虚拟机数量

输入以下命令检查 **virt-who** 当前报告的虚拟机数量：

```
# 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
```

从 RHEL 9 Beta 版开始，输入以下命令检查 **virt-who** 在最后一次成功运行周期期间报告的客户机数量：

```
# virt-who --status --json
```

## 第 7 章 VIRT-WHO 故障排除场景

### virt-who 无法连接到虚拟化平台

如果 `virt-who` 无法连接到 `hypervisor` 或虚拟化管理器，请检查 Red Hat Subscription Manager 日志文件 `/var/log/rhsm/rhsm.log`。如果您发现 `No route to host` 的消息，则虚拟机监控程序可能会侦听错误的端口。在这种情况下，修改虚拟机监控程序的 `virt-who` 配置文件，并将正确的端口号附加到服务器值。

在修改配置文件后，您必须重启 `virt-who` 服务。

### virt-who 无法通过本地网络上的 HTTP 代理连接到虚拟化平台

如果 `virt-who` 无法通过 HTTP 代理连接到虚拟机监控程序或虚拟化管理器，请将代理配置为允许本地流量通过，或通过向 `"/etc/virt-who.conf"` 中添加以下行来修改 `virt-who` 服务以使用代理：

```
[system_environment]
no_proxy=*
```

在修改配置文件后，您必须重启 `virt-who` 服务。



#### 注意

部分 `[system_environment]` 仅支持 `virt-who-0.30.x-1.el8` (RHEL 8.4)，如果您使用的是旧的 `virt-who` 版本，请通过 `/etc/sysconfig/virt-who` 设置 `NO_PROXY`。