Red Hat Enterprise Virtualization 3.6

REST API 指南

使用 Red Hat Enterprise Virtualization REST API

Last Updated: 2017-09-22

Red Hat Enterprise Virtualization 3.6 REST API 指南

使用 Red Hat Enterprise Virtualization REST API

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

法律通告

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.

摘要

本指南介绍了 Red Hat Enterprise Virtualization 的 REST API。

目录

第 1 章 介绍 **6**
1.1. REPRESENTATIONAL STATE TRANSFER - 表述性状态传输 6
1.2. RED HAT ENTERPRISE VIRTUALIZATION REST API 先决条件 6

第 2 章 用户身份验证和安全 **8**
2.1. TLS/SSL 认证 8
2.2. HTTP 验证 9
2.3. 验证会话 10

第 3 章 REST API 快速入门介绍 **12**
3.1. 实例：访问 API 进入点（API ENTRY POINT） 12
3.2. 实例：列出数据中心集合 14
3.3. 实例：列出主机集群集合 15
3.4. 实例：列出逻辑网络集合 16
3.5. 实例：列出组件集合 17
3.6. 示例：列出 CPU 配置集 19
3.7. 实例：批准主机 20
3.8. 实例：创建 NFS 数据存储 21
3.9. 实例：创建 NFS ISO 存储 22
3.10. 实例：为数据中心附加存储域 24
3.11. 实例：激活存储域 25
3.12. 实例：创建虚拟机 26
3.13. 实例：创建虚拟机 NIC 29
3.14. 实例：创建虚拟机存储磁盘 29
3.15. 实例：为虚拟机附加 ISO 镜像 30
3.16. 实例：启动虚拟机 31
3.17. 实例：检查系统事件 32

第 4 章 进入点 **34**
4.1. 产品信息 35
4.2. 连接项 35
4.3. 特殊项 37
4.4. 概述项 37
4.5. RESTFUL SERVICE DESCRIPTION LANGUAGE（RSDL） 38
4.6. RED HAT ENTERPRISE VIRTUALIZATION WINDOWS 客户端 VSS 支持 40
4.7. QEMU GUEST AGENT 介绍 40
4.8. VSS 工作流程 40

第 5 章 兼容级别版本 **42**
5.1. 升级兼容级别 42

第 6 章 能力 **44**
6.1. 与特定版本相关的能力 44
6.2. 当前版本 44
6.3. 功能 45

第 7 章 通用功能 **48**
7.1. 项属性图标 48
7.2. 表述 48
7.3. 集合 49
7.4. 资源 54

第 8 章 备份和恢复 API **61**

8.1. 备份一个虚拟机 61
8.2. 恢复一个虚拟机 62

第 9 章 数据中心 64
9.1. 数据中心项 64
9.2. 数据中心的 XML 表述 65
9.3. 数据中心的 JSON 表述 65
9.4. 方法 66
9.5. 子集合 67
9.6. 操作 72

第 10 章 集群 74
10.1. 集群项 74
10.2. 内存策略项 75
10.3. 调度策略项 76
10.4. 集群的 XML 表述 76
10.5. 集群的 JSON 表述 77
10.6. 方法 79
10.7. 子集合 80

第 11 章 网络 89
11.1. 网络项 89
11.2. 网络的 XML 表述 89
11.3. 网络资源的 JSON 表述 90
11.4. 方法 91
11.5. 子集合 91

第 12 章 存储域 94
12.1. 存储域项 94
12.2. 存储域的 XML 表述 95
12.3. 存储域的 JSON 表述 96
12.4. 方法 97
12.5. 存储类型 98
12.6. 导出存储域 100
12.7. GLANCE 镜像存储域 103
12.8. 子集合 104
12.9. 操作 105

第 13 章 存储连接 108
13.1. 存储连接项 108
13.2. 存储连接资源的 XML 表述 109
13.3. 方法 109

第 14 章 主机 113
14.1. 主机项 113
14.2. 主机的 XML 表述 115
14.3. 主机的 JSON 表述 117
14.4. 电源管理项 120
14.5. 内存管理项 122
14.6. 方法 123
14.7. 子集合 124
14.8. 操作 139

第 15 章 虚拟机 143
15.1. 虚拟机项 143

15.2. 虚拟机的 XML 表述 148
15.3. 虚拟机的 OVF 数据的 XML 表述 151
15.4. 虚拟机的 JSON 表述 153
15.5. 方法 156
15.6. 子集合 159
15.7. 操作 184

第 16 章 浮动磁盘 192
16.1. 浮动磁盘项 192
16.2. 浮动磁盘的 XML 表述 193
16.3. 方法 194
16.4. 子集合 194
16.5. 操作 196

第 17 章 模板 197
17.1. 虚拟机模板项 197
17.2. 虚拟机模板的 XML 表述 199
17.3. 方法 200
17.4. 操作 202

第 18 章 虚拟机池 203
18.1. 虚拟机池项 203
18.2. 虚拟机池的 XML 表述 203
18.3. 方法 204
18.4. 操作 205

第 19 章 域 206
19.1. 域项 206
19.2. 域资源的 XML 表述 206
19.3. 子集合 206

第 20 章 组 209
20.1. 导入的组项 209
20.2. 组资源的 XML 表述 209
20.3. 从目录服务添加组 209

第 21 章 角色 211
21.1. 角色项 211
21.2. 角色集合的 XML 表述 211
21.3. 方法 212
21.4. 角色权限子集合 213

第 22 章 用户 215
22.1. 用户项 215
22.2. 用户资源的 XML 表述 215
22.3. 方法 216

第 23 章 标签 218
23.1. 标签项 218
23.2. 标签资源的 XML 表述 218
23.3. 关联标签 218
23.4. 父标签 220

第 24 章 事件 223
24.1. 事件项 223
24.2. 事件集合的 XML 表述 223

24.3. 虚拟机创建事件的 XML 表述 223
24.4. 搜索事件 224
24.5. 对事件进行分页显示 225

附录 A. 在 CURL 中使用 API **227**
A.1. 在 CURL 中使用 API 227
A.2. 安装 CURL 227
A.3. 使用 CURL 227
A.4. 实例 228

附录 B. 列举值的映射 **231**
B.1. 列举值的映射 231

附录 C. 事件代码 **233**
C.1. 事件代码 233

附录 D. 时区 **340**
D.1. 时区 340

附录 E. 修订历史 **345**

第 1 章 介绍

Red Hat Enterprise Virtualization Manager 包括了一个 **REST （Representational State Transfer）** API。软件开发人员和系统管理员可以在自己定制的脚本中调用 API，或在外部的应用程序中使用 HTTP 来访问这个 API，从而达到不使用标准的网站管理界面来对 Red Hat Enterprise Virtualization 环境进行控制的目的。

使用 REST API 的好处包括：

- 支持大量客户端 - 任何可以使用 HTTP 协议调用 API 的编程语言、开发平台和系统都可以使用所提供的 REST API;
- 自描述 - 客户端应用程序只需要了解最基本的虚拟化平台的信息，REST API 中的多数信息都可以在运行时被自动实现;
- 基于资源的模式 - 这个基于资源的 REST 模式为管理虚拟化平台提供了一个自然的方法。

REST API 为软件开发人员和系统管理员提供了以下功能：

- 把虚拟环境集成到 IT 环境中。
- 与第三方虚拟化软件进行集成。
- 自动化维护和错误检查任务。
- 使用脚本在 Red Hat Enterprise Virtualization 环境中执行重复性的操作。

本文档可以作为 Red Hat Enterprise Virtualization Manager REST API 的一个参考信息，它包括了基本的指导信息和实例。开发人员和系统管理员可以通过这些内容来了解直接使用 REST API 或 Python 库来实现管理 Red Hat Enterprise Virtualization 环境的功能。

1.1. REPRESENTATIONAL STATE TRANSFER - 表述性状态传输

Representational State Transfer - 表述性状态传输（简称 REST） 是一个设计架构，它专注于特定服务资源以及这些服务的表述形式。资源的表述形式就是一个关键的信息抽象层，它代表了服务器上的一个特定管理项。客户端向服务器（以一个 URI 表示）发送一个请求，并执行标准的 HTTP 方法，如 **GET**、**POST**、**PUT** 和 **DELETE**。在客户端和服务器间的通讯是无状态的，每个请求包括了为了完成这个请求所需的所有信息，并独立于其它请求。

1.2. RED HAT ENTERPRISE VIRTUALIZATION REST API 先决条件

Red Hat Enterprise Virtualization REST API 先决条件

- 安装了一个联网的、包括了 REST API 的 Red Hat Enterprise Virtualization Manager。
- 可以发起并接收从 REST API 发出的 HTTP 请求的客户端或程序库，如：
  - Python 软件开发套件（SDK）
  - Java 软件开发套件（SDK）
  - **cURL** 命令行工具
  - **RESTClient**，使用 REST 的 web 服务的一个故障排除工具程序

- 对 REST API 所使用的 Hypertext Transfer Protocol（HTTP）有一定了解。http://www.ietf.org/rfc/rfc2616.txt 提供了与 HTTP 相关的信息；

- 对 API 用来创建资源表述的 XML 或 JSON 有一定了解。W3C 提供了 XML 的规格介绍（http://www.w3.org/TR/xml/）。ECMA International 提供了免费的 JSON 资料（http://www.ecma-international.org）。

第 2 章 用户身份验证和安全

2.1. TLS/SSL 认证

Red Hat Enterprise Virtualization Manager API 需要使用 Hypertext Transfer Protocol Secure（HTTPS）[1] 来和客户端软件（如 Manager 的 SDK 和 CLI）进行安全的通信。这需要从 Red Hat Enterprise Virtualization Manager 中获得一个证书，并把它导入到客户端系统的证书存储系统中。

重要

请使用一个安全的网络连接来从 Red Hat Enterprise Virtualization Manager 获得您的证书。

过程 2.1. 获得一个证书

可以使用以下 3 种方法中的一个来从 Red Hat Enterprise Virtualization Manager 获得证书，并把证书传输到客户端系统上：

1. **方法 1** - 使用一个命令行工具从 Manager 上下载证书。例如使用 **cURL** 或 **Wget**。多个平台都提供了这两个工具。

   a. 使用 **cURL**：

   ```
   $ curl -o rhevm.cer http://[rhevm-server]/ca.crt
   ```

   b. 使用 **Wget**：

   ```
   $ wget -O rhevm.cer http://[rhevm-server]/ca.crt
   ```

2. **方法 2** - 使用一个网络浏览器访问提供证书的位置：

   ```
   http://[rhevm-server]/ca.crt
   ```

   根据您所使用的浏览器，证书可能会被下载，也可能会被直接导入到浏览器的密钥库（keystore）中。

   a. **如果浏览器需要下载这个证书：**把文件保存为 **`rhevm.cer`**。

      如果浏览器导入了这个证书：把它从浏览器的证书选项中导出，并保存为 **`rhevm.cer`**。

3. **方法 3** - 登录到 Manager 所在的系统，把证书从 truststore 中导出，并复制到您的客户端系统上。

   a. 以 **`root`** 用户身份登录到 Manager 所在的机器上。

   b. 使用 Java **keytool** 工具把证书从 truststore 中导出：

   ```
   $ keytool -exportcert -keystore /etc/pki/ovirt-engine/.truststore
   -alias cacert -storepass mypass -file rhevm.cer
   ```

   它会创建一个名为 **`rhevm.cer`** 的证书文件。

   c. 使用 **scp** 命令把证书复制到客户端系统上：

```
$ scp rhevm.cer [username]@[client-machine]:[directory]
```

以上所有 3 个方法都会在您的客户端系统上创建一个名为 **rhevm.cer** 的证书文件。API 用户可以把这个文件导入到客户端系统的证书库中。

过程 2.2. 把证书导入到一个客户端系统上

- 把证书导入到客户端系统的方法取决于客户端系统如果保存和解析证书。以上包括的导入证书的实例是针对于使用 Network Security Services（NSS）或 Java KeyStore（JKS）的客户端系统的。如果您使用其它的客户端，请参阅它的文档来了解如何导入证书的方法。

2.2. HTTP 验证

具有 Red Hat Enterprise Virtualization 帐号的用户都可以访问 REST API。API 用户到 API 的请求都包括一个强制的 Red Hat Enterprise Virtualization Manager 用户名和密码。每个请求都使用 HTTP Basic Authentication [2] 来加密它们的身份信息。如果请求没有包括正确的 **Authorization** 头，API 会在结果中包括一个 **401 Authorization Required** 信息：

例 2.1. 访问 REST API 时没有包括正确的身份信息

```
HEAD [base] HTTP/1.1
Host: [host]

HTTP/1.1 401 Authorization Required
```

请求会获得一个特定域（realm）的 **Authorization** 头。API 用户在所提供的身份信息中指定一个加密的 Red Hat Enterprise Virtualization Manager 域（domain）和用户信息（格式是 **username@domain:password**）。

以下表格显示了基于 base64 的身份信息。

表 2.1. API 服务所需的身份信息

项	值
username	rhevmadmin
domain	domain.example.com
password	123456
未编码的用户信息	rhevmadmin@domain.example.com:123456
使用 base64 编码的用户信息	cmhldm1hZG1pbkBibGFjay5xdW1yYW5ldC5jb206MTIzNDU2

API 用户提供以下的 base64 加密身份信息：

例 2.2. 使用正确的身份信息访问 REST API

```
HEAD [base] HTTP/1.1
Host: [host]
Authorization: Basic cmhldm1hZG1pbkBibGFjay5xdW1yYW5ldC5jb206MTIzNDU2

HTTP/1.1 200 OK
...
```

重要

基本验证会以明码的形式发送可能需要保密的信息，如密码。REST API 需要使用 Hypertext Transfer Protocol Secure (HTTPS) 来处理明码请求。

重要

一些 base64 库会把加密结果分为多个行，每个行以一个 newline 符结束，这会破坏头数据并导致一个错误请求。Authorization 头需要加密的身份信息以一行的形式包括在头中。

2.3. 验证会话

API 同时提供了对验证会话的支持。API 用户在初始请求中包括验证信息，并使用一个会话 cookie 为其后的请求提供验证信息。以下介绍了使用验证会话的过程。

过程 2.3. 请求一个验证会话

1. 发送带有 **Authorization** 和 **Prefer: persistent-auth** 的请求

```
HEAD [base] HTTP/1.1
Host: [host]
Authorization: Basic
cmhldm1hZG1pbkBibGFjay5xdW1yYW5ldC5jb206MTIzNDU2
Prefer: persistent-auth

HTTP/1.1 200 OK
...
```

这会返回带有以下头数据的响应：

```
Set-Cookie: JSESSIONID=5dQja5ubr4yvI2MM2z+LZxrK; Path=/api; Secure
```

记录 **JSESSIONID=** 的值。在这个例子中，它的值是 **JSESSIONID=5dQja5ubr4yvI2MM2z+LZxrK**。

2. 在随后的所有请求中包括带有 **Prefer: persistent-auth** 和 **cookie**（包括 **JSESSIONID=** 的值） 的头。使用验证会话将不再需要 **Authorization**。

```
HEAD [base] HTTP/1.1
Host: [host]
Prefer: persistent-auth
cookie: JSESSIONID=5dQja5ubr4yvI2MM2z+LZxrK
```

```
HTTP/1.1 200 OK
...
```

3. 当不再需要会话时，执行一个没有 **`Prefer: persistent-auth`** 头的请求。

```
HEAD [base] HTTP/1.1
Host: [host]
Authorization: Basic
cmhldm1hZG1pbkBibGFjay5xdW1yYW5ldC5jb206MTIzNDU2

HTTP/1.1 200 OK
...
```

---

[1] RFC 2818 HTTP Over TLS 包括了 HTTPS 的信息。

[2] RFC 2617 HTTP Authentication: Basic and Digest Access Authentication 包括了用户验证的基本信息。

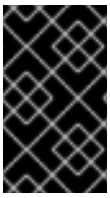
第 3 章 REST API 快速入门介绍

本章提供了一个实例，它展示了使用 REST API 设置一个基本的 Red Hat Enterprise Virtualization 环境并创建一个虚拟机的方法。

除了前面提到的标准先决条件外，这个实例还需要以下条件：

- 一个包括了 Red Hat Enterprise Virtualization Hypervisor 的、已经联网并配置的主机；
- 包括了需要安装的虚拟机操作系统的 ISO 文件。在这个实例中我们使用 Red Hat Enterprise Linux Server 6 作为虚拟机的操作系统；
- Red Hat Enterprise Virtualization 提供的、用来上传操作系统 ISO 文件的 **engine-iso-uploader** 工具程序。

这个实例使用 **cURL** 来演示到客户端应用程序的 REST 请求。任何支持 HTTP 请求的程序都可以替代这里的 **cURL**。



重要

为了简化，这个实例中的 HTTP 请求头中都省略掉了 **`Host:`** 和 **`Authorization:`** 项。这些项是强制的项，它们需要和您所安装的 Red Hat Enterprise Virtualization Manager 相对应。

重要

所有的 **cURL** 实例中的验证信息和证书位置都使用替换符（分别是 **USER:PASS** 和 **CERT**）。请确认 **cURL** 的所有请求都包括相关的验证信息。

注意

Red Hat Enterprise Virtualization Manager 会为每个资源的 **`id`** 属性生成一个 GUID。这里所使用的值可能会和您实际使用的 Red Hat Enterprise Virtualization 环境中的值不同。

3.1. 实例：访问 API 进入点（API ENTRY POINT）

以下请求会获得 API 主进入点的表述。

例 3.1. 访问 API 的进入点

请求：

```
GET /api HTTP/1.1
Accept: application/xml
```

cURL 命令：

```
# curl -X GET -H "Accept: application/xml" -u [USER:PASS] \
    --cacert [CERT] https://[RHEVM Host]:443/api
```

结果：

```
HTTP/1.1 200 OK
Content-Type: application/xml

<api>
    <link rel="capabilities" href="/api/capabilities"/>
    <link rel="clusters" href="/api/clusters"/>
    <link rel="clusters/search" href="/api/clusters?search={query}"/>
    <link rel="datacenters" href="/api/datacenters"/>
    <link rel="datacenters/search" href="/api/datacenters?search=
{query}"/>
    <link rel="events" href="/api/events"/>
    <link rel="events/search" href="/api/events?search={query}"/>
    <link rel="hosts" href="/api/hosts"/>
    <link rel="hosts/search" href="/api/hosts?search={query}"/>
    <link rel="networks" href="/api/networks"/>
    <link rel="roles" href="/api/roles"/>
    <link rel="storagedomains" href="/api/storagedomains"/>
    <link rel="storagedomains/search" href="/api/storagedomains?search=
{query}"/>
    <link rel="tags" href="/api/tags"/>
    <link rel="templates" href="/api/templates"/>
    <link rel="templates/search" href="/api/templates?search={query}"/>
    <link rel="users" href="/api/users"/>
    <link rel="groups" href="/api/groups"/>
    <link rel="domains" href="/api/domains"/>
    <link rel="vmpools" href="/api/vmpools"/>
    <link rel="vmpools/search" href="/api/vmpools?search={query}"/>
    <link rel="vms" href="/api/vms"/>
    <link rel="vms/search" href="/api/vms?search={query}"/>
    <special_objects>
        <link rel="templates/blank"
          href="/api/templates/00000000-0000-0000-0000-000000000000"/>
        <link rel="tags/root"
          href="/api/tags/00000000-0000-0000-0000-000000000000"/>
    </special_objects>
    <product_info>
        <name>Red Hat Enterprise Virtualization</name>
        <vendor>Red Hat</vendor>
        <version revision="0" build="0" minor="0" major="3"/>
    </product_info>
    <summary>
        <vms>
            <total>5</total>
            <active>0</active>
        </vms>
        <hosts>
            <total>1</total>
            <active>1</active>
        </hosts>
        <users>
            <total>1</total>
            <active>1</active>
        </users>
        <storage_domains>
            <total>2</total>
            <active>2</active>
```

```
        </storage_domains>
    </summary>
</api>
```

进入点为用户提供了虚拟环境中的集合连接。每个集合连接的 **rel=** 属性提供了到这个连接的参考点。这个实例的下一步会检查 **datacenter** 集合（连接是 **rel="datacenter"**）。

进入点还包括了 **product_info**、**special_objects** 和 **summary** 等其它信息。这些信息没有包括在这个实例中。

3.2. 实例：列出数据中心集合

Red Hat Enterprise Virtualization Manager 在安装时创建一个 **Default** 数据中心。这个实例使用 **Default** 数据中心作为虚拟环境的基础。

以下请求会获得数据中心集合的表述。

例 3.2. 列出数据中心集合

请求

```
GET /api/datacenters HTTP/1.1
Accept: application/xml
```

cURL 命令：

```
# curl -X GET -H "Accept: application/xml" -u [USER:PASS] \
    --cacert [CERT] \
    https://[RHEVM Host]:443/api/datacenters
```

结果：

```
HTTP/1.1 200 OK
Content-Type: application/xml

<data_centers>
    <data_center href="/api/datacenters/00000002-0002-0002-0002-
0000000003ab" id="00000002-0002-0002-0002-0000000003ab">
        <name>Default</name>
        <description>The default Data Center</description>
        <link rel="storagedomains"/>
          href="/api/datacenters/00000002-0002-0002-0002-
0000000003ab/storagedomains"
        <link rel="clusters"/>
          href="/api/datacenters/00000002-0002-0002-0002-
0000000003ab/clusters"
        <link rel="networks"/>
          href="/api/datacenters/00000002-0002-0002-0002-
0000000003ab/networks"
        <link rel="permissions"/>
          href="/api/datacenters/00000002-0002-0002-0002-
0000000003ab/permissions"
```

```
        <link rel="quotas"/>
          href="/api/datacenters/00000002-0002-0002-0002-
0000000003ab/quotas"
        <link rel="iscsibonds"/>
          href="/api/datacenters/00000002-0002-0002-0002-
0000000003ab/iscsibonds"
        <link rel="qoss"/>
          href="/api/datacenters/00000002-0002-0002-0002-
0000000003ab/qoss"
        <local>false</local>
        <storage_format>v3</storage_format>
        <version major="3" minor="5"/>
        <supported_versions>
            <version major="3" minor="5"/>
        </supported_versions>
        <status>
            <state>up</state>
        </status>
    </data_center>
</data_centers>
```

请记录 **Default** 数据中心的 **id** 值。访问虚拟环境中和这个数据中心相关的其它资源时需要指定这个值。

数据中心同时还包括了一个到 **storagedomains** 子集合的连接。数据中心使用这个子集合从 **storagedomains** 主集合中附加存储域。

3.3. 实例：列出主机集群集合

Red Hat Enterprise Virtualization Manager 在安装的时候会创建一个 **Default** 主机集群。这个实例使用 **Default** 集群来为 Red Hat Enterprise Virtualization 环境中的资源进行分组。

以下请求会获得集群集合的表述。

例 3.3. 列出主机集群集合

请求

```
GET /api/clusters HTTP/1.1
Accept: application/xml
```

cURL 命令：

```
# curl -X GET -H "Accept: application/xml" -u [USER:PASS] \
    --cacert [CERT] \
    https://[RHEVM Host]:443/api/clusters
```

结果：

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
<clusters>
    <cluster id="99408929-82cf-4dc7-a532-9d998063fa95"
      href="/api/clusters/99408929-82cf-4dc7-a532-9d998063fa95">
        <name>Default</name>
        <description>The default server cluster</description>
        <link rel="networks"
          href="/api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/networks"/>
        <link rel="permissions"
          href="/api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/permissions"/>
        <cpu id="Intel Penryn Family"/>
        <data_center id="01a45ff0-915a-11e0-8b87-5254004ac988"
          href="/api/datacenters/01a45ff0-915a-11e0-8b87-
5254004ac988"/>
        <memory_policy>
            <overcommit percent="100"/>
            <transparent_hugepages>
                <enabled>false</enabled>
            </transparent_hugepages>
        </memory_policy>
        <scheduling_policy/>
        <version minor="0" major="3"/>
        <error_handling>
            <on_error>migrate</on_error>
        </error_handling>
    </cluster>
</clusters>
```

请记录 **Default** 主机集群的 **id** 值。访问虚拟环境中和这个集群相关的其它资源时需要指定这个值。

Default 集群通过使用 **data_center** 项中的 **id** 和 **href** 属性来与 **Default** 数据中心建立联系。

networks 子集合包括了一组和这个集群相关联的网络资源。下面一节详细介绍了 **networks** 集群。

3.4. 实例：列出逻辑网络集合

Red Hat Enterprise Virtualization Manager 在安装时会创建默认的 **ovirtmgmt** 网络。这个网络作为管理网络被 Red Hat Enterprise Virtualization Manager 使用来访问 hypervisor 主机。

这个网络与 **Default** 集群项关联，而且是 **Default** 数据中心的一员。这个实例使用 **ovirtmgmt** 网络来与虚拟机进行连接。

以下请求会获得逻辑网络集合的表述。

例 3.4. 列出逻辑网络集合

请求：

```
GET /api/networks HTTP/1.1
Accept: application/xml
```

cURL 命令：

```
# curl -X GET -H "Accept: application/xml" -u [USER:PASS] \
    --cacert [CERT] \
    https://[RHEVM Host]:443/api/networks
```

结果：

```
HTTP/1.1 200 OK
Content-Type: application/xml

<networks>
    <network id="00000000-0000-0000-0000-000000000009"
      href="/api/networks/00000000-0000-0000-0000-000000000009">
        <name>ovirtmgmt</name>
        <description>Management Network</description>
        <data_center id="01a45ff0-915a-11e0-8b87-5254004ac988"
          href="/api/datacenters/01a45ff0-915a-11e0-8b87-
5254004ac988"/>
        <stp>false</stp>
        <status>
            <state>operational</state>
        </status>
        <display>false</display>
    </network>
</networks>
```

ovirtmgmt 网络通过使用数据中心的 **id** 与 **Default** 数据中心相关联。

ovirtmgmt 网络也通过集群的 **network** 子集合与 **Default** 集群建立关系。

3.5. 实例：列出组件集合

Red Hat Enterprise Virtualization Manager 会自动注册所有配置过的 Red Hat Enterprise Virtualization Hypervisor 主机。这个实例会获得主机集合的表述，并显示一个在虚拟环境中注册的、名为 **hypervisor** 的 Red Hat Enterprise Virtualization Hypervisor 主机。

例 3.5. 列出主机集合

请求：

```
GET /api/hosts HTTP/1.1
Accept: application/xml
```

cURL 命令：

```
# curl -X GET -H "Accept: application/xml" -u [USER:PASS] \
    --cacert [CERT] \
    https://[RHEVM Host]:443/api/hosts
```

结果：

```
HTTP/1.1 200 OK
Accept: application/xml
```

```
<hosts>
    <host id="0656f432-923a-11e0-ad20-5254004ac988"
      href="/api/hosts/0656f432-923a-11e0-ad20-5254004ac988">
        <name>hypervisor</name>
        <actions>
            <link rel="install"
              href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/install"/>
            <link rel="activate"
              href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/activate"/>
            <link rel="fence"
              href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/fence"/>
            <link rel="deactivate"
              href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/deactivate"/>
            <link rel="approve"
              href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/approve"/>
            <link rel="iscsilogin"
              href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/iscsilogin"/>
            <link rel="iscsidiscover"
              href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/iscsidiscover"/>
            <link rel="commitnetconfig"
              href="/api/hosts/0656f432-923a-11e0-ad20-5254004ac988/
              commitnetconfig"/>
        </actions>
        <link rel="storage"
          href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/storage"/>
        <link rel="nics"
          href="/api/hosts/0656f432-923a-11e0-ad20-5254004ac988/nics"/>
        <link rel="tags"
          href="/api/hosts/0656f432-923a-11e0-ad20-5254004ac988/tags"/>
        <link rel="permissions"
          href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/permissions"/>
        <link rel="statistics"
          href="/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/statistics"/>
        <address>10.64.14.110</address>
        <status>
            <state>non_operational</state>
        </status>
        <cluster id="99408929-82cf-4dc7-a532-9d998063fa95"
          href="/api/clusters/99408929-82cf-4dc7-a532-9d998063fa95"/>
        <port>54321</port>
        <storage_manager>true</storage_manager>
        <power_management>
            <enabled>false</enabled>
            <options/>
        </power_management>
```

```
        <ksm>
            <enabled>false</enabled>
        </ksm>
        <transparent_hugepages>
            <enabled>true</enabled>
        </transparent_hugepages>
        <iscsi>
            <initiator>iqn.1994-05.com.example:644949fe81ce</initiator>
        </iscsi>
        <cpu>
            <topology cores="2"/>
            <name>Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz</name>
            <speed>2993</speed>
        </cpu>
        <summary>
            <active>0</active>
            <migrating>0</migrating>
            <total>0</total>
        </summary>
    </host>
</hosts>
```

记录 **Default** 主机的 **id** 值。访问虚拟环境中和这个主机相关的其它资源时需要指定这个值。

这个主机是 **Default** 集群的成员，访问 **nics** 子集合会显示这个主机已经连接到 **ovirtmgmt** 网络。

3.6. 示例：列出 CPU 配置集

以下请求会获得 CPU 配置集的表述信息：

例 3.6. 列出 CPU 配置集

请求：

```
GET /api/cpuprofiles HTTP/1.1
Accept: application/xml
```

cURL 命令：

```
# curl -X GET -H "Accept: application/xml" -u [USER:PASS] --cacert
[CERT] https://[RHEVM Host]:443/api/cpuprofiles
```

结果：

```
HTTP/1.1 200 OK
Content-Type: application/xml

<cpu_profiles>
    <cpu_profile href="0000001a-001a-001a-001a-00000000035e"
id="0000001a-001a-001a-001a-00000000035e">
        <name>Default</name>
        <link href="/api/cpuprofiles/0000001a-001a-001a-001a-
00000000035e/permissions" rel="permissions"/>
```

```
        <cluster href= "/api/clusters/00000001-0001-0001-0001-
00000000021b" id="00000001-0001-0001-0001-00000000021b"/>
    </cpu_profile>
    <cpu_profile href="fc4b9188-f87f-44f9-b9c5-c7665e10e0a2"
id="fc4b9188-f87f-44f9-b9c5-c7665e10e0a2">
        <name>Premium</name>
        <description>Full service available</description>
        <link href="/api/cpuprofiles/fc4b9188-f87f-44f9-b9c5-
c7665e10e0a2/permissions" rel="permissions"/>
        <qos href= "/api/datacenters/00000002-0002-0002-0002-
0000000000f7/qoss/5afe49e3-aac4-4b7b-bb83-11b9aef285e1" id="5afe49e3-
aac4-4b7b-bb83-11b9aef285e1"/>
        <cluster href= "/api/clusters/00000001-0001-0001-0001-
00000000021b" id="00000001-0001-0001-0001-00000000021b"/>
    </cpu_profile>
    <cpu_profile href="48c600f4-6768-49ca-9c16-a877d0e586e5"
id="48c600f4-6768-49ca-9c16-a877d0e586e5">
        <name>Budget</name>
        <description>Limited CPU</description>
        <link href="/api/cpuprofiles/48c600f4-6768-49ca-9c16-
a877d0e586e5/permissions" rel="permissions"/>
        <cluster href= "/api/clusters/00000001-0001-0001-0001-
00000000021b" id="00000001-0001-0001-0001-00000000021b"/>
    </cpu_profile>
    <cpu_profile href="48c600f4-6768-49ca-9c16-a877d0e586e5"
id="48c600f4-6768-49ca-9c16-a877d0e586e5">
        <name>Backup</name>
        <link href="/api/cpuprofiles/d510b042-42f0-4cb2-9d2e-
25fcc28d6c5f/permissions" rel="permissions"/>
        <cluster href= "/api/clusters/668cab0c-9185-4eaa-9942-
658284eeecdd" id="668cab0c-9185-4eaa-9942-658284eeecdd"/>
    </cpu_profile>
</cpu_profiles>
```

3.7. 实例：批准主机

hypervisor 主机资源包括了一个**批准（approve）**操作。用户可以使用 **POST** 请求来访问这个操作的 URI。

例 3.7. 批准已经被配置的 Red Hat Enterprise Virtualization Hypervisor 主机

请求：

```
POST /api/hosts/0656f432-923a-11e0-ad20-5254004ac988/approve HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
```

```
    -u [USER:PASS] --cacert [CERT] \
    -d "<action/>" \
    https://[RHEVM Host]:443/api/hosts/0656f432-923a-11e0-ad20-
5254004ac988/approve
```

POST 请求需要一个信息的正文（body）来初始化一个操作。因为这个操作不需要额外的参数，所以信息的内容中包括一个空的 **action** 项。

批准操作只适用于 Red Hat Enterprise Virtualization Hypervisor 主机。Red Hat Enterprise Linux 主机需要使用不同的方法连接到虚拟环境。

这个操作批准并激活了主机，这个主机可以在您的环境中被使用。**hypervisor** 的**状**态从 **non_operational** 变为 **up**。

3.8. 实例：创建 NFS 数据存储

NFS 数据域就是附加到数据中心上的、一个导出的 NFS 共享，它为虚拟机镜像提供存储空间。创建新的存储域需要一个 **POST** 请求（包括了存储域的表述）发送到存储域集合的 URL。

在 Red Hat Enterprise Virtualization 3.6 以及更高的版本中，可以在存储域中默认启用“删除后清除数据”选项。您需要在 **POST** 请求中指定 **<wipe_after_delete>**。这个选项可以在域被创建后进行修改，但这不会改变那些已存在磁盘的“删除后清除数据”的设置。

例 3.8. 创建一个 NFS 数据存储域

请求：

```
POST /api/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_domain>
  <name>data1</name>
  <type>data</type>
  <storage>
    <type>nfs</type>
    <address>192.168.0.10</address>
    <path>/data1</path>
  </storage>
  <host>
    <name>hypervisor</name>
  </host>
</storage_domain>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
    -d "<storage_domain><name>data1</name><type>data</type> \
    <storage><type>nfs</type><address>192.168.0.10</address> \
```

```
    <path>/data1</path></storage> \
    <host><name>hypervisor</name></host></storage_domain>" \
    https://[RHEVM Host]:443/api/storagedomains
```

API 创建了一个 NFS 数据存储域（名为 **data1**，带有一个 **192.168.0.10:/data1** 导出路径），并可以通过 **hypervisor** 主机访问存储域。API 同时返回新建存储域资源的以下表述。

结果：

```
HTTP/1.1 200 OK
Accept: application/xml

<storage_domain id="9ca7cb40-9a2a-4513-acef-dc254af57aac"
  href="/api/storagedomains/9ca7cb40-9a2a-4513-acef-dc254af57aac">
    <name>data1</name>
    <link rel="permissions"
      href="/api/storagedomains/9ca7cb40-9a2a-4513-acef-dc254af57aac/
      permissions"/>
    <link rel="files"
      href="/api/storagedomains/9ca7cb40-9a2a-4513-acef-
dc254af57aac/files"/>
    <type>data</type>
    <master>false</master>
    <storage>
        <type>nfs</type>
        <address>192.168.0.10</address>
        <path>/data1</path>
    </storage>
    <available>175019917312</available>
    <used>27917287424</used>
    <committed>10737418240</committed>
    <storage_format>v1</storage_format>
    <host id="0656f432-923a-11e0-ad20-5254004ac988"
      href="/api/hosts/0656f432-923a-11e0-ad20-5254004ac988">
</storage_domain>
```

3.9. 实例：创建 NFS ISO 存储

NFS ISO 存储域就是附加到数据中心上的、一个挂载的 NFS 共享，它为 DVD/CD-ROM ISO 和虚拟软盘（VFD）镜像提供存储空间。创建新的存储域需要一个 **POST** 请求（包括了存储域的表述）发送到存储域集合的 URL。

在 Red Hat Enterprise Virtualization 3.6 以及更高的版本中，可以在存储域中默认启用“删除后清除数据”选项。您需要在 **POST** 请求中指定 **<wipe_after_delete>**。这个选项可以在域被创建后进行修改，但这不会改变那些已存在磁盘的“删除后清除数据”的设置。

例 3.9. 创建一个 NFS ISO 存储域

请求：

```
POST /api/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

```
<storage_domain>
  <name>iso1</name>
  <type>iso</type>
  <storage>
    <type>nfs</type>
    <address>192.168.0.10</address>
    <path>/iso1</path>
  </storage>
  <host>
    <name>hypervisor</name>
  </host>
</storage_domain>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
    -d "<storage_domain><name>iso1</name><type>iso</type> \
    <storage><type>nfs</type><address>192.168.0.10</address> \
    <path>/iso1</path></storage> \
    <host><name>hypervisor</name></host></storage_domain>" \
    https://[RHEVM Host]:443/api/storagedomains
```

API 创建了一个 NFS iso 存储域（名为 **iso1**，带有一个 **192.168.0.10:/iso1** 导出路径），并可以通过 **hypervisor** 主机访问存储域。API 同时返回新建存储域资源的以下表述。

结果：

```
HTTP/1.1 200 OK
Accept: application/xml

<storage_domain id="00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da"
  href="/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da">
    <name>iso1</name>
    <link rel="permissions"
      href="/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da/
      permissions"/>
    <link rel="files"
      href="/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-
3c898fa8b6da/files"/>
    <type>iso</type>
    <host id="" href="">
    <master>false</master>
    <storage>
        <type>nfs</type>
        <address>192.168.0.10</address>
        <path>/iso1</path>
    </storage>
    <available>82678120448</available>
    <used>18253611008</used>
    <committed>0</committed>
    <storage_format>v1</storage_format>
```

```
    <host id="0656f432-923a-11e0-ad20-5254004ac988"
      href="/api/hosts/0656f432-923a-11e0-ad20-5254004ac988">
</storage_domain>
```

3.10. 实例：为数据中心附加存储域

以下实例把 **data1** 和 **iso1** 存储域附加到 **Default** 数据中心。

例 3.10. 为 Default 数据中心附加 data1 存储域

请求：

```
POST /api/datacenters/01a45ff0-915a-11e0-8b87-
5254004ac988/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_domain>
  <name>data1</name>
</storage_domain>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
    -d "<storage_domain><name>data1</name></storage_domain>" \
    https://[RHEVM Host]:443/api/datacenters/01a45ff0-915a-11e0-8b87-
5254004ac988/storagedomains
```

例 3.11. 为 Default 数据中心附加 iso1 存储域

请求：

```
POST /api/datacenters/01a45ff0-915a-11e0-8b87-
5254004ac988/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_domain>
  <name>iso1</name>
</storage_domain>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
```

```
    -d "<storage_domain><name>iso1</name></storage_domain>" \
    https://[RHEVM Host]:443/api/datacenters/01a45ff0-915a-11e0-8b87-
5254004ac988/storagedomains
```

这些 **POST** 请求在 **Default** 数据中的 **storagedomains** 子集合中放置两个新 **storage_domain** 资源。这意味着 **storagedomains** 子集合包括了附加到数据中心中的存储域。

3.11. 实例：激活存储域

这个实例为 Red Hat Enterprise Virtualization Manager 激活 **data1** 和 **iso1** 存储域。

例 3.12. 激活 data1 存储域

请求：

```
POST /api/datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains/
9ca7cb40-9a2a-4513-acef-dc254af57aac/activate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
    -d "<action/>" \
    https://[RHEVM Host]:443/api/datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains/9ca7cb40-9a2a-4513-acef-
dc254af57aac/activate
```

例 3.13. 激活 iso1 存储域

请求：

```
POST /api/datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains/
00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da/activate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
```

```
    -d "<action/>"
    https://[RHEVM Host]:443/api/datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains/00f0d9ce-da15-4b9e-9e3e-
3c898fa8b6da/activate
```

这个实例激活了两个存储域，现在它们可以在数据中心中被使用。

3.12. 实例：创建虚拟机

以下实例使用虚拟环境中的 **Blank** 模板作为基础，在 **Default** 集群中创建了一个名为 **vm1** 的虚拟机。这个请求还定义了虚拟机的**内存**为 512 MB，并把一个虚拟硬盘指定为**引导**设备。

例 3.14. 创建一个虚拟机

请求：

```
POST /api/vms HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm>
    <name>vm1</name>
    <cluster>
        <name>default</name>
    </cluster>
    <template>
        <name>Blank</name>
    </template>
    <memory>536870912</memory>
    <os>
        <boot dev="hd"/>
    </os>
    <cpu_profile id="0000001a-001a-001a-001a-00000000035e"/>
</vm>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" -u [USER:PASS] --cacert [CERT] -d "<vm>
<name>vm1</name><cluster><name>default</name></cluster><template>
<name>Blank</name></template><memory>536870912</memory><os><boot
dev='hd'/></os><cpu_profile id='0000001a-001a-001a-001a-00000000035e'/>
</vm>" https://[RHEVM Host]:443/api/vms
```

结果：

```
HTTP/1.1 200 OK
Accept: application/xml

<vm id="6efc0cfa-8495-4a96-93e5-ee490328cf48"
  href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48">
    <name>vm1</name>
    <actions>
```

```
        <link rel="shutdown"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/shutdown"/>
        <link rel="start"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/start"/>
        <link rel="stop"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/stop"/>
        <link rel="reboot"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/reboot"/>
        <link rel="suspend"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/suspend"/>
        <link rel="detach"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/detach"/>
        <link rel="export"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/export"/>
        <link rel="move"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/move"/>
        <link rel="ticket"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/ticket"/>
        <link rel="migrate"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/migrate"/>
        <link rel="undo_snapshot"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/undo_snapshot"/>
        <link rel="commit_snapshot"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/commit_snapshot"/>
        <link rel="preview_snapshot"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/preview_snapshot"/>
        <link rel="logon"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/logon"/>
        <link rel="cancelmigration"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/cancelmigration"/>
        <link rel="maintenance"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/maintenance"/>
        <link rel="clone"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/clone"/>
    </actions>
    <link rel="applications"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/applications"/>
    <link rel="disks"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/disks"/>
    <link rel="nics"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/nics"/>
    <link rel="cdroms"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/cdroms"/>
    <link rel="snapshots"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/snapshots"/>
    <link rel="tags"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/tags"/>
```

```
    <link rel="permissions"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/permissions"/>
    <link rel="statistics"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/statistics"/>
    <link rel="reporteddevices"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/reporteddevices"/>
    <link rel="watchdogs"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/watchdogs"/>
    <link rel="sessions"
      href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/sessions"/>
    <type>desktop</type>
    <status>
        <state>down</state>
    </status>
    <memory>536870912</memory>
    <cpu>
        <topology cores="1" sockets="1"/>
    </cpu>
    <os type="Unassigned">
        <boot dev="cdrom"/>
    </os>
    <high_availability>
        <enabled>false</enabled>
        <priority>0</priority>
    </high_availability>
    <display>
        <type>spice</type>
        <monitors>1</monitors>
        <single_qxl_pci>false</single_qxl_pci>
        <allow_override>false</allow_override>
        <smartcard_enabled>false</smartcard_enabled>
        <file_transfer_enabled>true</file_transfer_enabled>
        <copy_paste_enabled>true</copy_paste_enabled>
    </display>
    <cluster id="99408929-82cf-4dc7-a532-9d998063fa95"
      href="/api/clusters/99408929-82cf-4dc7-a532-9d998063fa95"/>
    <template id="00000000-0000-0000-0000-000000000000"
      href="/api/templates/00000000-0000-0000-0000-000000000000"/>
    <stop_time>2011-06-15T04:48:02.167Z</stop_time>
    <creation_time>2011-06-15T14:48:02.078+10:00</creation_time>
    <origin>rhev</origin>
    <stateless>false</stateless>
    <delete_protected>false</delete_protected>
    <sso>
        <methods>
            <method id="GUEST_AGENT"/>
        </methods>
    </sso>
    <console enabled="false"/>
    <timezone>Etc/GMT</timezone>
    <initialization>
        <configuration>
            <type>ovf</type>
            <data>...</data>
        </configuration>
```

```
    </initialization>
    <placement_policy>
        <affinity>migratable</affinity>
    </placement_policy>
    <memory_policy>
        <guaranteed>536870912</guaranteed>
        <ballooning>true</ballooning>
    </memory_policy>
    <usb>
        <enabled>false</enabled>
    </usb>
    <soundcard_enabled>true</soundcard_enabled>
    <migration_downtime>-1</migration_downtime>
    <virtio_scsi enabled="true"/>
    <cpu_profile id="0000001a-001a-001a-001a-00000000035e"/>
    <next_run_configuration_exists>false</next_run_configuration_exists>
    <numa_tune_mode>interleave</numa_tune_mode>
</vm>
```

3.13. 实例：创建虚拟机 NIC

以下实例创建了一个虚拟网络接口（NIC）来把虚拟机连接到 **ovirtmgmt** 网络。

例 3.15. 创建一个虚拟机 NIC

请求：

```
POST /api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/nics HTTP/1.1
Accept: application/xml
Content-type: application/xml

<nic>
  <interface>virtio</interface>
  <name>nic1</name>
  <network>
    <name>ovirtmgmt</name>
  </network>
</nic>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
    -d "<nic><name>nic1</name><network><name>ovirtmgmt</name></network>
</nic>" \
    https://[RHEVM Host]:443/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/nics
```

3.14. 实例：创建虚拟机存储磁盘

以下实例为虚拟机创建了一个 8 GB Copy-On-Write 存储磁盘。

例 3.16. 创建一个虚拟机存储磁盘

请求：

```
POST /api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/disks HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk>
    <storage_domains>
        <storage_domain id="9ca7cb40-9a2a-4513-acef-dc254af57aac"/>
    </storage_domains>
    <size>8589934592</size>
    <type>system</type>
    <interface>virtio</interface>
    <format>cow</format>
    <bootable>true</bootable>
</disk>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
    -d "<disk><storage_domains> \
    <storage_domain id='9ca7cb40-9a2a-4513-acef-dc254af57aac'/> \
    </storage_domains><size>8589934592</size><type>system</type> \
    <interface>virtio</interface><format>cow</format> \
    <bootable>true</bootable></disk>" \
    https://[RHEVM Host]:443/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/disks
```

storage_domain 项告知 API 在 **data1** 存储域中保存磁盘。

3.15. 实例：为虚拟机附加 ISO 镜像

安装虚拟机操作系统时需要一个 CD-ROM 或 DVD ISO 镜像作为操作系统的安装介质。在这里我们使用 Red Hat Enterprise Serverr 6 ISO 镜像作为安装介质。

ISO 镜像需要存在于 **iso1** ISO 域中以供虚拟机使用。Red Hat Enterprise Virtualization 提供了一个上传工具来确保 ISO 镜像被上传到正确的目录中，并有正确的用户权限。

当 ISO 被上传后，API 用户就可以请求 ISO 存储域的 **files** 子集合来查看文件资源：

例 3.17. 在一个 ISO 存储域中查看文件子集合

请求：

```
GET /api/storagedomains/00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da/files
HTTP/1.1
Accept: application/xml
```

cURL 命令：

```
# curl -X GET -H "Accept: application/xml" -u [USER:PASS] --cacert
[CERT] \
    https://[RHEVM Host]:443/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-
3c898fa8b6da/files
```

API 返回文件子集合的以下表述：

```
<files>
    <file id="rhel-server-6.0-x86_64-dvd.iso"
      href="/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da/
      files/rhel-server-6.0-x86_64-dvd.iso.iso">
        <name>rhel-server-6.0-x86_64-dvd.iso.iso</name>
        <storage_domain id="00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da"
          href="/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-
3c898fa8b6da"/>
    </file>
</files>
```

API 用户把 **rhel-server-6.0-x86_64-dvd.iso** 附加到示例中使用的虚拟机。附加一个 ISO 镜像等同于使用管理门户或用户门户中的**更换 CD** 按钮。

例 3.18. 为虚拟机附加一个 ISO 镜像

请求：

```
POST /api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/cdroms HTTP/1.1
Accept: application/xml
Content-type: application/xml

<cdrom>
  <file id="rhel-server-6.0-x86_64-dvd.iso"/>
</cdrom>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
    -d "<cdrom><file id='rhel-server-6.0-x86_64-dvd.iso'/></cdrom>" \
    https://[RHEVM Host]:443/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/cdroms
```

3.16. 实例：启动虚拟机

一个虚拟环境已经被创建，虚拟机包括了正常操作所需的所有组件。这个实例使用 **start** 操作启动虚拟机。

例 3.19. 启动虚拟机

请求：

```
POST /api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48/start HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
  <vm>
    <os>
      <boot dev="cdrom"/>
    </os>
  </vm>
</action>
```

cURL 命令：

```
# curl -X POST -H "Accept: application/xml" -H "Content-Type:
application/xml" \
    -u [USER:PASS] --cacert [CERT] \
    -d "<action><vm><os><boot dev='cdrom'/></os></vm></action>" \
    https://[RHEVM Host]:443/api/vms/6efc0cfa-8495-4a96-93e5-
ee490328cf48/start
```

额外的信息项指定了虚拟机本次启动的引导设备是 CD-ROM，这使虚拟机可以使用附加的 ISO 镜像安装 Red Hat Enterprise Server 6。以后的启动都将重新使用 **disk** 作为引导设备。

3.17. 实例：检查系统事件

对 **vm1** 的 **start** 操作会在 **events** 集合中创建多个项。这个实例列出了 events 集合中的内容，并找出针对于 API 启动一个虚拟机的事件信息。

例 3.20. 列出事件集合

请求：

```
GET /api/events HTTP/1.1
Accept: application/xml
```

cURL 命令：

```
# curl -X GET -H "Accept: application/xml" -u [USER:PASS] \
    --cacert [CERT] \
    https://[RHEVM Host]:443/api/events
```

结果：

```
<events>
    ...
    <event id="103" href="/api/events/103">
```

```
        <description>User admin logged out.</description>
        <code>31</code>
        <severity>normal</severity>
        <time>2011-06-29T17:42:41.544+10:00</time>
        <user id="80b71bae-98a1-11e0-8f20-525400866c73"
          href="/api/users/80b71bae-98a1-11e0-8f20-525400866c73"/>
    </event>
    <event id="102" href="/api/events/102">
        <description>vm1 was started by admin (Host: hypervisor).
</description>
        <code>153</code>
        <severity>normal</severity>
        <time>2011-06-29T17:42:41.499+10:00</time>
        <user id="80b71bae-98a1-11e0-8f20-525400866c73"
          href="/api/users/80b71bae-98a1-11e0-8f20-525400866c73"/>
        <vm id="6efc0cfa-8495-4a96-93e5-ee490328cf48"
          href="/api/vms/6efc0cfa-8495-4a96-93e5-ee490328cf48"/>
        <host id="0656f432-923a-11e0-ad20-5254004ac988"
          href="/api/hosts/0656f432-923a-11e0-ad20-5254004ac988"/>
    </event>
    <event id="101" href="/api/events/101">
        <description>User admin logged in.</description>
        <code>30</code>
        <severity>normal</severity>
        <time>2011-06-29T17:42:40.505+10:00</time>
        <user id="80b71bae-98a1-11e0-8f20-525400866c73"
          href="/api/users/80b71bae-98a1-11e0-8f20-525400866c73"/>
    </event>
    ...
</events>
```

从结果中可以知道发生了以下事件：

- **id="101"** - API 使用 **admin** 用户的用户名和密码进行验证。
- **id="102"** - API 以 **admin** 用户身份在 **hypervisor** 主机上启动 **vm1**。
- **id="103"** - API 登出 **admin** 用户帐号。

第 4 章 进入点

用户通过一个到进入点 URI（由 **host** 和 **base** 组成）的 **GET** 请求来开始使用 API。

例 4.1. 访问 API 进入点

以下请求包括了一个进入点（**host** 是 **www.example.com**，**base** 是 **/api**）：

```
GET /api HTTP/1.1
Accept: application/xml
Host: www.example.com
Authorization: [base64 encoded credentials]

HTTP/1.1 200 OK
Content-Type: application/xml

<api>
    <link rel="hosts" href="/api/hosts"/>
    <link rel="vms" href="/api/vms"/>
    ...
    <product_info>
        <name>Red Hat Enterprise Virtualization</name>
        <vendor>Red Hat</vendor>
        <version revision="0" build="0" minor="1" major="3"/>
    </product_info>
    <special_objects>
        <link rel="templates/blank" href="..."/>
        <link rel="tags/root" href="..."/>
    </special_objects>
    <summary>
        <vms>
            <total>10</total>
            <active>3</active>
        </vms>
        <hosts>
            <total>2</total>
            <active>2</active>
        </hosts>
        <users>
            <total>8</total>
            <active>2</active>
        </users>
        <storage_domains>
            <total>2</total>
            <active>2</active>
        </storage_domains>
    </summary>
</api>
```

注意

为了简化，所有其它实例都会省略 **Host:** 和 **Authorization:** 请求头，并假设 **base** 是默认的 **/api** 路径。base 路径的实际值由您的具体情况所决定。

4.1. 产品信息

进入点有一个 **product_info** 项，它包括了产品的 **name**、**vendor** 和 **version**，用户可以使用这些信息确定 Red Hat Enterprise Virtualization 环境的真实性。

例 4.2. 验证 Red Hat Enterprise Virtualization 环境的真实性

以下项代表了一个真实的 Red Hat Enterprise Virtualization 3.2 环境：

```
<api>
    ...
    <product_info>
        <name>Red Hat Enterprise Virtualization</name>
        <vendor>Red Hat</vendor>
        <version revision="0" build="0" minor="2" major="3"/>
    </product_info>
    ...
</api>
```

4.2. 连接项

访问进入点会获得 API 可以使用的资源集合的 **link**（连接）项和 URI，每个集合都使用关系类型来指定客户端需要的 URI。

表 4.1. 有效的关系类型

关系	描述
capabilities	支持的 Red Hat Enterprise Virtualization Manager 能力。
datacenters	数据中心
clusters	主机集群
networks	虚拟网络
storagedomains	存储域
hosts	主机
vms	虚拟机
disks	虚拟机磁盘
templates	模板
vmpools	虚拟机池
domains	服务域

关系	描述
`groups`	导入的服务组
`roles`	角色
`users`	用户
`tags`	标签
`events`	事件

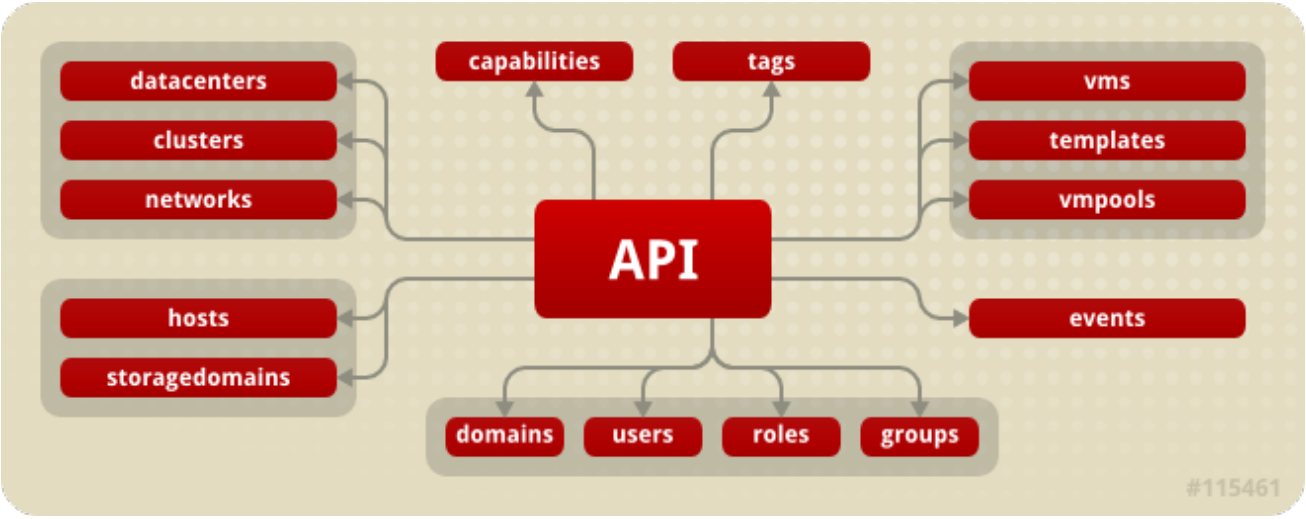
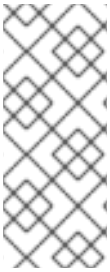


图 4.1. API 进入点和 API 访问的资源集合的关系



注意

这个实例中所使用的 URI 只作为演示，服务器实际返回的 URI 的格式会和实例中的 URI 不同。客户端通过进入点指定资源，并使用关系类型访问 URI。

服务器会在 `link` 项的 `href` 属性中使用绝对 URI 或绝对路径 [3]，所以客户端需要支持这两种形式。

`link` 项也为特定集合包括了一组 `search` URI，这些 URI 使用 URI 模板[4] 来集成搜索查询。URI 模板的目的是接受使用查询参数的自然 HTTP 特征的查询表述。客户端不需要预先知道 URI 的结构，但是需要使用 URI 模板库来访问这些模板。

每个搜索查询 URI 模板都使用关系类型来代表（格式是 `"collection/search"`）。

表 4.2. 与搜索查询 URI 相关的关系

关系	描述
`datacenters/search`	查询数据库

关系	描述
clusters/search	查询主机集群
storagedomains/search	查询存储域
hosts/search	查询主机
vms/search	查询虚拟机
disks/search	查询磁盘
templates/search	查询模板
vmpools/search	查询虚拟机池
events/search	查询事件
users/search	查询用户

4.3. 特殊项

特殊项（special_object）定义了虚拟环境中的特殊资源。

表 4.3. 特殊项

关系	描述
templates/blank	虚拟环境中的默认 **blank** 虚拟机模板。这个模板会存在于每个集群中，而标准模板只存在于某个特定的集群中。
tags/root	**root** 标签是虚拟环境中的标签分级机构的基础点。

4.4. 概述项

概述项（summary）显示了系统的统计数据概况。

表 4.4. 概述项

元素	描述
vms	虚拟机的总数，以及活跃虚拟机的总数。
hosts	主机的总数，以及活跃主机的总数。
users	用户的总数，以及活跃用户的总数。

元素	描述
storage_domains	存储域的总数，以及活跃存储域的总数。

4.5. RESTFUL SERVICE DESCRIPTION LANGUAGE（RSDL）

RESTful Service Description Language（RSDL）提供了在整个 XML 文件中的 REST API 中的结构和项描述。使用以下请求来调用 RSDL。

```
GET /api?rsdl HTTP/1.1
Accept: application/xml
```

产生以下格式的 XML 文件：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rsdl href="/api?rsdl" rel="rsdl">
    <description>...</description>
    <version major="3" minor="1" build="0" revision="0"/>
    <schema href="/api?schema" rel="schema">
        <name>...</name>
        <description>...</description>
    </schema>
    <links>
        <link href="/api/capabilities" rel="get">
            ...
        </link>
        ...
    </links>
</rsdl>
```

表 4.5. RSDL 结构项

项	描述
description	RSDL 文件的描述。
version	API 版本（包括 **major** 版本、**minor** 版本、**build** 和 **revision**）。
schema	到 XML schema（XSD）文件的链接。
links	定义 API 中的每个 **link**。

每个 **link** 项包括以下结构：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rsdl href="/api?rsdl" rel="rsdl">
    ...
    <links>
        <link href="/api/..." rel="...">
            <request>
```

```
                <http_method>...</http_method>
                <headers>
                    <header>
                        <name>...</name>
                        <value>...</value>
                    </header>
                    ...
                </headers>
                <body>
                    <type>...</type>
                    <parameters_set>
                        <parameter required="..." type="...">
                            <name>...</name>
                        </parameter>
                        ...
                    </parameters_set>
                </body>
            </request>
            <response>
                <type>...</type>
            </response>
        </link>
        ...
    </links>
</rsdl>
```

表 4.6. RSDL 连接结构项

项	描述
link	API 请求的 URI，它包括一个 URI 属性（**href**）和一个关系类型属性（**rel**）。
request	定义了连接所需的请求属性。
http_method	访问这个连接的方法类型，它包括 REST API 访问的标准 HTTP 方法：**GET**、**POST**、**PUT** 和 **DELETE**。
headers	定义 HTTP 请求的头。它包括一组 **header** 项，每个项中包括了一个头 **name** 和 **value** 来定义这个头。
body	定义 HTTP 请求的内容。它包括一个资源 **type** 和 **parameter_set**（包括一组 **parameter** 项）。每个 **parameter** 项都包括了一个属性来指定这个项对于请求和数据 **type** 是否是**必需的**；这个项还包括了一个 **name** 项来定义要修改的 Red Hat Enterprise Virtualization Manager 属性。另外，如果 **type** 被设置为 **collection**，还会有一个 **parameter_set** 子集。
response	定义 HTTP 请求的输出。它包括一个 **type** 项来定义输出的资源结构。

在应用程序中使用 RSDL 作为一个方法，来映射控制 Red Hat Enterprise Virtualization 环境所需的所有连接和参数。

4.6. RED HAT ENTERPRISE VIRTUALIZATION WINDOWS 客户端 VSS 支持

Red Hat Enterprise Virtualization 备份和恢复 API 使用 **qemu-ga** 与 Microsoft Windows Volume Shadow Copy Service（VSS）进行集成。作为 Guest Tools 实施的一部分，VSS 的注册功能在客户端一级实现。

qemu-ga 提供了 VSS 支持。

4.7. QEMU GUEST AGENT 介绍

在 Red Hat Enterprise Linux 6.4 中，QEMU Guest Agent（QEMU GA）提供了 Linux 虚拟机的数据破坏保护功能。在发送一个快照请求或创建磁盘备份前，libvirt 会通过 virtio-serial 端口向 QEMU GA 发送一个 **`guest-fsfreeze-freeze QMP`** 命令。这个命令会使用 **`FIFREEZE ioctl()`** 内核功能冻结虚拟机的所有文件系统。**`ioctl()`** 功能是通过虚拟机的 Linux 内核实现的。它会清除虚拟机内核中的文件缓存，并拒绝所有用户空间线程对文件系统的写请求。

只有在 **QEMU GA** 报告命令已经成功完成后，**libvirt** 才会进行快照操作。当快照操作完成后，**libvirt** 通过 virtio-serial 端口发送 **`guest-fsfreeze-thaw QMP`** 命令到 **QEMU GA**。这个命令告诉 **QEMU GA** 使用 **`FITHAW ioctl()`** 取消以前对用户空间线程的写请求的封锁，并恢复正常操作。这个过程无法保证，在进行虚拟磁盘快照时应用程序一级数据处于一致的状态。因此，在某些情况下，虽然 fsck 程序没有发现通过恢复快照所获得的文件系统有问题，但应用程序仍然可能还无法从快照的创建点恢复操作，用户空间进程可能也还没有把它们内部的缓冲区数据写入到磁盘上的文件中。

Red Hat Enterprise Linux 6.5 允许客户端虚拟机系统的管理员开发并安装针对于特定应用程序的冻结/解除冻结操作的 hook 脚本，从而可以确保数据在文件和应用程序一级的同步。在冻结文件系统前，**QEMU GA** 会调用一个主 hook 脚本（它包括在 **QEMU GA** 软件包中）。这个主 hook 脚本会调用单独的、针对于特定应用程序的脚本（由客户端虚拟机系统的管理员提供）来暂时停止所有客户端虚拟机的应用程序。这些操作会在模式变为 "freeze" 后发生。

在文件系统被冻结前，客户端虚拟机系统管理员的脚本会把数据库和其它文件系统应用程序的工作缓冲区数据移到虚拟磁盘，并停止接受其它客户连接。然后，应用程序会把它们的数据文件变为一个一致的状态，这将使应用程序可以在虚拟磁盘数据恢复完成后恢复正常运行。当所有的脚本都完成了对相关应用程序的操作后，主 hook 脚本会返回，**QEMU GA** 将冻结文件系统，管理软件将对系统进行快照。在这些操作都完成后，文件系统将会重新开始接受写请求。这个步骤被称为取消冻结。

取消冻结的过程和冻结的过程完成相反。**libvirt** 要求 **QEMU GA** 取消解冻客户端虚拟机的文件系统。然后，它会通过主 hook 脚本调用独立的 hook 脚本来恢复在冻结过程中被暂停的应用程序。

4.8. VSS 工作流程

在处理备份时，请求者（requester）和写操作之间需要进行协调来完成以下操作：提供一个稳定的系统镜像来备份数据；根据使用情况对文件进行分组；存储需要被保持的数据信息。在进行这些操作时，需要对正常的写操作有尽可能小的影响。

请求者（在这里是 Backup Vendor）会查询写操作的元数据、处理这个数据、并在开始进行备份操作前通知写操作进程。当备份完成后，再次通知写操作进程。

以下显示了在安装完 Guest Tools 程序后， QEMU VSS 如何在 Windows 操作系统中注册：

```
C:\Users\Administrator>vssadmin list providers
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2005 Microsoft Corp.

Provider name: 'QEMU Guest Agent VSS Provider'
   Provider type: Software
```

```
    Provider Id: {3629d4ed-ee09-4e0e-9a5c-6d8ba2872aef}
    Version: 0.12.1
```

[3] The RFC describing Uniform Resource Locator Generic Syntax provides a Collected ABNF for URI that explains the difference between these forms.

[4] The Internet-Draft describing the format of a URI Template is available at http://tools.ietf.org/html/draft-gregorio-uritemplate-03.

第 5 章 兼容级别版本

每个连接到 Red Hat Enterprise Virtualization Manager 的主机都包括一个 VDSM 版本。VDSM 是虚拟环境中的一个代理程序，它运行在 hypervisor 或主机上，为虚拟机、网络和存储提供本地管理。Red Hat Enterprise Virtualization Manager 可以使用最新的 VDSM 版本来控制 hypervisor 和主机，也可以使用老的 VDSM 版本。

Manager 需要在一个集群的不同主机间迁移虚拟机，这意味着如果同一个集群中的主机安装了不同版本的 VDSM，其中所安装的最新 VDSM 版本的一些功能可能无法使用。

API 使用 **compatibility level** 来代表每个主机所安装的 VDSM 版本。**version** 包括了 **major** 和 **minor** 这两个与兼容级别相关的属性。

当管理员把集群中的所有主机都升级到特定级别时，**version** 级别值会出现在 **supported_versions** 项中。它代表了集群 **version** 现在可以被更新到的级别。当管理员把一个数据中心中的所有集群都更新到一个特定级别时，数据中心就可以升级到那个级别。

5.1. 升级兼容级别

例 5.1. 升级兼容级别

API 显示 Red Hat Enterprise Virtualization Manager 3.4 的以下兼容级别：

```
<host ...>
    ...
    <version major="4" minor="14" build="11" revision="0"
full_version="vdsm-4.14.11-5.el6ev"/>
    ...
</host>

<cluster ...>
    ...
    <version major="3" minor="4"/>
    ...
</cluster>

<data_center ...>
    ...
    <version major="3" minor="4"/>
    </supported_versions>
    ...
</data_center>
```

一个集群中的所有主机都被更新到 **3.5**，API 显示：

```
<host ...>
    ...
    <version major="4" minor="16" build="7" revision="4"
full_version="vdsm-4.16.7.4-1.el6ev"/>
    ...
</host>

<cluster ...>
```

```
    ...
    <version major="3" minor="4"/>
    <supported_versions>
        <version major="3" minor="5"/>
    </supported_versions>
    ...
</cluster>

<data_center ...>
    ...
    <version major="3" minor="4"/>
    <supported_versions/>
    ...
</data_center>
```

集群被升级到 **3.5**。当集群升级完成后，API 显示：

```
<cluster ...>
    ...
    <version major="3" minor="5"/>
    <supported_versions/>
    ...
</cluster>

<data_center ...>
    ...
    <version major="3" minor="4"/>
    <supported_versions>
        <version major="3" minor="5"/>
    </supported_versions>
    ...
</data_center>
```

API 用户把数据中心升级到 **3.5**。当升级完成后，API 显示这个数据中心可用的、Red Hat Enterprise Virtualization 3.5 的功能。

第 6 章 能力

capabilities 集合提供了所使用的 Red Hat Enterprise Virtualization 支持的能力。这些能力包括活跃的功能，以及特定属性的完整有效值列表。

要获取从 Red Hat Enterprise Virtualization 版本 3.2 到最新版本的完整能力列表，发送以下请求：

```
GET /api/capabilities/ HTTP/1.1
Content-Type: application/xml
Accept: application/xml
```

6.1. 与特定版本相关的能力

capabilities 项中包括了 **version** 项，它显示了和一个兼容级别相关的能力。

version 项包括了 **major** 和 **minor** 版本号属性，它们标识了当前的版本级别。

以下表述分别显示了 Red Hat Enterprise Virtualization Manager **3.0**、**3.1**、**3.2**、**3.3**、**3.4**、**3.5** 和 **3.6** 的能力：

```
<capabilities>
    <version major="3" minor="0">
        ...
    </version>
    <version major="3" minor="1">
        ...
    </version>
    <version major="3" minor="2">
        ...
    </version>
    <version major="3" minor="3">
        ...
    </version>
    <version major="3" minor="4">
        ...
    </version>
    <version major="3" minor="5">
        ...
    </version>
    <version major="3" minor="6">
        ...
    </version>
    ...
</capabilities>
```

每个 **version** 项包括了一组和这个版本相关的能力。

6.2. 当前版本

current 项指定了 **version** 的值是否是兼容级别支持的最新版本。这个项的值是一个布尔值（**true** 或 **false**）。

```
<capabilities>
```

```
    <version major="3" minor="5">
        ...
        <current>true</current>
        ...
    </version>
</capabilities>
```

6.3. 功能

每个版本都包括了一组兼容的功能，以下表格显示了与 Red Hat Enterprise Virtualization 3.6 相兼容的功能。

表 6.1. 功能类型

功能	描述
透明超大内存页策略	允许您为主机指定有效的透明超大内存页。它的值是 **`true`** 或 **`false`**。
Gluster 支持	这个功能支持使用 Gluster 卷（Volume）和数据块（Brick）作为内存。
POSIX-FS 存储类型	这个功能提供了对 POSIX-FS 存储类型的支持。
端口镜像	允许您指定在虚拟网络接口卡上是否支持端口镜像。它的值是 **`true`** 或 **`false`**。
显示服务器时间	在 API 中显示当前的日期和时间。
显示主机内存	显示特定主机的总内存数。
显示主机的插槽数	允许您定义一个主机 CPU 的拓扑结构。它包括 3 个属性 - **`sockets`**（显示的主机插槽数）、**`threads`**（线程数）和 **`cores`**（每个插槽的内核数）。
搜索条件是否区分大小写	允许您指定搜索查询是否区分大小写（使用 **`case-sensitive=`*`true`*`|`*`false`*** URL 参数）。
GET 请求的最大结果	允许您指定一个 **GET** 请求可以返回的最大结果数。
JSON 内容类型	允许您定义一个头来为 **POST** 和 **PUT** 请求设置一个相关性 ID。
激活和取消激活磁盘	允许您通过在特定磁盘上指定 **`activate`** 或 **`deactivate`** 操作来激活或取消激活它。
激活和取消激活网络接口卡	允许您通过在特定网络接口卡上指定 **`activate`** 或 **`deactivate`** 操作来激活或取消激活它。
快照重构	允许您为虚拟机重构快照。

功能	描述
从特定存储域中删除模板磁盘	允许您使用 **DELETE** 请求来从一个特定存储域中删除虚拟机模板磁盘。
浮动磁盘	浮动磁盘就是那些没有被附加到任何虚拟机上的磁盘。在这个功能中，浮动磁盘存在于 root 集合中，而不是存在于某些特定虚拟机中。
异步删除	允许您通过使用 **async** URL 参数来指定 **DELETE** 请求以异步的形式执行。
基于会话的验证	通过提供一个适当的头来保持客户端/服务器间的会话，从而可以不需要在每次请求时都进行登录验证。
虚拟机应用程序	允许您查看特定虚拟机上安装的应用程序列表。这个列表位于指定虚拟机的 **applications** 项中。
VirtIO-SCSI 支持	这个功能支持准虚拟化 SCSI 控制器设备。
自定义资源注释	允许您为数据中心和其它资源添加自定义注释。
刷新主机能力	允许您同步主机上的数据，并刷新对特定主机有效的网络接口列表。
内存快照	允许您在快照中包括内存的状态。
Watchdog 设备	允许您为虚拟机创建 watchdog 设备。
SSH 验证方法	允许您指定通过 SSH 和主机验证时使用管理员用户的密码还是使用 SSH 公共密钥。
强制选择 SPM	允许您强制选择一个主机作为 SPM。
控制台设备	允许您控制附加到虚拟机上的控制台设备。
存储域的存储服务器连接	允许您查看和特定存储域相连接的存储服务器。
附加或取消存储服务器的连接	允许您附加或取消存储服务器和特定存储域的连接。
Qxl 的单独 PCI	允许您通过一个单独的 PCI 客户设备查看多个视频设备。
从 OVF 配置中添加虚拟机	允许您从一个提供的 OVF 配置中添加虚拟机。
虚拟网络接口卡配置集	允许您配置一个配置集来为特定的虚拟网络接口卡定义服务质量、自定义属性和端口镜像。

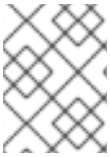
功能	描述
镜像存储域（技术预览）	允许您从镜像存储域（如 OpenStack 镜像服务 Glance）导入镜像，或向存储域输出镜像。
虚拟机全称域名（FQDN）	允许您获取一个指定虚拟机的全称域名。
为虚拟机附加磁盘快照	这个功能支持为虚拟机附加磁盘快照。
Cloud-Init	允许您使用 Cloud Init 初始化一个虚拟机。
Gluster 数据库管理	允许您使用 **`migrate`** 和 **`DELETE`** 操作在数据迁移时删除 Gluster 数据块。**`migrate`** 操作和 **`stopmigrate`** 操作允许您迁移数据后重复使用数据块。
复制并移动后台磁盘	允许您复制并移动磁盘。
网络标签	允许您在使用标签的主机上配置网络。
重启虚拟机	允许您通过一个单独的操作重启虚拟机。

完整的功能项和它们的属性列表位于相关版本数据段的上面。

```
<capabilities>
    <version major="3" minor="4">
        ...
        <features>
            <feature>
             <name>Transparent-Huge-Pages Memory Policy</name>
             <transparent_huepages/>
            </feature>
        </features>
        ...
    </version>
</capabilities>
```

第 7 章 通用功能

7.1. 项属性图标



注意

在本文档中，每个资源的项会在相应的表中进行说明。这些表中会包括一个属性栏，它会显示项属性图标。这些图标的含义包括在 表 7.1 “项属性图标” 中。

表 7.1. 项属性图标

属性	描述	图标
创建时需要	这些项必须包括在客户端所提供的、创建资源的表述中，但在更新资源时不是必需的。	
不能更新	这些项的值不能在更新资源时被修改。只有在 API 用户不修改它们的情况下才可以把这些项包括在客户端提供的表述中，否则 API 会报告一个错误。	
只读	这些项是只读的。只读项的值不能被创建或修改。	

7.2. 表述

7.2.1. 表述

API 使用以下的 XML 文档结构构建资源表述：

```
<resource id="resource_id" href="/api/collection/resource_id">
    <name>Resource-Name</name>
    <description>A description of the resource</description>
    ...
</resource>
```

以一个虚拟机为例，它的表述会以以下形式出现：

```
<vm id="5b9bbce5-0d72-4f56-b931-5d449181ee06"
  href="/api/vms/5b9bbce5-0d72-4f56-b931-5d449181ee06">
    <name>RHEL6-Machine</name>
    <description>Red Hat Enterprise Linux 6 Virtual Machine</description>
    ...
</vm>
```

7.2.2. 资源表述的通用属性

所有资源表述都包括了一组通用的属性

表 7.2. 资源表述的通用属性

属性	类型	描述	属性
id	GUID	虚拟环境中的每个资源都包括一个 **id**，它被作为一个全局唯一标识（GUID）使用。GUID 是代表资源的主要方法。	
href	字符串	作为绝对路径的资源的位置。	

7.2.3. 资源表述的通用项

所有资源表述都包括了一组通用项。

表 7.3. 资源表述的通用项

项	类型	描述	属性
name	字符串	用户提供的资源名。在一个类型中，每个资源的 **name** 必须是是唯一的。	
description	字符串	用户提供的、对资源的描述信息。	

7.3. 集合

7.3.1. 集合

集合就是相同类型的一组资源。API 提供了顶级集合和子集合。**hosts** 集合是一个顶级集合的例子，它包括了虚拟环境中的所有虚拟主机。**host.nics** 集合是子集合的一个例子，它包括了附加到一个主机资源上的所有网络接口卡。

7.3.2. 列出一个集合中的所有资源

使用一个从进入点获得的集合 URI 的 **GET** 请求来得到一个集合中的资源列表。

包括一个 **Accept** HTTP 头来定义响应格式的 MIME 类型。

```
GET /api/[collection] HTTP/1.1
Accept: [MIME type]
```

7.3.3. 列出扩展的资源子集合

当 **Accept** 头包括了 **detail** 参数时，API 会扩展它的集合表述来包括子集合信息。

```
GET /api/collection HTTP/1.1
Accept: application/xml; detail=subcollection
```

使用多个 **detail** 参数来包括多个子集合请求：

```
GET /api/collection HTTP/1.1
Accept: application/xml; detail=subcollection1; detail=subcollection2
```

或在一个 **detail** 参数中包括由 **+** 操作符分隔的多个子集合：

```
GET /api/collection HTTP/1.1
Accept: application/xml;
detail=subcollection1+subcollection2+subcollection3
```

API 支持以下主集合中的扩展子集合。

表 7.4. 使用扩展子集合的集合

集合	支持的扩展子集合
hosts	**statistics**
vms	**statistics**、**nics**、**disks**

例 7.1. 对 vms 集合中的 statistics、NICs 和 disks 扩展子集合的请求

```
GET /api/vms HTTP/1.1
Accept: application/xml; detail=statistics+nics+disks
```

7.3.4. 使用查询条件来搜索集合

对 **"collection/search"** 链接的 **GET** 请求会执行一个对所指定集合的查询。API 只返回那些满足查询条件的集合。

```
GET /api/collection?search={query} HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<collection>
    <resource id="resource_id" href="/api/collection/resource_id">
        ...
    </resource>
    ...
</collection>
```

7.3.5. 最大结果参数

使用 **max** URL 参数可以限制返回结果的数量。在 Red Hat Enterprise Virtualization 3.4 以前，返回结果的数量限制是由 SearchResultsLimit 参数指定的。从 Red Hat Enterprise Virtualization 3.4 开始，这个参数将不对 REST API 起作用，如果 API 搜索查询中没有指定 **max** 参数，它会返回所有结果。我们推荐用户使用 **max** 参数来防止因大量 API 搜索查询结果可能造成的、对 UI 性能的影响。

```
GET /api/collection;max=1 HTTP/1.1
Accept: application/xml
```

```
HTTP/1.1 200 OK
Content-Type: application/xml

<collection>
    <resource id="resource_id" href="/api/collection/resource_id">
        <name>Resource-Name</name>
        <description>A description of the resource</description>
        ...
    </resource>
</collection>
```

7.3.6. 区分大小写

在默认的情况下，所有搜索查询都是区分大小写的。URL 提供了一个布尔选项来切换区分大小写或不区分大小写的查询。

例 7.2. 不区分大小写的搜索查询

```
GET /api/collection;case-sensitive=false?search={query} HTTP/1.1
Accept: application/xml
```

7.3.7. 查询语法

API 使用 URI 模板来执行带有搜索 **query** 的 **GET** 请求：

```
GET /api/collection?search={query} HTTP/1.1
Accept: application/xml
```

query 模板值指向 API 对 **collection** 的搜索查询。这个 **query** 使用和 Red Hat Enterprise Virtualization Query Language 相同的格式：

```
(criteria) [sortby (element) asc|desc]
```

sortby 是可选的，它只在需要对结果进行排序时才使用。

表 7.5. 搜索查询实例

集合	条件	结果
hosts	**vms.status=up**	显示所有运行的虚拟机的状态为 **up** 的主机列表。
vms	**domain=qa.company.com**	显示运行于指定域中的所有虚拟机列表。
vms	**users.name=mary**	显示属于用户 **mary** 的所有虚拟机列表。

集合	条件	结果
events	**severity>normal sortby time**	显示所有 severity 的值高于 **normal** 的**事件**，并以**时间**顺序排序。
events	**severity>normal sortby time desc**	显示所有 severity 的值高于 **normal** 的**事件**，并以**时间**进行倒序排序。

API 需要 **query** 模板使用 URL 编码来翻译那些保留的字符，如操作符和空格。

例 7.3. URL 编码的搜索查询

```
GET /api/vms?search=name%3Dvm1 HTTP/1.1
Accept: application/xml
```

7.3.8. 通配符

在搜索查询中可以使用星号作为通配符。

例 7.4. 在搜索查询中使用通配符 name=vm*

```
GET /api/vms?search=name%3Dvm* HTTP/1.1
Accept: application/xml
```

这个查询的结果将包括所有名称以 **vm** 开头的虚拟机，如 **vm1**、**vm2**、**vma** 和 **vm-webserver**。

例 7.5. 在搜索查询中使用通配符 name=v*1

```
GET /api/vms?search=name%3Dv*1 HTTP/1.1
Accept: application/xml
```

这个查询的结果将包括所有名称以 **v** 开头并以**1** 结束的虚拟机，如 **vm1**、**vr1** 和 **virtualmachine1**。

7.3.9. 分页

一些 Red Hat Enterprise Virtualization 环境会包括大量资源集合，但是 API 只会显示默认数量的集合搜索查询结果。要显示更多的结果，在搜索查询中包括 **page** 命令，API 会把返回的集合分为不同的页。

例 7.6. 为资源分页

这个实例对一个机器中的资源进行分页。使用 URL 编码的请求是：

```
GET /api/collection?search=page%201 HTTP/1.1
Accept: application/xml
```

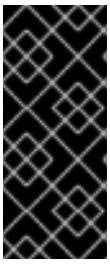
增加 **page** 的值来查看结果的下一页：

```
GET /api/collection?search=page%202 HTTP/1.1
Accept: application/xml
```

在其它命令中使用 **page** 命令。例如：

```
GET /api/collection?search=sortby%20element%20asc%20page%202 HTTP/1.1
Accept: application/xml
```

这个查询会显示集合的第 2 页（结果以所选项排序）。



重要

REST API 是无状态的，每个请求都独立于其它请求，它们无法保持其它请求的状态。因此，如果在不同的请求间发生了状态改变，结果页的内容可能会不一致。

例如，您请求了一组 VM 的一个特定页，而在您请求下一页前，发生了一些状态改变，您所获得的结果可能就会缺少一些数据，或包括了一些重复的数据。

7.3.10. 在集合中创建一个资源

使用一个到包括了一个新资源表述的集合 URI 的 **POST** 请求来创建一个新资源。

POST 请求需要一个 **Content-Type** 头。它会告诉 API 在内容数据中使用的表述 MIME 类型。

包括一个 **Accept** HTTP 头来定义响应格式的 MIME 类型。

每个资源类型都有它们特定的属性。客户端需要在创建新资源时提供这些属性。请参阅具体资源的相关文档来获得详细的信息。

如果缺少了一个所需的属性，创建操作会失败，一个包括了所缺属性信息的表述会返回。

```
POST /api/[collection] HTTP/1.1
Accept: [MIME type]
Content-Type: [MIME type]

[body]
```

7.3.11. 异步请求

如果用户没有使用 **Expect: 201-created** 头，API 将会执行异步 **POST** 请求。

例如，特定资源（如虚拟机、磁盘、快照和模板）会以异步的形式创建。异步创建资源的请求会获得一个 **202 Accepted** 状态。**202 Accepted** 资源的初始文档结构中也包括了一个 **creation_status** 项，以及到创建状态更新的链接。例如：

```
POST /api/collection HTTP/1.1
Accept: application/xml
Content-Type: application/xml
```

```
<resource>
    <name>Resource-Name</name>
</resource>

HTTP/1.1 202 Accepted
Content-Type: application/xml

<resource id="resource_id" href="/api/collection/resource_id">
    <name>Resource-Name</name>
    <creation_status>
        <state>pending</state>
    </creation status>
    <link rel="creation_status"

href="/api/collection/resource_id/creation_status/creation_status_id"/>
      ...
</resource>
```

到 **creation_status** 链接的 **GET** 请求提供了创建状态的更新信息：

```
GET /api/collection/resource_id/creation_status/creation_status_id
HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<creation id="creation_status_id"
  href="/api/collection/resource_id/creation_status/creation_status_id">
    <status>
        <state>complete</state>
    </status>
</creation>
```

如果不以异步的形式创建资源，使用 **Expect: 201-created** 头：

```
POST /api/collection HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Expect: 201-created

<resource>
    <name>Resource-Name</name>
</resource>
```

7.4. 资源

7.4.1. 资源

资源就是支持 REST 的网站服务的数据资源。每个资源类型包括一组通用的参数，REST API 使用这些参数构成一个 XML 或 JSON **资源表述（resource representation）**。用户可以查看资源的表述，然后编辑参数后使用 API 把表述发送到资源的 URL。用户也可以使用 REST 删除资源。

支持 REST 的网站服务也会把资源组合成**集合（collections）**。用户可以查看一个集合中的所有资源的表述。用户也可以向特定的集合发送资源表述来在相应的集合中创建一个新资源。

7.4.2. 获取一个资源

使用从一个集合列表获得的 URI 的 **GET** 请求来得到一个资源状态列表。

包括一个 **Accept** HTTP 头来定义响应格式的 MIME 类型。

```
GET /api/[collection]/[resource_id] HTTP/1.1
Accept: [MIME type]
```

对于一些资源，可以使用带有 **All-Content: true** 的头来获取它们的额外信息。RESTful Service Description Language 介绍了哪些链接支持这个头。

```
GET /api/[collection]/[resource_id] HTTP/1.1
Accept: [MIME type]
All-Content: true
```

7.4.3. 更新一个资源

使用一个 **PUT** 请求修改资源的属性，这个请求需要包括一个从前一个对资源 URI 的 **GET** 请求所获得的更新描述。请参阅相关资源类型的文档来了解更多关于更新资源的信息。

POST 请求需要一个 **Content-Type** 头。它会告诉 API 在内容数据中使用的表述 MIME 类型。

包括一个 **Accept** HTTP 头来定义响应格式的 MIME 类型。

```
PUT /api/collection/resource_id HTTP/1.1
Accept: [MIME type]
Content-Type: [MIME type]

[body]
```

这不包括 API 用户试图修改的“不可变”的资源属性。如果试图修改一个*强制*不可变的资源属性，API 会在响应的内容中包括一个错误信息。

不包括在表述中的属性将不会被修改。

7.4.4. 删除一个资源

使用发送到 URI 的 **DELETE** 请求来删除一个资源。

包括一个 **Accept** HTTP 头来定义响应格式的 MIME 类型。

```
DELETE /api/[collection]/[resource_id] HTTP/1.1
Accept: [MIME type]
```

在一些情况下，需要在 **DELETE** 请求的可选内容项中指定额外的属性。当 **DELETE** 请求带有可选内容项时，它将需要一个 **Content-Type** 头来告诉 API 内容项中的 MIME 类型表述。如果 **DELETE** 请求不包括内容项，将不需要 **Content-Type** 头。

7.4.5. 子集合关系

子集合关系定义了资源和子集合间的分级结构。子集合存在于它们的父资源中，或它们所具有的含义需要放在它们的父资源中来理解。例如，一个虚拟机包括网络接口，这意味着 API 对虚拟机资源和这个网络接口子集合建立映射关系。

子集合有以下的关系类型：

- 一个父资源可以包括多个子资源，一个子资源也可以有多个父资源。例如，一个虚拟机可以包括多个磁盘，而一些磁盘也可以在多个虚拟机间共享。
- 被映射的资源只依赖于一个父资源，如果没有父资源，这个资源将不存在。例如，一个虚拟机和它的快照间的关系。
- 被映射的资源独立存在于父资源，但是数据间仍然保持着联系。例如，一个集群和一个网络的关系。

API 使用 **`link rel=`** 属性定义了一个资源和一个子集合的关系：

```
GET /api/collection/resource_id HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<resource id="resource_id" href="/api/collection/resource_id">
    ...
    <link rel="subcollection"
      href="/api/collection/resource_id/subcollection"/>
    ...
</resource>
```

API 用户查询子集合。

```
GET /api/collection/resource_id/subcollection HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<subcollection>
    <subresource id="subresource_id"
      href="/api/collection/resource_id/subcollection/subresource_id">
        ...
    </subresource>
    ...
</subcollection>
```

7.4.6. XML 项关系

XML 项中的链接是除子集合以外的另外一种代表资源间关系的方式。XML 项中的链接就是那些使用 "href" 属性的项。

XML 项中的链接被用来代表没有依赖关系，并且没有关联的资源间的 1:N 映射，如一个主机和一个集群的关系。

这种关系的实例包括：

- 子资源中的一个资源到它的父资源的链接；或
- 带有随意性关系的资源间的链接。

例 7.7. 从一个子集合资源到一个使用 XML 项的资源链接

```
GET /api/collection/resource_id/subcollection/subresource_id HTTP/1.1

HTTP/1.1 200 OK
Content-Type: application/xml

<subcollection>
    <subresource id="subresource_id"
      href="/api/collection/resource_id/subcollection/subresource_id">
        <resource id="resource_id" href="/api/collection/resource_id"/>
        ...
    </subresource>
</subcollection>
```

7.4.7. 操作

多数资源都会包括一组无法通过标准 HTTP 方法实现的操作。

```
<resource>
    ...
    <actions>
        <link rel="start" href="/api/collection/resource_id/start"/>
        <link rel="stop" href="/api/collection/resource_id/stop"/>
        ...
    </actions>
    ...
</resource>
```

API 通过一个对所提供的 URI 的 **POST** 请求来调用一个操作。**POST** 请求的正文（body）需要一个 **action** 表述来封装通用的参数和针对于特定操作的参数。

表 7.6. 通用操作参数

项	描述
async	**true** 如果服务器马上返回 **202 Accepted**，并且操作表述中包括了一个 **href** 链接。
grace_period	宽限期（以毫秒为单位）。在初始化操作前需要等待的时间。

操作和它们的参数在相应的资源类型文档中有详细介绍。一些参数对于某些操作是必需的，如果没有使用它们就会导致一个 **fault** 响应。

因为 **POST** 请求需要在内容项中包括一个 XML 表述，所以操作也需要一个 **Content-Type: application/xml** 头。

当一个操作是异步进行的，**202 Accepted** 响应会提供一个用来监测操作状态的链接：

```
POST /api/collection/resource_id/action HTTP/1.1
Content-Type: application/xml
Accept: application/xml

<action>
    <async>true</async>
</action>

HTTP/1.1 202 Accepted
Content-Type: application/xml

<action id="action_id"
  href="/api/collection/resource_id/action/action_id">
    <async>true</async>
    ...
</action>
```

接下来的一个对操作 URI 的 **GET** 请求提供了异步操作的状态信息。

表 7.7. 操作状态

状态	描述
pending	任务还没有开始。
in_progress	任务在进行中。
complete	任务已经成功完成。
failed	任务失败。返回的 **action** 表述将包括一个 **fault** 信息。

在任务完成后，操作会被保留一段时间。当这段保留时间过去后，随后的 **GET** 会是 **301 Moved Permanently** 来重定向到目标资源。

```
GET /api/collection/resource_id/action/action_id HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<action id="action_id"
  href="/api/collection/resource_id/action/action_id">
    <status>
        <state>pending</state>
    </status>
    <link rel="parent" /api/collection/resource_id"/>
    <link rel="replay" href="/api/collection/resource_id/action"/>
</action>
```

一个操作的表述还会包括由 **rel** 属性指定的一些链接：

表 7.8. 操作关系

类型	描述
parent	连接回这个操作资源的链接。
replay	连接回原始操作 URI 的链接。到这些 URI 的 POST 请求会重新初始化这个操作。

7.4.8. 权限

每个资源都包括了一个 **permissions** 子集合。每个 **permission** 都会包括一个 **user**、一个分配的 **role** 以及特定的资源。例如：

```
GET /api/collection/resource_id/permissions HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<permissions>
    <permission id="permission-id"
      href="/api/collection/resource_id/permissions/permission_id">
        <role id="role_id" href="/api/roles/role_id"/>
        <user id="user_id" href="/api/users/user_id"/>
        <resource id="resource_id" href="/api/collection/resource_id"/>
    </permission>
    ...
</permissions>
```

当一个 API 用户向资源的 **permissions** 子集合发送一个带有 **permission** 表述和一个 **Content-Type: application/xml** 头的 **POST** 请求时，资源会获得一个新权限。每个新权限都需要一个 **role** 和一个 **user**：

```
POST /api/collection/resource_id/permissions HTTP/1.1
Content-Type: application/xml
Accept: application/xml

<permission>
    <role id="role_id"/>
    <user id="user_id"/>
</permission>

HTTP/1.1 201 Created
Content-Type: application/xml

<permission id="permission_id"
  href="/api/resources/resource_id/permissions/permission_id">
    <role id="role_id" href="/api/roles/role_id"/>
```

```
    <user id="user_id" href="/api/users/user_id"/>
    <resource id="resource_id" href="/api/collection/resource_id"/>
</permission>
```

7.4.9. 错误处理

一些错误需要除标准 HTTP 状态代码外的其它一些信息。例如，API 会在响应信息的正文中包括一个 **fault** 表述来包括一个失败的资源状态更新或操作。这个表述会包括一个 **reason** 和 **detail** 字符串。客户端需要根据响应状态代码，*或* **fault** 信息中的资源表述来对错误进行处理。相关的详细信息可在相应资源的文档中获得。

```
PUT /api/collection/resource_id HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<resource>
    <id>id-update-test</id>
</resource>

HTTP/1.1 409 Conflict
Content-Type: application/xml

<fault>
    <reason>Broken immutability constraint</reason>
    <detail>Attempt to set immutable field: id</detail>
</fault>
```

第 8 章 备份和恢复 API

备份和恢复 API 就是一组功能调用，您可以使用它们来执行虚拟机的完全或文件级备份和恢复操作。API 包括了 Red Hat Enterprise Virtualization 的多个组件（如实时快照和 REST API）来创建并使用临时卷，这个卷可以被附加到包括第三方软件商提供的备份软件的虚拟机上。

如需了解支持的第三方备份厂商，请参阅 Red Hat Enterprise Virtualization Ecosystem（Red Hat Marketplace）。

8.1. 备份一个虚拟机

使用备份和恢复 API 来备份一个虚拟机。这个步骤假设您有两个虚拟机：一个需要备份的虚拟机和一个安装了备份管理软件的虚拟机。

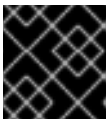
过程 8.1. 备份一个虚拟机

1. 使用 REST API 创建需要备份的虚拟机的一个快照：

```
POST /api/vms/11111111-1111-1111-1111-111111111111/snapshots/
HTTP/1.1
Accept: application/xml
Content-type: application/xml

<snapshot>
    <description>BACKUP</description>
</snapshot>
```

注意

在创建一个虚拟机快照时，虚拟机当时的配置数据会被保存到快照下的 **initialization** 中的 **configuration** 属性的 **data** 属性中。



重要

您不能对标记为“可共享”的磁盘或基于直接 LUN 的磁盘进行快照。

2. 从快照下的 **data** 属性中获取配置数据：

```
GET /api/vms/11111111-1111-1111-1111-
111111111111/snapshots/11111111-1111-1111-1111-111111111111 HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

3. 记录快照的磁盘 ID 和快照 ID：

```
GET /api/vms/11111111-1111-1111-1111-
111111111111/snapshots/11111111-1111-1111-1111-111111111111/disks
HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

4. 把快照附加到用于备份的虚拟机，并激活磁盘：

```
POST /api/vms/22222222-2222-2222-2222-222222222222/disks/ HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk id="11111111-1111-1111-1111-111111111111">
    <snapshot id="11111111-1111-1111-1111-111111111111"/>
    <active>true</active>
</disk>
```

5. 使用备份软件在用于备份的虚拟机上备份快照磁盘中的数据。

6. 把快照磁盘从用于备份的虚拟机上断开：

```
DELETE /api/vms/22222222-2222-2222-2222-222222222222/disks/11111111-
1111-1111-1111-111111111111 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <detach>true</detach>
</action>
```

7. 或者，删除快照：

```
DELETE /api/vms/11111111-1111-1111-1111-
111111111111/snapshots/11111111-1111-1111-1111-111111111111 HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

您使用在一个独立虚拟机上安装的备份软件备份了一个虚拟机。

8.2. 恢复一个虚拟机

恢复使用备份和恢复 API 备份的虚拟机。这个过程假设您有一个安装了备份虚拟机软件的、用于进行备份的虚拟机。

过程 8.2. 恢复一个虚拟机

1. 把磁盘附加到用于备份的虚拟机上：

```
POST /api/vms/22222222-2222-2222-2222-222222222222/disks/ HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk id="11111111-1111-1111-1111-111111111111">
</disk>
```

2. 使用备份软件在磁盘上恢复备份。

3. 把磁盘从用于备份的虚拟机上断开：

```
DELETE /api/vms/22222222-2222-2222-2222-222222222222/disks/11111111-
1111-1111-1111-111111111111 HTTP/1.1
```

```
Accept: application/xml
Content-type: application/xml

<action>
    <detach>true</detach>
</action>
```

4. 使用被恢复虚拟机的配置信息创建一个新虚拟机。

```
POST /api/vms/ HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm>
    <cluster>
        <name>cluster_name</name>
    </cluster>
    <name>NAME</name>
    ...
</vm>
```

5. 把磁盘附加到新虚拟机上：

```
POST /api/vms/33333333-3333-3333-3333-333333333333/disks/ HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk id="11111111-1111-1111-1111-111111111111">
</disk>
```

您通过使用备份和恢复 API 所创建的备份恢复了一个虚拟机。

第 9 章 数据中心

9.1. 数据中心项

datacenters 集合提供了 Red Hat Enterprise Virtualization 环境中的数据中心的信息。API 用户可以通过从进入点 URI 获得的 **rel="datacenters"** 链接来访问这些信息。

以下表格包括了数据中心资源表述中的特定项的信息。

表 9.1. 数据中心项

项	类型	描述	属性
name	字符串	数据中心的名称。数据中心资源中的每个数据中心的 **name** 值必须是唯一的。	
description	字符串	数据中心的描述信息	
link rel="storagedomains"	关系	附加到这个数据中心的存储域子集合的链接。	
link rel="clusters"	关系	附加到这个数据中心的集群子集合的链接。	
link rel="networks"	关系	这个数据中心可用的网络子集合的链接。	
link rel="permissions"	关系	数据中心权限子集合的链接。	
link rel="quotas"	关系	与数据中心相关的配额子集合的链接。	
local	布尔值：true 或 false	指定数据中心是否是一个本地的数据中心。	
storage_format	列举	描述数据中心的存储格式版本。在 **capabilities** 中有效的一组值。	
version major= minor=	复杂数据	数据中心的兼容级别。	
supported_versions	复杂数据	数据中心中有效的版本级别，包括 **version major= minor=**。	
status	参阅以下信息	数据中心状态。	

status 包括了以下值：**uninitialized**、**up**、**maintenance**、**not_operational**、**problematic** 和 **contend**。这些状态在 **data_center_states** 下面的 **capabilities** 中列出。

9.2. 数据中心的 XML 表述

例 9.1. 数据中心的 XML 表述

```
<data_center href="/api/datacenters/00000000-0000-0000-0000-
000000000000"
    id="00000000-0000-0000-0000-000000000000">
  <name>Default</name>
  <description>The default Data Center</description>
  <link href="/api/datacenters/00000000-0000-0000-0000-
000000000000/storagedomains" rel="storagedomains"/>
  <link href="/api/datacenters/00000000-0000-0000-0000-
000000000000/clusters" rel="clusters"/>
  <link href="/api/datacenters/00000000-0000-0000-0000-
000000000000/networks" rel="networks"/>
  <link href="/api/datacenters/00000000-0000-0000-0000-
000000000000/permissions" rel="permissions"/>
  <link href="/api/datacenters/00000000-0000-0000-0000-
000000000000/quotas" rel="quotas"/>
  <local>false</local>
  <storage_format>v3</storage_format>
  <version major="3" minor="4"/>
  <supported_versions>
    <version major="3" minor="4"/>
  </supported_versions>
  <status>
    <state>up</state>
  </status>
</data_center>
```

9.3. 数据中心的 JSON 表述

例 9.2. 数据中心的 JSON 表述

```
{
  "data_center" : [ {
    "local" : "false",
    "storage_format" : "v3",
    "version" : {
      "major" : "3",
      "minor" : "5"
    },
    "supported_versions" : {
      "version" : [ {
        "major" : "3",
        "minor" : "5"
      } ]
    },
    "status" : {
```

```
      "state" : "up"
    },
    "name" : "Default",
    "description" : "The default Data Center",
    "href" : "/api/datacenters/00000002-0002-0002-0002-000000000255",
    "id" : "00000002-0002-0002-0002-000000000255",
    "link" : [ {
      "href" : "/api/datacenters/00000002-0002-0002-0002-
000000000255/storagedomains",
      "rel" : "storagedomains"
    }, {
      "href" : "/api/datacenters/00000002-0002-0002-0002-
000000000255/clusters",
      "rel" : "clusters"
    }, {
      "href" : "/api/datacenters/00000002-0002-0002-0002-
000000000255/networks",
      "rel" : "networks"
    }, {
      "href" : "/api/datacenters/00000002-0002-0002-0002-
000000000255/permissions",
      "rel" : "permissions"
    }, {
      "href" : "/api/datacenters/00000002-0002-0002-0002-
000000000255/quotas",
      "rel" : "quotas"
    }, {
      "href" : "/api/datacenters/00000002-0002-0002-0002-
000000000255/iscsibonds",
      "rel" : "iscsibonds"
    }, {
      "href" : "/api/datacenters/00000002-0002-0002-0002-
000000000255/qoss",
      "rel" : "qoss"
    } ]
  } ]
}
```

9.4. 方法

9.4.1. 创建数据中心

创建新数据中心需要 **name** 和 **local** 项。

例 9.3. 创建数据中心

```
POST /api/datacenters HTTP/1.1
Accept: application/xml
Content-type: application/xml

<data_center>
```

```
    <name>NewDatacenter</name>
    <local>false</local>
</data_center>
```

9.4.2. 更新数据中心

在创建完成后，**name**、**description**、**storage_type**、**version** 和 **storage_format** 项可以被更新。

例 9.4. 更新数据中心

```
PUT /api/datacenters/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<data_center>
    <name>UpdatedName</name>
    <description>An updated description for the data
center</description>
</data_center>
```

9.4.3. 删除数据中心

删除数据中心操作需要一个 **DELETE** 请求。

例 9.5. 删除一个数据中心

```
DELETE /api/datacenters/00000000-0000-0000-0000-000000000000 HTTP/1.1

HTTP/1.1 204 No Content
```

9.5. 子集合

9.5.1. 存储域子集合

9.5.1.1. 存储域子集合

每个数据中心都包括了附加到它上面的存储域的子集合。API 用户使用标准的 REST 方法来和这些子集合进行交互。

附加到数据中心的存储域和顶级存储域有相似的表述，唯一的不同是附加的存储域表述包括了所附加的数据中心的 **status**，以及一组操作。**status** 项的值在 **capabilities** 下面的 **storage_domain_states** 中被列出。

重要

本节中所使用的 API 还处于试用阶段，今后它们可能会有变化。产品的向下兼容声明不适用于这些 API。

9.5.1.2. 附加和取消附加一个存储域

数据中心只有在最少一个存储域被附加后才可以使用，API 用户使用 **POST** 来对数据中心的子集合进行操作。

当附加一个存储域时，**id** 或 **name** 必须被提供。以下是一个为数据中心附加存储域的实例：

例 9.6. 为数据中心附加存储域

```
POST /api/datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"/>

HTTP/1.1 201 Created
Location: /datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed
Content-Type: application/xml

<storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"
  href="/api/datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains/
  fabe0451-701f-4235-8f7e-e20e458819ed">
    <name>images0</name>
    <type>data</type>
    <status>
        <state>inactive</state>
    </status>
    <master>true</master>
    <storage>
        <type>nfs</type>
        <address>172.31.0.6</address>
        <path>/exports/RHEVX/images/0</path>
    </storage>
    <data_center id="d70d5e2d-b8ad-494a-a4d2-c7a5631073c4"
      href="/api/datacenters/d70d5e2d-b8ad-494a-a4d2-c7a5631073c4"/>
    <actions>
        <link rel="activate"
          href="/api/datacenters/d70d5e2d-b8ad-494a-a4d2-c7a5631073c4/
          storagedomains/fabe0451-701f-4235-8f7e-
e20e458819ed/activate"/>
        <link rel="deactivate"
          href="/api/datacenters/d70d5e2d-b8ad-494a-a4d2-c7a5631073c4/
          storagedomains/fabe0451-701f-4235-8f7e-
e20e458819ed/deactivate"/>
    </actions>
</storage_domain>
```

使用 **DELETE** 请求从数据中心中取消附加一个存储域。这个请求可以使用可选的 **async** 项来指定这个请求以异步的方式进行。

例 9.7. 从数据中心中取消附加一个存储域

```
DELETE /api/datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed
HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
  <async>true</async>
</action>
```

9.5.1.3. 操作

9.5.1.3.1. 激活存储域操作

附加到数据中心的存储域需要在激活后才可以被使用。激活操作不需要任何特殊的参数。

例 9.8. 激活数据中心中的一个存储域操作

```
POST /api/datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains/fabe0451-701f-4235-8f7e-
e20e458819ed/activate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

9.5.1.3.2. 取消激活存储域操作

附加到数据中心的存储域需要被取消激活后才可以被删除。取消激活的操作不需要特殊的参数。

例 9.9. 取消激活数据中心中的一个存储域

```
POST /api/datacenters/d70d5e2d-b8ad-494a-a4d2-
c7a5631073c4/storagedomains/fabe0451-701f-4235-8f7e-
e20e458819ed/deactivate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

9.5.2. 网络子集合

9.5.2.1. 网络子集合

与数据中心相关联的网络以 **networks** 子集合代表。数据中心的 **network** 子集合表述包括以下项：

表 9.2. 网络项

项	类型	描述
name	字符串	网络的名称。
description	字符串	网络的描述。
rel="permissions"	关系	到这个网络的 **permissions** 子集合的链接。
rel="vnicprofiles"	关系	到这个网络的 **vnicprofiles** 子集合的链接。
rel="labels"	关系	到这个网络的 **labels** 子集合的链接。
data_center id=	关系	网络所在数据中心的标识。
stp	布尔值：true 或 false	指定 spanning tree 协议是否在网络上被启用。
mtu	整数	指定网络的最大传输单元。
usages	复杂数据	为网络定义一组 **usage** 项。用户可以在这一级把网络定义为 **vm** 网络和 **display** 网络。

您可以在 REST API 中使用标准的 REST 方法来处理 **networks** 子集合。例如，可以使用 **POST** 方法更新网络的 **id** 或 **name**

例 9.10. 为数据中心关联一个网络

```
POST /api/datacenters/00000000-0000-0000-0000-000000000000/networks
HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<network id="da05ac09-00be-45a1-b0b5-4a6a2438665f">
    <name>ovirtmgmt</name>
</network>

HTTP/1.1 201 Created
Location: http://{host}/clusters/00000000-0000-0000-0000-
000000000000/networks/00000000-0000-0000-0000-000000000000
Content-Type: application/xml

<network href="/api/networks/00000000-0000-0000-0000-000000000000"
    id="00000000-0000-0000-0000-000000000000">
  <name>Network_001</name>
```

```
  <link href="/api/networks/00000000-0000-0000-0000-
000000000000/permissions"
    rel="permissions"/>
  <link href="/api/networks/00000000-0000-0000-0000-
000000000000/vnicprofiles"
    rel="vnicprofiles"/>
  <link href="/api/networks/00000000-0000-0000-0000-000000000000/labels"
    rel="labels"/>
  <data_center href="/api/datacenters/00000000-0000-0000-0000-
000000000000"
    id="00000000-0000-0000-0000-000000000000"/>
  <stp>false</stp>
  <mtu>0</mtu>
  <usages>
    <usage>vm</usage>
  </usages>
</network>
```

使用**PUT** 请求更新资源。使用 **PUT** 请求设置 **mtu** 项的值来指定网络的最大传输单元。

例 9.11. 设置网络最大传输单元

```
PUT /api/datacenters/00000000-0000-0000-0000-
000000000000/networks/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<network>
    <mtu>1500</mtu>
</network>
```

使用 **DELETE** 请求删除集合中相关项的关联。

例 9.12. 从数据中心中删除一个网络关联

```
DELETE /api/datacenters/00000000-0000-0000-0000-
000000000000/networks/00000000-0000-0000-0000-000000000000 HTTP/1.1

HTTP/1.1 204 No Content
```

9.5.3. 配额子集合

9.5.3.1. 配额子集合

配额子集合列出了 Red Hat Enterprise Virtualization Manager 在资源上实施的限制。API 用户可以使用 **GET** 方法查看这个子集合以及它的资源。

例 9.13. 配额的 XML 表述

```
<quota href="/api/datacenters/56087282-d7a6-11e1-af44-001a4a400e0c
  /quotas/e13ff85a-b2ba-4f7b-8010-e0d057c03dfe"
  id="e13ff85a-b2ba-4f7b-8010-e0d057c03dfe">
    <name>MyQuota</name>
    <description>A quota for my Red Hat Enterprise
      Virtualization environment</description>
    <data_center href= "/api/datacenters/56087282-d7a6-11e1-af44-
001a4a400e0c"
    id="56087282-d7a6-11e1-af44-001a4a400e0c"/>
</quota>
```

创建新配额需要 **name** 和 **description** 项。

例 9.14. 创建一个配额

```
POST /api/datacenters/56087282-d7a6-11e1-af44-001a4a400e0c/quotas
HTTP/1.1
Accept: application/xml
Content-type: application/xml

<quota>
    <name>VMQuota</name>
    <description>My new quota for virtual machines</description>
</quota>
```

删除配额需要一个 **DELETE** 请求。

例 9.15. 删除一个配额

```
DELETE /api/datacenters/01a45ff0-915a-11e0-8b87-
5254004ac988/quotas/e13ff85a-b2ba-4f7b-8010-e0d057c03dfe HTTP/1.1

HTTP/1.1 204 No Content
```

9.6. 操作

9.6.1. 强制删除数据中心操作

API 用户可以在遇到无法解决的存储域问题时（如无法连接到主存储域，或删除存储域时没有可用的主机），强制删除数据中心。API 包括了一个 **force** 操作。

这个操作会在 API 从 Red Hat Enterprise Virtualization 环境中删除数据中心前先删除数据中心的数据库。这意味着无论相关的存储域是什么状态，API 都会删除数据中心。

这个操作需要一个 **DELETE** 方法。请求的内容项中包括了一个 **action** 表述，其中的 **force** 参数被设为 **true**。这个请求还需要一个额外的 **Content-type: application/xml** 头来处理内容项中的 XML 表述。

例 9.16. 数据中心中的强制删除操作

```
DELETE /api/datacenters/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
  <force>true</force>
</action>
```

这个操作：

- 删除与数据中心相关联的 **data** 存储域的所有数据库信息；
- 从数据库中删除与数据中心相关联的 **data** 存储域中的资源（如虚拟机和模板）信息；
- 取消 **iso** 存储域和 **export** 存储域和数据中心的关联；
- 删除数据中心的数据库信息。

这个操作不需要在删除数据中心前清空数据中心。

重要

这个操作只删除与数据中心相关资源在数据库中的信息。与数据中心相关联的 **data** 存储域在可以被重新使用前需要被手动格式化。另外，**iso** 和 **export** 域的元数据需要被手工清除后才可以被其它数据中心使用。

第 10 章 集群

10.1. 集群项

clusters 集合提供了 Red Hat Enterprise Virtualization 环境中的集群的信息。API 用户可以通过从进入点 URI 获得的 **rel="clusters"** 链接来访问这些信息。

以下表格包括了集群资源表述中的特定项的信息。

表 10.1. 集群项

项	类型	描述	属性
name	字符串	集群的名称。集群资源中的每个集群的 **name** 值必须是唯一的。	
description	字符串	用户提供的、对集群的描述信息。	
link rel="networks"	关系	与集群相关的网络子集合的链接。	
link rel="permissions"	关系	集群权限子集合的链接。	
link rel="glustervolumes"	关系	与这个集群相关的 Red Hat Gluster Storage 卷子集合的链接。	
link rel="storagedomains"	关系	与这个集群相关的 Red Hat Gluster Storage 卷 hook 子集合的链接。	
link rel="affinitygroups"	关系	与集群相关的虚拟机关联组子集合的链接。	
cpu id=	复杂数据	定义集群中的所有主机都必须支持的 CPU 类型的服务器 CPU 标识。	
data_center id=	GUID	集群所在数据中心的标识。	
memory_policy	复杂数据	定义关于主机内存利用的集群策略。	

项	类型	描述	属性
scheduling_policy	复杂数据	定义集群中主机的负载均衡或电源节省模式。	
version major= minor=	复杂数据	集群的兼容级别。	
supported_versions	复杂数据	这个集群可能的 **version** 级别。	
error_handling	复杂数据/列表	指定当集群中的某个主机无法正常工作时需要进行的操作。它需要一个包括了在 **capabilities** 中列出的类型属性的 **on_error** 项。	
virt_service	布尔值	指定这个集群是否可以使用虚拟化服务。	
gluster_service	布尔值	指定这个集群是否可以使用 Red Hat Storage Gluster 服务。	
threads_as_cores	布尔值	指定是否允许主机上所运行的处理器核的数量大于主机本身的处理器核的数量。	
tunnel_migration	布尔值	指定在迁移过程中，虚拟机是否使用一个 libvirt-to-libvirt 通道。	
trusted_service	布尔值	指定是否使用 OpenAttestation 服务器来验证主机。	
ballooning_enabled	布尔值	指定是否在这个集群中使用"内存气球（ballooning）"功能。	
ksm	布尔值	指定是否在这个集群中启用 ksm	

注意

当一个主机的可用内存降到低于 20% 时，ballooning 命令（如 **mom.Controllers.Balloon - INFO Ballooning guest:half1 from 1096400 to 1991580**）会被存到 **/etc/vdsm/mom.conf** 文件中。**/etc/vdsm/mom.conf** 是 Memory Overcommit Manager 的日志文件。

10.2. 内存策略项

memory_policy 项包括了以下项：

表 10.2. 内存策略项

项	类型	描述	属性
overcommit percent=	复杂数据	指定主机内存的百分比，当所使用的内存超过这个百分比时，主机将无法再运行更多的虚拟机。因为使用了 KSM 内存共享技术，虚拟机可以使用比主机物理内存更多的内存。推荐的值包括 **100** (None)、**150** (Server Load) 和 **200** (Desktop Load)。	
transparent_hugepages	复杂数据	指定 Transparent Hugepages 的 **enabled** 状态，它的值是 true 或 false。请检查 **capabilities** 的功能设置来确定您的版本支持 **transparent hugepages**。	

10.3. 调度策略项

scheduling_policy 项包括以下项：

表 10.3. 调度策略项

项	类型	描述	属性
policy	列举	集群中的主机上的虚拟机调度模式，类型列表包括在 **capabilities** 中。	
thresholds low= high= duration=	复杂数据	指定主机的 CPU 限制。**high** 属性指定了在主机被认为“过载”前最大的 CPU 利用率（以百分数的形式表示）；**low** 属性指定了在主机被认为“没有被充分利用”前最小的 CPU 利用率（以百分数的形式表示）；**duration** 属性指定了当主机处于“过载”状态多长时间后（以秒为单位），调度进程开始把负载迁移到其它主机上。	

10.4. 集群的 XML 表述

例 10.1. 集群的 XML 表述

```
<cluster id="00000000-0000-0000-0000-000000000000"
  href="/api/clusters/00000000-0000-0000-0000-000000000000">
    <name>Default</name>
    <description>The default server cluster</description>
    <link rel="networks"
      href="/api/clusters/00000000-0000-0000-0000-
000000000000/networks"/>
    <link rel="permissions"
      href="/api/clusters/00000000-0000-0000-0000-
```

```
000000000000/permissions"/>
          <link rel="glustervolumes"
      href="/api/clusters/00000000-0000-0000-0000-
000000000000/glustervolumes"/>
          <link rel="glusterhooks"
      href="/api/clusters/00000000-0000-0000-0000-
000000000000/glusterhooks"/>
          <link rel="affinitygroups"
      href="/api/clusters/00000000-0000-0000-0000-
000000000000/affinitygroups"/>
    <cpu id="Intel Penryn Family"/>
        <architecture>X86_64<architecture/>
    <data_center id="00000000-0000-0000-0000-000000000000"
      href="/api/datacenters/00000000-0000-0000-0000-000000000000"/>
    <memory_policy>
        <overcommit percent="100"/>
        <transparent_hugepages>
            <enabled>false</enabled>
        </transparent_hugepages>
    </memory_policy>
    <scheduling_policies>
      <policy>evenly_distributed</policy>
      <thresholds low="10" high="75" duration="120"/>
    </scheduling_policies>
    <version minor="0" major="3"/>
    <supported_versions>
        <version minor="0"<usage> major="3"/>
    </supported_versions>
    <error_handling>
        <on_error>migrate</on_error>
    </error_handling>
    <virt_service>true</virt_service>
    <gluster_service>false</gluster_service>
    <threads_as_cores>false</threads_as_cores>
    <tunnel_migration>false</tunnel_migration>
    <trusted_service>false</trusted_service>
    <ha_reservation>false</ha_reservation>
    <ballooning_enabled>false</ballooning_enabled>
    <ksm>
        <enabled>true</enabled>
    </ksm>
</cluster>
```

10.5. 集群的 JSON 表述

例 10.2. 集群的 JSON 表述

```
{
  "cluster" : [ {
    "cpu" : {
      "architecture" : "X86_64",
      "id" : "Intel Penryn Family"
    },
    "data_center" : {
```

```
      "href" : "/api/datacenters/00000002-0002-0002-0002-000000000255",
      "id" : "00000002-0002-0002-0002-000000000255"
    },
    "memory_policy" : {
      "overcommit" : {
        "percent" : "100"
      },
      "transparent_hugepages" : {
        "enabled" : "true"
      }
    },
    "scheduling_policy" : {
      "policy" : "none",
      "name" : "none",
      "href" : "/api/schedulingpolicies/b4ed2332-a7ac-4d5f-9596-
99a439cb2812",
      "id" : "b4ed2332-a7ac-4d5f-9596-99a439cb2812"
    },
    "version" : {
      "major" : "3",
      "minor" : "5"
    },
    "error_handling" : {
      "on_error" : "migrate"
    },
    "virt_service" : "true",
    "gluster_service" : "false",
    "threads_as_cores" : "false",
    "tunnel_migration" : "false",
    "trusted_service" : "false",
    "ha_reservation" : "false",
    "optional_reason" : "false",
    "ballooning_enabled" : "false",
    "ksm" : {
      "enabled" : "true"
    },
    "required_rng_sources" : { },
    "name" : "Default",
    "description" : "The default server cluster",
    "href" : "/api/clusters/00000001-0001-0001-0001-0000000002fb",
    "id" : "00000001-0001-0001-0001-0000000002fb",
    "link" : [ {
      "href" : "/api/clusters/00000001-0001-0001-0001-
0000000002fb/networks",
      "rel" : "networks"
    }, {
      "href" : "/api/clusters/00000001-0001-0001-0001-
0000000002fb/permissions",
      "rel" : "permissions"
    }, {
      "href" : "/api/clusters/00000001-0001-0001-0001-
0000000002fb/glustervolumes",
      "rel" : "glustervolumes"
    }, {
      "href" : "/api/clusters/00000001-0001-0001-0001-
0000000002fb/glusterhooks",
```

```
      "rel" : "glusterhooks"
    }, {
      "href" : "/api/clusters/00000001-0001-0001-0001-
0000000002fb/affinitygroups",
      "rel" : "affinitygroups"
    }, {
      "href" : "/api/clusters/00000001-0001-0001-0001-
0000000002fb/cpuprofiles",
      "rel" : "cpuprofiles"
    } ]
  } ]
}
```

10.6. 方法

10.6.1. 创建集群

创建新集群需要 **name**、**cpu id=** 和 **datacenter** 项。**datacenter** 可以由 **id** 属性代表，也可以由 **name** 项代表。

例 10.3. 创建一个集群

```
POST /api/clusters HTTP/1.1
Accept: application/xml
Content-type: application/xml

<cluster>
    <name>cluster1</name>
    <cpu id="Intel Penryn Family"/>
    <data_center id="00000000-0000-0000-0000-000000000000"/>
</cluster>
```

10.6.2. 更新集群

在创建完成后，**name**、**description**、**cpu id=** 和 **error_handling** 项可以被更新。

例 10.4. 更新一个集群

```
PUT /api/clusters/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<cluster>
    <description>Cluster 1</description>
</cluster>
```

10.6.3. 删除集群

删除集群需要一个 **DELETE** 请求。

例 10.5. 删除一个集群

```
DELETE /api/clusters/00000000-0000-0000-0000-000000000000 HTTP/1.1

HTTP/1.1 204 No Content
```

10.7. 子集合

10.7.1. 网络子集合

10.7.1.1. 网络子集合

与集群相关联的网络由 **networks** 子集合代表。集群中的每个主机都和这些网络相连接。

除了以下项外，集群的 **network** 子集合表述和标准的 **network** 资源表述相同：

表 10.4. 额外的网络项

元素	类型	描述	项
cluster id=	关系	网络所在集群的标识。	
required	布尔值	指定网络是“必需的”网络还是“可选的”网络。	
display	布尔值	定义显示网络状态。	
usages	复杂数据	为网络定义一组 **usage** 项。用户可以在这一级把网络定义为 **VM** 网络和 **DISPLAY** 网络。	

API 用户使用标准的 REST 方法来处理 **networks** 子集合。在 **POST** 中使用网络 **id** 或 **name** 来指定和集群相关联的 **networks** 子集合。

例 10.6. 为集群关联一个网络资源

```
POST /api/clusters/99408929-82cf-4dc7-a532-9d998063fa95/networks
HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<network id="da05ac09-00be-45a1-b0b5-4a6a2438665f">
    <name>ovirtmgmt</name>
</network>

HTTP/1.1 201 Created
```

```
Location: http://{host}/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/networks/da05ac09-00be-45a1-b0b5-4a6a2438665f
Content-Type: application/xml

<network id="da05ac09-00be-45a1-b0b5-4a6a2438665f"
  href="/api/clusters/99408929-82cf-4dc7-a532-9d998063fa95/networks/
  da05ac09-00be-45a1-b0b5-4a6a2438665f">
    <name>ovirtmgmt</name>
    <status>
        <state>operational</state>
    </status>
    <description>Display Network</description>
    <cluster id="99408929-82cf-4dc7-a532-9d998063fa95"
      href="/api/clusters/99408929-82cf-4dc7-a532-9d998063fa95"/>
    <data_center id="d70d5e2d-b8ad-494a-a4d2-c7a5631073c4"
      href="/api/datacenters/d70d5e2d-b8ad-494a-a4d2-c7a5631073c4"/>
    <required>true</required>
    <usages>
        <usage>VM</usage>
    </usages>
</network>
```

使用 **PUT** 请求更新资源。

例 10.7. 设置显示网络状态

```
PUT /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/networks/da05ac09-00be-45a1-b0b5-4a6a2438665f HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<network>
    <required>false</required>
    <usages>
        <usage>VM</usage>
        <usage>DISPLAY</usage>
    </usages>
</network>
```

使用 **PUT** 请求指定 **required** 项的值（true 或 false）可以把网络设置为“必需的”网络或“可选的”网络。

例 10.8. 设置“可选的”网络

```
PUT /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/networks/da05ac09-00be-45a1-b0b5-4a6a2438665f HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<network>
    <required>false</required>
</network>
```

使用 **DELETE** 请求删除集合中相关项的关联。

例 10.9. 从集群中删除一个网络关联

```
DELETE /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/networks/da05ac09-00be-45a1-b0b5-4a6a2438665f HTTP/1.1

HTTP/1.1 204 No Content
```

10.7.2. 存储卷子集合

10.7.2.1. Red Hat Gluster Storage 卷子集合

Red Hat Enterprise Virtualization 提供了创建和管理 Red Hat Gluster Storage 卷的方法。Red Hat Gluster Storage 卷可以和集群相关联，并由 **glustervolumes** 子集合代表。

Red Hat Gluster Storage 卷资源在 **glustervolumes** 子集合中的表述由以下项定义：

表 10.5. Gluster 卷项

项	类型	描述	属性
volume_type	列举	定义卷类型。**capabilities** 集合中包括了卷类型列表。	
bricks	关系	Red Hat Gluster Storage 数据库（brick）的子集合。在创建一个新卷时，请求需要一组 **brick** 项。使用 **server_id** 代表 Red Hat Gluster Storage 服务器；使用 **brick_dir** 项代表数据块目录。	
transport_types	复杂数据	定义一组卷 **transport_type** 项。**capabilities** 集合中包括了有效传输类型的列表。	
replica_count	整数	指定一个重复的卷所具有的副本数量。	
stripe_count	整数	指定一个条带卷的条带数量	
options	复杂数据	一组额外的 Red Hat Gluster Storage **option** 项。每个 **option** 都包括一个选项 **name** 和相应的 **value**。	

例 10.10. Red Hat Gluster Storage 卷的 XML 表述

```
<gluster_volume id="99408929-82cf-4dc7-a532-9d998063fa95"
  href="/api/clusters/99408929-82cf-4dc7-a532-9d998063fa95
  /glustervolume/e199f877-900a-4e30-8114-8e3177f47651">
    <name>GlusterVolume1</name>
    <link rel="bricks"
      href="/api/clusters/99408929-82cf-4dc7-a532-9d998063fa95
      /glustervolume/e199f877-900a-4e30-8114-8e3177f47651/bricks"/>
    <volume_type>DISTRIBUTED_REPLICATE</volume_type>
    <transport_types>
        <transport_type>TCP</transport_type>
    </transport_types>
    <replica_count>2</replica_count>
    <stripe_count>1</stripe_count>
    <options>
        <option>
            <name>cluster.min-free-disk</name>
            <value>536870912</value>
        </option>
    </options>
</gluster_volume>
```

通过带有子集合的 **name**、**volume_type** 和 **bricks** 的 **POST** 请求创建一个 Red Hat Gluster Storage 卷

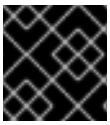
例 10.11. 创建一个 Red Hat Gluster Storage 卷

```
POST /api/clusters/99408929-82cf-4dc7-a532-9d998063fa95/glustervolumes
HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<gluster_volume>
    <name>GlusterVolume1</name>
    <volume_type>DISTRIBUTED_REPLICATE</volume_type>
    <bricks>
        <brick>
            <server_id>server1</server_id>
            <brick_dir>/exp1</brick_dir>
        </brick>
    <bricks>
</gluster_volume>
```

使用 **DELETE** 请求删除一个 Red Hat Gluster Storage 卷。

例 10.12. 删除一个 Red Hat Gluster Storage 卷

```
DELETE /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/glustervolumes/e199f877-900a-4e30-8114-8e3177f47651
HTTP/1.1

HTTP/1.1 204 No Content
```



重要

glustervolumes 子集合中的资源不能被更新。

10.7.2.2. 数据块子集合

glustervolumes 子集合包括自己的 **bricks** 子集合来定义 Red Hat Gluster Storage 卷中的单独的数据块。额外信息可以通过使用 **All-Content: true** 头的 **GET** 请求获得。

一个卷的 **bricks** 子集合的表述使用以下元素定义：

表 10.6. 数据块项

项	类型	描述	属性
server_id	字符串	到 Red Hat Gluster Storage 服务器的参考。	
brick_dir	字符串	定义 Red Hat Gluster Storage 服务器上的一个数据块。	
replica_count	整数	指定卷中的数据块的文件副本数量。	
stripe_count	整数	指定卷中数据块的条带数量	

通过带有子集合的 **server_id** 和 **brick_id** 的 **POST** 请求创建新数据块。

例 10.13. 添加一个数据块

```
POST /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/glustervolumes/e199f877-900a-4e30-8114-8e3177f47651/bricks
HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<brick>
    <server_id>server1</server_id>
    <brick_dir>/exp1</brick_dir>
</brick>
```

使用 **DELETE** 请求删除一个数据块。

例 10.14. 删除一个数据块

```
DELETE /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/glustervolumes/e199f877-900a-4e30-8114-
8e3177f47651/bricks/0a473ebe-01d2-444d-8f58-f565a436b8eb HTTP/1.1

HTTP/1.1 204 No Content
```

重要

bricks 子集合中的资源不能被更新。

10.7.2.3. 操作

10.7.2.3.1. 启动操作

start 操作使 Gluster 卷可以被使用。

例 10.15. 启动一个卷

```
POST /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/glustervolumes/e199f877-900a-4e30-8114-8e3177f47651/start
HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<action/>
```

使用可选的 **force** 项可以强制运行一个卷。如需在一个运行的卷中启动禁用的数据块，可以使用这个选项。

10.7.2.3.2. 停止操作

stop 操作会取消激活 Gluster 卷。

例 10.16. 停止一个卷

```
POST /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/glustervolumes/e199f877-900a-4e30-8114-8e3177f47651/stop
HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<action/>
```

使用可选的 **force** 项来强制执行停止操作。

10.7.2.3.3. 设置选项操作

setoption 操作会设置一个卷选项。

例 10.17. 设置一个选项

```
POST /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/glustervolumes/e199f877-900a-4e30-8114-
8e3177f47651/setoption HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<action>
    <option>
        <name>cluster.min-free-disk</name>
        <value>536870912</value>
    </option>
</action>
```

10.7.2.3.4. 重置选项操作

resetoption 操作会重置一个卷选项。

例 10.18. 重置一个选项

```
POST /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/glustervolumes/e199f877-900a-4e30-8114-
8e3177f47651/resetoption HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<action>
        <option>
            <name>cluster.min-free-disk</name>
        </option>
</action>
```

10.7.2.3.5. 重置所有选项操作

resetalloptions 操作会重置所有卷选项。

例 10.19. 重置所有选项

```
POST /api/clusters/99408929-82cf-4dc7-a532-
9d998063fa95/glustervolumes/e199f877-900a-4e30-8114-
8e3177f47651/resetalloptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<action/>
```

10.7.3. 关联组子集合

10.7.3.1. 关联组子集合

affinitygroups 子集合中的虚拟机关联组的表述由以下项定义：

表 10.7. 关联组项

项	类型	描述	属性
name	字符串	关联组的名称。	
cluster	关系	应用关联组的集群的标识。	
positive	布尔值：true 或 false	指定关联组中的虚拟机是正关联（positive affinity）还是负关联（negative affinity）。	
enforcing	布尔值：true 或 false	指定对关联组中的虚拟机硬实施（hard enforcement）还是软实施（soft enforcement）关联策略。	

例 10.20. 虚拟机关联组的 XML 表述

```
<affinity_group href="/api/clusters/00000000-0000-0000-0000-
000000000000/affinitygroups/00000000-0000-0000-0000-000000000000"
id="00000000-0000-0000-0000-000000000000">
  <name>AF_GROUP_001</name>
  <cluster href="/api/clusters/00000000-0000-0000-0000-000000000000"
    id="00000000-0000-0000-0000-000000000000"/>
  <positive>true</positive>
  <enforcing>true</enforcing>
</affinity_group>
```

使用带有 **name** 属性的 **POST** 请求创建虚拟机关联组。

例 10.21. 创建一个虚拟机关联组

```
POST https://XX.XX.XX.XX/api/clusters/00000000-0000-0000-0000-
000000000000/affinitygroups HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<affinity_group>
  <name>AF_GROUP_001</name>
  <positive>true</positive>
  <enforcing>true</enforcing>
</affinity_group>
```

使用 **DELETE** 请求删除一个虚拟机关联组。

例 10.22. 删除一个虚拟机关联组

```
DELETE https://XX.XX.XX.XX/api/clusters/00000000-0000-0000-0000-
000000000000/affinitygroups/00000000-0000-0000-0000-000000000000
HTTP/1.1

HTTP/1.1 204 No Content
```

第 11 章 网络

11.1. 网络项

networks 集合提供了 Red Hat Enterprise Virtualization 环境中的逻辑网络的信息。API 用户可以通过从进入点 URI 获得的 **rel="networks"** 链接来访问这些信息。

以下表格包括了网络资源表述中的特定项的信息。

表 11.1. 网络项

项	类型	描述	属性
link rel="vnicprofiles"	关系	附加到这个逻辑网络的 VNIC 配置集的子集合链接。	
link rel="labels"	关系	附加到这个逻辑网络的标签的子集合链接。	
data_center id=	GUID	集群所在数据中心的标识。	
vlan id=	整数	VLAN 标签（tag）。	
stp	布尔值：true 或 false	当把这个参数设为 **true** 时，Spanning Tree Protocol 将在这个网络中被启用。	
mtu	整数	为逻辑网络设置最大传输单元。如果没有使用它，逻辑网络会使用默认的值。	
status	**operational** 或 **non_operational**	网络的状态。这些状态在 **capabilities** 下面的 **network_states** 中列出。	
usages	复杂数据	为网络定义一组 **usage** 项。用户可以在这一级把网络定义为 **vm** 网络。	

重要

本节中所使用的 API 还处于试用阶段，今后它们可能会有变化。产品的向下兼容声明不适用于这些 API。

11.2. 网络的 XML 表述

例 11.1. 网络资源的 XML 表述

```
<network href="/api/networks/00000000-0000-0000-0000-000000000000"
  id="00000000-0000-0000-0000-000000000000">
  <name>ovirtmgmt</name>
  <description>Management Network</description>
  <link href="/api/networks/00000000-0000-0000-0000-
000000000000/permissions" rel="permissions"/>
  <link href="/api/networks/00000000-0000-0000-0000-
000000000000/vnicprofiles" rel="vnicprofiles"/>
  <link href="/api/networks/00000000-0000-0000-0000-000000000000/labels"
rel="labels"/>
  <data_center href="/api/datacenters/00000000-0000-0000-0000-
000000000000"
    id="00000000-0000-0000-0000-000000000000"/>
  <stp>false</stp>
  <mtu>0</mtu>
  <usages>
    <usage>vm</usage>
  </usages>
</network>
```

11.3. 网络资源的 JSON 表述

例 11.2. 网络资源的 JSON 表述

```
{
  "network" : [ {
    "data_center" : {
      "href" : "/api/datacenters/00000002-0002-0002-0002-000000000255",
      "id" : "00000002-0002-0002-0002-000000000255"
    },
    "stp" : "false",
    "mtu" : "0",
    "usages" : {
      "usage" : [ "vm" ]
    },
    "name" : "ovirtmgmt",
    "description" : "Management Network",
    "href" : "/api/networks/00000000-0000-0000-0000-000000000009",
    "id" : "00000000-0000-0000-0000-000000000009",
    "link" : [ {
      "href" : "/api/networks/00000000-0000-0000-0000-
000000000009/permissions",
      "rel" : "permissions"
    }, {
      "href" : "/api/networks/00000000-0000-0000-0000-
000000000009/vnicprofiles",
      "rel" : "vnicprofiles"
    }, {
      "href" : "/api/networks/00000000-0000-0000-0000-
000000000009/labels",
      "rel" : "labels"
```

```
      } ]
    } ]
}
```

11.4. 方法

11.4.1. 创建网络资源

创建新网络需要 **name** 和 **datacenter** 项。

例 11.3. 创建一个网络资源

```
POST /api/networks HTTP/1.1
Accept: application/xml
Content-type: application/xml

<network>
    <name>network 1</name>
    <data_center id="00000000-0000-0000-0000-000000000000"/>
</network>
```

11.4.2. 更新网络资源

在创建完成后，**name**、**description**、**ip**、**vlan**、**stp** 和 **display** 项可以被修改。

例 11.4. 更新网络资源

```
PUT /api/networks/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<network>
    <description>Network 1</description>
</network>
```

11.4.3. 删除网络资源

删除网络需要一个 **DELETE** 请求。

例 11.5. 删除一个网络

```
DELETE /api/networks/00000000-0000-0000-0000-000000000000 HTTP/1.1

HTTP/1.1 204 No Content
```

11.5. 子集合

11.5.1. 网络 VNIC 配置集子集合

VNIC（Virtual Network Interface Controller - 虚拟网络接口控制器）配置集，也被称为虚拟机接口配置集，它是一组应用于用户和组的自定义配置，可以被用来限制对网络带宽的使用。每个 **vnicprofile** 都包括以下项：

表 11.2. VNIC 配置集的项

元素	类型	描述
name	字符串	配置集的唯一标识。
description	字符串	配置集的描述信息。
network	字符串	应用配置集的逻辑网络 ID。
port_mirroring	布尔值：**true** 或 **false**	默认值是 **false**。

例 11.6. 网络 VNIC 配置集子集合的 XML 表述

```
<vnic_profile href= "/api/vnicprofiles/f9c2f9f1-3ae2-4100-a9a5-
285ebb755c0d" id="f9c2f9f1-3ae2-4100-a9a5-285ebb755c0d">
 <name>Peanuts</name>
 <description>shelled</description>
 <network href= "/api/networks/00000000-0000-0000-0000-000000000009"
id="00000000-0000-0000-0000-000000000009"/>
 <port_mirroring>false</port_mirroring>
 </vnic_profile>
</vnic_profiles>
```

11.5.2. 网络标签子集合

网络标签就是一个标签，它可以自动地把逻辑网络和物理主机网络接口进行关联。每个 **label** 都包括以下项：

表 11.3. 标签项

项	类型	描述
network	字符串	标签被附加到的网络 **href** 和 **id**。

例 11.7. 网络标签子集合的 XML 表述

```
<labels>
  <label href="/api/networks/00000000-0000-0000-0000-
000000000000/labels/eth0" id="eth0">
    <network href="/api/networks/00000000-0000-0000-0000-000000000000"
```

```
        id="00000000-0000-0000-0000-000000000000"/>
    </label>
</labels>
```

11.5.3. 方法

11.5.3.1. 为逻辑网络附加标签操作

您可以为网络附加一个标签，使用这个标签可以把逻辑网络和有相同标签的物理主机网络接口进行关联。

例 11.8. 为逻辑网络添加标签的操作

```
POST /api/networks/00000000-0000-0000-0000-000000000000/labels/ HTTP/1.1
Accept: application/xml
Content-type: application/xml

<label id="Label_001" />
```

11.5.3.2. 从逻辑网络中删除一个标签

从逻辑网络中删除一个标签需要一个 **DELETE** 请求。

例 11.9. 从逻辑网络中删除一个标签

```
DELETE /api/networks/00000000-0000-0000-0000-
000000000000/labels/[label_id] HTTP/1.1

HTTP/1.1 204 No Content
```

第 12 章 存储域

12.1. 存储域项

storagedomains 集合提供了 Red Hat Enterprise Virtualization 环境中的存储域信息。API 用户可以通过从进入点 URI 获得的 **rel="storagedomains"** 链接来访问这些信息。

以下表格包括了存储域表述中的特定项的信息。

表 12.1. 存储域项

项	类型	描述	属性
link rel="permissions"	关系	存储域权限子集合的链接。	
link rel="files"	关系	存储域的 **files** 子集合链接。	
link rel="vms"	关系	**type** 被设置为 **export** 的存储域的 **vms** 子集合链接。	
link rel="templates"	关系	**type** 被设置为 **export** 的存储域的 **templates** 子集合链接。	
type	列举	存储域类型。**capabilities** 包括了有效值列表。	
master	布尔值：true 或 false	**true**：如果这个存储域是数据中心的主存储域。	
host	复杂数据	存储域进行初始化所在的主机。这个主机需要可以访问到指定的物理存储设备。	
storage	复杂数据	存储域所基于的存储设备的描述。	
available	整数	可用的空间（以字节为单位）。	
used	整数	使用的空间（以字节为单位）。	

项	类型	描述	属性
committed	整数	分配的空间（以字节为单位）。	
storage_format	列举	描述存储域的存储格式版本。**capabilities** 包括了有效值列表。	
wipe_after_delete	布尔值：true 或 false	在存储域中默认设置删除后清除数据的选项。这个选项可以在域被创建后进行修改，但这不会改变那些已存在磁盘的删除后清除数据的设置。	
warning_low_space_indicator	整数	警告级低存储空间选项设置（一个百分比值）。当存储域中的空闲空间低于这个值时，用户登录时就会显示一个警告信息。	
critical_space_action_blocker	整数	严重的空间操作限制阈值选项设置（一个以 GB 为单位的值）。当存储域中的空闲空间低于这个值时，用户登录时会显示一个错误信息，任何需要使用存储空间的操作（包括临时使用存储空间的操作）都会被限制。	

重要

本节中所使用的 API 还处于试用阶段，今后它们可能会有变化。产品的向下兼容声明不适用于这些 API。

12.2. 存储域的 XML 表述

例 12.1. 存储域的 XML 表述

```
<storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"
  href="/api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed">
    <name>data0</name>
    <link rel="permissions"
      href="/api/storagedomains/be24cd98-8e23-49c7-b425-
1a12bd12abb0/permissions"/>
    <link rel="files"
      href="/api/storagedomains/be24cd98-8e23-49c7-b425-
1a12bd12abb0/files"/>
    <type>data</type>
    <master>true</master>
    <storage>
        <type>nfs</type>
        <address>172.31.0.6</address>
        <path>/exports/RHEVX/images/0</path>
```

```
    </storage>
    <available>156766306304</available>
    <used>433791696896</used>
    <committed>617401548800</committed>
    <storage_format>v1</storage_format>
    <wipe_after_delete>true</wipe_after_delete>
    <warning_low_space_indicator>10</warning_low_space_indicator>
    <critical_space_action_blocker>5</critical_space_action_blocker>
</storage_domain>
```

12.3. 存储域的 JSON 表述

例 12.2. 存储域的 JSON 表述

```
{
  "storage_domain" : [ {
    "type" : "data",
    "master" : "false",
    "storage" : {
      "address" : "192.0.2.0",
      "type" : "nfs",
      "path" : "/storage/user/nfs"
    },
    "available" : 193273528320,
    "used" : 17179869184,
    "committed" : 0,
    "storage_format" : "v3",
    "name" : "NFS_01",
    "href" : "/api/storagedomains/8827b158-6d2e-442d-a7ee-c6fd4718aaba",
    "id" : "8827b158-6d2e-442d-a7ee-c6fd4718aaba",
    "link" : [ {
      "href" : "/api/storagedomains/8827b158-6d2e-442d-a7ee-
c6fd4718aaba/permissions",
      "rel" : "permissions"
    }, {
      "href" : "/api/storagedomains/8827b158-6d2e-442d-a7ee-
c6fd4718aaba/disks",
      "rel" : "disks"
    }, {
      "href" : "/api/storagedomains/8827b158-6d2e-442d-a7ee-
c6fd4718aaba/storageconnections",
      "rel" : "storageconnections"
    }, {
      "href" : "/api/storagedomains/8827b158-6d2e-442d-a7ee-
c6fd4718aaba/disksnapshots",
      "rel" : "disksnapshots"
    }, {
      "href" : "/api/storagedomains/8827b158-6d2e-442d-a7ee-
c6fd4718aaba/diskprofiles",
      "rel" : "diskprofiles"
    } ]
  } ]
}
```

12.4. 方法

12.4.1. 创建存储域

创建新存储域需要 **name**、**type**、**host** 和 **storage** 项。使用 **id** 属性或 **name** 项来标识 **host** 项。

在 Red Hat Enterprise Virtualization 3.6 以及更高的版本中，可以在存储域中默认启用“删除后清除数据”选项。您需要在 **POST** 请求中指定 **<wipe_after_delete>**。这个选项可以在域被创建后进行修改，但这不会改变那些已存在磁盘的“删除后清除数据”的设置。

例 12.3. 创建一个存储域

```
POST /api/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_domain>
    <name>data1</name>
    <type>data</type>
    <host id="2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"/>
    <storage>
        <type>nfs</type>
        <address>172.31.0.6</address>
        <path>/exports/RHEVX/images/0</path>
    </storage>
</storage_domain>
```

API 用户在创建存储域后把它加入到一个数据中心。

12.4.2. 更新存储域

在存储域创建完成后，只有 **name** 和 **wipe after delete** 项可以被修改。修改 **wipe after delete** 项的值不会改变那些已存在磁盘的删除后清除数据的设置。

例 12.4. 更新一个存储域

```
PUT /api/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_domain>
    <name>data2</name>
    ...
    <wipe_after_delete>true</wipe_after_delete>
    ...
</storage_domain>
```

12.4.3. 删除存储域

删除存储域需要一个 **DELETE** 请求。

例 12.5. 删除一个存储域

```
DELETE /api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed HTTP/1.1

HTTP/1.1 204 No Content
```

12.5. 存储类型

12.5.1. 存储类型

storage 项包括了一个 **type** 项，它的值包括在 **capabilities** 集合中。

storage 项还会包括和每个特定 **type** 相关的项。以下介绍了这些存储的 **type** 项。

12.5.2. NFS 存储

下表包括了 **storage** 表述中的针对于 **nfs** 的项。

表 12.2. 针对于 NFS 的项

项	类型	描述	属性
address	字符串	NFS 服务器的主机名或 IP 地址。	
path	字符串	服务器中的 NFS 挂载路径。	

12.5.3. PosixFS 存储

下表包括了 **storage** 表述中的针对于 **posixfs** 的项。

表 12.3. 针对于 PosixFS 的项

项	类型	描述	属性
address	字符串	PosixFS 服务器的主机名或 IP 地址。	
path	字符串	服务器中的 PosixFS 挂载路径。	
vfs_type	字符串	Linux 支持的 PosixFS 共享文件系统类型。	

项	类型	描述	属性
--mount_options	字符串	挂载 PosixFS 共享的选项。	

12.5.4. iSCSI 和 FCP 存储

下表包括了 **storage** 表述中的针对于 **iscsi** 和 **fcp** 的项。

表 12.4. 针对于 iSCSI 和 FCP 的项

项	类型	描述	属性
logical_unit id=	复杂数据	逻辑单元的 **id**。一个存储域可以接受多个 iSCSI 或 FCP 逻辑单元。	
override_luns	布尔值	指定是否使用新的设置来替换所有逻辑单元设置。把它的值设为 **true** 来进行覆盖。	

logical_unit 包括一组子集合。

表 12.5. 逻辑单元项

项	类型	描述	属性
address	字符串	包括存储设备的服务器地址。	
port	整数	服务器的端口号。	
target	字符串	存储设备的目标 IQN。	
username	字符串	登录到目标的 CHAP 用户名。	
password	字符串	登录到目标的 CHAP 密码。	
serial	字符串	目录的序列 ID。	

项	类型	描述	属性
vendor_id	字符串	目标的厂商名。	
product_id	字符串	目标的产品代码。	
lun_mapping	整数	映射为目标的 LUN（Logical Unit Number） 设备。	

对于 iSCSI，如果一个 **logical_unit** 表述中也包括了带有 LUN 的 iSCSI 目标，存储域在被创建时，目标会自动执行一个登录操作。

12.5.5. LocalFS 存储

下表包括了 **storage** 表述中的针对于 **localfs** 的项。

表 12.6. 针对于 Localfs 的项

项	类型	描述	属性
path	字符串	主机本地存储的路径。	

localfs 存储需要一个 **storage_type** 被设置为 **localfs** 的数据中心。这个数据中心只包括一个主机集群，这个主机集群只包括一个主机。

12.6. 导出存储域

12.6.1. 导出存储域

type 被设置为 **export** 的存储域会包括 **vms** 和 **templates** 子集合，它们包括了可以被所在存储域导出的虚拟机和模板。

> **例 12.6. 列出一个导出存储域的虚拟机子集合。**
>
> ```
> GET /api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed/vms
> Accept: application/xml
>
> HTTP/1.1 200 OK
> Content-Type: application/xml
>
> <vms>
>     <vm id="082c794b-771f-452f-83c9-b2b5a19c0399"
>       href="/api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed/
>       vms/082c794b-771f-452f-83c9-b2b5a19c0399">
>         <name>vm1</name>
> ```

```
        ...
        <storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"
          href="/api/storagedomains/fabe0451-701f-4235-8f7e-
e20e458819ed"/>
        <actions>
            <link rel="import" href="/api/storagedomains/
              fabe0451-701f-4235-8f7e-e20e458819ed/vms/
              082c794b-771f-452f-83c9-b2b5a19c0399/import"/>
        </actions>
    </vm>
</vms>
```

这些集合中的虚拟机和模板的表述与顶级虚拟机和模板的表述相似，唯一的不同是它们包括一个 **storage_domain**，以及一个 **import** 操作。

import 操作从一个 **export** 存储域导入一个虚拟机或模板。使用 **cluster** 和 **storage_domain** 来指定目标集群和存储域。

包括一个可选的 **name** 项来指定虚拟机或模板的名称。

例 12.7. 从一个导出域中导入一个虚拟机的操作

```
POST /api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed/vms/
082c794b-771f-452f-83c9-b2b5a19c0399/import HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <storage_domain>
        <name>images0</name>
    </storage_domain>
    <cluster>
        <name>Default</name>
    </cluster>
</action>
```

例 12.8. 从一个导出存储域中导入一个模板的操作

```
POST /api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed/templates/
082c794b-771f-452f-83c9-b2b5a19c0399/import HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <storage_domain>
        <name>images0</name>
    </storage_domain>
    <cluster>
        <name>Default</name>
    </cluster>
</action>
```

包括一个可选的 **clone** 布尔项来把导入的虚拟机作为一个全新的项。

例 12.9. 把导入的虚拟机作为一个全新项的操作

```
POST /api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed/vms/
082c794b-771f-452f-83c9-b2b5a19c0399/import HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <storage_domain>
        <name>images0</name>
    </storage_domain>
    <cluster>
        <name>Default</name>
    </cluster>
    <clone>true</clone>
    <vm>
        <name>MyVM</name>
    </vm>
    ...
</action>
```

包括一个可选的 **disks** 项来使用 **disk id** 项指定需要导入的磁盘。

例 12.10. 在导入操作中指定磁盘

```
POST /api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed/vms/
082c794b-771f-452f-83c9-b2b5a19c0399/import HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <cluster>
        <name>Default</name>
    </cluster>
    <vm>
        <name>MyVM</name>
    </vm>
    ...
    <disks>
        <disk id="4825ffda-a997-4e96-ae27-5503f1851d1b"/>
    </disks>
</action>
```

使用 **DELETE** 请求从一个 **export** 存储域中删除虚拟机或模板。

例 12.11. 从一个导出存储域中删除虚拟机

```
DELETE /api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed/vms/
082c794b-771f-452f-83c9-b2b5a19c0399 HTTP/1.1
Accept: application/xml
```

```
HTTP/1.1 204 No Content
```

12.7. GLANCE 镜像存储域

12.7.1. Glance 镜像存储域

类型被设置为 **Image** 的存储域代表了一个 OpenStack 镜像服务实例，它已经被添加到 Red Hat Enterprise Virtualization 环境中作为一个外部供应商。这些 Glance 镜像存储域包括了一个 **images** 子集合，它包括了可以导出到 Glance 镜像存储域或从 Glance 镜像存储域导入的虚拟机镜像。

例 12.12. 列出 Glance 镜像存储域的镜像子集合

```
GET /api/storagedomains/00000000-0000-0000-0000-000000000000/images
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<images>
  <image href="/api/storagedomains/00000000-0000-0000-0000-
000000000000/images/
    00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-
000000000000">
    <actions>
      <link href="/api/storagedomains/00000000-0000-0000-0000-
000000000000/images/
        00000000-0000-0000-0000-000000000000/import" rel="import"/>
    </actions>
    <name>RHEL_65_Disk_001</name>
    <storage_domain href="/api/storagedomains/00000000-0000-0000-0000-
000000000000"
      id="00000000-0000-0000-0000-000000000000"/>
  </image>
  <image href="/api/storagedomains/00000000-0000-0000-0000-
000000000000/images/
    00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-
000000000000">
    <actions>
      <link href="/api/storagedomains/00000000-0000-0000-0000-
000000000000/images/
        00000000-0000-0000-0000-000000000000/import" rel="import"/>
    </actions>
    <name>RHEL_65_Disk_002</name>
    <storage_domain href="/api/storagedomains/00000000-0000-0000-0000-
000000000000"
      id="00000000-0000-0000-0000-000000000000"/>
  </image>
</images>
```

import 操作会从一个 Glance 镜像存储域中导入虚拟机镜像。使用 **storage_domain** 指定目标存储域，使用 **cluster** 指定目标集群。

包括一个可选的 **name** 项来指定虚拟机或模板的名称。

例 12.13. 从一个 Glance 镜像存储域中导入一个虚拟机的操作

```
POST /api/storagedomains/00000000-0000-0000-000000000000/images/
00000000-0000-0000-000000000000/import HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <storage_domain>
        <name>images0</name>
    </storage_domain>
    <cluster>
        <name>images0</name>
    </cluster>
</action>
```

您也可以使用 **import_as_template** 把镜像作为模板导入：

例 12.14. 从 Glance 镜像域中导入虚拟机作为一个模板

```
POST /api/storagedomains/00000000-0000-0000-000000000000/images/
00000000-0000-0000-000000000000/import HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <storage_domain>
        <name>images0</name>
    </storage_domain>
    <cluster>
        <name>images0</name>
    </cluster>
    </import_as_template>true</import_as_template>
</action>
```

12.8. 子集合

12.8.1. 文件子集合

每个存储域中的 **files** 子集合可以被客户端使用来列出有效的文件。这个子集合只针对于那些包括了管理员通过 Red Hat Enterprise Virtualization Manager 上传的 ISO 镜像和虚拟软硬（VFD）的 ISO 存储域。

添加到虚拟机上的 CD-ROM 设备需要 ISO 存储域中的 **files** 子集合中的一个 ISO 镜像。

例 12.15. 列出一个 ISO 存储域中的文件子集合

```
GET /api/storagedomains/00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da/files
HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<files>
    <file id="en_winxp_pro_with_sp2.iso"
      href="/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-
3c898fa8b6da/files/
      en_winxp_pro_with_sp2.iso">
        <name>en_winxp_pro_with_sp2.iso</name>
        <type>iso</type>
        <storage_domain id="00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da"
          href="/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-
3c898fa8b6da"/>
    </file>
    <file id="boot.vfd"
      href="/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-
3c898fa8b6da/files/
      boot.vfd">
        <name>boot.vfd</name>
        <type>vfd</type>
        <storage_doman id="00f0d9ce-da15-4b9e-9e3e-3c898fa8b6da"
          href="/api/storagedomains/00f0d9ce-da15-4b9e-9e3e-
3c898fa8b6da"/>
    </file>
</files>
```

和其它资源一样，**files** 包括了 **id** 和 **href** 属性。**name** 项包括了文件名。

12.9. 操作

12.9.1. 导入一个存在的存储域

API 为用户提供了在不需要对存储设备进行重新格式化，就可以从一个 Red Hat Enterprise Virtualization Manager 实例上删除一个 ISO 存储域或导出存储域，并把它导入到其它实例上的功能。导入操作和添加新存储域的过程相似，唯一的不同是不需要指定 **name**。

例 12.16. 导入一个存在的存储域。

```
POST /api/storagedomains HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<storage_domain>
    <type>export</type>
    <storage>
        <type>nfs</type>
        <address>172.31.0.6</address>
```

```
            <path>/exports/RHEVX/export-domain</path>
        </storage>
        <host id="2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"/>
    </storage_domain>

    HTTP/1.1 201 Created
    Content-Type: application/xml

    <storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"
      href="/api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed">
        <name>export1</name>
        ...
    </storage_domain>
```

12.9.2. 删除存储域

在 **DELETE** 请求的内容中使用 **storage_domain** 指定要被删除的存储域。**storage_domain** 使用以下格式：

```
<storage_domain>
    <host id="..."/>
</storage_domain>
```

或

```
<storage_domain>
    <host>
        <name>...</name>
    </host>
</storage_domain>
```

格式化存储域

API 包括了一个可选的 **format** 项来指定在删除后是否对存储域进行格式化。

例 12.17. 在删除后格式化存储域

```
<storage_domain>
    <host id="..."/>
    <format>true</format>
</storage_domain>
```

如果没有使用 **format** 项，存储域将不会被格式化。

逻辑删除存储域

API 提供了一个逻辑删除存储域的功能（存储域的数据会被保存，并可以被用来进行其它的导入操作）。使用 **destroy** 项来逻辑删除存储域并保留它上面的数据。

例 12.18. 逻辑删除一个存储域

```
<storage_domain>
```

```
    <host id="..."/>
    <destroy>true</destroy>
</storage_domain>
```

第 13 章 存储连接

13.1. 存储连接项

表 13.1. 存储连接基本项

项	类型	描述	属性
`type`	`nfs`、`posixfs`、`local` 或 `iscsi` 之一	存储域类型。	
`address`	字符串	存储域的主机名或 IP 地址。	只限于 NFS 和 iSCSI
`host`	字符串	hypervisor 的 `id` 或 `name`。`host` 是可选的，使用它可以通过主机连接到存储，如果没有指定它，存储详情会保存在数据库中。	

表 13.2. 存储连接项（基于文件的存储）

项	类型	描述	属性
`path`	字符串	存储域的挂载文件路径。已经被存储连接使用的 `path` 不能被更新。	
`mount_options`	字符串	挂载 PosixFS 共享的选项。	
`vfs_type`	字符串	Linux 支持的 PosixFS 共享文件系统类型。	
`nfs_version`	字符串	NFS 使用的版本。	
`nfs_timeo`	整数	NFS 客户端为了完成一个请求可以等待的时间（以 0.1 秒为单位）。	
`nfs_retrans`	整数	NFS 客户端为了完成一个请求尝试重新传输的次数。	

表 13.3. 存储连接项（iSCSI 存储）

项	类型	描述	属性
port	整数	iSCSI 存储域使用的 TCP 端口。	
target	字符串	存储设备的目标 IQN。	
username	字符串	登录到一个目标的 CHAP 用户名。	
password	字符串	登录到一个目标的 CHAP 密码。	

13.2. 存储连接资源的 XML 表述

例 13.1. 存储连接资源的 XML 表述

```
<storage_connections>
  <storage_connection href= "/api/storageconnections/608c5b96-9939-4331-
96b5-197f28aa2e35"    id="608c5b96-9939-4331-96b5-197f28aa2e35">
    <address>domain.example.com</address>
    <type>nfs</type>
    <path>/var/lib/exports/iso</path>
  </storage_connection>
  <storage_connection href= "/api/storageconnections/2ebb3f78-8c22-4666-
8df4-e4bb7fec6b3a" id="2ebb3f78-8c22-4666-8df4-e4bb7fec6b3a">
    <address>domain.example.com</address>
    <type>posixfs</type>
    <path>/export/storagedata/username/data</path>
    <vfs_type>nfs</vfs_type>
  </storage_connection>
</storage_connections>
```

13.3. 方法

13.3.1. 创建新存储连接

创建新存储连接需要一个 **POST** 请求。

新存储连接可以在没有添加存储域的情况下被创建。主机的 **id** 和 **name** 是可选的，使用它们将会通过主机连接到存储。

例 13.2. 创建新存储连接

```
POST /api/storageconnections HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_connection>
```

```
    <type>nfs</type>
    <address>domain.example.com</address>
    <path>/export/storagedata/username/data</path>
    <host>
      <name>Host_Name</name>
    </host>
</storage_connection>
```

13.3.2. 删除存储连接

删除存储连接需要一个 **DELETE** 请求。存储连接只有在存储域和 LUN 磁盘都没有使用它时才可以被删除。

主机 **name** 和 **id** 都是可选的，使用它们可以从主机上卸载连接。

例 13.3. 删除存储连接

```
DELETE /api/storageconnections/Storage_Connection_ID HTTP/1.1
Accept: application/xml
Content-type: application/xml

<host>
  <name>Host_Name</name>
</host>
```

13.3.3. 更新存储连接

更新已经存在的存储连接需要一个 **PUT** 请求。存储域必须在维护模式或没有被附加的情况下才可以更新连接。

主机 **name** 和 **id** 都是可选的，使用它们时，主机会试图通过一个连接来更新存储详情。

例 13.4. 更新存储连接

```
PUT /api/storageconnections/Storage_Connection_ID HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_connection>
  <address>updated.example.domain.com</address>
  <host>
      <name>Host_name</name>
   </host>
</storage_connection>
```

13.3.4. 更新 iSCSI 存储连接

更新已经存在的 iSCSI 存储连接需要一个 **PUT** 请求。iSCSI 存储域必须在维护模式或没有被附加的情况下才可以更新连接。

例 13.5. 更新存储连接

```
PUT /api/storageconnections/Storage_Connection_ID HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_connection>
  <port>3456</port>
</storage_connection>
```

13.3.5. 使用存在的存储连接添加新存储域

使用存在的存储连接添加新存储域需要一个 **POST** 请求。这只适用于基于文件的存储域：**NFS**、**POSIX** 和 **local**。

例 13.6. 使用存在的存储连接添加新存储域

```
POST /api/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_domain>
  <name>New_Domain</name>
  <type>data</type>
 <storage id="Storage_Connection_ID"/>
  <host>
    <name>Host_Name</name>
  </host>
</storage_domain>
```

13.3.6. 为 iSCSI 存储附加一个额外的存储连接

为 iSCSI 存储附加一个额外的存储连接需要一个 **POST** 请求。

例 13.7. 为 iSCSI 存储附加一个额外的存储连接

```
POST /api/storagedomains/iSCSI_Domain_ID/storageconnections HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_connection id="Storage_Connection_ID">
</storage_connection>
```

13.3.7. 从 iSCSI 存储上删除一个存储连接

从 iSCSI 存储上删除一个存储连接需要一个 **DELETE** 请求。

例 13.8. 从 iSCSI 存储上删除一个存储连接

```
DELETE
/api/storagedomains/iSCSI_Domain_ID/storageconnections/Storage_Connectio
n_ID HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

第 14 章 主机

14.1. 主机项

hosts 集合提供了 Red Hat Enterprise Virtualization 环境中的主机的信息。API 用户可以通过从进入点 URI 获得的 **rel="hosts"** 链接来访问这些信息。

使用带有 **All-Content: true** 头的 **GET** 请求可以获得额外信息。

以下表格包括了主机资源表述中的特定项的信息。

表 14.1. 主机项

项	类型	描述	属性
link rel="storage"	关系	主机存储的 **storage** 子集合链接。	
link rel="nics"	关系	主机网络接口的 **nics** 子集合链接。	
link rel="tags"	关系	主机标签的 **tags** 子集合链接。	
link rel="permissions"	关系	主机权限的 **permissions** 子集合链接。	
link rel="statistics"	关系	主机统计数据的 **statistics** 子集合链接。	
link rel="hooks"	关系	主机 hook 的 **hooks** 子集合链接。	
name	字符串	主机的唯一标识。	
root_password	字符串	主机的 root 密码，只包括在创建时由客户端提供的主机表述中。	
comment	字符串	主机的备注信息。	
address	字符串	主机的 IP 地址或主机名。	
certificate	复杂数据	主机证书的详情，包括 **organization** 和 **subject**。	

项	类型	描述	属性
status	参阅以下信息	主机状态。	
cluster id=	GUID	包括这个主机的集群标识。	
port	整数	在这个主机上运行的 VDSM 守护进程监听的端口。	
type	**rhel** 或 **rhev-h**	主机类型。	
storage_manager priority=	布尔值：true 或 false	指定主机是否是一个存储管理器。	
version major= minor= build= revision= full_version=	复杂数据	主机的兼容级别。	
hardware_information	复杂数据	主机硬件相关的信息，包括 **manufacturer**、**version**、**serial_number**、**product_name**、**uuid** 和 **family**。	
power_management type=	复杂数据	主机电源管理的配置选项，包括 **enabled**、**options**、**kdump_detection**、**automatic_pm_enabled** 和 **agents**。如需了解更多与主机电源管理选项相关的信息，请参阅 第 14.4 节 “电源管理项”。	
ksm	布尔值：true 或 false	**true**：Kernel SamePage Merging（KSM）被启用。	
transparent_hugepages	布尔值：true 或 false	**true**：Transparent Hugepages 被启用。	
iscsi	复杂数据	主机的 SCSI **initiator**。	
ssh	复杂数据	使用 SSH 与主机进行连接的详情，包括 **port** 和 **fingerprint**。	

项	类型	描述	属性
cpu	复杂数据	主机 CPU 的统计数据，包括 CPU 的 **name**、**topology cores=**、**topology sockets=**、**topology threads=** 和 **speed** 子项。**topology cores=** 是内核的总和，而 **topology sockets=** 是物理 CPU 的总和。虚拟机可用的内核总数是物理 CPU 的数量乘以每个物理 CPU 的内核数。	
memory	整数	主机内存的总量（以字节为单位）。	
max_scheduling_memory	整数	可以在调度中使用的最大内存数量（以字节为单位）。	
summary	复杂数据	主机上运行的虚拟机的统计数据总结，包括 **active**、**migrating** 和 **total** 虚拟机子项。	
os type=	复杂数据	安装在主机上的操作系统详情，包括 **version full_version=**。	
libvirt_version major= minor= build= revision= full_version=	复杂数据	主机的 libvirt 兼容级别。	

status 包括以下值之一：**down**、**error**、**initializing**、**installing**、**install_failed**、**maintenance**、**non_operational**、**non_responsive**、**pending_approval**、**preparing_for_maintenance**、**connecting**、**reboot**、**unassigned** 和 **up**。这些状态包括在 **capabilities** 下的 **host_states** 中。

14.2. 主机的 XML 表述

例 14.1. 主机的 XML 表述

```
<host id="00000000-0000-0000-0000-000000000000"
  href="/api/hosts/00000000-0000-0000-0000-000000000000">
    <actions>
        <link rel="install"
          href="/api/hosts/00000000-0000-0000-0000-
000000000000/install"/>
        <link rel="forceselectspm"
          href="/api/hosts/00000000-0000-0000-0000-
000000000000/forceselectspm"/>
        <link rel="activate"
          href="/api/hosts/00000000-0000-0000-0000-
```

```
000000000000/activate"/>
        <link rel="fence"
          href="/api/hosts/00000000-0000-0000-0000-
000000000000/fence"/>
        <link rel="deactivate"
          href="/api/hosts/00000000-0000-0000-0000-
000000000000/deactivate"/>
        <link rel="approve"
          href="/api/hosts/00000000-0000-0000-0000-
000000000000/approve"/>
        <link rel="iscsilogin"
          href="/api/hosts/00000000-0000-0000-0000-
000000000000/iscsilogin"/>
        <link rel="iscsidiscover"
          href="/api/hosts/00000000-0000-0000-0000-
000000000000/iscsidiscover"/>
        <link rel="commitnetconfig"
          href="/api/hosts/00000000-0000-0000-0000-
000000000000/commitnetconfig"/>
    </actions>
    <name>host1</name>
    <link rel="storage"
      href="/api/hosts/00000000-0000-0000-0000-000000000000/storage"/>
    <link rel="nics"
      href="/api/hosts/00000000-0000-0000-0000-000000000000/nics"/>
    <link rel="tags"
      href="/api/hosts/00000000-0000-0000-0000-000000000000/tags"/>
    <link rel="permissions"
      href="/api/hosts/00000000-0000-0000-0000-
000000000000/permissions"/>
    <link rel="statistics"
      href="/api/hosts/00000000-0000-0000-0000-
000000000000/statistics"/>
    <link rel="hooks"
      href="/api/hosts/00000000-0000-0000-0000-000000000000/hooks"/>
    <address>host1.example.com</address>
    <certificate>
        <organization>exampleorg</organization>
        <subject>O=exampleorg,CN=XX.XX.XX.XX</subject>
    </certificate>
    <status>
        <state>up</state>
    </status>
    <cluster href="/api/clusters/00000000-0000-0000-0000-000000000000"
      id="00000000-0000-0000-0000-000000000000"/>
    <port>54321</port>
    <type>rhel</type>
    <storage_manager priority="2">true</storage_manager>
    <version major="4" minor="10" build="2" revision="0"
full_version="vdsm-4.10.2-1.13.el6ev"/>
    <power_management type="apc">
        <enabled>false</enabled>
        <options>
          <option name="secure" value="false"/>
        <options>
        <automatic_pm_enabled>true</automatic_pm_enabled>
```

```
        <kdump_detection>true</kdump_detection>
    </power_management>
    <ksm>
        <enabled>true</enabled>
    </ksm>
    <transparent_hugepages>
        <enabled>true</enabled>
    </transparent_hugepages>
    <iscsi>
        <initiator>iqn.2001-04.com.example:diskarrays-sn-
a8675309</initiator>
    </iscsi>
    <ssh>
        <port>22</port>

<fingerprint>00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00</fingerprin
t>
    </ssh>
    <cpu>
        <topology cores="2" sockets="1"/>
        <name>Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz</name>
        <speed>2993</speed>
    </cpu>
    <summary>
        <active>2</active>
        <migrating>0</migrating>
        <total>3</total>
    </summary>
    <os type="RHEL">
        <version full_version="6Server-6.5.0.1.el6"/>
    </os>
    <libvirt_version major="0" minor="10" build="2" revision="0"
full_version="libvirt-0.10.2-15.el6"/>
</host>
```

14.3. 主机的 JSON 表述

例 14.2. 主机的 JSON 表述

```
{
  "host" : [ {
    "address" : "198.51.100.0",
    "certificate" : {
      "organization" : "example.com",
      "subject" : "O=example.com,CN=192.0.2.0"
    },
    "status" : {
      "state" : "up"
    },
    "cluster" : {
      "href" : "/api/clusters/00000001-0001-0001-0001-0000000002fb",
      "id" : "00000001-0001-0001-0001-0000000002fb"
    },
    "port" : "54321",
```

```
"type" : "rhel",
"storage_manager" : {
  "value" : "true",
  "priority" : "5"
},
"spm" : {
  "priority" : "5"
},
"version" : {
  "major" : "4",
  "minor" : "16",
  "build" : "8",
  "revision" : "1",
  "full_version" : "vdsm-4.16.8.1-6.el6ev"
},
"hardware_information" : {
  "manufacturer" : "System Manufacturer To Be Filled By O.E.M.",
  "version" : "System Version To Be Filled By O.E.M.",
  "serial_number" : "Serial Number To Be Filled By O.E.M.",
  "product_name" : "Product Name To Be Filled By O.E.M.",
  "uuid" : "9fa0a1a2-a3a4-a5a6-a7a8-a9aaabacadae",
  "family" : "Family To Be Filled By O.E.M.",
  "supported_rng_sources" : {
    "source" : [ "RANDOM" ]
  }
},
"power_management" : {
  "enabled" : "false",
  "options" : {
    "option" : [ {
      "name" : "secure",
      "value" : "false"
    } ]
  },
  "automatic_pm_enabled" : "true",
  "kdump_detection" : "true",
  "type" : "apc"
},
"ksm" : {
  "enabled" : "false"
},
"transparent_hugepages" : {
  "enabled" : "true"
},
"iscsi" : {
  "initiator" : "iqn.1994-05.com.example:795610ff2632"
},
"ssh" : {
  "port" : "22",
  "fingerprint" : "77:27:38:25:8f:60:8d:93:9c:2c:b0:cb:5e:19:f4:53"
},
"cpu" : {
  "topology" : {
    "sockets" : "1",
    "cores" : "4",
    "threads" : "1"
```

```
      },
      "name" : "Intel(R) Core(TM)2 Quad CPU    Q9550  @ 2.83GHz",
      "speed" : 2833
    },
    "memory" : 2989490176,
    "max_scheduling_memory" : 2584739840,
    "summary" : {
      "active" : "0",
      "migrating" : "0",
      "total" : "0"
    },
    "protocol" : "stomp",
    "os" : {
      "version" : {
        "full_version" : "6Server - 6.6.0.2.el6"
      },
      "type" : "RHEL"
    },
    "libvirt_version" : {
      "major" : "0",
      "minor" : "10",
      "build" : "2",
      "revision" : "0",
      "full_version" : "libvirt-0.10.2-46.el6_6.2"
    },
    "kdump_status" : "disabled",
    "selinux" : {
      "mode" : "enforcing"
    },
    "auto_numa_status" : "unknown",
    "numa_supported" : "false",
    "live_snapshot_support" : "true",
    "actions" : {
      "link" : [ {
        "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/fence",
        "rel" : "fence"
      }, {
        "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/approve",
        "rel" : "approve"
      }, {
        "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/forceselectspm",
        "rel" : "forceselectspm"
      }, {
        "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/iscsilogin",
        "rel" : "iscsilogin"
      }, {
        "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/iscsidiscover",
        "rel" : "iscsidiscover"
      }, {
        "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/commitnetconfig",
```

```
        "rel" : "commitnetconfig"
      }, {
        "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/deactivate",
        "rel" : "deactivate"
      }, {
        "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/install",
        "rel" : "install"
      }, {
        "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/activate",
        "rel" : "activate"
      } ]
    },
    "name" : "Host-07",
    "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-d86f005c93fe",
    "id" : "ea7aa772-d2af-4a5c-9350-d86f005c93fe",
    "link" : [ {
      "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/storage",
      "rel" : "storage"
    }, {
      "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-d86f005c93fe/nics",
      "rel" : "nics"
    }, {
      "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/numanodes",
      "rel" : "numanodes"
    }, {
      "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-d86f005c93fe/tags",
      "rel" : "tags"
    }, {
      "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/permissions",
      "rel" : "permissions"
    }, {
      "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-
d86f005c93fe/statistics",
      "rel" : "statistics"
    }, {
      "href" : "/api/hosts/ea7aa772-d2af-4a5c-9350-d86f005c93fe/hooks",
      "rel" : "hooks"
    } ]
  } ]
}
```


14.4. 电源管理项

用户可以使用 **power_management** 项来设置主机隔离所需的电源管理配置。在配置 **power_management** 时需要一些子项。

表 14.2. 电源管理选项

项	类型	描述	属性
type=	隔离设备代码	**capabilities** 集合中包括了有效的隔离设备代码。	
enabled	布尔值：true 或 false	指定是否启用电源管理配置。	
address	字符串	主机的主机名或 IP 地址。	
username	字符串	电源管理的用户名。	
password	字符串	电源管理的密码。	
options	复杂数据	所选 **type=** 的隔离选项是由选项 **name=""** 和 **value=""** 指定的。	
agents	复杂数据	当使用多个隔离时，用来指定代理的选项。使用 **order** 子项来对隔离代理进行优先级排序。代理会根据这个顺序被使用，直到隔离操作成功进行。如果两个或多个隔离代理有相同的 **order**，它们会同时运行。其它子项包括 **type**、**ip**、**user**、**password** 和 **options**。	
automatic_pm_enabled	布尔值：true 或 false	开启或关闭用于节省能源的主机自动电源管理功能。如果设置为 **true**，当集群的负载低时，主机会被自动关闭；当需要时，主机会被自动开启。在创建一个主机时，这个选项被设置为 **true**，除非被用户禁用。	
kdump_detection	布尔值：true 或 false	在关闭主机前，检查是否有正在主机上运行的 kdump。当设置为 **true** 时，主机不会在有 kdump 进程的情况下关闭。在启用主机的电源管理功能时，这个选择被设置为 **true**，除非被用户禁用。	

options 项需要一组 **option** 子项。每个 **option** 子项需要一个 **name** 和一个 **type** 属性。一些特定的选择只适用于特定的隔离类型（在 **capabilities** 集合中定义）。

当对主机资源使用 **POST** 请求时，新主机会包括一个可选的 **power_management** 配置。**power_management** 配置可以通过使用 **PUT** 请求进行更新。

例 14.3. 主机电源管理的 XML 表述

```
<host id="2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"
  href="/api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3">
    <name>host1</name>
    ...
    <power_management type="ilo">
        <enabled>true</enabled>
        <address>192.168.1.107</address>
        <username>admin</username>
        <password>p@55w0Rd!</password>
        <options>
            <option name="secure" value="true"/>
            <option name="port" value="54345"/>
            <option name="slot" value="3"/>
        </options>
        <agents>
            <agent id="07f0b9ce-923a-4a96-a532-3c898fa8b6da">
                <type>apc</type>
                <order>1</order>
                <ip>192.168.1.111</ip>
                <user>example</user>
                <password>p@55w0rd!</password>
                <port>9</port>
                <options>
                    <option name="power_wait" value="5"/>
                    <option name="secure" value="false"/>
                </options>
            </agent>
            <agent id="50c71ba2-8495-11e0-b931-e20e458819ed">
                <type>rsa</type>
                <order>2</order>
                <ip>192.168.1.112</ip>
                <user>example</user>
                <password>p@55w0rd!</password>
                <port>9</port>
                <options>
                    <option name="power_wait" value="5"/>
                    <option name="secure" value="false"/>
                </options>
            </agent>
        </agents>
        <automatic_pm_enabled>true</automatic_pm_enabled>
        <kdump_detection>true</kdump_detection>
    </power_management>
    ...
</host>
```

14.5. 内存管理项

API 为主机内存管理提供了两个配置选项。

Kernel SamePage Merging (KSM) 通过把多个相同的内存页合并为一个来减小内存页的标识数量，这可以帮助对内存的密度进行优化。KSM 使用 **ksm** 项。

例 14.4. 设置 KSM 内存管理

```
PUT /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3 HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<host id="2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"
  href="/api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3">
    <ksm>true</ksm>
</host>
```

Transparent Hugepage support 可以把内存页扩展到超过标准 4kB 限制的大小。这可以减少对内存的消耗，并提高主机的性能。Transparent Hugepage 支持使用 **transparent_hugepages** 项。

例 14.5. 设置 Transparent Hugepage 内存管理

```
PUT /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3 HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<host id="2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"
  href="/api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3">
    <transparent_hugepages>true</transparent_hugepages>
</host>
```

capabilities 集合中包括了支持 Transparent Hugepage 的信息。

14.6. 方法

14.6.1. 创建主机

创建新主机需要 **name**、**address** 和 **root_password** 项。

例 14.6. 创建一个主机

```
POST /api/hosts HTTP/1.1
Accept: application/xml
Content-type: application/xml

<host>
    <name>host2</name>
    <address>host2.example.com</address>
    <root_password>p@55w0Rd!</root_password>
</host>
```

创建新主机只适用于创建 Red Hat Enterprise Linux 主机。Red Hat Enterprise Virtualization Manager 会自动发现 hypervisor 主机，但需要在使用前批准它。

root_password 项只包括在客户端提供的初始表述中，它不会包括在以后请求的返回表述中。

14.6.2. 更新主机

在创建完成
后，**name**、**description**、**cluster**、**power_management**、**transparent_hugepages** 和 **ksm** 项可以被更新。

例 14.7. 更新一个主机

```
POST /api/hosts/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<host>
    <name>host3</name>
</host>
```

14.6.3. 删除主机

删除主机需要一个 **DELETE** 请求。

例 14.8. 删除一个主机

```
DELETE /api/hosts/00000000-0000-0000-0000-000000000000 HTTP/1.1

HTTP/1.1 204 No Content
```

14.7. 子集合

14.7.1. 主机网络接口子集合

14.7.1.1. 主机网络接口子集合

nics 子集合代表了主机的物理网络接口。使用带有 **All-Content: true** 头的 **GET** 请求可以获取额外的信息。表述中的每个 **host_nic** 项都代表一个网络接口，并包括以下项：

表 14.3. 主机网络接口的项

项	类型	描述	属性
name	字符串	主机网络接口名，如 **eth0**。	[a]
link rel="statistics"	关系	主机网络接口统计数据的 **statistics** 子集合链接。	

项	类型	描述	属性
link rel="labels"	关系	主机网络接口标签的 **labels** 子集合链接。	
link rel="master"	关系	主绑定接口的标识（如果有从接口）。	
host id=	GUID	主机的标识。	
network id=	GUID	指定网卡所连接的网络。	[b]
mac address=	字符串	接口的 MAC 地址。	
ip address= netmask= gateway= mtu=	复杂数据	接口的 IP 级别配置。	
mtu	复杂数据	网络接口的最大传输单位。	
--boot_protocol	列举	在主机启动时分配 IP 地址的协议，它的有效值包括在 **capabilities** 中。	
status	列举	网络接口的连接状态。这些状态在 **capabilities** 下的 **host_nic_states** 中列出。	
vlan id	整数	接口所代表的 VLAN。	
bonding	复杂数据	绑定接口的 **options** 和 **slave** NIC 列表。	[c]
bridged	布尔值	指定是否是桥接网络。**true** 代表桥接网络，**false** 代表非桥接网络。	
properties	复杂数据	为网络的网桥选项定义自定义属性。每个 **property** 都包括 **name** 和 **value** 子项。	

项	类型	描述	属性

[a] 只在添加绑定接口时需要。其它接口是只读的，不需要被添加。

[b] 只在添加绑定接口时需要。其它接口是只读的，不需要被添加。

[c] 只在添加绑定接口时需要。其它接口是只读的，不需要被添加。

例 14.9. 主机上的网络接口的 XML 表述

```
<host_nic id="00000000-0000-0000-0000-000000000000"
  href="/api/hosts/00000000-0000-0000-0000-000000000000/nics/
  00000000-0000-0000-0000-000000000000">
  <actions>
    <link rel="attach"
      href="/api/hosts/00000000-0000-0000-0000-000000000000/nics/
      00000000-0000-0000-0000-000000000000/attach"/>
    <link rel="detach"
      href="/api/hosts/00000000-0000-0000-0000-000000000000/nics/
      00000000-0000-0000-0000-000000000000/detach"/>
  </actions>
  <name>bond0</name>
  <link rel="labels"
    href= "/api/hosts/00000000-0000-0000-0000-000000000000/nics/
    00000000-0000-0000-0000-000000000000/labels"/>
  <link rel="statistics"
    href="/api/hosts/00000000-0000-0000-0000-000000000000/nics/
    00000000-0000-0000-0000-000000000000/statistics"/>
  <host id="00000000-0000-0000-0000-000000000000"
    href="/api/hosts/00000000-0000-0000-0000-000000000000"/>
  <network id="00000000-0000-0000-0000-000000000000"
    href="/api/networks/00000000-0000-0000-0000-000000000000"/>
  <mac address="00:00:00:00:00:00"/>
  <ip address="XX.XX.XX.XX" netmask="255.255.255.0"
gateway="XX.XX.XX.XX"/>
  <boot_protocol>dhcp</boot_protocol>
  <status>
    <state>up</state>
  </status>
  <bonding>
    <options>
      ...
    </options>
    <slaves>
      <host_nic id="00000000-0000-0000-0000-000000000000"/>
      <host_nic id="00000000-0000-0000-0000-000000000000"/>
    </slaves>
  </bonding>
  <mtu>1500</mtu>
  <bridged>true</bridged>
  <custom_configuration>false</custom_configuration>
  <properties>
    <property>
      <name>bridge_opts</name>
```

```
        <value>
          forward_delay=1500 group_fwd_mask=0x0 multicast_snooping=1
        </value>
      </property>
    </properties>
  </host_nic>
```

您在使用 REST API 时，只能创建绑定接口。其它网络接口可以通过使用 **PUT** 请求更新它们的 **network**、**ip** 和 **boot_protocol** 项。

当添加一个网络接口时，**name** 和 **network** 项是必需的。使用 **id** 属性或 **name** 项来指定 **network** 项。

```
POST /api/hosts HTTP/1.1
Accept: application/xml
Content-type: application/xml

<host_nic>
    <name>MyNIC</name>
    <network id="00000000-0000-0000-0000-000000000000">
        <name>MyNetwork</name>
    </network>
</host_nic>
```

您在使用 REST API 时，可以通过一个 **PUT** 请求修改网络接口。

```
PUT /api/hosts/00000000-0000-0000-0000-000000000000/nics/
00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<host_nic>
    <ip address="XX.XX.XX.XX" netmask="255.255.255.0"
gateway="XX.XX.XX.XX"/>
</host_nic>
```

您在使用 REST API 时，可以通过一个 **DELETE** 请求删除网络接口。

```
DELETE /api/hosts/00000000-0000-0000-0000-000000000000/nics/
00000000-0000-0000-0000-000000000000 HTTP/1.1

HTTP/1.1 204 No Content
```

14.7.1.2. 绑定接口

绑定接口由带有 **bonding** 项的 **host_nic** 资源来代表。

表 14.4. 绑定接口属性

项	类型	描述	属性

项	类型	描述	属性
options	复杂数据	绑定接口的 **option** 项列表。每个 **option** 都包括了 **name** 和 **value** 属性。	[a]
slaves	复杂数据	绑定接口的从 **host_nic id=** 项列表。	[b]

[a] 只在添加绑定接口时需要。其它接口是只读的，不需要被添加。

[b] 只在添加绑定接口时需要。其它接口是只读的，不需要被添加。

API 用户在创建新的 **host_nic**（**POST**）或更新 **host_nic**（**PUT**）时可以创建新的绑定。使用 **id** 或 **name** 项指定从 **host_nic** 项。

例 14.10. 创建绑定接口

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/nics HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<host_nic>
    <name>bond4</name>
    <network id="e657d631-657d-42bb-a536-73501a085d85"/>
    <bonding>
        <options>
            ...
        </options>
        <slaves>
            <host_nic id="eb14e154-5e73-4f7f-bf6b-7f52609d94ec"/>
            <host_nic id="6aede5ca-4c54-4b37-a81b-c0d6b53558ea"/>
        </slaves>
    </bonding>
</host_nic>
```

重要

绑定接口的有效值只包括 **bond0**、**bond1**、**bond2**、**bond3** 和 **bond4**。

使用一个对绑定接口的 **DELETE** 请求可以删除它。

重要

对绑定接口配置的改变必须被明确提交。

14.7.1.3. 网络接口自定义属性

自定义属性可以被应用到网络接口。每个属性都包括 **name** 和 **value** 子项。为了修改网络接口的自定义属性，需要执行一个带有 **setupnetworks** 操作的 **POST** 请求。

表 14.5. 主机网络接口的自定义桥接选项的项

项	类型	描述
name	字符串	属性的唯一标识。桥接选项有 **bridge_opts** 的一组名称。
value	字符串	桥接选项，有一个有效的关键字和值（格式是 [key]=[value]）代表。使用白空格符可以分隔多个项。以下列出的是有效的关键字，以及相关的值示例： **forward_delay**=*1500* **gc_timer**=*3765* **group_addr**=*1:80:c2:0:0:0* **group_fwd_mask**=*0x0* **hash_elasticity**=*4* **hash_max**=*512* **hello_time**=*200* **hello_timer**=*70* **max_age**=*2000* **multicast_last_member_count**=*2* **multicast_last_member_interval**=*100* **multicast_membership_interval**=*26000* **multicast_querier**=*0* **multicast_querier_interval**=*25500* **multicast_query_interval**=*13000* **multicast_query_response_interval**=*1000* **multicast_query_use_ifaddr**=*0* **multicast_router**=*1* **multicast_snooping**=*1* **multicast_startup_query_count**=*2* **multicast_startup_query_interval**=*3125*

例 14.11. 主机网络接口属性子集合的 XML 表述

```
<host_nic>
  ...
  <properties>
    <property>
      <name>bridge_opts</name>
      <value>
        forward_delay=1500 group_fwd_mask=0x0 multicast_snooping=1
      </value>
    </property>
```

```
  </properties>
  ...
</host_nic>
```

14.7.1.4. 网络接口统计数据

每个主机网络接口都有一个提供主机网络统计数据的 **statistics** 子集合。每个 **statistic** 都包括以下项：

表 14.6. 主机网络接口统计数据项

项	类型	描述
name	字符串	统计数据记录的唯一标识。
description	字符串	统计数据的描述。
unit	字符串	统计数据值所使用的单位。
type	**GAUGE** 或 **COUNTER**	统计数据类型。
values type=	**INTEGER** 或 **DECIMAL**	统计数据值的数据类型。
value	复杂数据	包括 **datum** 的数据集。
datum	请参阅 **values type**	**value** 中的一条数据。
host_nic id=	关系	与包括 **host_nic** 资源的关系。

下表列出了主机的网络接口的统计数据类型。

表 14.7. 主机 NIC 统计数据类型

名称	描述
data.current.rx	数据接收频率（字节/每秒）
data.current.tx	数据传输频率（字节/每秒）
errors.total.rx	接收数据中的错误总数
errors.total.tx	传输数据中的错误总数

例 14.12. 网络接口统计数据子集合的 XML 表述

```
<statistics>
    <statistic id="ecd0559f-e88f-3330-94b4-1f091b0ffdf7"
      href="/api/hosts/25fcdd2e-d452-11e0-bb4d-525400d75548/nics/
      c34728e8-4338-4540-ac9b-86b8582e602e/statistics/
      ecd0559f-e88f-3330-94b4-1f091b0ffdf7">
        <name>data.current.rx</name>
        <description>Receive data rate</description>
        <values type="DECIMAL">
            <value>
                <datum>0</datum>
            </value>
        </values>
        <type>GAUGE</type>
        <unit>BYTES_PER_SECOND</unit>
        <host_nic id="c34728e8-4338-4540-ac9b-86b8582e602e"
          href="/api/hosts/25fcdd2e-d452-11e0-bb4d-525400d75548/nics/
          c34728e8-4338-4540-ac9b-86b8582e602e"/>
    </statistic>
    ...
</statistics>
```

注意

这个 **statistics** 子集合是只读的。

14.7.1.5. 操作

14.7.1.5.1. 为主机附加网络接口的操作

把主机网络接口附加到一个网络，这意味着这个网络可以通过被附加的网络接口卡进行访问。使用 **id** 或 **name** 项指定网络接口卡要被附加到的网络。

例 14.13. 把主机网络接口卡附加到网络的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/nics/e8f02fdf-3d7b-
4135-86e1-1bf185570cd8/attach HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <network id="e657d631-657d-42bb-a536-73501a085d85"/>
</action>
```

重要

对网络配置的改变必须被明确提交。

14.7.1.5.2. 从主机上删除网络接口卡的操作

从主机上删除网络接口卡。使用 **id** 或 **name** 项来指定网络接口卡要从哪个网络上删除。

例 14.14. 从网络上删除网络接口卡的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/nics/e8f02fdf-3d7b-
4135-86e1-1bf185570cd8/detach HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <network id="e657d631-657d-42bb-a536-73501a085d85"/>
</action>
```

重要

对网络配置的改变必须被明确提交。

14.7.1.5.3. 多网络设置操作

主机的 **nics** 集合中包括了一个设置多网络接口卡的操作。在 **setupnetworks** 操作中执行一个 **POST** 请求。

例 14.15. 设置多主机网络接口的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/nics/setupnetworks
HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <host_nics>
        <host_nic id="41561e1c-c653-4b45-b9c9-126630e8e3b9">
            <name>em1</name>
            <network id="00000000-0000-0000-0000-000000000009"/>
            <boot_protocol>dhcp</boot_protocol>
        </host_nic<
        <host_nic id="3c3f442f-948b-4cdc-9a48-89bb0593cfbd">
            <name>em2</name>
            <network id="00000000-0000-0000-0000-000000000010"/>
            <ip address="10.35.1.247" netmask="255.255.254.0"
              gateway="10.35.1.254"/>
            <boot_protocol>static</boot_protocol>
        </host_nic>
        <checkConnectivity>true</checkConnectivity>
        <connectivityTimeout>60</connectivityTimeout>
        <force>false</false>
    </host_nics>
</action>
```

这个操作会更新所有指定的、带有标准 NIC 项的网络接口资源。这个请求包括了下表中介绍的额外项。

表 14.8. 多主机网络接口设置的额外项

项	类型	描述
checkConnectivity	布尔值	设置为 **true** 来对主机和 Red Hat Enterprise Virtualization Manager 间的连接进行验证。如果连接断开，Red Hat Enterprise Virtualization Manager 会恢复它的设置。
connectivityTimeout	整数	连接的断开超时时间。
force	布尔值	设置为 **true** 来强制应用改变（即使连接已被断开）。

14.7.1.5.4. 为网络接口卡附加标签的操作

您可以为主机网络接口卡附加一个标签，使用这个标签可以把逻辑网络和有相同标签的物理主机网络接口进行关联。

例 14.16. 为网络接口卡附加标签的操作

```
POST /api/hosts/00000000-0000-0000-0000-000000000000/nics/00000000-0000-
0000-0000-000000000000/labels HTTP/1.1
Accept: application/xml
Content-type: application/xml

<label id="Label_001" />
```

14.7.1.5.5. 为网络接口卡删除标签

从物理主机网络接口中删除一个标签需要一个 **DELETE** 请求。

例 14.17. 从网络接口卡中删除标签

```
DELETE /api/hosts/00000000-0000-0000-0000-000000000000/nics/00000000-
0000-0000-0000-000000000000/labels/[label_id] HTTP/1.1

HTTP/1.1 204 No Content
```

14.7.2. 存储子集合

14.7.2.1. 存储子集合

storage 子集合提供了一组主机可用的 iSCSI 和 FCP 存储表述列表。这个存储被用来创建存储域。

子集合中的每个 **storage** 表述都代表了一个 SCSI LUN。

例 14.18. 主机上的存储子集合的 XML 表述

```
<host_storage>
```

```
    <storage id="82fb123b-321e-40a1-9889-95dcd2654463"
      href="/api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/storage/
      82fb123b-321e-40a1-9889-95dcd2654463">
        <name>LUN0</name>
        <type>iscsi</type>
        <logical_unit id="LUN0">
            <address>mysan.example.com</address>
            <target>iqn.2009-08.com.example:mysan.foobar</target>
        </logical_unit>
    </storage>
</host_storage>
```

注意

host_storage 集合是只读的。

重要

本节中所使用的 API 还处于试用阶段，今后它们可能会有变化。产品的向下兼容声明不适用于这些 API。

14.7.3. 主机 NUMA 节点子集合

14.7.3.1. NUMA 节点子集合

numanodes 子集合代表了主机的 NUMA 拓扑。子集合中的每个 **host_numa_node** 项都代表了一个 NUMA 节点。

例 14.19. 主机上的 numanodes 子集合的 XML 表述

```
<host_numa_nodes>
    <host_numa_node href="/api/hosts/f6735fa9-4ee5-47ce-b750-
a87863736cc2/numanodes/91d8537c-699e-460b-9a70-285f651e7d68"
id="91d8537c-699e-460b-9a70-285f651e7d68">
        <link href="/api/hosts/f6735fa9-4ee5-47ce-b750-
a87863736cc2/numanodes/91d8537c-699e-460b-9a70-285f651e7d68/statistics"
rel="statistics"/>
        <host href="/api/hosts/f6735fa9-4ee5-47ce-b750-a87863736cc2"
id="f6735fa9-4ee5-47ce-b750-a87863736cc2"/>
        <index>0</index>
        <memory>8157</memory>
        <cpu>
            <cores>
                <core index="0"/>
                <core index="2"/>
                <core index="4"/>
                <core index="6"/>
            </cores>
        </cpu>
        <node_distance>10 16</node_distance>
    </host_numa_node>
    <host_numa_node href="/api/hosts/f6735fa9-4ee5-47ce-b750-
```

```
a87863736cc2/numanodes/4b18926e-6faf-43f5-9fc2-0503f1531562"
id="4b18926e-6faf-43f5-9fc2-0503f1531562">
        <link href="/api/hosts/f6735fa9-4ee5-47ce-b750-
a87863736cc2/numanodes/4b18926e-6faf-43f5-9fc2-0503f1531562/statistics"
rel="statistics"/>
        <host href="/api/hosts/f6735fa9-4ee5-47ce-b750-a87863736cc2"
id="f6735fa9-4ee5-47ce-b750-a87863736cc2"/>
        <index>2</index>
        <memory>8175</memory>
        <cpu>
            <cores>
                <core index="1"/>
                <core index="3"/>
                <core index="5"/>
                <core index="7"/>
            </cores>
        </cpu>
        <node_distance>16 10</node_distance>
    </host_numa_node>
</host_numa_nodes>
```

注意

host_numa_nodes 子集合是只读的。

14.7.3.2. NUMA 节点统计数据

每个主机 NUMA 节点都包括了一个 NUMA 统计数据的 **statistics** 子集合。**statistic** 包括以下项：

表 14.9. 主机的 NUMA 节点统计数据的项

项	类型	描述
name	字符串	统计数据记录的唯一标识。
description	字符串	统计数据的描述信息。
unit	字符串	统计数据值所使用的单位。
type	**GAUGE** 或 **COUNTER**	统计数据类型。
values type=	**INTEGER** 或 **DECIMAL**	统计数据值的数据类型。
value	复杂数据	包括 **datum** 的数据集。
datum	请参阅 **values type**	**value** 中的一条数据。

项	类型	描述
host_numa_node id=	关系	与包括 **numanode** 资源的关系。

下表列出了主机 NUMA 节点的统计数据类型。

表 14.10. 主机 NUMA 节点统计数据

名称	描述
memory.total	NUMA 节点的总内存数（以字节为单位）。
memory.used	NUMA 节点中使用的内存数（以字节为单位）。
memory.free	NUMA 节点中空闲的内存数（以字节为单位）。
cpu.current.user	用户的 CPU 使用百分比。
cpu.current.system	系统的 CPU 使用百分比。
cpu.current.idle	空闲 CPU 使用百分比。

例 14.20. 主机 NUMA 统计数据子集合的 XML 表述

```
<statistics>
    <statistic href="/api/hosts/f6745fa9-4ee5-47ce-b750-
a87863736cc2/numanodes/91d8537c-689e-460b-9a70-
285f651e7d68/statistics/7816602b-c05c-3dc7-a4da-3769f7ad8896"
id="7816602b-c05c-3dc7-a4da-3769f7ad8896">
        <name>memory.total</name>
        <description>Total memory</description>
        <values type="INTEGER">
            <value>
                <datum>8157</datum>
            </value>
        </values>
        <type>GAUGE</type>
        <unit>BYTES</unit>
        <host_numa_node href="/api/hosts/f6745fa9-4ee5-47ce-b750-
a87863736cc2/numanodes/91d8537c-689e-460b-9a70-285f651e7d68"
id="91d8537c-689e-460b-9a70-285f651e7d68"/>
    </statistic>
    ...
</statistics>
```

注意

主机 NUMA 的 **statistics** 子集合是只读的。

14.7.4. 主机统计数据子集合

14.7.4.1. 主机统计数据子集合

每个主机资源都包括了一个主机统计数据的 **statistics** 子集合。**statistic** 包括以下项：

表 14.11. 主机统计数据项

项	类型	描述
name	字符串	统计数据记录的唯一标识。
description	字符串	统计数据的描述。
unit	字符串	统计数据值所使用的单位。
type	**GAUGE** 或 **COUNTER**	统计数据类型。
values type=	**INTEGER** 或 **DECIMAL**	统计数据值的数据类型。
value	复杂数据	包括 **datum** 的数据集。
datum	请参阅 **values type**	**value** 中的一条数据。
host id=	关系	与包括 **host** 资源的关系。

下表列出了主机统计数据类型。

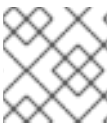
表 14.12. 主机统计数据类型

名称	描述
memory.total	主机的总内存数（以字节为单位）。
memory.used	主机使用的内存数量（以字节为单位）。
memory.free	主机上空闲的内存数量（以字节为单位）。
memory.shared	主机共享的内存数量（以字节为单位）。
memory.buffers	I/O 缓冲区大小（以字节为单位）。

名称	描述
memory.cached	OS 缓存（以字节为单位）。
swap.total	主机的交换内存数量（以字节为单位）。
swap.free	主机上空闲的交换内存数量（以字节为单位）。
swap.used	主机使用的交换内存数量（以字节为单位）。
swap.cached	在主机内存中缓存的交换内存数量（以字节为单位）。
ksm.cpu.current	Kernel SamePage Merging 的 CPU 使用百分比。
cpu.current.user	用户的 CPU 使用百分比。
cpu.current.system	系统的 CPU 使用百分比。
cpu.current.idle	空闲 CPU 使用百分比。
cpu.load.avg.5m	CPU 负载 5 分钟平均值。

例 14.21. 主机统计数据子集合的 XML 表述

```
<statistics>
    <statistic id="4ae97794-f56d-3f05-a9e7-8798887cd1ac"
      href="/api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/
      statistics/4ae97794-f56d-3f05-a9e7-8798887cd1ac">
        <name>memory.total</name>
        <description>Total memory</description>
        <unit>BYTES</unit>
        <type>GUAGE</type>
        <values type="INTEGER">
            <value>
                <datum>3983540224<datum>
            </value>
        </values>
        <host id="2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"
          href="/api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"/>
    </statistic>
    ...
</statistics>
```



注意

主机的 **statistics** 子集合是只读的。

14.8. 操作

14.8.1. 安装 VDSM 操作

在主机上安装 VDSM 和相关的软件。主机的类型决定了操作所需的额外参数。

- **Red Hat Enterprise Linux 主机** - 这个主机类型需要一个 **root_password** 项来指定主机 **root** 用户的密码。
- **Red Hat Enterprise Virtualization Hypervisor 主机** - 这个主机类型需要一个 **image** 项来指定保存在 Red Hat Enterprise Virtualization Manager 服务器上的 ISO 文件。

例 14.22. 在 Red Hat Enterprise Linux 主机上安装 VDSM 操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/install HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <root_password>p@55w0Rd!</root_password>
</action>
```

例 14.23. 在 Red Hat Enterprise Virtualization Hypervisor 主机上安装 VDSM 的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/install HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <image>/usr/share/rhev-hypervisor/rhev-hypervisor.iso</image>
</action>
```

14.8.2. 激活主机操作

激活主机

例 14.24. 激活主机的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/activate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

14.8.3. 隔离主机操作

API 用户可以使用 **fence** 操作来控制主机的电源管理设备。**capabilities** 列出了可用的 **fence_type** 选项。

-

例 14.25. 隔离主机的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/fence
Accept: application/xml
Content-Type: application/xml

<action>
    <fence_type>start</fence_type>
</action>
```

14.8.4. 取消激活主机操作

取消激活主机来进行维护操作。

例 14.26. 取消激活主机的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/deactivate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

14.8.5. 批准主机操作

在虚拟环境中使用预安装的 Red Hat Enterprise Virtualization Hypervisor 主机前，需要批准它。这个批准操作可以使用一个可选的 **cluster** 项来指定这个主机的目标集群。

例 14.27. 批准一个主机的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/approve HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <cluster id="99408929-82cf-4dc7-a532-9d998063fa95"/>
</action>
```

14.8.6. 主机 iSCSI 登录操作

iscsilogin 操作将允许主机登录到一个 iSCSI 目标上。这可以使目标中所包括的 LUN 在 **host_storage** 集合中有效。

例 14.28. 允许主机登录到 iSCSI 目标的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/iscsilogin HTTP/1.1
Accept: application/xml
Content-Type: application/xml
```

```
<action>
    <iscsi>
        <address>mysan.example.com</address>
        <target>iqn.2009-08.com.example:mysan.foobar</target>
        <username>jimmy</username>
        <password>s3kr37</password>
    </iscsi>
</action>
```

14.8.7. 主机 iSCSI 的 Discover 操作

iscsidiscover 操作允许对 iSCSI 门户进行目录列表查询。

例 14.29. 对 iSCSI 门户进行目录列表查询操作

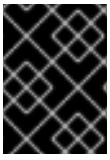
```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/iscsidiscover
HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<action>
    <iscsi>
        <address>mysan.example.com</address>
        <port>3260</port>
    </iscsi>
</action>
```

14.8.8. 提交主机网络配置操作

API 用户提交网络配置的改变。

例 14.30. 提交网络配置

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/commitnetconfig
HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```



重要

网络配置提交只会在 Manager 和主机建立了连接后，而且连接没有被断开时才可以进行。在连接断开时，主机会重启，并会自动恢复以前的网络配置。

14.8.9. 设置 SPM

把主机手动设置为 Storage Pool Manager（SPM）。

例 14.31. 把主机设置为 SPM 的操作

```
POST /api/hosts/2ab5e1da-b726-4274-bbf7-0a42b16a0fc3/forceselectspm
HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

第 15 章 虚拟机

15.1. 虚拟机项

vms 集合提供了 Red Hat Enterprise Virtualization 环境中的虚拟机信息。API 用户可以通过从进入点 URI 获得的 **rel="vms"** 链接来访问这些信息。

使用带有 **All-Content: true** 头的 **GET** 请求可以获得额外信息。

以下表格包括了虚拟机表述中的特定项的信息。

表 15.1. 虚拟机项

项	类型	描述	属性
link rel="applications"	关系	到虚拟机资源的 **applications** 子集合的链接，它显示了虚拟机上安装的应用程序。	
link rel="disks"	关系	虚拟机资源的 **disks** 子集合。	
link rel="nics"	关系	虚拟机资源的 **nics** 子集合。	
link rel="numanodes"	关系	到虚拟机资源的 **numanodes** 子集合的链接。	
link rel="cdroms"	关系	虚拟机资源的 **cdroms** 子集合。	
link rel="snapshots"	关系	虚拟机资源的 **snapshots** 子集合。	
link rel="tags"	关系	虚拟机资源的 **tags** 子集合。	
link rel="permissions"	关系	虚拟机资源的 **permissions** 子集合。	
link rel="statistics"	关系	虚拟机资源的 **statistics** 子集合。	
link rel="reporteddevices"	关系	到虚拟机资源的 **reporteddevices** 子集合的链接。	

项	类型	描述	属性
link rel="watchdogs"	关系	虚拟机资源的 **watchdogs** 子集合。	
link rel="sessions"	关系	虚拟机资源的 **sessions** 子集合。	
type	列举	虚拟机类型。**capabilities** 包括了有效值列表。	
status	参阅以下信息	虚拟机状态。	
memory	整数	分配给虚拟机的内存大小（以字节为单位）	
cpu	复杂数据	定义虚拟机的 CPU 详情。**topology** 子项定义了虚拟机可用的逻辑 **sockets** 的数量，以及每个插槽中的 **cores** 的数量。虚拟机可用的内核总数是插槽数量乘以每个插槽的内核数。 **cputune** 子项使用一组 **vcpupin** 项来把虚拟 CPU 和物理主机 CPU 进行映射。每个 **vcpupin** 项包括一个虚拟 CPU 属性（**vcpu**）以及一个指定虚拟 CPU 所使用的物理 CPU 属性（**cpuset**）。**cpuset** 可以被设置为单一 CPU（**cpuset="0"**）、多个 CPU（**cpuset="0,2"**）、一个 CPU 范围（**cpuset="0-3"**）或排除特定 CPU 的 CPU 范围（**cpuset="0-3,^2"**）。 **cpu_mode** 子项定义了虚拟 CPU 和相关主机 CPU 的紧密关系程度。它包括 3 个值：**custom**（默认值，没有指定任何模式）、**host_model**（复制 libvirt 支持的主机 CPU ）和 **host_passthrough**（把主机的所有功能都复制到虚拟机，包括 libvirt 不支持的功能。但是，**host_passthrough** 会使虚拟机无法被迁移）。	
os type=	字符串，如 **RHEL5** 或 **WindowsXP**	虚拟机操作系统类型。	
os boot dev=	列举	**boot** 项的 **dev** 属性所描述的引导设备列表。**capabilities** 包括了它的有效值。	
os kernel	字符串	到虚拟机引导内核镜像的路径。这个选项支持直接引导 Linux 内核，而不需要 BIOS bootloader。	

项	类型	描述	属性
os initrd	字符串	到一个指定内核所使用的 initrd 镜像的路径。这个选项支持直接引导 Linux 内核，而不需要 BIOS bootloader。	
os cmdline	字符串	内核需要使用的内核命令行参数。这个选项支持直接引导一个 Linux 内核，而不需要使用 BIOS bootloader。	
high_availability	复杂数据	如果需要在虚拟机或它所在的主机出现系统故障时自动重启，请把 **enabled** 设置为 **true**。一个 **priority** 项被用来控制虚拟机重启的顺序。	
display	复杂数据	显示的**类型**（**vnc** 或 **spice**）、端口以及**显示器**的数量。**allow_reconnect** 的值指定了客户端是否可以通过显示重新连接到机器上。 **smartcard_enabled** 子项是一个布尔值（**true** 或 **false**），它被用来指定附加到客户端系统上的 Smartcard 是否可以被传递到虚拟机上。这个子项的默认值是 **false**。	
cluster id=	GUID	虚拟机所在主机的集群标识。	
template id=	GUID	虚拟机所基于的模板标识。	
domain id=	GUID	虚拟机域的标识。	
start_time	**xsd:dateTime** 格式：**YYYY-MM-DDThh:mm:ss**	虚拟机被启动的时间。	
stop_time	**xsd:dateTime** 格式：**YYYY-MM-DDThh:mm:ss**	虚拟机被停止的时间。	
creation_time	**xsd:dateTime** 格式：**YYYY-MM-DDThh:mm:ss**	虚拟机被创建的时间。	
origin	**rhev**、**ovirt**、**vmware** 或 **xen**	虚拟机的原始系统。	

项	类型	描述	属性
stateless	布尔值：true 或 false	如果虚拟机是无状态的（stateless），把这个参数设为 **true**。无状态的虚拟机会在系统启动时对磁盘做一个快照，并在关机时删除对系统所做的所有改变。这意味着每次对系统所做的改变会在系统重启后丢失。	
delete_protected	布尔值：true 或 false	如果设为 **true**，虚拟机将无法被删除。	
sso	字符串	到虚拟机的单点登录方法的一个指代，它包括一个带有 **ip** 属性的 **method** 项。	
placement_policy	复杂数据	设置虚拟机迁移的替换策略。需要一个默认的 **host=** 和一个 **affinity**（**migratable**、**user_migratable** 或 **pinned**）。如果没有设置 **host** 项的值，则意味着没有首选的主机。使用多个 **host** 项可以在集群中指定一组首选的主机。	
memory_policy	复杂数据	为虚拟机设置内存策略。定义为了运行这个虚拟机所需要的最小 **guaranteed** 主机内存。	
quota id=	GUID	为虚拟机设置配额。	
custom_properties	复杂数据	传递给脚本的一组用户定义的环境变量。每个 **custom_property** 都包括 **name** 和 **value** 属性。**capabilities** 包括了有效值的列表。	
usb	复杂数据	定义虚拟机的 USB 策略。需要一个 **enabled** 项（布尔值）和一个 **type** 项（值是 **native** 或 **legacy**）。	
migration_downtime	整数	代表在实时迁移时，虚拟机可以下线的最长时间（以毫秒为单位）。如果它的值为 0，则代表使用 VDSM 的默认设置。	
cpu_profile id=	GUID	到虚拟机的 CPU 配置集的参考。	
next_run_configuration	布尔值：true 或 false	如果对虚拟机的配置改变需要在虚拟机下一次重启时生效，这个值应为 **true**。	
numa_tune_mode	字符串	到主机 NUMA 节点的内存分配模式（**interleave**、**strict** 或 **preferred**）的参考。	
guest_info	复杂数据	虚拟机客户端的信息。包括一个带有 **address=** 属性的 **ip** 项。	

项	类型	描述	属性
vmpool	复杂数据	虚拟机池的标识。这个项只会出现在属于某个虚拟机池的虚拟机上。	
timezone	时区数据库格式：**Area/Location**	Windows 虚拟机在 Sysprep 中的时区设置。	
domain	复杂数据	Windows 虚拟机模板在 Sysprep 中的时区设置。需要 **domains** 集合中的一个 **name**。	
initialization	复杂数据	定义一组 Linux 虚拟机使用 Cloud-Init 引导时，或 Windows 虚拟机使用 Sysprep 时使用的值。 **Cloud-Init** • **host_name**：虚拟机的主机名。 • **timezone**：虚拟机使用的时区。 • **user_name**：虚拟机的用户名。 • **root_password**：指定用户的密码。如果没有指定用户，则代表 root 密码。 • **authorized_ssh_keys**：指定 SSH 密钥并把它加入到虚拟机的验证密钥文件中。您可以通过在一个新行中输入 SSH 密钥来指定多个 SSH 密钥。 • **regenerate_ssh_keys**：是否为虚拟机重新产生 SSH 密钥。它的值是 **true** 或 **false**。 • **dns_servers**：由空格分隔的 DNS 服务器列表。 • **dns_search**：由空格分隔的 DNS 搜索域列表。 • **nic_configurations**：定义虚拟机的网络接口控制器。网络接口控制器在这个集合中被定义为 **nic_configuration** 对象，每个对象都指定了 **name**、**ip**、**boot_protocol** 和 **on_boot**。 • **custom_script**：虚拟机启动时运行的自定义脚本。 **Sysprep** • **host_name**：虚拟机的主机名。 • **domain**：虚拟机所属的域。 • **authorized_ssh_keys**：指定 SSH 密钥并把它加入到虚拟机的验证密钥文件中。	

项	类型	描述	您可以通过在一个新行中输入 SSH 密钥来指定多个 SSH 密钥。	属性
			• **regenerate_ssh_keys**：是否为虚拟机重新产生 SSH 密钥。它的值是 **true** 或 **false**。 • **timezone**：虚拟机使用的时区。 • **root_password**：虚拟机 admin 用户的密码。 • **custom_script**：虚拟机启动时运行的自定义脚本。 • **input_locale**：用户输入的语言环境。 • **ui_language**：用户界面项（如按钮、菜单）所使用的语言。 • **system_locale**：系统整体的语言环境。 • **user_locale**：用户的语言环境。 • **active_directory_ou**：虚拟机在所属 Active Directory 域中的 OU（organizational unit）。 • **org_name**：虚拟机所属的机构（organization）。	
payloads	复杂数据	定义一组在引导时传递给虚拟机的 **payload** 项。每个 **payload** 需要一个 **type** 属性（**cdrom** 或 **floppy**）和一组 **file** 属性。每个 **file** 项中包括一个 **name** 项（定义了文件的名称和位置），以及一个 **content** 项（定义了传递给文件的内容）。 **payloads** 项被 **cloud-init** 使用。当使用 **cloud-init** 配置虚拟机时，一个 payload 会被自动创建。它的 **type** 属性被设置为 **cd-rom**，并有两个 **file** 子项（**openstack/latest/meta_data.json** 和 **openstack/latest/user_data**），它们把配置参数传递给虚拟机。		

status 包括以下值之一：**unassigned**、**down**、**up**、**powering_up**、**powered_down**、**paused**、**migrating_from**、**migrating_to**、**unknown**、**not_responding**、**wait_for_launch**、**reboot_in_progress**、**saving_state**、**restoring_state**、**suspended**、**image_illegal**、**image_locked** 或 **powering_down**。这些状态在 **capabilities** 下的 **vm_states** 中列出。

15.2. 虚拟机的 XML 表述

例 15.1. 虚拟机的 XML 表述

```
<vm id="70b4d9a7-4f73-4def-89ca-24fc5f60e01a"
  href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a">
    <actions>
```

```
        <link rel="move"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/move"/>
        <link rel="ticket"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/ticket"/>
        <link rel="reboot"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/reboot"/>
        <link rel="undo_snapshot"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/undo_snapshot"/>
        <link rel="commit_snapshot"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-
24fc5f60e01a/commit_snapshot"/>
        <link rel="preview_snapshot"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-
24fc5f60e01a/preview_snapshot"/>
        <link rel="logon"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/logon"/>
        <link rel="cancelmigration"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-
24fc5f60e01a/cancelmigration"/>
        <link rel="maintenance"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/maintenance"/>
        <link rel="clone"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/clone"/>
        <link rel="migrate"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/migrate"/>
        <link rel="detach"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/detach"/>
        <link rel="export"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/export"/>
        <link rel="shutdown"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/shutdown"/>
        <link rel="start"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/start"/>
        <link rel="stop"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/stop"/>
        <link rel="suspend"
   href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/suspend"/>
    </actions>
    <name>VM_01</name>
    <description>Testing Virtual Machine</description>
    <link rel="applications"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-
24fc5f60e01a/applications"/>
    <link rel="disks"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/disks"/>
    <link rel="nics"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/nics"/>
    <link rel="numanodes"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/numanodes"/>
    <link rel="cdroms"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/cdroms"/>
    <link rel="snapshots"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/snapshots"/>
    <link rel="tags"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/tags"/>
    <link rel="permissions"
```

```
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/permissions"/>
    <link rel="statistics"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/statistics"/>
    <link rel="reporteddevices"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-
24fc5f60e01a/reporteddevices"/>
    <link rel="watchdogs"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/watchdogs"/>
    <link rel="sessions"
      href="/api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a/sessions"/>
    <type>server</type>
    <status>
        <state>down</state>
    </status>
    <memory>1073741824</memory>
    <cpu>
        <topology sockets="1" cores="1"/>
        <architecture>X86_64</architecture>
    </cpu>
    <cpu_shares>0</cpu_shares>
    <bios>
        <boot_menu>
            <enabled>false</enabled>
        </boot_menu>
    </bios>
    <os type="other">
        <boot dev="hd"/>
    </os>
    <high_availability>
        <enabled>false</enabled>
        <priority>1</priority>
    </high_availability>
    <display>
        <type>spice</type>
        <monitors>1</monitors>
        <single_qxl_pci>false</single_qxl_pci>
        <allow_override>true</allow_override>
        <smartcard_enabled>false</smartcard_enabled>
        <file_transfer_enabled>true</file_transfer_enabled>
        <copy_paste_enabled>true</copy_paste_enabled>
    </display>
    <cluster href="/api/clusters/00000001-0001-0001-0001-0000000002fb"
id="00000001-0001-0001-0001-0000000002fb"/>
    <template href="/api/templates/00000000-0000-0000-0000-000000000000"
id="00000000-0000-0000-0000-000000000000"/>
    <stop_time>2014-12-03T14:25:45.588+10:00</stop_time>
    <creation_time>2014-12-03T14:25:45.535+10:00</creation_time>
    <origin>ovirt</origin>
    <stateless>false</stateless>
    <delete_protected>false</delete_protected>
    <sso>
        <methods>
            <method id="GUEST_AGENT"/>
        </methods>
    </sso>
    <timezone>Etc/GMT</timezone>
```

```
    <placement_policy>
        <affinity>migratable</affinity>
    </placement_policy>
    <memory_policy>
        <guaranteed>1073741824</guaranteed>
    </memory_policy>
    <usb>
        <enabled>false</enabled>
    </usb>
    <migration_downtime>-1</migration_downtime>
    <cpu_profile href="/api/cpuprofiles/0000001a-001a-001a-001a-
0000000002e3" id="0000001a-001a-001a-001a-0000000002e3"/>
    <next_run_configuration_exists>false</next_run_configuration_exists>
    <numa_tune_mode>interleave</numa_tune_mode>
</vm>
```

15.3. 虚拟机的 OVF 数据的 XML 表述

使用带有 **All-Content: true** 头的、对虚拟机的 **GET** 请求可以获得包括额外的、代表快照的 OVF 数据。

如果 **Accept** 头为空，它会被默认设置为 **application/xml**，数据由 HTML 项来代表，因此不会影响到 XML tag。如果指定 **Accept: application/json** 头，返回的数据会使用标准的 XML tag。这个示例使用标准块格式构成。

例 15.2. 虚拟机的 OVF 数据的 XML 表述

```
GET /api/vms/70b4d9a7-4f73-4def-89ca-24fc5f60e01a HTTP/1.1
All-Content: true

<?xml version='1.0' encoding='UTF-8'?>
<ovf:Envelope xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1/"
  xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
  xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_VirtualSystemSettingData"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  ovf:version="3.5.0.0">
  <References/>
  <Section xsi:type="ovf:NetworkSection_Type">
    <Info>List of networks</Info>
    <Network ovf:name="Network 1"/>
  </Section>
  <Section xsi:type="ovf:DiskSection_Type">
    <Info>List of Virtual Disks</Info>
  </Section>
  <Content ovf:id="out" xsi:type="ovf:VirtualSystem_Type">
    <CreationDate>2014/12/03 04:25:45</CreationDate>
    <ExportDate>2015/02/09 14:12:24</ExportDate>
    <DeleteProtected>false</DeleteProtected>
    <SsoMethod>guest_agent</SsoMethod>
    <IsSmartcardEnabled>false</IsSmartcardEnabled>
    <TimeZone>Etc/GMT</TimeZone>
    <default_boot_sequence>0</default_boot_sequence>
```

```
    <Generation>1</Generation>
    <VmType>1</VmType>
    <MinAllocatedMem>1024</MinAllocatedMem>
    <IsStateless>false</IsStateless>
    <IsRunAndPause>false</IsRunAndPause>
    <AutoStartup>false</AutoStartup>
    <Priority>1</Priority>
    <CreatedByUserId>fdfc627c-d875-11e0-90f0-
83df133b58cc</CreatedByUserId>
    <IsBootMenuEnabled>false</IsBootMenuEnabled>
    <IsSpiceFileTransferEnabled>true</IsSpiceFileTransferEnabled>
    <IsSpiceCopyPasteEnabled>true</IsSpiceCopyPasteEnabled>
    <Name>VM_export</Name>
    <TemplateId>00000000-0000-0000-0000-000000000000</TemplateId>
    <TemplateName>Blank</TemplateName>
    <IsInitilized>false</IsInitilized>
    <Origin>3</Origin>
    <DefaultDisplayType>1</DefaultDisplayType>
    <TrustedService>false</TrustedService>
    <OriginalTemplateId>00000000-0000-0000-0000-
000000000000</OriginalTemplateId>
    <OriginalTemplateName>Blank</OriginalTemplateName>
    <UseLatestVersion>false</UseLatestVersion>
    <Section ovf:id="70b4d9a7-4f73-4def-89ca-24fc5f60e01a"
      ovf:required="false"
      xsi:type="ovf:OperatingSystemSection_Type">
      <Info>Guest Operating System</Info>
      <Description>other</Description>
    </Section>
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Info>1 CPU, 1024 Memeory</Info>
      <System>
        <vssd:VirtualSystemType>ENGINE 3.5.0.0</vssd:VirtualSystemType>
      </System>
      <Item>
        <rasd:Caption>1 virtual cpu</rasd:Caption>
        <rasd:Description>Number of virtual CPU</rasd:Description>
        <rasd:InstanceId>1</rasd:InstanceId>
        <rasd:ResourceType>3</rasd:ResourceType>
        <rasd:num_of_sockets>1</rasd:num_of_sockets>
        <rasd:cpu_per_socket>1</rasd:cpu_per_socket>
      </Item>
      <Item>
        <rasd:Caption>1024 MB of memory</rasd:Caption>
        <rasd:Description>Memory Size</rasd:Description>
        <rasd:InstanceId>2</rasd:InstanceId>
        <rasd:ResourceType>4</rasd:ResourceType>
        <rasd:AllocationUnits>MegaBytes</rasd:AllocationUnits>
        <rasd:VirtualQuantity>1024</rasd:VirtualQuantity>
      </Item>
      <Item>
        <rasd:Caption>USB Controller</rasd:Caption>
        <rasd:InstanceId>3</rasd:InstanceId>
        <rasd:ResourceType>23</rasd:ResourceType>
        <rasd:UsbPolicy>DISABLED</rasd:UsbPolicy>
      </Item>
```

```
    </Section>
  </Content>
</ovf:Envelope>
```

15.4. 虚拟机的 JSON 表述

例 15.3. 虚拟机的 JSON 表述

```
{
  "type" : "server",
  "status" : {
    "state" : "down"
  },
  "stop_reason" : "",
  "memory" : 1073741824,
  "cpu" : {
    "topology" : {
      "sockets" : "1",
      "cores" : "1"
    },
    "architecture" : "X86_64"
  },
  "cpu_shares" : "0",
  "bios" : {
    "boot_menu" : {
      "enabled" : "false"
    }
  },
  "os" : {
    "boot" : [ {
      "dev" : "hd"
    } ],
    "type" : "other"
  },
  "high_availability" : {
    "enabled" : "false",
    "priority" : "1"
  },
  "display" : {
    "type" : "spice",
    "monitors" : "1",
    "single_qxl_pci" : "false",
    "allow_override" : "false",
    "smartcard_enabled" : "false",
    "file_transfer_enabled" : "true",
    "copy_paste_enabled" : "true"
  },
  "cluster" : {
    "href" : "/api/clusters/00000001-0001-0001-0001-0000000002fb",
    "id" : "00000001-0001-0001-0001-0000000002fb"
  },
  "template" : {
    "href" : "/api/templates/00000000-0000-0000-0000-000000000000",
    "id" : "00000000-0000-0000-0000-000000000000"
```

```
  },
  "stop_time" : 1423550982110,
  "creation_time" : 1423490033647,
  "origin" : "ovirt",
  "stateless" : "false",
  "delete_protected" : "false",
  "sso" : {
    "methods" : {
      "method" : [ {
        "id" : "GUEST_AGENT"
      } ]
    }
  },
  "timezone" : "Etc/GMT",
  "initialization" : {
    "regenerate_ssh_keys" : "false",
    "nic_configurations" : { }
  },
  "placement_policy" : {
    "affinity" : "migratable"
  },
  "memory_policy" : {
    "guaranteed" : 1073741824,
    "ballooning" : "true"
  },
  "usb" : {
    "enabled" : "false"
  },
  "migration_downtime" : "-1",
  "cpu_profile" : {
    "href" : "/api/cpuprofiles/0000001a-001a-001a-001a-0000000002e3",
    "id" : "0000001a-001a-001a-001a-0000000002e3"
  },
  "next_run_configuration_exists" : "false",
  "numa_tune_mode" : "interleave",
  "actions" : {
    "link" : [ {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/ticket",
      "rel" : "ticket"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/move",
      "rel" : "move"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/clone",
      "rel" : "clone"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-
973a44b2562e/commit_snapshot",
      "rel" : "commit_snapshot"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-
973a44b2562e/preview_snapshot",
      "rel" : "preview_snapshot"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/logon",
      "rel" : "logon"
```

```
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-
973a44b2562e/cancelmigration",
      "rel" : "cancelmigration"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-
973a44b2562e/maintenance",
      "rel" : "maintenance"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/reboot",
      "rel" : "reboot"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-
973a44b2562e/undo_snapshot",
      "rel" : "undo_snapshot"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/migrate",
      "rel" : "migrate"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/detach",
      "rel" : "detach"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/export",
      "rel" : "export"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/shutdown",
      "rel" : "shutdown"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/start",
      "rel" : "start"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/stop",
      "rel" : "stop"
    }, {
      "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/suspend",
      "rel" : "suspend"
    } ]
  },
  "name" : "VM_01",
  "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e",
  "id" : "42ec2621-7ad6-4ca2-bd68-973a44b2562e",
  "link" : [ {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-
973a44b2562e/applications",
    "rel" : "applications"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/disks",
    "rel" : "disks"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/nics",
    "rel" : "nics"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/numanodes",
    "rel" : "numanodes"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/cdroms",
```

```
    "rel" : "cdroms"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/snapshots",
    "rel" : "snapshots"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/tags",
    "rel" : "tags"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-
973a44b2562e/permissions",
    "rel" : "permissions"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/statistics",
    "rel" : "statistics"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-
973a44b2562e/reporteddevices",
    "rel" : "reporteddevices"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/watchdogs",
    "rel" : "watchdogs"
  }, {
    "href" : "/api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/sessions",
    "rel" : "sessions"
  } ]
}
```

15.5. 方法

15.5.1. 创建虚拟机

创建一个新虚拟机需要 **name**、**template** 和 **cluster** 项。使用 **id** 属性或 **name** 项指定 **template** 和 **cluster** 项。使用 **cpuprofiles** 属性指定 CPU 配置集的 ID。

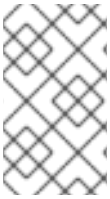
例 15.4. 创建一个带有 512 MB 内存的、从 CD-ROM 引导的虚拟机。

```
POST /api/vms HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm>
    <name>vm2</name>
    <description>Virtual Machine 2</description>
    <type>desktop</type>
    <memory>536870912</memory>
    <cluster>
        <name>default</name>
    </cluster>
    <template>
        <name>Blank</name>
    </template>
    <os>
        <boot dev="cdrom"/>
```

```
    </os>
    <cdroms>
        <cdrom>
            <file id="example_windows_7_x64_dvd_u_677543.iso"/>
        </cdrom>
    </cdroms>
    <cpu_profile id="0000001a-001a-001a-001a-00000000035e"/>
</vm>
```

例 15.5. 创建一个带有 512 MB 内存的、从虚拟硬盘引导的虚拟机。

```
POST /api/vms HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm>
    <name>vm2</name>
    <description>Virtual Machine 2</description>
    <type>desktop</type>
    <memory>536870912</memory>
    <cluster>
        <name>default</name>
    </cluster>
    <template>
        <name>Blank</name>
    </template>
    <os>
      <boot dev="hd"/>
    </os>
    <cpu_profile id="0000001a-001a-001a-001a-00000000035e"/>
</vm>
```



注意

以上实例中的内存是通过以下公式计算的：

512MB * 1024 ^{2} = 536870912 字节

15.5.2. 更新虚拟机

在创建
后，**name**、**description**、**cluster**、**type**、**memory**、**cpu**、**os**、**high_availability**、**display**、**timezone**、**domain**、**stateless**、**placement_policy**、**memory_policy**、**usb**、**payloads**、**origin** 和 **custom_properties** 项可以被更新。

例 15.6. 更新一个带有 1 GB 内存的虚拟机

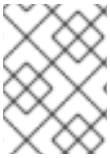
```
PUT /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399 HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

```
<vm>
    <memory>1073741824</memory>
</vm>
```

注意

以上实例中的内存是通过以下公式计算的：

1024MB * 1024 ^{2} = 1073741824 字节



注意

从 Red Hat Enterprise Virtualization 3.6 开始，内存热插功能开始被支持。您可以使用上面的示例来在虚拟机运行时增加内存的数量。

例 15.7. 把一个虚拟机固定到多个主机

一个固定到多个主机的虚拟机不能被实时迁移，但在特定的主机间具有高可用性。多主机固定设置可以用来把虚拟机限制到一组主机中（如有相同硬件配置的一组主机）。

```
PUT /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm>
    <high_availability>
        <enabled>true</enabled>
        <priority>1</priority>
    </high_availability>
    <placement_policy>
        <hosts>
            <host><name>Host1</name></host>
            <host><name>Host2</name></host>
        </hosts>
        <affinity>pinned</affinity>
    </placement_policy>
</vm>
```

15.5.3. 删除虚拟机

删除虚拟机需要一个 **DELETE** 请求。

例 15.8. 删除一个虚拟机

```
DELETE /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399 HTTP/1.1

HTTP/1.1 204 No Content
```

15.5.4. 删除虚拟机，但不删除虚拟磁盘

在删除虚拟机前断开附加的虚拟磁盘，这会保留虚拟磁盘中的数据。删除虚拟机需要一个 **DELETE** 请求。

例 15.9. 删除一个虚拟机

```
DELETE /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <vm>
        <disks>
            <detach_only>true</detach_only>
        </disks>
    </vm>
</action>
```

15.6. 子集合

15.6.1. 磁盘子集合

15.6.1.1. 磁盘子集合

disks 子集合代表了虚拟机上的所有虚拟硬盘。一个 **disk** 表述包括以下项：

表 15.2. 虚拟机磁盘的项

项	类型	描述	属性
link rel="statistics"	关系	虚拟机磁盘统计数据的 **statistics** 子集合链接。	
link rel="permissions"	关系	到 **permissions** 子集合的链接。	
alias	字符串	磁盘的唯一 ID。使用 **alias** 而不是使用 **name**。	
image_id	字符串	保存在所定义的存储域中的虚拟机镜像标识。	
storage_domains	复杂数据	与磁盘相关联的存储域。每个 **storage_domain** 项都包括一个带有存储域 GUID 的 **id** 属性。使用 **POST** 更新这个项可以执行从一个数据存储域到另一个数据存储域的磁盘实时迁移操作。	[a]

项	类型	描述	属性
size	整数	磁盘大小（以字节为单位）。已被 **provisioned_size** 所替代。	
provisioned_size	整数	配置的磁盘大小（以字节为单位）	
actual_size	整数	磁盘的实际大小（以字节为单位）	
status	**illegal**、**invalid**、**locked** 或 **ok**	磁盘设备的状态。**capabilities** 下的 **disk_states** 列出了它的值。	
interface	列举	连接到磁盘设备的接口驱动类型。**capabilities** 包括了有效的值。	
format	列举	底层的存储格式。**capabilities** 包括了有效的值。使用 Copy On Write（COW）可以创建快照，但对系统性能会有一些影响；Raw 不能创建快照，但可以提供更好的性能。	
sparse	布尔值：true 或 false	**true**：磁盘所使用的物理存储无法被预先分配。	
bootable	布尔值：true 或 false	**true**：磁盘被设为可引导。	
--shareable	布尔值：true 或 false	**true**：磁盘可以被多个虚拟机共享。	
wipe_after_delete	布尔值：true 或 false	如果为 **true**，在磁盘被删除后，这个磁盘的底层物理存储会被清零。此选项会增加数据的安全性，但同时会消耗大量的操作资源，并会增加删除过程所需的时间。	
propagate_errors	布尔值：true 或 false	**true**：当磁盘出现错误时，不会导致虚拟机暂停，而是把磁盘错误传递给虚拟机的 OS。	
vm id=	GUID	包括虚拟机的 ID。	
quota id=	GUID	为磁盘设置一个配额。	

项	类型	描述	属性
lun_storage	复杂数据	为存储使用的直接 LUN 的标识。需要一个包括 iSCSI 或 FCP 设备详情的 **logical_unit** 项。	
active	布尔值（Boolean）	指定磁盘是否连接到虚拟机。	
read_only	布尔值（Boolean）	定义磁盘是否为只读。	
link rel="disk_profile"	关系	到 **disk_profile** 子集合的链接。	

[a] 这个项只有在磁盘被添加到虚拟机，而不是通过虚拟机模板创建时才需要

例 15.10. 磁盘设备的 XML 表述

```
<disk id="ed7feafe-9aaf-458c-809a-ed789cdbd5b4"
  href="/api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/disks/
  ed7feafe-9aaf-458c-809a-ed789cdbd5b4">
    <link rel="statistics"
      href="/api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/disks/
      ed7feafe-9aaf-458c-809a-ed789cdbd5b4/statistics"/>
    <link rel="permissions"
      href="/api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/disks/
      ed7feafe-9aaf-458c-809a-ed789cdbd5b4/permissions"/>
    <vm id="082c794b-771f-452f-83c9-b2b5a19c0399"
      href="/api/vms/082c794b-771f-452f-83c9-b2b5a19c0399"/>
    <alias>Classic_VM</alias>
    <image_id>cac69a29-ccff-49d4-8a26-e4cdacd83e34</image_id>
    <storage_domains>
        <storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"/>
    </storage_domains>
    <size>12884901888</size>
    <provisioned_size>12884901888</provisioned_size>
    <actual_size>1073741824</actual_size>
    <type>system</type>
    <status>
        <state>ok</state>
    </status>
    <interface>virtio</interface>
    <format>raw</format>
    <bootable>true</bootable>
    <shareable>true</shareable>
    <wipe_after_disk>true</wipe_after_disk>
    <propagate_errors>false</propagate_errors>
    <active>true</active>
    <read_only>false</read_only>
    <disk_profile id="23fb2e0d-3062-4819-8165-3be88f2f587e"
      href="/api/diskprofiles/23fb2e0d-3062-4819-8165-3be88f2f587e"/>
```

```
    <lun_storage>
        <logical_unit id="lun1">
                ...
        </logical_unit>
    </lun_storage>
</disk>
```

添加一个新虚拟磁盘。添加新虚拟磁盘需要 **provisioned_size** 项。使用 **storage_domains** 项指定磁盘要在哪个存储域中创建。同一个虚拟机的多个磁盘可以存在于不同的存储域中。

例 15.11. 在虚拟机上创建一个新磁盘设备

```
POST /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/disks HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk>
    <storage_domains>
        <storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"/>
    </storage_domains>
    <provisioned_size>8589934592</provisioned_size>
    <type>system</type>
    <interface>virtio</interface>
    <format>cow</format>
    <bootable>true</bootable>
</disk>
```

为虚拟机添加一个新的外部（direct LUN）磁盘。这个调用需要 **lun_storage** 和 **logical_unit** 项来包括 iSCSI 或 FCP 设备的详细信息。

例 15.12. 在虚拟机上创建一个新的 direct LUN 磁盘设备

```
POST /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/disks HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk>
    <interface>virtio</interface>
    <lun_storage>
        <type>iscsi</type>
        <logical_unit id="lun1">
            <address>iscsi.example.com</address>
            <port>3260</port>
            <target>iqn.2010.05.com.example:iscsi.targetX</target>
        </logical_unit>
    </lun_storage>
</disk>
```

在创建完成
后，**alias**、**description**、**storage_domains**、**provisioned_size**、**interface**、**bootable**、

shareable、**wipe_after_delete** 和 **propagate_errors** 项都可以被更新。

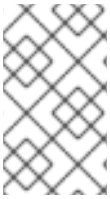
用户可以在不需要暂停、休眠、重启虚拟机的情况下调整虚拟磁盘的大小。

例 15.13. 更新虚拟机磁盘

```
PUT /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/disks/ed7feafe-9aaf-
458c-809a-ed789cdbd5b4 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk>
    <bootable>false</bootable>
    <shareable>false</shareable>
</disk>
```

例 15.14. 把虚拟机磁盘的大小改为 20GB。

```
PUT /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/disks/ed7feafe-9aaf-
458c-809a-ed789cdbd5b4 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk>
    <provisioned_size>21474836480</provisioned_size>
</disk>
```



注意

以上实例中的磁盘大小是通过以下公式计算的：

20480MB * 1024^2 = 21474836480 字节

例 15.15. 重新命名一个虚拟机磁盘

```
PUT /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/disks/ed7feafe-9aaf-
458c-809a-ed789cdbd5b4 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk>
    <alias>Classic_VM2</alias>
</disk>
```

删除虚拟机磁盘需要一个 **DELETE** 请求。

例 15.16. 删除虚拟机磁盘

```
DELETE /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/disks/ed7feafe-
9aaf-458c-809a-ed789cdbd5b4 HTTP/1.1

HTTP/1.1 204 No Content
```

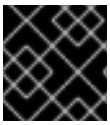
15.6.1.2. 磁盘克隆

使用带有 **clone** 项的模板克隆磁盘。在创建虚拟机时，把 **disks** 子集合的 **clone** 项设置为 **true**。这会使用基本模板克隆磁盘，并把它附加到虚拟机上。

例 15.17. 使用模板克隆一个磁盘

以下实例在创建虚拟机时使用模板克隆一个磁盘。

```
POST /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm>
    <name>cloned_vm</name>
    <template id="64d4aa08-58c6-4de2-abc4-89f19003b886"/>
    <cluster id="99408929-82cf-4dc7-a532-9d998063fa95"/>
    <disks>
        <clone>true</clone>
        <disk id="4825ffda-a997-4e96-ae27-5503f1851d1b">
            <format>COW</format>
        </disk>
        <disk id="42aef10d-3dd5-4704-aa73-56a023c1464c">
            <format>COW</format>
        </disk>
    </disks>
</vm>
```



重要

基于磁盘名称对虚拟机磁盘进行搜索需要 **alias** 搜索参数，而不是 **name**。

15.6.1.3. 磁盘统计数据子集合

每个虚拟机都包括一个 **statistics** 子集合来提供与磁盘相关的统计数据。每个 **statistic** 包括以下项：

表 15.3. 虚拟机磁盘统计数据的项

项	类型	描述
name	字符串	统计数据记录的唯一标识。
description	字符串	统计数据的描述。

项	类型	描述
unit	字符串	统计数据值所使用的单位。
type	**GAUGE** 或 **COUNTER**	统计数据类型。
values type=	**INTEGER** 或 **DECIMAL**	统计数据值的数据类型。
value	复杂数据	包括 **datum** 的数据集。
datum	请参阅 **values type**	**value** 中的一条数据。
disk id=	关系	与包括 **disk** 资源的关系。

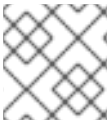
下表列出了虚拟机磁盘的统计数据类型信息。

表 15.4. 虚拟机磁盘统计数据类型

名称	描述
data.current.read	读磁盘操作的数据传输频率（字节/每秒）
data.current.write	写磁盘操作的数据传输频率（字节/每秒）

例 15.18. 虚拟机统计数据子集合的 XML 表述

```
<statistics>
    <statistic id="33b9212b-f9cb-3fd0-b364-248fb61e1272"
      href="/api/vms/3a42530e-3bc5-4094-829d-489257894c2a/disks/
      f28ec14c-fc85-43e1-818d-96b49d50e27b/statistics/
      33b9212b-f9cb-3fd0-b364-248fb61e1272">
        <name>data.current.read</name>
        <description>Read data rate</description>
        <values type="DECIMAL">
            <value>
                <datum>0</datum>
            </value>
        </values>
        <type>GAUGE</type>
        <unit>BYTES_PER_SECOND</unit>
        <disk id="f28ec14c-fc85-43e1-818d-96b49d50e27b"
          href="/api/vms/3a42530e-3bc5-4094-829d-489257894c2a/
          disks/f28ec14c-fc85-43e1-818d-96b49d50e27b"/>
    </statistic>
    ...
</statistics>
```



注意

这个 **statistics** 子集合是只读的。

15.6.1.4. 附加浮动磁盘和取消附加浮动磁盘的操作

使用 **POST** 请求在虚拟机的 **disks** 子集合中附加主 **rel="disks"** 集合中的一个磁盘。包括被附加的磁盘 **id**。

例 15.19. 附加浮动磁盘

```
POST /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/disks HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk id="d135f1c5-b5e1-4238-9381-b3277f5a3742">
</disk>
```

使用对磁盘资源的 **DELETE** 请求来从虚拟机的 **disks** 子集合中取消附加一个磁盘，但需要包括一个 **detach** 项来确保磁盘不会被破坏。

例 15.20. 从虚拟机上取消附加的磁盘

```
DELETE /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/disks/
  d135f1c5-b5e1-4238-9381-b3277f5a3742 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <detach>true</detach>
</action>
```

15.6.1.5. 激活和取消激活操作

每个虚拟机磁盘都提供了一组 **activate** 和 **deactivate** 操作来为虚拟机添加磁盘，或从虚拟机上删除磁盘。

例 15.21. 激活虚拟机磁盘操作

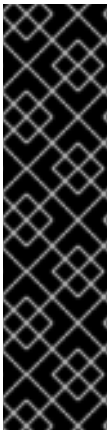
```
POST /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/disks/a42ada0e-1d69-
410d-a392-a6980d873e5d/activate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

例 15.22. 取消激活虚拟机磁盘操作

```
POST /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/disks/a42ada0e-1d69-
```

```
410d-a392-a6980d873e5d/deactivate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

使用这些操作来为虚拟机“热插”磁盘，并激活新添加的磁盘。



重要

“热插”功能只支持 **VirtIO** 磁盘，虚拟机操作系统必须支持“热插”操作。支持这个功能的操作系统包括：

- Red Hat Enterprise Linux 6；
- Red Hat Enterprise Linux 5；
- Windows Server 2008； 和
- Windows Server 2003。

15.6.2. 网络接口子集合

15.6.2.1. 网络接口子集合

nics 子集合代表了虚拟机上的所有网络接口设备。一个 **nics** 表述包括以下项：

表 15.5. 虚拟机网络接口项

项	类型	描述	属性
link rel="statistics"	关系	虚拟机网络接口统计数据的 **statistics** 子集合链接。	
network id=	GUID	接口需要连接到的网络标识。可以是一个空的网络 ID	
interface	列举	nic 使用的驱动类型。**capabilities** 包括了有效的值。	
mac address=	字符串	接口的 MAC 地址。	
port_mirroring	复杂数据	指定 NIC 是否接收镜像的网络数据。使用一组 **network id=** 来定义 **networks** 项。	
plugged	布尔值	指定 NIC 是否被插到虚拟机上。	

项	类型	描述	属性
linked	布尔值	指定 NIC 是否被连接到虚拟机上。	

例 15.23. 网络接口的 XML 表述

```
<nic id="7a3cff5e-3cc4-47c2-8388-9adf16341f5e"
  ref="/api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/nics/
  7a3cff5e-3cc4-47c2-8388-9adf16341f5e">
    <link rel="statistics"
      href="/api/vms/082c794b-771f-452f-83c9-b2b5a19c0399/nics/
      7a3cff5e-3cc4-47c2-8388-9adf16341f5e/statistics"/>
    <name>nic1</name>
    <interface>virtio</interface>
    <mac address="00:1a:4a:16:84:07"/>
    <network id="00000000-0000-0000-0000-000000000009"
      href="/api/networks/00000000-0000-0000-0000-000000000009"/>
    <vm id="cdc0b102-fbfe-444a-b9cb-57d2af94f401"
      href="/api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401"/>
    <port_mirroring>
        <networks>
            <network id="56087282-d7a6-11e1-af44-001a4a400e0c"
              href="/api/networks/56087282-d7a6-11e1-af44-
001a4a400e0c"/>
        </networks>
    </port_mirroring>
</nic>
```

当添加一个网络接口时，**name** 和 **network** 项是必需的。使用 **id** 属性或 **name** 项来指定 **network** 项。

例 15.24. 创建一个虚拟机 NIC

```
POST /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/nics HTTP/1.1
Accept: application/xml
Content-type: application/xml

<nic>
    <name>nic1</name>
    <network id="00000000-0000-0000-0000-000000000009"/>
</nic>
```

API 用户可以使用 **PUT** 请求修改网络接口。

例 15.25. 更新虚拟机 NIC

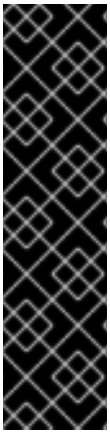
```
PUT /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/nics/
7a3cff5e-3cc4-47c2-8388-9adf16341f5e HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

```
<nic>
    <name>nic2</name>
    <network id="00000000-0000-0000-0000-000000000010"/>
    <type>e1000</type>
</nic>
```

API 用户可以使用 **DELETE** 请求删除网络接口。

例 15.26. 删除虚拟机 NIC

```
DELETE /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/nics/
7a3cff5e-3cc4-47c2-8388-9adf16341f5e HTTP/1.1

HTTP/1.1 204 No Content
```



重要

“热插”功能只在虚拟机操作系统支持“热插”操作时才被支持。支持这个功能的操作系统包括：

- Red Hat Enterprise Linux 6；
- Red Hat Enterprise Linux 5；
- Windows Server 2008 和
- Windows Server 2003。

15.6.2.2. 网络接口统计数据子集合

每个虚拟机网络接口都有一个提供网络接口统计数据的 **statistics** 子集合。每个 **statistic** 都包括以下项：

表 15.6. 虚拟机网络接口统计数据项

项	类型	描述
name	字符串	统计数据记录的唯一标识。
description	字符串	统计数据的描述。
unit	字符串	统计数据值所使用的单位。
type	**GAUGE** 或 **COUNTER**	统计数据类型。
values type=	**INTEGER** 或 **DECIMAL**	统计数据值的数据类型。

项	类型	描述
value	复杂数据	包括 **datum** 的数据集。
datum	请参阅 **values type**	**value** 中的一条数据。
nic id=	关系	与包括 **nic** 资源的关系。

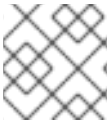
下表列出了虚拟机网络接口的统计数据类型。

表 15.7. 虚拟机 NIC 统计数据类型

名称	描述
data.current.rx	数据接收频率（字节/每秒）
data.current.tx	数据传输频率（字节/每秒）
errors.total.rx	接收数据中的错误总数
errors.total.tx	传输数据中的错误总数

例 15.27. 虚拟机 NIC 统计数据子集合的 XML 表述

```
<statistics>
    <statistic id="ecd0559f-e88f-3330-94b4-1f091b0ffdf7"
      href="/api/vms/3a42530e-3bc5-4094-829d-489257894c2a/nics/
      6cd08e76-57c0-41ba-a728-7eba46ae1e36/statistics/
      ecd0559f-e88f-3330-94b4-1f091b0ffdf7">
        <name>data.current.rx</name>
        <description>Receive data rate</description>
        <values type="DECIMAL">
            <value>
                <datum>0</datum>
            </value>
        </values>
        <type>GAUGE</type>
        <unit>BYTES_PER_SECOND</unit>
        <nic id="6cd08e76-57c0-41ba-a728-7eba46ae1e36"
          href="/api/vms/3a42530e-3bc5-4094-829d-489257894c2a/
          nics/6cd08e76-57c0-41ba-a728-7eba46ae1e36"/>
    </statistic>
    ...
</statistics>
```



注意

这个 **statistics** 子集合是只读的。

15.6.3. 虚拟 NUMA 节点子集合

numanodes 子集合代表了一个虚拟机上的所有虚拟 NUMA 节点。一个 **vm_numa_node** 会包括以下项：

表 15.8. 虚拟 NUMA 节点的项

项	类型	描述	属性
index	整数	虚拟 NUMA 节点的索引号。	
memory	整数	分配给虚拟 NUMA 节点的内存数量（以 MB 为单位）。	
cpu	复杂数据	与这个虚拟 NUMA 节点相关联的 CPU 拓扑。每个 **core** 项包括一个 **index** 属性代表相关联的内核索引号。	
vm id=	GUID	包括虚拟机的 ID。	
numa_node_pins	复杂数据	把虚拟 NUMA 节点固定到一个主机的 NUMA 节点。每个 **numa_node_pin** 项包括一个 **pinned="true"** 布尔值，以及主机 NUMA 节点的 **index** 号。	

例 15.28. 一个虚拟 NUMA 节点的 XML 表述

```
<vm_numa_node href="/api/vms/c7ecd2dc-dbd3-4419-956f-
1249651c0f2b/numanodes/3290b973-ed3e-4f0b-bbf5-9be10d229e50"
id="3290b973-ed3e-4f0b-bbf5-9be10d229e50">
        <index>0</index>
        <memory>1024</memory>
        <cpu>
            <cores>
                <core index="0"/>
            </cores>
        </cpu>
        <vm href="/api/vms/c7ecd2dc-dbd3-4419-956f-1249651c0f2b"
id="c7ecd2dc-dbd3-4419-956f-1249651c0f2b"/>
        <numa_node_pins>
            <numa_node_pin pinned="true" index="0">
                <host_numa_node id="417cdefb-8c47-4838-87f3-
dd0498fdf6c7"/>
```

```
            </numa_node_pin>
        </numa_node_pins>
</vm_numa_node>
```

在添加一个新虚拟 NUMA 节点时，需要 **index**、**memory** 和 **cpu** 项。

例 15.29. 把一个新的虚拟 NUMA 节点添加到一个虚拟机

```
POST /api/vms/c7ecd2dc-dbd3-4419-956f-1249651c0f2b/numanodes HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm_numa_node>
  <index>0</index>
  <memory>1024</memory>
  <cpu>
    <cores>
      <core index="0"/>
    </cores>
  </cpu>
</vm_numa_nodes>
```

使用 **PUT** 请求来更新一个虚拟 NUMA 节点。您可以使用一个 **PUT** 请求来把一个虚拟 NUMA 节点固定到一个主机上的一个物理 NUMA 节点上。

例 15.30. 更新一个虚拟 NUMA 节点

```
PUT /api/vms/c7ecd2dc-dbd3-4419-956f-1249651c0f2b/numanodes/3290b973-
ed3e-4f0b-bbf5-9be10d229e50 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm_numa_node>
  <numa_node_pins>
    <numa_node_pin pinned="true" index="0">
      <host_numa_node id="417cdefb-8c47-4838-87f3-dd0498fdf6c7"/>
    </numa_node_pin>
  </numa_node_pins>
</vm_numa_node>
```

使用 **DELETE** 请求删除一个虚拟 NUMA 节点。

例 15.31. 删除一个虚拟 NUMA 节点

```
DELETE /api/vms/c7ecd2dc-dbd3-4419-956f-1249651c0f2b/numanodes/3290b973-
ed3e-4f0b-bbf5-9be10d229e50 HTTP/1.1

HTTP/1.1 204 No Content
```

15.6.4. CD-ROM 子集合

cdroms 子集合代表了虚拟机上的 CD-ROM 设备。一个 **cdrom** 表述包括以下项：

表 15.9. 虚拟机 CD-ROM 项

项	类型	描述	属性
file id=	字符串/文件名	ISO 镜像的标识。	

例 15.32. CD-ROM 设备的 XML 表述

```
<cdrom id="00000000-0000-0000-0000-000000000000"
  href="/api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/cdroms/
  00000000-0000-0000-0000-000000000000">
    <file id="rhel-server-6.0-x86_64-dvd.iso"/>
    <vm id="cdc0b102-fbfe-444a-b9cb-57d2af94f401"
      href="/api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401"/>
</cdrom>
```

使用带有 **file id** 项的 **PUT** 请求来添加一个新 CD-ROM 资源。

例 15.33. 添加一个新 CD-ROM 文件

```
PUT /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/cdroms/00000000-0000-
0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<cdrom>
    <file id="fedora-15-x86_64-dvd.iso"/>
</cdrom>
```

API 使用 **PUT** 请求更换 CD-ROM：

例 15.34. 更换 CD-ROM 文件

```
PUT /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/cdroms/00000000-0000-
0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<cdrom>
    <file id="fedora-15-x86_64-dvd.iso"/>
</cdrom>
```

API 使用一个带有额外 **current** URI 参数的 **PUT** 请求来更换当前会话中使用的 CD-ROM：

例 15.35. 更换当前会话中的 CD-ROM 文件

```
PUT /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/cdroms/00000000-0000-
0000-0000-000000000000;current=true HTTP/1.1
Accept: application/xml
Content-type: application/xml

<cdrom>
    <file id="fedora-15-x86_64-dvd.iso"/>
</cdrom>
```

要临时弹出 CD-ROM，发送一个到虚拟机的 **cdroms** 子集合的 **PUT** 请求，这个请求中需要使用 **current=true** 参数：

例 15.36. 在当前会话中弹出 CD-ROM 文件

```
PUT /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/cdroms/00000000-0000-
0000-0000-000000000000;current=true HTTP/1.1
Accept: application/xml
Content-type: application/xml
<cdrom>
  <file id=""/>
</cdrom>
```

注意

重启虚拟机后会自动连接到 CD-ROM。

要永久弹出 CD-ROM，向虚拟机的 **cdroms** 子集合发送 **PUT** 请求：

例 15.37. 永久弹出 CD-ROM 文件

```
PUT /api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/cdroms/00000000-0000-
0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml
<cdrom>
  <file id=""/>
</cdrom>
```

注意

虚拟机只包括一个 CD-ROM 设备。

15.6.5. 快照子集合

15.6.5.1. 快照子集合

虚拟机使用快照来保存和恢复磁盘的状态。和其它集合相似，用户可以使用 **rel="snapshot"** 子集合表述和管理快照。

每个虚拟机快照都由一个单独的、带有以下项的 **snapshot** 项来表述：

表 15.10. 虚拟机快照的项

项	类型	描述	属性
vm id=	GUID	快照所属虚拟机的 ID 和 URI。	
link rel="restore"	关系	到恢复虚拟机快照的一个链接。	
link rel="prev"	关系	虚拟机前一个快照的标识。	
type	字符串	快照的类型。例如，**active** 或 **regular**。	
date	**xsd:dateTime** 格式：**YYYY-MM-DDThh:mm:ss**	快照被创建的日期和时间。	
snapshot_status	字符串	快照的当前状态。	
persist_memorystate	布尔值	指定在进行快照时，是否在快照中包括虚拟机内存的状态。	

注意

不能使用 **PUT** 修改快照。

例 15.38. 虚拟机快照的 XML 表述

```
<snapshot id="00000000-0000-0000-0000-000000000000"
  href="/api/vms/00000000-0000-0000-0000-000000000000/snapshots/
  00000000-0000-0000-0000-000000000000">
    <actions>
      <link rel="restore"
      href="/api/vms/00000000-0000-0000-0000-000000000000/snapshots/
      00000000-0000-0000-0000-000000000000/restore"/>
    <link rel="prev"
      href="/api/vms/00000000-0000-0000-0000-000000000000/snapshots/
    </actions>
    <vm id="00000000-0000-0000-0000-000000000000"
      href="/api/vms/00000000-0000-0000-0000-000000000000"/>
    <description>Virtual Machine 1 - Snapshot A</description>
    <type>active</type>
```

```
    <date>2010-08-16T14:24:29</date>
    <snapshot_status>ok</snapshot_status>
    <persist_memorystate>false</persist_memorystate>
</snapshot>
```

使用带有 **All-Content: true** 头的、对虚拟机快照的 **GET** 请求可以获得包括额外的、代表快照的 OVF 数据。

如果 **Accept** 头为空，它会被默认设置为 **application/xml**，数据由 HTML 项来代表，因此不会影响到 XML tag。如果指定 **Accept: application/json** 头，返回的数据会使用标准的 XML tag。这个示例使用标准块格式构成。

例 15.39. 代表一个快照的 ovf 数据的 XML

```
GET /api/vms/42ec2621-7ad6-4ca2-bd68-973a44b2562e/snapshots HTTP/1.1
All-Content: true

<?xml version='1.0' encoding='UTF-8'?>
<ovf:Envelope xmlns:ovf=\"http://schemas.dmtf.org/ovf/envelope/1/\"
  xmlns:rasd=\"http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData\"
  xmlns:vssd=\"http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_VirtualSystemSettingData\"
  xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
ovf:version=\"3.5.0.0\">
  <References>
  <File ovf:href=\"ad353554-f668-46cf-aa3c-e57383de2c92/40456d92-3687-
4a85-bab3-87b4cc7af459\"
    ovf:id=\"40456d92-3687-4a85-bab3-87b4cc7af459\"
ovf:size=\"10737418240\"
    ovf:description=\"Active VM\"/>
  <Nic ovf:id=\"be14bfc8-3dbd-4ac1-ba02-c6dfa7fc707c\"/>
  </References>
  <Section xsi:type=\"ovf:NetworkSection_Type\">
    <Info>List of networks</Info><Network ovf:name=\"Network 1\"/>
  </Section>
  <Section
    xsi:type=\"ovf:DiskSection_Type\">
    <Info>List of Virtual Disks</Info>
    <Disk ovf:diskId=\"40456d92-3687-4a85-bab3-87b4cc7af459\"
    ovf:size=\"10\" ovf:actual_size=\"0\"
    ovf:vm_snapshot_id=\"a209216d-2909-4802-8886-02aad55dccc8\"
    ovf:parentRef=\"\"
    ovf:fileRef=\"ad353554-f668-46cf-aa3c-e57383de2c92/40456d92-3687-
4a85-bab3-87b4cc7af459\"
    ovf:format=\"http://www.vmware.com/specifications/vmdk.html#sparse\"
    ovf:volume-format=\"RAW\"
    ovf:volume-type=\"Preallocated\"
    ovf:disk-interface=\"VirtIO\"
    ovf:boot=\"true\"
    ovf:disk-alias=\"VM_01_Disk1\"
    ovf:wipe-after-delete=\"false\"/>
  </Section>
  <Content
```

```
    ovf:id=\"out\"
    xsi:type=\"ovf:VirtualSystem_Type\">
    <CreationDate>2015/02/09 13:53:53</CreationDate>
    <ExportDate>2015/02/10 00:39:24</ExportDate>
    <DeleteProtected>false</DeleteProtected>
    <SsoMethod>guest_agent</SsoMethod>
    <IsSmartcardEnabled>false</IsSmartcardEnabled>
    <TimeZone>Etc/GMT</TimeZone>
<default_boot_sequence>0</default_boot_sequence>
    <Generation>1</Generation>
    <VmType>1</VmType>
    <MinAllocatedMem>1024</MinAllocatedMem>
    <IsStateless>false</IsStateless>
    <IsRunAndPause>false</IsRunAndPause>
    <AutoStartup>false</AutoStartup>
    <Priority>1</Priority>
    <CreatedByUserId>fdfc627c-d875-11e0-90f0-
83df133b58cc</CreatedByUserId>
    <IsBootMenuEnabled>false</IsBootMenuEnabled>
    <IsSpiceFileTransferEnabled>true</IsSpiceFileTransferEnabled>
    <IsSpiceCopyPasteEnabled>true</IsSpiceCopyPasteEnabled>
    <Name>VM_01</Name>
    <TemplateId>00000000-0000-0000-0000-000000000000</TemplateId>
    <TemplateName>Blank</TemplateName>
    <IsInitilized>true</IsInitilized>
    <Origin>3</Origin>
    <DefaultDisplayType>1</DefaultDisplayType>
    <TrustedService>false</TrustedService>
    <OriginalTemplateId>00000000-0000-0000-0000-
000000000000</OriginalTemplateId>
    <OriginalTemplateName>Blank</OriginalTemplateName>
    <UseLatestVersion>false</UseLatestVersion>
    <Section ovf:id=\"42ec2621-7ad6-4ca2-bd68-973a44b2562e\"
ovf:required=\"false\" xsi:type=\"ovf:OperatingSystemSection_Type\">
      <Info>Guest Operating System</Info>
      <Description>other</Description>
    </Section>
    <Section xsi:type=\"ovf:VirtualHardwareSection_Type\">
      <Info>1 CPU, 1024 Memeory</Info>
      <System>
        <vssd:VirtualSystemType>ENGINE 3.5.0.0</vssd:VirtualSystemType>
      </System>
      <Item>
        <rasd:Caption>1 virtual cpu</rasd:Caption>
        <rasd:Description>Number of virtual CPU</rasd:Description>
        <rasd:InstanceId>1</rasd:InstanceId>
        <rasd:ResourceType>3</rasd:ResourceType>
        <rasd:num_of_sockets>1</rasd:num_of_sockets>
        <rasd:cpu_per_socket>1</rasd:cpu_per_socket>
      </Item>
      <Item>
        <rasd:Caption>1024 MB of memory</rasd:Caption>
        <rasd:Description>Memory Size</rasd:Description>
        <rasd:InstanceId>2</rasd:InstanceId>
        <rasd:ResourceType>4</rasd:ResourceType>
        <rasd:AllocationUnits>MegaBytes</rasd:AllocationUnits>
```

```
        <rasd:VirtualQuantity>1024</rasd:VirtualQuantity>
      </Item>
      <Item>
        <rasd:Caption>VM_01_Disk1</rasd:Caption>
        <rasd:InstanceId>40456d92-3687-4a85-bab3-
87b4cc7af459</rasd:InstanceId>
        <rasd:ResourceType>17</rasd:ResourceType>
        <rasd:HostResource>ad353554-f668-46cf-aa3c-
e57383de2c92/40456d92-3687-4a85-bab3-87b4cc7af459</rasd:HostResource>
        <rasd:Parent>00000000-0000-0000-0000-000000000000</rasd:Parent>
        <rasd:Template>00000000-0000-0000-0000-
000000000000</rasd:Template>
        <rasd:ApplicationList></rasd:ApplicationList>
        <rasd:StoragePoolId>00000002-0002-0002-0002-
000000000255</rasd:StoragePoolId>
        <rasd:CreationDate>2015/02/09 13:54:41</rasd:CreationDate>
        <rasd:LastModified>1970/01/01 00:00:00</rasd:LastModified>
        <rasd:last_modified_date>2015/02/10
00:39:22</rasd:last_modified_date>
        <Type>disk</Type>
        <Device>disk</Device>
        <rasd:Address>{slot=0x06, bus=0x00, domain=0x0000, type=pci,
function=0x0}</rasd:Address>
        <BootOrder>1</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>false</IsReadOnly>
        <Alias>virtio-disk0</Alias>
      </Item>
      <Item>
        <rasd:Caption>Ethernet adapter on ovirtmgmt</rasd:Caption>
        <rasd:InstanceId>be14bfc8-3dbd-4ac1-ba02-
c6dfa7fc707c</rasd:InstanceId>
        <rasd:ResourceType>10</rasd:ResourceType>
        <rasd:OtherResourceType>ovirtmgmt</rasd:OtherResourceType>
        <rasd:ResourceSubType>3</rasd:ResourceSubType>
        <rasd:Connection>ovirtmgmt</rasd:Connection>
        <rasd:Linked>true</rasd:Linked>
        <rasd:Name>nic1</rasd:Name>
        <rasd:MACAddress>00:1a:4a:87:cb:00</rasd:MACAddress>
        <rasd:speed>1000</rasd:speed>
        <Type>interface</Type>
        <Device>bridge</Device>
        <rasd:Address>{slot=0x03, bus=0x00, domain=0x0000, type=pci,
function=0x0}</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>false</IsReadOnly>
        <Alias>net0</Alias>
      </Item>
      <Item>
        <rasd:Caption>USB Controller</rasd:Caption>
        <rasd:InstanceId>3</rasd:InstanceId>
        <rasd:ResourceType>23</rasd:ResourceType>
        <rasd:UsbPolicy>DISABLED</rasd:UsbPolicy>
      </Item>
      <Item>
```

```
        <rasd:Caption>Graphical Controller</rasd:Caption>
        <rasd:InstanceId>17bbf0db-7cf0-4529-9b53-
dee6dee41cfd</rasd:InstanceId>
        <rasd:ResourceType>20</rasd:ResourceType>
        <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
        <rasd:SinglePciQxl>false</rasd:SinglePciQxl>
        <Type>video</Type>
        <Device>qxl</Device>
        <rasd:Address>{slot=0x02, bus=0x00, domain=0x0000, type=pci,
function=0x0}</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>true</IsReadOnly>
        <Alias>video0</Alias>
        <SpecParams>
          <vram>32768</vram>
          <heads>1</heads>
        </SpecParams>
      </Item>
      <Item>
        <rasd:Caption>CDROM</rasd:Caption>
        <rasd:InstanceId>7ce1bd14-d98a-43ba-beee-
520bdfd9c698</rasd:InstanceId>
        <rasd:ResourceType>15</rasd:ResourceType>
        <Type>disk</Type>
        <Device>cdrom</Device>
        <rasd:Address>{bus=1, controller=0, type=drive, target=0,
unit=0}</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>true</IsReadOnly>
        <Alias>ide0-1-0</Alias></Item>
      <Item>
        <rasd:ResourceType>0</rasd:ResourceType>
        <rasd:InstanceId>8758c42f-7523-461b-82bb-
41d91e46fd36</rasd:InstanceId>
        <Type>controller</Type>
        <Device>usb</Device>
        <rasd:Address>{slot=0x01, bus=0x00, domain=0x0000, type=pci,
function=0x2}</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>false</IsReadOnly>
        <Alias>usb0</Alias>
      </Item>
      <Item>
        <rasd:ResourceType>0</rasd:ResourceType>
        <rasd:InstanceId>58f1a596-553e-4e95-9331-
64b5d8cebe2e</rasd:InstanceId>
        <Type>controller</Type>
        <Device>ide</Device>
        <rasd:Address>{slot=0x01, bus=0x00, domain=0x0000, type=pci,
function=0x1}</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>false</IsReadOnly>
```

```
        <Alias>ide0</Alias>
      </Item>
      <Item>
        <rasd:ResourceType>0</rasd:ResourceType>
        <rasd:InstanceId>2f4f8aa5-25eb-4a31-b841-
50dc48fce4a7</rasd:InstanceId>
        <Type>channel</Type>
        <Device>unix</Device>
        <rasd:Address>{bus=0, controller=0, type=virtio-serial, port=1}
</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>false</IsReadOnly>
        <Alias>channel0</Alias>
      </Item>
      <Item>
        <rasd:ResourceType>0</rasd:ResourceType>
        <rasd:InstanceId>edaac3ed-2ab6-48b1-ae77-
cc98f8b45bd8</rasd:InstanceId>
        <Type>channel</Type>
        <Device>unix</Device>
        <rasd:Address>{bus=0, controller=0, type=virtio-serial, port=2}
</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>false</IsReadOnly>
        <Alias>channel1</Alias>
      </Item>
      <Item>
        <rasd:ResourceType>0</rasd:ResourceType>
        <rasd:InstanceId>8dfed248-5164-41d3-8b6e-
46aef9798d84</rasd:InstanceId>
        <Type>channel</Type>
        <Device>spicevmc</Device>
        <rasd:Address>{bus=0, controller=0, type=virtio-serial, port=3}
</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>false</IsReadOnly>
        <Alias>channel2</Alias>
      </Item>
      <Item>
        <rasd:ResourceType>0</rasd:ResourceType>
        <rasd:InstanceId>d184185e-ee19-442a-88f5-
6a48f14164e1</rasd:InstanceId>
        <Type>controller</Type>
        <Device>virtio-scsi</Device>
        <rasd:Address>{slot=0x04, bus=0x00, domain=0x0000, type=pci,
function=0x0}</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>false</IsReadOnly>
        <Alias>scsi0</Alias>
      </Item>
      <Item>
        <rasd:ResourceType>0</rasd:ResourceType>
```

```
        <rasd:InstanceId>374d219e-e2ff-4755-a544-
d537c87e82df</rasd:InstanceId>
        <Type>controller</Type>
        <Device>virtio-serial</Device>
        <rasd:Address>{slot=0x05, bus=0x00, domain=0x0000, type=pci,
function=0x0}</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>false</IsReadOnly>
        <Alias>virtio-serial0</Alias>
      </Item>
      <Item>
        <rasd:ResourceType>0</rasd:ResourceType>
        <rasd:InstanceId>cf3d7121-9db0-4fd1-bd12-
50ce4e1ce379</rasd:InstanceId>
        <Type>balloon</Type>
        <Device>memballoon</Device>
        <rasd:Address>{slot=0x07, bus=0x00, domain=0x0000, type=pci,
function=0x0}</rasd:Address>
        <BootOrder>0</BootOrder>
        <IsPlugged>true</IsPlugged>
        <IsReadOnly>true</IsReadOnly>
        <Alias>balloon0</Alias>
        <SpecParams>
          <model>virtio</model>
        </SpecParams>
      </Item>
    </Section>
  </Content>
</ovf:Envelope>
```

您可以为正在运行的主机进行快照（实时快照），或使用 **POST** 来关闭虚拟机：

例 15.40. 创建一个虚拟机的快照

```
POST /api/vms/00000000-0000-0000-0000-000000000000/snapshots/ HTTP/1.1
Accept: application/xml
Content-type: application/xml

<snapshot>
<description>Snapshot description</description>
</snapshot>
```

您可以在快照表述中使用 **rel="restore"** 操作链接来恢复虚拟机快照：

例 15.41. 恢复虚拟机快照

```
POST /api/vms/00000000-0000-0000-0000-000000000000/snapshots/00000000-
0000-0000-0000-000000000000/restore HTTP/1.1
Accept: application/xml
```

```
Content-type: application/xml

<action/>
```

15.6.5.2. 通过快照克隆虚拟机

API 提供了一个使用其它虚拟机快照创建虚拟机的功能。API 用户可以在创建新虚拟机时保留原始虚拟机的所有快照。

使用快照创建虚拟机需要在标准的虚拟机表述中添加一个 **snapshots** 项。用户需要向 **vms** 集合发送一个 **POST** 请求。

snapshots 项包括了一个 **snapshot id=** 项来指定创建虚拟机需要基于的快照。

例 15.42. 从快照克隆虚拟机

```
POST /api/vms HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm>
  ...
  <snapshots>
    <snapshot id="3f68ee63-0016-4f8c-9b8a-11d9f28f7c9e"/>
  </snapshots>
  ...
</vm>
```

15.6.6. 统计数据子集合

每个虚拟机都包括一个 **statistics** 子集合来提供与虚拟机相关的统计数据。每个 **statistic** 包括以下项：

表 15.11. 虚拟机磁盘统计数据的项

项	类型	描述
name	字符串	统计数据记录的唯一标识。
description	字符串	统计数据的描述。
unit	字符串	统计数据值所使用的单位。
type	**GAUGE** 或 **COUNTER**	统计数据类型。
values type=	**INTEGER** 或 **DECIMAL**	统计数据值的数据类型。
value	复杂数据	包括 **datum** 的数据集。

项	类型	描述
datum	请参阅 **values type**	**value** 中的一条数据。
vm id=	关系	与包括 **vm** 资源的关系。

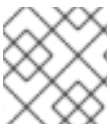
下表列出了虚拟机的统计数据类型。

表 15.12. 虚拟机统计数据类型

名称	描述
memory.installed	为虚拟机使用分配的总内存数量（以字节为单位）。
memory.used	虚拟机当前使用的内存（以字节为单位）。
cpu.current.guest	虚拟机使用的 CPU 百分数。
cpu.current.hypervisor	hypervisor 消耗的 CPU 百分数。
cpu.current.total	当前使用的 CPU 百分比。

例 15.43. 虚拟机统计数据子集合 XML 表述

```
<statistics>
    <statistic id="ef802239-b74a-329f-9955-be8fea6b50a4"
      href="/api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401/
      statistics/ef802239-b74a-329f-9955-be8fea6b50a4">
        <name>memory.installed</name>
        <description>Total memory configured</description>
        <unit>BYTES</unit>
        <type>GUAGE</type>
        <values type="DECIMAL">
            <value>
                <datum>1073741824<datum>
            </value>
        </values>
        <vm id="cdc0b102-fbfe-444a-b9cb-57d2af94f401"
          href="/api/vms/cdc0b102-fbfe-444a-b9cb-57d2af94f401"/>
    </statistic>
    ...
</statistics>
```



注意

虚拟机的 **statistics** 子集合是只读的。

15.6.7. 显示虚拟机对话信息

为虚拟机发一个 **GET** 请求并使用 **session** 子集合来查看初始 SPICE 控制台会话的用户的会话信息，以及登录到这个虚拟机的用户的会话信息。

虚拟机的 **session** 信息作为子集合被列出：

例 15.44. 显示一个虚拟机的会话信息

```
GET /api/roles/a1a701f1-aa06-4f02-af17-158be31489b3/sessions HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<sessions>
  <session id="37a6259c-c0c1-dae2-99a7-866489dff0bd"
    href= "/api/vms/a1a701f1-aa06-4f02-af17-
158be31489b3/sessions/37a6259c-c0c1-dae2-99a7-866489dff0bd">
  <vm href= "/api/vms/a1a701f1-aa06-4f02-af17-158be31489b3"
id="a1a701f1-aa06-4f02-af17-158be31489b3"/>
  <ip address="192.0.2.0"/>
  <user href= "/api/users/fdfc627c-d875-11e0-90f0-83df133b58cc"
id="fdfc627c-d875-11e0-90f0-83df133b58cc">
    <domain href= "/api/domains/696e7465-726e-616c-696e-7465726e616c"
id="696e7465-726e-616c-696e-7465726e616c">
      <name>internal</name>
    </domain>
    <user_name>admin</user_name>
    </user>
    <console_user>true</console_user>
  </session>
  <session id="37a6259c-c0c1-dae2-99a7-866489dff0bd"
    href= "/api/vms/a1a701f1-aa06-4f02-af17-
158be31489b3/sessions/37a6259c-c0c1-dae2-99a7-866489dff0bd" >
    <vm href= "/api/vms/a1a701f1-aa06-4f02-af17-158be31489b3"
id="a1a701f1-aa06-4f02-af17-158be31489b3"/>
    <user>
      <user_name>root</user_name>
    </user>
  </session>
</sessions>
```

15.7. 操作

15.7.1. 启动虚拟机操作

启动操作会启动一个停止的、关闭的或挂起的虚拟机。

例 15.45. 启动虚拟机的操作

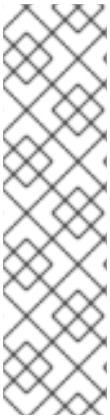
```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/start HTTP/1.1
Accept: application/xml
```

```
Content-type: application/xml

<action/>
```

在启动操作中可以使用一个 **vm** 项作为参数。如果使用了这个 **vm** 项，虚拟机会在启动时使用这个项中的值来覆盖系统设置。在 REST API 中使用带有 **vm** 项的启动操作与在管理门户或用户门户中使用**只运行一次**窗口的作用相同。这些设置会在虚拟机被停止前一直有效。这些设置包括 **os**、**domain**、**placement_policy**、**cdroms**、**stateless** 和 **display type**。

例 15.46. 使用指定的参数启动虚拟机

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/start HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <pause>true</pause>
    <vm>
        <stateless>true</stateless>
        <display>
            <type>spice</type>
        </display>
        <os>
            <boot dev="cdrom"/>
        </os>
        <cdroms>
            <cdrom>
                <file id="windows-xp.iso"/>
            </cdrom>
        </cdroms>
        <floppies>
            <floppy>
                <file id="virtio-win_x86.vfd"/>
            </floppy>
       </floppies>
        <domain>
            <name>domain.example.com</name>
            <user>
                <user_name>domain_user</user_name>
                <password>domain_password</password>
            </user>
        </domain>
        <placement_policy>
            <host id="02447ac6-bcba-448d-ba2b-f0f453544ed2"/>
        </placement_policy>
    </vm>
</action>
```

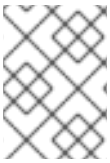


注意

- **domain** 项只用于运行 Windows 操作系统的虚拟机，来使它在使用 **start** 操作引导时覆盖系统的相应参数。**domain** 项决定了 Windows 虚拟机需要加入的域。如果域没有存在于 **domains** 集合中，这个项将需要额外的 **user** 身份验证信息，包括一个 **user_name** 和 **password**。如果域存在于 **domains** 集合中，这个操作将不需要额外的 **user** 身份验证信息。
- CD 镜像和软盘文件需要已存在于 ISO 域中。如果还没有存在于 ISO 域中，使用 ISO 上传工具来它们上传到域中。如需了解更多相关信息，请参阅 The ISO Uploader Tool。

15.7.2. 启动虚拟机时使用 Cloud-Init 操作

Cloud-Init 是一个自动化虚拟机初始设置的工具程序。您可以使用这个工具来配置主机名、网络接口、DNS 服务、认证密钥，以及设置用户名和密码。您还可以使用 **custom_script** 来指定一个在虚拟机引导时运行的自定义脚本。



注意

cloud-init 项只能被用来启动安装了 cloud-init 软件包的虚拟机。当 **cloud-init** 项被使用时，在 **initialization** 项内，但在 **cloud-init** 项外的所有项都会被忽略。

例 15.47. 在启动虚拟机时使用 Cloud-Init 的操作

以下示例演示了如何在启动虚拟机时使用 Cloud-Init 工具来设置主机名、修改 root 密码、为 **eth0** 接口设置静态 IP、配置 DNS，以及为 **root** 用户添加一个 SSH 密钥。

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/start HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <vm>
       <initialization>
        <cloud_init>
          <host>
           <address>MyHost.MyDomain.com</address>
          </host>
          <users>
           <user>
            <user_name>root</user_name>
            <password>p@55w0rd!</password>
           </user>
          </users>
          <network_configuration>
           <nics>
            <nic>
              <name>eth0</name>
              <boot_protocol>static</boot_protocol>
              <network>
                <ip address="192.168.122.31" netmask="255.255.255.0"
gateway="192.168.122.1"/>
              </network>
```

```
            <on_boot>true</on_boot>
          </nic>
        </nics>
        <dns>
          <servers>
            <host>
              <address>192.168.122.1</address>
            </host>
          </servers>
          <search_domains>
            <host>
              <address>MyDomain.com</address>
            </host>
          </search_domains>
        </dns>
      </network_configuration>
      <authorized_keys>
        <authorized_key>
          <user>
            <user_name>root</user_name>
          </user>
          <key>ssh-rsa AAAAB3Nza[...]75zkdD root@MyDomain.com</key>
        </authorized_key>
      </authorized_keys>
    </cloud_init>
    <custom_script><![CDATA[your script]]></custom_script>
  </initialization>
 </vm>
</action>
```

15.7.3. 停止虚拟机操作

停止操作会强制停止虚拟机的运行。

例 15.48. 停止虚拟机的操作

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/stop HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

15.7.4. 关闭虚拟机操作

关闭（shutdown）操作向虚拟机发送一个关闭请求。

例 15.49. 向虚拟机发送一个关闭请求操作

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/shutdown HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

```
<action/>
```

15.7.5. 挂起虚拟机操作

挂起操作会把虚拟机状态保存到磁盘后停止它。使用启动操作可以启动被挂起的虚拟机，并恢复其状态。

例 15.50. 保存虚拟机的当前状态后挂起虚拟机的操作

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/suspend HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

15.7.6. 重启虚拟机操作

重启操作向虚拟机发出重启请求。

例 15.51. 向虚拟机发出重启请求的操作

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/reboot HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

15.7.7. 允许用户从一个外部控制台登录来访问虚拟机

这个登录操作将允许用户从 Red Hat Enterprise Virtualization 环境以外的控制台访问虚拟机。

这个操作需要安装 rhevm-guest-agent-gdm-plugin 和 rhevm-guest-agent-pam-module 软件，**ovirt-guest-agent** 服务在虚拟机上运行。

为了可以通过一个外部控制台访问虚拟机，用户需要有虚拟机的适当权限。

例 15.52. 登录到一台虚拟机

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/logon HTTP/1.1
Content-Type: application/json
Content-Length: 2

{}
```

15.7.8. 从虚拟机池中删除虚拟机的操作

从虚拟机池中删除（detach）虚拟机的操作

例 15.53. 从虚拟机池中删除虚拟机的操作

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/detach HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

15.7.9. 迁移虚拟机操作

迁移操作把一个虚拟机迁移到另外一个物理主机。目标 **host** 项是一个可选项，如果没有使用，Red Hat Enterprise Virtualization Manager 会自动选择一个默认的主机进行迁移。如果 API 用户需要指定 **host**，用户可以使用 **id** 或 **name** 参数。

例 15.54. 把虚拟机迁移到另外一个主机的操作。

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/migrate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <host id="2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"/>
</action>
```

15.7.10. 取消虚拟机迁移操作

取消虚拟机迁移操作会停止所有把虚拟机迁移到其它主机的操作。

例 15.55. 取消虚拟机迁移操作

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/cancelmigration
HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

15.7.11. 导出虚拟机操作

导出操作会把虚拟机导出到一个 **export** 存储域。目标存储域需要使用 **storage_domain** 来指定。

如果在目标域中存在同名的虚拟机，导出操作会报告一个错误。把 **exclusive** 参数设置为 **true** 可以改变这个行为，它会把虚拟机导入到目标域中，并覆盖其中同名的虚拟机。

如果不需要把虚拟机的快照包括在导出的虚拟机中，把 **discard_snapshots** 参数设置为 **true**。

例 15.56. 把虚拟机导出到一个导出存储域的操作

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/export HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <storage_domain>
        <name>export1</name>
    </storage_domain>
    <exclusive>true</exclusive>
    <discard_snapshots>true</discard_snapshots>
</action>
```

15.7.12. 虚拟机票证（ticket）操作

票证操作会产生一个与时间相关的、用来为访问虚拟机进行验证的验证票证。客户端提供的 **action** 操作可以包括一个可选的 **ticket** 表述，它包括一个 **value**（如果令牌字符串需要使用特殊的格式）和（或）一个 **expiry** 值（以分钟代表）。这个请求会返回实际票证的值和过期的时间。

例 15.57. 为虚拟机创建用户验证令牌

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/ticket HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <ticket>
        <expiry>120</expiry>
    </ticket>
</action>

200 OK
Content-Type: application/xml

<action id="94e07552-14ba-4c27-8ce6-2cc75190d3ef"
  href="/api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/ticket/
  94e07552-14ba-4c27-8ce6-2cc75190d3ef">
    <status>
        <state>complete</state>
    </status>
    <ticket>
        <value>5c7CSzK8Sw41</value>
        <expiry>120</expiry>
    </ticket>
    <link rel="parent"
      href="/api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720"/>
    <link rel="replay"
      href="/api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/ticket"/>
</action>
```

15.7.13. 强制删除虚拟机操作

API 用户可以使用 **force** 操作强制删除一个出现问题的虚拟机。这个操作需要一个 **DELETE** 方法。请求的内容项中包括了一个 **action** 表述，其中的 **force** 参数被设为 **true**。这个请求还需要一个额外的 **Content-type: application/xml** 头来处理内容项中的 XML 表述。

例 15.58. 强制删除虚拟机操作

```
DELETE /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <force>true</force>
</action>
```

第 16 章 浮动磁盘

16.1. 浮动磁盘项

disks 集合提供了 Red Hat Enterprise Virtualization 环境中的磁盘的信息。用户可以把磁盘从任何虚拟机上取消关联，并把它们附加到其它虚拟机上，从而实现磁盘在不同虚拟机间"浮动"。API 用户可以通过从进入点 URI 获得的 **rel="disks"** 链接来访问这些信息。

以下表格包括了 **disks** 资源表述中的特定项的信息。

表 16.1. 浮动磁盘项

项	类型	描述	属性
link rel="statistics"	关系	虚拟机磁盘统计数据的 **statistics** 子集合链接。	
image_id	GUID	保存在所定义的存储域中的虚拟机镜像标识。	
storage_domains	复杂数据	与磁盘相关联的存储域。每个 **storage_domain** 项都包括一个带有存储域 GUID 的 **id** 属性。使用 **POST** 更新这个项可以执行从一个数据存储域到另一个数据存储域的磁盘实时迁移操作。	
size	整数	磁盘大小（以字节为单位）。	
provisioned_size	整数	配置的磁盘大小（以字节为单位）。	
actual_size	整数	磁盘的实际大小（以字节为单位）。	
status	**illegal**、**invalid**、**locked** 或 **ok**	磁盘设备的状态。**capabilities** 下的 **disk_states** 列出了它的值。	
interface	列举	连接到磁盘设备的接口驱动类型。**capabilities** 包括了有效的值。	
format	列举	底层的存储格式。**capabilities** 包括了有效的值。使用 Copy On Write（COW）可以创建快照，但对系统性能会有一些影响；Raw 不能创建快照，但可以提供更好的性能。	

项	类型	描述	属性
sparse	布尔值：true 或 false	**true**：磁盘所使用的物理存储无法被预先分配。	
bootable	布尔值：true 或 false	**true**：磁盘被设为可引导。	
--shareable	布尔值：true 或 false	**true**：磁盘可以被多个虚拟机共享。	
wipe_after_delete	布尔值：true 或 false	如果为 **true**，在磁盘被删除后，这个磁盘的底层物理存储会被清零。此选项会增加数据的安全性，但同时会消耗大量的操作资源，并会增加删除过程所需的时间。	
propagate_errors	布尔值：true 或 false	**true**：当磁盘出现错误时，不会导致虚拟机暂停，而是把磁盘错误传递给虚拟机的 OS。	
quota id=	GUID	为磁盘设置一个配额。	
lunStorage	复杂数据	为存储使用的直接 LUN 的标识。需要一个包括 iSCSI 或 FCP 设备详情的 **storage** 项。	
active	布尔值	指定磁盘是否连接到虚拟机。	

重要

基于名称对磁盘进行搜索需要 **alias** 搜索参数，而不是 **name**。

16.2. 浮动磁盘的 XML 表述

例 16.1. 浮动磁盘的 XML 表述

```
<disk id="ed7feafe-9aaf-458c-809a-ed789cdbd5b4"
  href="/api/disks/ed7feafe-9aaf-458c-809a-ed789cdbd5b4">
    <link rel="statistics"
      href="/api/disks/ed7feafe-9aaf-458c-809a-
ed789cdbd5b4/statistics"/>
    <storage_domains>
        <storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"/>
    </storage_domains>
    <size>10737418240</size>
    <type>system</type>
    <status>
        <state>ok</state>
    </status>
    <interface>virtio</interface>
```

```
    <format>raw</format>
    <bootable>true</bootable>
    <shareable>true</shareable>
    <lunStorage>
        <storage>
            <logical_unit id="lun1">
                ...
            </logical_unit>
        <storage>
    </lunStorage>
</disk>
```

16.3. 方法

16.3.1. 创建浮动磁盘

API 需要 **size** 和 **storage_domains** 项来创建新浮动磁盘。

例 16.2. 创建一个新浮动磁盘设备

```
POST /api/disks HTTP/1.1
Accept: application/xml
Content-type: application/xml

<disk>
    <storage_domains>
        <storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"/>
    </storage_domains>
    <size>8589934592</size>
    <type>system</type>
    <interface>virtio</interface>
    <format>cow</format>
</disk>
```

16.4. 子集合

16.4.1. 统计数据子集合

每个浮动磁盘都包括一个 **statistics** 子集合来提供与磁盘相关的统计数据。每个 **statistic** 包括以下项：

表 16.2. 虚拟机磁盘统计数据的项

项	类型	描述
name	字符串	统计数据记录的唯一标识。
description	字符串	统计数据的描述。

项	类型	描述
unit	字符串	统计数据值所使用的单位。
type	**GAUGE** 或 **COUNTER**	统计数据类型。
values type=	**INTEGER** 或 **DECIMAL**	统计数据值的数据类型。
value	复杂数据	包括 **datum** 的数据集。
datum	请参阅 **values type**	**value** 中的一条数据。
disk id=	关系	与包括 **disk** 资源的关系。

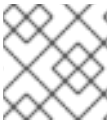
下表列出了浮动磁盘的统计数据类型信息。

表 16.3. 磁盘统计数据类型

名称	描述
data.current.read	读磁盘操作的数据传输频率（字节/每秒）
data.current.write	写磁盘操作的数据传输频率（字节/每秒）

例 16.3. 虚拟机统计数据子集合的 XML 表述

```
<statistics>
    <statistic id="33b9212b-f9cb-3fd0-b364-248fb61e1272"
      href="/api/disks/f28ec14c-fc85-43e1-818d-96b49d50e27b/statistics/
      33b9212b-f9cb-3fd0-b364-248fb61e1272">
        <name>data.current.read</name>
        <description>Read data rate</description>
        <values type="DECIMAL">
            <value>
                <datum>0</datum>
            </value>
        </values>
        <type>GAUGE</type>
        <unit>BYTES_PER_SECOND</unit>
        <disk id="f28ec14c-fc85-43e1-818d-96b49d50e27b"
          href="/api/disks/f28ec14c-fc85-43e1-818d-96b49d50e27b"/>
    </statistic>
    ...
</statistics>
```



注意

这个 **statistics** 子集合是只读的。

16.5. 操作

16.5.1. 复制一个浮动磁盘

在复制浮动磁盘时，API 需要 **storage_domain**。可选的 **name** 被用来指定磁盘的别名。

例 16.4. 复制一个浮动磁盘

```
POST /api/disks/54a81464-b758-495a-824b-1e7937116ae5/copy HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <storage_domain id="c8e108f7-c049-40d2-ad3d-620e4638828e"/>
    <disk>
        <name>rhel_disk2</name>
    </disk>
</action>
```

第 17 章 模板

17.1. 虚拟机模板项

templates 集合提供了 Red Hat Enterprise Virtualization 环境中的虚拟机模板信息。API 用户可以通过从进入点 URI 获得的 **rel="templates"** 链接来访问这些信息。

使用带有 **All-Content: true** 头的 **GET** 请求可以获得额外信息。

以下表格包括了虚拟机模板表述中的特定项的信息。

表 17.1. 虚拟机模板项

项	类型	描述	属性
link rel="disks"	关系	虚拟机模板资源的 **disks** 子集合。	
link rel="nics"	关系	虚拟机模板资源的 **nics** 子集合。	
link rel="cdroms"	关系	虚拟机模板资源的 **cdroms** 子集合。	
link rel="permissions"	关系	虚拟机模板权限的 **permissions** 子集合。	
type	列举	模板提供的虚拟机类型。**capabilities** 包括了有效值列表。	
status	**illegal**、**locked** 或 **ok**	模板状态。这些状态在 **capabilities** 下面的 **template_states** 中列出。	
memory	整数	分配给虚拟机的内存大小（以字节为单位）。	
cpu	复杂数据	虚拟机可用的 CPU **topology**（如 **sockets** 数量和 **cores** 数量）。	
os type=	字符串，如 **RHEL5** 或 **WindowsXP**	虚拟机操作系统类型。	
os boot dev=	列举	**boot** 项的 **dev** 属性所描述的引导设备列表。**capabilities** 包括了它的有效值。	
os kernel	字符串	用来引导模板的内核镜像路径。	

项	类型	描述	属性
os initrd	字符串	以上内核使用的 initrd 镜像。	
os cmdline	字符串	以上内核使用的内核命令行参数。	
cluster id=	GUID	模板所属主机的集群标识。	
vm id=	GUID	模板所基于的虚拟机的标识。	
domain id=	GUID	模板域的标识。	
creation_time	**xsd:dateTime** 格式：**YYYY-MM-DDThh:mm:ss**	模板被创建的日期和时间。	
origin	**rhev**、**ovirt**、**vmware** 或 **xen**	模板的原始系统。	
high_availability	复杂数据	如果需要在虚拟机或它所在的主机出现系统故障时自动重启，请把 **enabled** 设置为 **true**。一个 **priority** 项被用来控制虚拟机重启的顺序。	
display	复杂数据	显示的 **type**（**vnc** 或 **spice**）、端口以及显示器的数量。**allow_reconnect** 的值指定了客户端是否可以通过显示重新连接到机器上。	
stateless	布尔值：true 或 false	无状态模板会在系统启动时对磁盘做一个快照，并在关机时删除对系统所做的所有改变。这意味着每次对系统所做的改变会在系统重启后丢失。	
usb	复杂数据	定义虚拟机模板的 USB 策略。需要一个 **enabled** 项（布尔值）和一个 **type** 项（值是 **native** 或 **legacy**）。	

项	类型	描述	属性
placement_policy	复杂数据	设置虚拟机迁移的替换策略。需要一个默认的 **host=** 和一个 **affinity**（**migratable**、**user_migratable** 或 **pinned**）。如果没有设置 **host** 项的值，则意味着没有首选的主机。	
custom_properties	复杂数据	传递给脚本的一组用户自定义的环境变量。每个 **custom_property** 都包括 **name** 和 **value** 属性。**capabilities** 包括了有效值的列表。	
timezone	时区数据库格式：**Area/Location**	Windows 虚拟机模板在 Sysprep 中的时区设置。	
domain	复杂数据	Windows 虚拟机模板在 Sysprep 中的时区设置。需要 **domains** 集合中的一个 **name**。	

17.2. 虚拟机模板的 XML 表述

例 17.1. 虚拟机模板的 XML 表述

```
<template href="/api/templates/00000000-0000-0000-0000-000000000000"
  id="00000000-0000-0000-0000-000000000000">
    <actions>
        <link href="/api/templates/00000000-0000-0000-0000-
000000000000/export"
          rel="export"/>
    </actions>
    <name>Blank</name>
    <description>Blank template</description>
    <comment>Blank template</comment>
    <link href="/api/templates/00000000-0000-0000-0000-
000000000000/disks"
      rel="disks"/>
    <link href="/api/templates/00000000-0000-0000-0000-
000000000000/nics"
      rel="nics"/>
    <link href="/api/templates/00000000-0000-0000-0000-
000000000000/cdroms"
      rel="cdroms"/>
    <link href="/api/templates/00000000-0000-0000-0000-
000000000000/permissions"
      rel="permissions"/>
    <link href="/api/templates/00000000-0000-0000-0000-
000000000000/watchdogs"
      rel="watchdogs"/>
```

```
    <type>server</type>
    <status>
        <state>ok</state>
    </status>
    <memory>536870912</memory>
    <cpu>
        <topology sockets="1" cores="1"/>
        <architecture>X86_64<architecture/>
    </cpu>
    <cpu_shares>0</cpu_shares>
    <os type="rhel_6x64">
        <boot dev="hd"/>
        <boot dev="cdrom"/>;
    </os>
    <cluster id="00000000-0000-0000-0000-000000000000"
      href="/api/clusters/00000000-0000-0000-0000-000000000000"/>
    <creation_time>2010-08-16T14:24:29</creation_time>
    <origin>ovirt</origin>
    <high_availability>
        <enabled>true</enabled>
        <priority>100</priority>
    </high_availability>
    <display>
        <type>spice</type>
        <monitors>1</monitors>
        <single_qxl_pci>false</single_qxl_pci>
        <allow_override>true</allow_override>
        <smartcard_enabled>true</smartcard_enabled>
    </display>
    <stateless>false</stateless>
    <delete_protected>false</delete_protected>
    <sso>
      <methods>
        <method id="GUEST_AGENT">true</enabled>
      </methods>
    </sso>
    <usb>
        <enabled>true</enabled>
    </usb>
    <migration_downtime>-1</migration_downtime>
    <version>
      <base_template href="/api/templates/00000000-0000-0000-0000-
000000000000"
        id="00000000-0000-0000-0000-000000000000"/>
      <version_number>2</version_number>
      <version_name>RHEL65_TMPL_001</version_name>
    </version>
</template>
```

17.3. 方法

17.3.1. 创建新模板

创建新模板需要 **name** 和 **vm** 项。使用 **id** 属性或 **name** 项来指定 **vm**。

例 17.2. 通过虚拟机创建一个模板

```
POST /api/templates HTTP/1.1
Accept: application/xml
Content-type: application/xml

<template>
    <name>template1</name>
    <vm id="00000000-0000-0000-0000-000000000000"/>
</template>
```

17.3.2. 创建新的模板子版本

创建新的模板子版本需要新模板的 **name** 和 **vm** 项，以及新模板版本的 **base_template** 和 **version_name** 项。**base_template** 和 **version_name** 项必须在 **template** 项中的 **version** 中指定。使用 **id** 属性或 **name** 项来指定 **vm**。

例 17.3. 通过虚拟机创建一个模板子版本

```
POST /api/templates HTTP/1.1
Accept: application/xml
Content-type: application/xml

<template>
  <name>template1_001</name>
  <vm id="00000000-0000-0000-0000-000000000000"/>
  <version>
    <base_template id="00000000-0000-0000-0000-000000000000"/>
    <version_name>"template1_001"</version_name>
  </version>
</template>
```

17.3.3. 更新模板

在模板被创建后，**name**、**description**、**type**、**memory**、**cpu** **topology**、**os**、**high_availability**、**display**、**stateless**、**usb** 和 **timezone** 项可以被更新。

例 17.4. 更新一个带有 1 GB 内存的虚拟机模板

```
PUT /api/templates/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<template>
    <memory>1073741824</memory>
</template>
```

17.3.4. 更新模板子版本

当一个模板子版本被创建后，只有 **version_name** 项可以被更新。

例 17.5. 更新一个虚拟机模板子版本名

```
PUT /api/templates/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<template>
  <version>
    <version_name>template1_002</version_name>
  </version>
</template>
```

17.3.5. 删除模板

删除虚拟机模板需要一个 **DELETE** 请求。

例 17.6. 删除虚拟机模板

```
DELETE /api/templates/00000000-0000-0000-0000-000000000000 HTTP/1.1

HTTP/1.1 204 No Content
```

17.4. 操作

17.4.1. 导出模板操作

templates 集合包括一个 **export** 操作。

导出操作会把模板导出到一个 **export** 存储域。目标存储域需要使用 **storage_domain** 来指定。

如果在目标域中存在同名的虚拟机模板，导出操作会报告一个错误。把 **exclusive** 参数设置为 **true** 可以改变这个行为，它会把虚拟机模板导入到目标域中，并覆盖其中同名的虚拟机模板。

例 17.7. 把模板导出到一个导出存储域中

```
POST /api/templates/00000000-0000-0000-0000-000000000000/export HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
    <storage_domain id="00000000-0000-0000-0000-000000000000"/>
    <exclusive>true<exclusive/>
</action>
```

第 18 章 虚拟机池

18.1. 虚拟机池项

vmpools 集合提供了 Red Hat Enterprise Virtualization 环境中的虚拟机池的信息。API 用户可以通过从进入点 URI 获得的 **rel="vmpools"** 链接来访问这些信息。

以下表格包括了虚拟机池表述中的特定项的信息。

表 18.1. 虚拟机池项

项	类型	描述	属性
name	字符串	虚拟机池的名称。所有虚拟机池资源中的每个虚拟机池的 **name** 需要是唯一的。	
description	字符串	用户提供的、对虚拟机池的描述信息。	
link rel="permissions"	关系	虚拟机池权限的 **permissions** 子集合。	
size	整数	虚拟机池中的虚拟机数量。	
cluster id=	GUID	虚拟机池中的虚拟机所属的集群资源。	
template id=	GUID	虚拟机池中的虚拟机所基于的模板资源。	
prestarted_vms	整数	虚拟机池中的预启动虚拟机的数量。	
max_user_vms	整数	一个用户可以从虚拟机池中获得虚拟机的最大数量。	

重要

本节中所使用的 API 还处于试用阶段，今后它们可能会有变化。产品的向下兼容声明不适用于这些 API。

18.2. 虚拟机池的 XML 表述

例 18.1. 虚拟机池的 XML 表述

```
<vmpool href="/api/vmpools/2d2d5e26-1b6e-11e1-8cda-001320f76e8e">
  id="2d2d5e26-1b6e-11e1-8cda-001320f76e8e"
    <actions>
```

```
     <link href="/api/vmpools/2d2d5e26-1b6e-11e1-8cda-
001320f76e8e/allocatevm"
       rel="allocatevm"/>
    </actions>
    <name>VMPool1</name>
    <description>Virtual Machine Pool 1</description>
    <size>2</size>
    <cluster href="/api/clusters/99408929-82cf-4dc7-a532-9d998063fa95"/>
      id="99408929-82cf-4dc7-a532-9d998063fa95"
    <template href="/api/templates/00000000-0000-0000-0000-
000000000000"/>
      id="00000000-0000-0000-0000-000000000000"
    <prestarted_vms>0</prestarted_vms>
    <max_user_vms>1</max_user_vms>
</vmpool>
```

18.3. 方法

18.3.1. 创建虚拟机池

新虚拟机池需要 **name**、**cluster** 和 **template** 项。使用 **id** 属性或 **name** 项来指定 **cluster** 和 **template**。

例 18.2. 创建虚拟机池

```
POST /api/vmpools HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vmpool>
    <name>VM_Pool_A</name>
    <cluster href="/api/clusters/99408929-82cf-4dc7-a532-9d998063fa95"/>
      id="99408929-82cf-4dc7-a532-9d998063fa95"
    <template href="/api/templates/00000000-0000-0000-0000-
000000000000"/>
      id="00000000-0000-0000-0000-000000000000"
</vmpool>
```

18.3.2. 更新虚拟机池

在虚拟机池被创建后，**name**、**description**、**size**、**prestarted_vms** 和 **max_user_vms** 可以被更新。

例 18.3. 更新一个虚拟机池

```
PUT /api/vmpools/2d2d5e26-1b6e-11e1-8cda-001320f76e8e HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vmpool>
    <name>VM_Pool_B</name>
```

```
    <description>Virtual Machine Pool B</description>
    <size>3</size>
    <prestarted_vms>1</size>
    <max_user_vms>2</size>
</vmpool>
```

18.3.3. 删除虚拟机池

删除虚拟机池需要一个 **DELETE** 请求。

例 18.4. 删除一个虚拟机

```
DELETE /api/vmpools/2d2d5e26-1b6e-11e1-8cda-001320f76e8e HTTP/1.1

HTTP/1.1 204 No Content
```

18.4. 操作

18.4.1. 分配虚拟机操作

分配虚拟机操作会在虚拟机池中分配一个虚拟机。

例 18.5. 从虚拟机池中分配一个虚拟机

```
POST /api/vmpools/2d2d5e26-1b6e-11e1-8cda-001320f76e8e/allocatevm
HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

第 19 章 域

19.1. 域项

API 提供了通过使用 **domains** 集合从机构的目录服务中访问用户和组信息的功能。域信息通过 **rel="domains"** 链接进行标识。

表 19.1. 域项

项	类型	描述
name	字符串	域名。
link rel="users"	关系	与域相关的用户子集合的链接。
link rel="groups"	关系	与域相关的组子集合的链接。

到 **users** 和 **groups** 子集合的链接也支持搜索查询。

注意

domains 集合和它的子集合是只读的。

19.2. 域资源的 XML 表述

例 19.1. 域资源的 XML 表述

```
<domain id="77696e32-6b38-7268-6576-2e656e676c61"
  href="/api/domains/77696e32-6b38-7268-6576-2e656e676c61">
    <name>domain.example.com</name>
    <link rel="users"
      href="/api/domains/77696e32-6b38-7268-6576-2e656e676c61/users"/>
    <link rel="groups"
      href="/api/domains/77696e32-6b38-7268-6576-2e656e676c61/groups"/>
    <link rel="users/search"
      href="/api/domains/77696e32-6b38-7268-6576-2e656e676c61/
      users?search={query}"/>
    <link rel="groups/search"
      href="/api/domains/77696e32-6b38-7268-6576-2e656e676c61/
      groups?search={query}"/>
</domain>
```

19.3. 子集合

19.3.1. 域用户子集合

users 子集合包括了目录服务中的所有用户。这些信息被用来把新用户添加到 Red Hat Enterprise Virtualization 环境中。

表 19.2. 域用户项

项	类型	描述
name	字符串	用户名。
last_name	字符串	用户的姓。
user_name	字符串	目录服务中的用户名。
domain id	GUID	提供目录服务的域。
groups	复杂数据	这个用户的目录服务组。

例 19.2. 用户子集合中的一个用户的 XML 表述

```
<user id="225f15cd-e891-434d-8262-a66808fcb9b1"
  href="/api/domains/77696e32-6b38-7268-6576-2e656e676c61/users/
  d3b4e7be-5f57-4dac-b937-21e1771a501f">
    <name>RHEV-M Admin</name>
    <user_name>rhevmadmin@domain.example.com</user_name>
    <domain id="77696e32-6b38-7268-6576-2e656e676c61"
      href="/api/domains/77696e32-6b38-7268-6576-2e656e676c61"/>
    <groups>
        <group>
            <name>domain.example.com/Users/Enterprise Admins</name>
        </group>
        <group>
            <name>domain.example.com/Users/Domain Admins</name>
        </group>
        ...
    </groups>
</user>
```

19.3.2. 域组子集合

groups 子集合包括了目录服务中的所有组。一个域 **group** 资源包括了一组项。

表 19.3. 域组项

项	类型	描述
name	字符串	组名。
domain id	GUID	提供目录服务的域。

例 19.3. 组子集合中的一个组的 XML 表述

```
<group id="85bf8d97-273c-4a5c-b801-b17d58330dab"
```

```
  href="/api/domains/77696e32-6b38-7268-6576-2e656e676c61/groups/
  85bf8d97-273c-4a5c-b801-b17d58330dab">
    <name>example.com/Users/Enterprise Admins</name>
    <domain id="77696e32-6b38-7268-6576-2e656e676c61"
      href="/api/domains/77696e32-6b38-7268-6576-2e656e676c61"/>
</group>
```

第 20 章 组

20.1. 导入的组项

groups 集合包括了从目录服务中导入的组。一个 **group** 资源包括了一组项。

表 20.1. 导入的组项

项	类型	描述
link rel="tags"	关系	附加到这个组的标签的 **tags** 子集合。
link rel="permissions"	关系	和这个组相关的权限 **permissions** 子集合链接。
link rel="roles"	关系	和这个组相关的角色的 **roles** 子集合链接。

20.2. 组资源的 XML 表述

例 20.1. 组资源的 XML 表述

```
<group id="85bf8d97-273c-4a5c-b801-b17d58330dab"
  href="/api/groups/85bf8d97-273c-4a5c-b801-b17d58330dab">
    <name>Everyone</name>
    <link rel="tags"
      href="/api/groups/85bf8d97-273c-4a5c-b801-b17d58330dab/tags"/>
    <link rel="permissions"
      href="/api/groups/85bf8d97-273c-4a5c-b801-
b17d58330dab/permissions"/>
    <link rel="roles"
      href="/api/groups/85bf8d97-273c-4a5c-b801-b17d58330dab/roles"/>
    <domain_entry_id>
          656565303030302D303030302D303030302D303030
    </domain_entry_id>
    <namespace>*</namespace>
</group>
```

20.3. 从目录服务添加组

API 使用一个对 **groups** 的 **POST** 请求来把存在的目录服务组添加到 Red Hat Enterprise Virtualization Manager 数据库中。

例 20.2. 从目录服务中添加一个组

```
POST /api/group HTTP/1.1
Content-Type: application/xml
Accept: application/xml

<group>
    <name>www.example.com/accounts/groups/mygroup</name>
```

```
    <domain>
     <name>example.com</name>
    </domain>
</group>
```

第 21 章 角色

21.1. 角色项

进入点 URI 所提供的 **rel="roles"** 链接可以用来访问系统的一组角色。每个 **role** 项都包括以下项：

表 21.1. 角色项

项	类型	描述	属性
link="permits"	关系	角色权限的 **permits** 子集合链接。	
mutable	布尔值：true 或 false	定义是否可以更新或删除角色。**mutable** 被设置为 **false** 的角色是内建到 Red Hat Enterprise Virtualization 环境中的角色。	
administrative	布尔值：true 或 false	指定角色是否是一个管理角色。	

21.2. 角色集合的 XML 表述

例 21.1. 角色集合的 XML 表述

```
<roles>
    <role id="00000000-0000-0000-0000-000000000001"
      href="/api/roles/00000000-0000-0000-0000-000000000001">
        <name>SuperUser</name>
        <description>Roles management administrator</description>
        <link rel="permits"
          href="/api/roles/00000000-0000-0000-0000-
000000000001/permits"/>
        <mutable>false</mutable>
        <administrative>true</administrative>
    </role>
    <role id="00000000-0000-0000-0001-000000000001"
      href="/api/roles/00000000-0000-0000-0001-000000000001">
        <name>RHEVMUser</name>
        <description>RHEVM user</description>
        <link rel="permits"
          href="/api/roles/00000000-0000-0000-0001-
000000000001/permits"/>
        <mutable>false</mutable>
        <administrative>false</administrative>
    </role>
    <role id="00000000-0000-0000-0001-000000000002"
       href="/api/roles/00000000-0000-0000-0001-000000000002">
        <name>RHEVMPowerUser</name>
        <description>RHEVM power user</description>
        <link rel="permits"
          href="/api/roles/00000000-0000-0000-0001-
000000000002/permits"/>
```

```
        <mutable>false</mutable>
        <administrative>false</administrative>
    </role>
</roles>
```

21.3. 方法

21.3.1. 创建角色

创建角色需要 **name**、**administrative** 和一组初始的 **permits**。

例 21.2. 创建一个角色

```
POST /api/roles HTTP/1.1
Accept: application/xml
Content-type: application/xml

<role>
    <name>Finance Role</name>
    <administrative>true</administrative>
    <permits>
        <permit id="1"/>
    </permits>
</role>
```

21.3.2. 更新角色

在创建完成后，**name**、**description** 和 **administrative** 项可以被更新。

例 21.3. 更新一个角色

```
PUT /api/roles/8de42ad7-f307-408b-80e8-9d28b85adfd7 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<role>
    <name>Engineering Role</name>
    <description>Standard users in the Engineering Role</description>
    <administrative>false</administrative>
</role>
```

21.3.3. 删除角色

删除角色需要一个 **DELETE** 请求。

例 21.4. 删除一个角色

```
DELETE /api/roles/8de42ad7-f307-408b-80e8-9d28b85adfd7 HTTP/1.1

HTTP/1.1 204 No Content
```

21.4. 角色权限子集合

21.4.1. 角色权限子集合

每个角色都包括了一组允许的操作或 **permits**，这些信息包括在 API 的 **capabilities** 中。

角色的 **permits** 作为子集合被列出。

例 21.5. 列出角色的权限

```
GET /api/roles/b67dfbe2-0dbc-41e4-86d3-a2fbef02cfa9/permits HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<permits>
    <permit id="1"
      href="/api/roles/b67dfbe2-0dbc-41e4-86d3-a2fbef02cfa9/permits/1">
        <name>create_vm</name>
        <administrative>false</administrative>
        <role id="b67dfbe2-0dbc-41e4-86d3-a2fbef02cfa9"
          href="/api/roles/b67dfbe2-0dbc-41e4-86d3-a2fbef02cfa9"/>
    </permit>
    ...
</permits>
```

21.4.2. 为角色分配权限

到 **permits** 子集合的 **POST** 请求可以为角色分配 **permit**。使用 **id** 属性或 **name** 项来指定要分配的 **permit**。

例 21.6. 为角色分配一个权限

```
POST /api/roles/b67dfbe2-0dbc-41e4-86d3-a2fbef02cfa9/permits HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<permit id="1"/>

HTTP/1.1 201 Created
Content-Type: application/xml

<permits>
    <permit id="1"
      href="/api/roles/b67dfbe2-0dbc-41e4-86d3-a2fbef02cfa9/permits/1">
```

```
        <name>create_vm</name>
        <administrative>false</administrative>
        <role id="b67dfbe2-0dbc-41e4-86d3-a2fbef02cfa9"
          href="/api/roles/b67dfbe2-0dbc-41e4-86d3-a2fbef02cfa9"/>
    </permit>
</permits>
```

21.4.3. 为角色删除权限

使用到 **permit** 资源的 **DELETE** 请求来为角色删除一个权限。

例 21.7. 为角色删除权限

```
DELETE /api/roles/b67dfbe2-0dbc-41e4-86d3-a2fbef02cfa9/permits/1
HTTP/1.1

HTTP/1.1 204 No Content
```

第 22 章 用户

22.1. 用户项

用户被包括在顶级集合中，并使用 **rel="users"** 链接进行标识。每个 **user** 项包括以下项：

表 22.1. 用户项

项	类型	描述	属性
user_name	字符串	UPN（User Principal Name - 用户主体名称）。为方便起见，在添加新用户时使用 UPN。	
link rel="tags"	关系	用户资源的 **tags** 子集合。	
link rel="roles"	关系	用户资源的 **roles** 子集合链接。	
name	字符串	用户名。	
domain	字符串	提供目录服务的域。	
groups	复杂数据	这个用户的目录服务组。	

22.2. 用户资源的 XML 表述

例 22.1. 用户资源的 XML 表述

```
GET /api/users HTTP/1.1
Accept: application/xml

<user id="225f15cd-e891-434d-8262-a66808fcb9b1"
  href="/api/users/225f15cd-e891-434d-8262-a66808fcb9b1">
    <name>RHEV-M Admin</name>
    <actions/>
    <link rel="roles"
      href="/api/users/225f15cd-e891-434d-8262-a66808fcb9b1/roles"/>
    <link rel="tags"
      href="/api/users/225f15cd-e891-434d-8262-a66808fcb9b1/tags"/>
    <domain>domain.example.com</domain>
    <logged_in>false</logged_in>
    <user_name>rhevmadmin@domain.example.com</user_name>
    <groups>
        <group>Group Policy Creator
Owners@domain.example.com/Users</group>
        <group>Domain Admins@domain.example.com/Users</group>
        <group>Enterprise Admins@domain.example.com/Users</group>
```

```
        <group>Schema Admins@domain.example.com/Users</group>
        <group>Administrators@domain.example.com/Builtin</group>
    </groups>
</user>
```

22.3. 方法

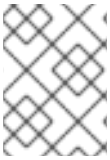
22.3.1. 添加用户

通过使用对 **users** 集合的 **POST** 请求，API 可以把存在的目录服务用户添加到 Red Hat Enterprise Virtualization Manager 数据库中。客户端所提供的新用户表述需要包括一个 **roles** 列表，以及最少一个分配给用户的初始 **role**。例如，下面的实例为用户 **joe@domain.example.com** 分配了两个初始角色：

例 22.2. 从目录服务中添加一个用户，并为用户分配两个角色

```
POST /api/users HTTP/1.1
Content-Type: application/xml
Accept: application/xml

<user>
    <user_name>joe@domain.example.com</user_name>
    <roles>
        <role>
            <name>RHEVMPowerUser</name>
        </role>
        <role id="00000000-0000-0000-0001-000000000003"/>
    </roles>
</user>
```

新用户可以使用 Red Hat Enterprise Virtualization Manager 用户 ID 或目录服务的 UPN 来代表。目录服务域中的用户 ID 格式可能和 Red Hat Enterprise Virtualization Manager 所希望获得的格式不同，例如，LDIF [5]，它的 ID 的字节顺序是相反的，并基于 base-64 编码。因此，为了方便起见，一般使用 UPN 来代表新用户。



注意

用户在被加入到 Red Hat Enterprise Virtualization Manager 数据库前，已经存在于目录服务域中。API 用户可以在创建用户前通过 **domains** 集合对它们进行查询。

角色可以由名称或 ID 来代表。以上的实例显示了这两种方法。

22.3.2. 为用户添加角色

使用到一个用户的角色子集合的 **POST** 或 **DELETE** 请求可以为用户添加或删除角色。以下实例显示了 API 如何为一个用户添加一个 **RHEVMVDIUser** 角色。

注意

内建的 **user** 项用户角色列表只在初始创建时使用，以后的所有为用户分配角色的操作都是通过 **roles** 子集合进行的。

例 22.3. 为一个用户添加角色

```
POST /api/users/225f15cd-e891-434d-8262-a66808fcb9b1/roles HTTP/1.1
Content-Type: application/xml
Accept: application/xml

<role>
    <name>RHEVMVDIUser</name>
</role>
```

[5] RFC 2849 中包括了对 LDAP Data Interchange Format （LDIF）的描述。

第 23 章 标签

23.1. 标签项

tags 集合提供了 Red Hat Enterprise Virtualization 环境中的标签的信息。API 用户可以通过从进入点 URI 获得的 **rel="tags"** 链接来访问这些信息。

以下表格包括了标签资源表述中的特定项的信息。

表 23.1. 标签项

元素	项	描述	属性
host	GUID	标签附加到的主机的标识。	
user	GUID	标签附加到的用户的标识。	
vm	GUID	标签附加到的虚拟机的标识。	
parent	复杂数据	标签附加到的虚拟机的标识。	

23.2. 标签资源的 XML 表述

例 23.1. 标签资源的 XML 表述

```
<tag id="f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e"
  href="/api/tags/f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e">
    <name>Finance</name>
    <description>Resources for the Finance department</description>
    <parent>
        <tag id="-1" href="/api/tags/-1"/>
    </parent>
</tag>
```

23.3. 关联标签

23.3.1. 为主机、用户或虚拟机附加标签

host、**user** 或 **vms** 中的 **link rel="tags"** 代表了附加到它们上面的标签。

这些 **tag** 表述也包括了一个 **host id**、**user id** 或 **vm id** 来代表附加了这个标签的主机、用户或虚拟机。

通过对某个对象的集合发送一个带有标签标识（标签 **id** 或 **name**）的 **POST** 请求来为这个对象附加标签。

例 23.2. 为一个虚拟机附加标签

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/tags HTTP/1.1
Accept: application/xml
Content-Type: application/xml

<tag>
    <name>Finance</name>
</tag>

HTTP/1.1 201 Created
Content-Type: application/xml

<tag id="f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e"
  href="/api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/tags/
  f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e">
    <name>Finance</name>
    <description>Resources for the Finance department</description>
    <vm id="5114bb3e-a4e6-44b2-b783-b3eea7d84720"
      href="/api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720"/>
</tag>
```

23.3.2. 删除标签

使用对集合中的相应项的 **DELETE** 请求删除和这个项相关联的标签。

例 23.3. 从虚拟机上删除标签

```
DELETE /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/tags/f436ebfc-67f2-
41bd-8ec6-902b6f7dcb5e HTTP/1.1

HTTP/1.1 204 No Content
```

23.3.3. 对带有标签的资源进行查询

要对附加了特定标签的资源进行查询，可以使用相应集合的 **collection/search** URI 模板，搜索的条件是 **tag=MyTag**。

例 23.4. 对带有标签的资源进行查询

```
GET /api/vms?search=tag%3DFinance HTTP/1.1
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<vms>
    <vm id="5114bb3e-a4e6-44b2-b783-b3eea7d84720"
      href="/api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720">
        ...
```

```
    </vm>
    ...
</vms>
```

23.4. 父标签

23.4.1. 父标签

API 用户通过为标签分配一个 **parent** 项来为父标签创建一个分级结构链接。所有标签在 **root** 标签下面以一个平面的结构存在，而标签会包括一个到父标签的链接项。

注意

在没有指定父标签时，**root** 标签被假设为默认的父标签。**root** 标签不能被删除，也不能为它分配一个父标签。

标签的分级结构以以下形式存在：

例 23.5. 标签分级结构

```
<tags>
    <tag id="-1" href="/api/tags/-1">
        <name>root</name>
        <description>root</description>
        <parent>
            <tag id="-1" href="/api/tags/-1"/>
        </parent>
    </tag>
    <tag id="f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e"
      href="/api/tags/f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e">
        <name>Finance</name>
        <description>Resources for the Finance department</description>
        <parent>
            <tag id="-1" href="/api/tags/-1"/>
        </parent>
    </tag>
    <tag id="ac18dabf-23e5-12be-a383-a38b165ca7bd"
      href="/api/tags/ac18dabf-23e5-12be-a383-a38b165ca7bd">
        <name>Billing</name>
        <description>Billing Resources</description>
        <parent>
            <tag id="f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e"
              href="/api/tags/f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e"/>
        </parent>
    </tag>
</tags>
```

在 XML 表述中，标签有以下分级结构：

```
root              (id: -1)
  - Finance       (id: f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e)
      - Billing   (id: ac18dabf-23e5-12be-a383-a38b165ca7bd)
```

23.4.2. 设置一个父标签

使用带有 **parent** 项的 **POST** 请求可以为新标签分配一个父标签。使用 **id** 属性或 **name** 项来代表父标签。

例 23.6. 使用 id 属性设置一个父标签

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/tags HTTP/1.1
Accept: application/xml
Content-Type: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<tag>
    <name>Billing</name>
    <description>Billing Resources</description>
    <parent>
        <tag id="f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e"/>
    </parent>
</tag>
```

例 23.7. 使用 name 项设置一个父标签

```
POST /api/vms/5114bb3e-a4e6-44b2-b783-b3eea7d84720/tags HTTP/1.1
Accept: application/xml
Content-Type: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<tag>
    <name>Billing</name>
    <description>Billing Resources</description>
    <parent>
        <tag>
            <name>Finance</name>
        </tag>
    </parent>
</tag>
```

23.4.3. 修改父标签

使用 **PUT** 请求可以修改父标签：

例 23.8. 修改父标签

```
PUT /api/tags/ac18dabf-23e5-12be-a383-a38b165ca7bd HTTP/1.1
```

```
Accept: application/xml
Content-Type: application/xml

<tag>
    <parent>
        <tag id="f436ebfc-67f2-41bd-8ec6-902b6f7dcb5e"/>
    </parent>
</tag>
```

第 24 章 事件

24.1. 事件项

从进入点 URI 获得的 **events** 集合可以被用来访问 **events**，并列出 Red Hat Enterprise Virtualization Manager 中的事件。

表 24.1. 事件项

项	类型	描述
description	字符串	系统事件描述
code	整数	以整数代表的事件代码。
severity	**normal**、**warning**、**error** 或 **alert**	事件的严重性级别。
time	**xsd:dateTime** 格式：**YYYY-MM-DDThh:mm:ss**	事件发生时的时间戳。
user id	GUID	触发事件的用户的 ID 代码。

注意

events 集合是只读的。

24.2. 事件集合的 XML 表述

例 24.1. 事件集合的 XML 表述

```
<events>
    <event id="537" href="/api/events/537">
        <description>User vdcadmin logged in.</description>
        <code>30</code>
        <severity>normal</severity>
        <time>2011-01-12T10:48:27.827+02:00</time>
        <user id="9b9002d1-ec33-4083-8a7b-31f6b8931648"
          href="/api/users/9b9002d1-ec33-4083-8a7b-31f6b8931648"/>
    </event>
    ...
</events>
```

24.3. 虚拟机创建事件的 XML 表述

除了 **user**，**event** 表述还包括了一组 XML 项，它们代表了和事件相关资源的关系。

例 24.2. 一个虚拟机创建事件的 XML 表述

```
<event id="635" href="/api/events/635">
    <description>VM bar was created by rhevadmin.</description>
    <code>34</code>
    <severity>normal</severity>
    <time>2011-07-11T16:32:03.172+02:00</time>
    <user id="4621b611-43eb-4d2b-ae5f-1180850268c4"
      href="/api/users/4621b611-43eb-4d2b-ae5f-1180850268c4"/>
    <vm id="9b22d423-e16b-4dd8-9c06-c8e9358fbc66"
      href="/api/vms/9b22d423-e16b-4dd8-9c06-c8e9358fbc66"/>
    <storage_domain id="a8a0e93d-c570-45ab-9cd6-3c68ab31221f"
      href="/api/storagedomains/a8a0e93d-c570-45ab-9cd6-3c68ab31221f"/>
</event>
```

以上实例包括了一个 XML 项，它代表了这个事件与虚拟机资源和存储域资源的关系。

24.4. 搜索事件

events 集合提供了和其它资源相似的搜索查询。在对 **events** 集合进行搜索时，可以只对特定的事件进行搜索，它会查询从一个特定事件后发生的所有事件。

从一个事件开始查询需要在搜索查询前添加一个额外的 **from** 参数。**from** 的参数值代表了一个事件的 **id** 代码。

例 24.3. 从一个事件开始搜索

```
GET /api/events;from=1012?search=type%3D30 HTTP/1.1
Accept: application/xml
```

显示事件 **id="1012"** 后 **type** 为 30 的所有事件

```
HTTP/1.1 200 OK
Content-Type: application/xml
<events>
    <event id="1018" href="/api/events/1018">
        <description>User admin logged in.</description>
        <code>30</code>
        <severity>normal</severity>
        <time>2011-07-11T14:03:22.485+10:00</time>
        <user id="80b71bae-98a1-11e0-8f20-525400866c73"
          href="/api/users/80b71bae-98a1-11e0-8f20-525400866c73"/>
    </event>
    <event id="1016" href="/api/events/1016">
        <description>User admin logged in.</description>
        <code>30</code>
        <severity>normal</severity>
        <time>2011-07-11T14:03:07.236+10:00</time>
        <user id="80b71bae-98a1-11e0-8f20-525400866c73"
          href="/api/users/80b71bae-98a1-11e0-8f20-525400866c73"/>
    </event>
    <event id="1014" href="/api/events/1014">
        <description>User admin logged in.</description>
```

```
        <code>30</code>
        <severity>normal</severity>
        <time>2011-07-11T14:02:16.009+10:00</time>
        <user id="80b71bae-98a1-11e0-8f20-525400866c73"
          href="/api/users/80b71bae-98a1-11e0-8f20-525400866c73"/>
    </event>
</events>
```

例 24.4. 使用一个特定的事件严重性进行搜索

```
GET /api/events?search=severity>normal HTTP/1.1
Accept: application/xml
```

这会显示所有严重性高于 **normal** 的事件。严重性级别包括 **normal**、**warning**、**error** 和 **alert**。

```
HTTP/1.1 200 OK
Content-Type: application/xml
<events>
    <event id="2823" href="/api/events/2823">
        <description>Host Host-05 has time-drift of 36002 seconds while
maximum configured value is 300 seconds.</description>
        <code>604</code>
        <severity>warning</severity>
        <time>2015-07-11T14:03:22.485+10:00</time>
        <host href= "/api/hosts/44e52bb2-27d6-4d35-8038-0c4b4db89789"
id="44e52bb2-27d6-4d35-8038-0c4b4db89789"/>
        <cluster href= "/api/clusters/00000001-0001-0001-0001-
00000000021b" id="00000001-0001-0001-0001-00000000021b"/>
        <origin>oVirt</origin>
        <custom_id>-1</custom_id>
        <flood_rate>30</flood_rate>
    </event>
...
</events>
```

24.5. 对事件进行分页显示

虚拟环境会在一定时间内产生大量事件，而 API 只会为每个搜索查询显示默认数量的事件。要显示比默认数量更多的事件，可以在搜索查询中使用 **page** 命令，API 会根据这个命令把搜索结果进行分页。

以下的搜索查询使用一个 **page** 值和 **sortby** 子句告诉 API 对结果进行分页显示：

```
sortby time asc page 1
```

sortby 子句指定了基于哪个项对结果进行排序，以及排序的顺序（升序或降序）。在 **events** 搜索查询中，把排序所基于的项设置为 **time**，把排序的顺序设置为升序（**asc**），API 会显示您的虚拟环境被创建后的所有事件。

page 定义了页的数量，而每一页会包括默认数量的事件。分页从 **page 1** 开始。增加 **page** 的值可以显示更多页：

sortby time asc page 2

sortby time asc page 3

sortby time asc page 4

例 24.5. 对事件进行分页显示

这个实例对 **event** 资源进行分页。URL 编码的请求是：

```
GET /api/events?search=sortby%20time%20asc%20page%201 HTTP/1.1
Accept: application/xml
```

增加 **page** 的值来查看结果的下一页：

```
GET /api/events?search=sortby%20time%20asc%20page%202 HTTP/1.1
Accept: application/xml
```

使用 **from** 参数指定开始事件的 **id**。

```
GET /api/events?search=sortby%20time%20asc%20page%202&from=30 HTTP/1.1
Accept: application/xml
```

附录 A. 在 CURL 中使用 API

A.1. 在 CURL 中使用 API

本附录介绍了如何在 **cURL** 中使用 REST 请求。**cURL** 是一个命令行工具，可以用来进行包括 HTTP 在内的跨协议和跨平台（如 Linux、Windows、Mac 和 Solaris）的数据传输。多数 Linux 版本中都包括 **cURL** 软件包。

A.2. 安装 CURL

Red Hat Enterprise Linux 用户可以使用以下命令安装 **cURL**：

```
yum install curl
```

如需在其它平台上安装，请参阅 **cURL** 网站（http://curl.haxx.se/）上的相应信息。

A.3. 使用 CURL

cURL 使用一个命令行接口向 HTTP 服务器发送请求。它的命令语法是：

```
Usage: curl [options] uri
```

uri 代表请求发送到的目标 HTTP 地址。它是您的 Red Hat Enterprise Virtualization Manager 主机中包括 API 进入点路径（**/api**）所在的位置。

cURL 选项

-X *COMMAND*, --request *COMMAND*

要使用的请求命令。对于 REST API，使用 **GET**、**POST**、**PUT** 或 **DELETE**。

例如：**-X GET**

-H *LINE*, --header *LINE*

请求中包括的 HTTP 头数据（使用多个 header 选项可以指定多个头数据）。

例如：**-H "Accept: application/xml" -H "Content-Type: application/xml"**

-u *USERNAME:PASSWORD*, --user *USERNAME:PASSWORD*

Red Hat Enterprise Virtualization 用户的用户名和密码。这个信息可以替代 **Authorization:** 头数据。

例如：**-u admin@internal:p@55w0rd!**

--cacert *CERTIFICATE*

和 REST API 进行 SSL 通信所需要的证书文件位置。这个证书文件应该保存在客户端系统的本地。使用 **-k** 属性可以忽略 SSL。

例如：**--cacert ~/Certificates/rhevm.cer**

-d *BODY*, --data *BODY*

为请求发送的正文（body）。它可以和 **POST**、**PUT** 和 **DELETE** 请求一起使用。如果请求中包括了正文，请确认指定了 **Content-Type: application/xml** 头数据。

例如：**-d "<cdrom><file id='rhel-server-6.0-x86_64-dvd.iso'/></cdrom>"**

A.4. 实例

A.4.1. 在 cURL 中使用 GET 请求

例 A.1. GET 请求

以下 **GET** 请求列出了 **vms** 集合中的虚拟机。请注意，**GET** 请求不包括正文（body）项。

```
GET /api/vms HTTP/1.1
Accept: application/xml
```

在 **cURL** 命令中使用方法（**GET**）、头数据（**Accept: application/xml**）和 URI（**https://*[RHEVM-Host]*:443/api/vms**）：

```
$ curl -X GET -H "Accept: application/xml" -u [USER:PASS] --cacert
[CERT] https://[RHEVM-Host]:443/api/vms
```

一个 **vms** 集合的表述会被显示。

A.4.2. 在 cURL 中使用 POST 请求

例 A.2. POST 请求

以下 **POST** 请求会在 **vms** 集合中创建一个虚拟机。请注意，**POST** 请求需要一个正文（body）。

```
POST /api/vms HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm>
  <name>vm1</name>
  <cluster>
    <name>default</name>
  </cluster>
  <template>
    <name>Blank</name>
  </template>
  <memory>536870912</memory>
  <os>
    <boot dev="hd"/>
  </os>
</vm>
```

在 **cURL** 命令中使用方法（**POST**），头（**Accept: application/xml** 和 **Content-type: application/xml**）、URI（**https://*[RHEVM-Host]*:443/api/vms**）以及一个请求正文：

```
$ curl -X POST -H "Accept: application/xml" -H "Content-type:
application/xml" -u [USER:PASS] --cacert [CERT] -d "<vm>
<name>vm1</name><cluster><name>default</name></cluster><template>
<name>Blank</name></template><memory>536870912</memory><os><boot
dev='hd'/></os></vm>" https://[RHEVM-Host]:443/api/vms
```

REST API 创建一个新虚拟机，并显示资源的表述。

A.4.3. 在 cURL 中使用 PUT 请求

例 A.3. PUT 请求

以下 **PUT** 请求会更新一个虚拟机的内存。请注意，**PUT** 请求需要一个正文。

```
PUT /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<vm>
    <memory>1073741824</memory>
</vm>
```

在 **cURL** 命令中使用方法（**PUT**），标头（**Accept: application/xml** 和 **Content-type: application/xml**）、URI（**https://[RHEVM-Host]:443/api/vms/082c794b-771f-452f-83c9-b2b5a19c0399**）以及一个请求正文：

```
$ curl -X PUT -H "Accept: application/xml" -H "Content-type:
application/xml" -u [USER:PASS] --cacert [CERT] -d "<vm>
<memory>1073741824</memory></vm>" https://[RHEVM-
Host]:443//api/vms/082c794b-771f-452f-83c9-b2b5a19c039
```

REST API 更新了虚拟机的内存配置。

A.4.4. 在 cURL 中使用 DELETE 请求

例 A.4. DELETE 请求

以下 **DELETE** 请求会删除一个虚拟机资源。

```
DELETE /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399 HTTP/1.1
```

在 **cURL** 命令中使用方法（**DELETE**）和 URI（**https://[RHEVM-Host]:443/api/vms/082c794b-771f-452f-83c9-b2b5a19c0399**）：

```
$ curl -X DELETE -u [USER:PASS] --cacert [CERT] https://[RHEVM-
Host]:443//api/vms/082c794b-771f-452f-83c9-b2b5a19c039
```

REST API 删除了虚拟机。请注意，因为 **DELETE** 请求的结果是空，**Accept: application/xml** 对标头的需求是可选的。

A.4.5. 在 cURL 中使用带有正文的 DELETE 请求

例 A.5. 带有正文的 DELETE 请求

以下 **DELETE** 请求包括了一个正文来强制删除一个虚拟机。

```
DELETE /api/vms/082c794b-771f-452f-83c9-b2b5a19c0399 HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action>
  <force>true</force>
</action>
```

在 **cURL** 命令中使用方法（**DELETE**），标头（**Accept: application/xml** 和 **Content-type: application/xml**）、URI（**https://*[RHEVM-Host]*:443/api/vms/082c794b-771f-452f-83c9-b2b5a19c0399**）以及一个请求正文：

```
$ curl -X DELETE -H "Accept: application/xml" -H "Content-type:
application/xml" -u [USER:PASS] --cacert [CERT] -d "<action>
<force>true</force></action>" https://[RHEVM-
Host]:443//api/vms/082c794b-771f-452f-83c9-b2b5a19c039
```

REST API 强制删除虚拟机。

附录 B. 列举值的映射

B.1. 列举值的映射

API 使用 Red Hat Enterprise Virtualization 查询语言来进行搜索查询。如需了解更多关于查询语言的信息，请参阅 *Red Hat Enterprise Virtualization 管理指南* 中的 *在 Red Hat Enterprise Virtualization 中执行搜索* 一节。

请注意，API 中的特定列举值和搜索语言中的值会有所不同。以下表格提供了这些列举值的映射信息。

表 B.1. 列举值映射

资源类型	API 列举类型	API 列举值	查询语言属性	查询语言值
数据中心	`data_center_states`	`not_operational`	`status`	`notoperational`
主机	`host_states`	`non_responsive`	`status`	`nonresponsive`
		`install_failed`		`installfailed`
		`preparing_for_maintenance`		`preparingformaintenance`
		`non_operational`		`nonoperational`
		`pending_approval`		`pendingapproval`
虚拟机	`vm_states`	`powering_up`	`status`	`poweringup`
		`powering_down`		`poweringdown`
		`migrating`		`migratingfrom`
		`migrating`		`migratingto`
		`not_responding`		`notresponding`
		`wait_for_launch`		`waitforlaunch`

资源类型	API 列举类型	API 列举值	查询语言属性	查询语言值
		reboot_in_progress		rebootinprogress
		saving_state		savingstate
		restoring_state		restoringstate
		image_locked		imagelocked

附录 C. 事件代码

C.1. 事件代码

这个表格列出了所有事件代码。

表 C.1. 事件代码

代码	名称	严重性	信息
0	UNASSIGNED	Info	
1	VDC_START	Info	Starting oVirt Engine.
2	VDC_STOP	Info	Stopping oVirt Engine.
12	VDS_FAILURE	Error	Host ${VdsName} is non responsive.
13	VDS_DETECTED	Info	Status of host ${VdsName} was set to ${HostStatus}.
14	VDS_RECOVER	Info	Host ${VdsName} is rebooting.
15	VDS_MAINTENANCE	Normal	Host ${VdsName} was switched to Maintenance Mode.
16	VDS_ACTIVATE	Info	Activation of host ${VdsName} initiated by ${UserName}.
17	VDS_MAINTENANCE_FAILED	Error	Failed to switch Host ${VdsName} to Maintenance mode.
18	VDS_ACTIVATE_FAILED	Error	Failed to activate Host ${VdsName}.(User: ${UserName}).
19	VDS_RECOVER_FAILED	Error	Host ${VdsName} failed to recover.
20	USER_VDS_START	Info	Host ${VdsName} was started by ${UserName}.
21	USER_VDS_STOP	Info	Host ${VdsName} was stopped by ${UserName}.

代码	名称	严重性	信息
22	IRS_FAILURE	Error	Failed to access Storage on Host ${VdsName}.
23	VDS_LOW_DISK_SPACE	Warning	Warning, Low disk space. Host ${VdsName} has less than ${DiskSpace} MB of free space left on: ${Disks}.
24	VDS_LOW_DISK_SPACE_ERROR	Error	Critical, Low disk space. Host ${VdsName} has less than ${DiskSpace} MB of free space left on: ${Disks}. Low disk space might cause an issue upgrading this host.
25	VDS_NO_SELINUX_ENFORCEMENT	Warning	Host ${VdsName} does not enforce SELinux. Current status: ${Mode}
26	IRS_DISK_SPACE_LOW	Warning	Warning, Low disk space. ${StorageDomainName} domain has ${DiskSpace} GB of free space.
27	VDS_STATUS_CHANGE_FAILED_DUE_TO_STOP_SPM_FAILURE	Warning	Failed to change status of host ${VdsName} due to a failure to stop the spm.
28	VDS_PROVISION	Warning	Installing OS on Host ${VdsName} using Hostgroup ${HostGroupName}.
29	USER_ADD_VM_TEMPLATE_SUCCESS	Info	Template ${VmTemplateName} was created successfully.
31	USER_VDC_LOGOUT	Info	User ${UserName} logged out.
32	USER_RUN_VM	Info	VM ${VmName} started on Host ${VdsName}
33	USER_STOP_VM	Info	VM ${VmName} powered off by ${UserName} (Host: ${VdsName}) (Reason: ${Reason}).
34	USER_ADD_VM	Info	VM ${VmName} was created by ${UserName}.

代码	名称	严重性	信息
35	USER_UPDATE_VM	Info	VM ${VmName} configuration was updated by ${UserName}.
36	USER_ADD_VM_TEMPLATE_FAILURE	Error	Failed creating Template ${VmTemplateName}.
37	USER_ADD_VM_STARTED	Info	VM ${VmName} creation was initiated by ${UserName}.
38	USER_CHANGE_DISK_VM	Info	CD ${DiskName} was inserted to VM ${VmName} by ${UserName}.
39	USER_PAUSE_VM	Info	VM ${VmName} was suspended by ${UserName} (Host: ${VdsName}).
40	USER_RESUME_VM	Info	VM ${VmName} was resumed by ${UserName} (Host: ${VdsName}).
41	USER_VDS_RESTART	Info	Host ${VdsName} was restarted by ${UserName}.
42	USER_ADD_VDS	Info	Host ${VdsName} was added by ${UserName}.
43	USER_UPDATE_VDS	Info	Host ${VdsName} configuration was updated by ${UserName}.
44	USER_REMOVE_VDS	Info	Host ${VdsName} was removed by ${UserName}.
45	USER_CREATE_SNAPSHOT	Info	Snapshot '${SnapshotName}' creation for VM '${VmName}' was initiated by ${UserName}.
46	USER_TRY_BACK_TO_SNAPSHOT	Info	Snapshot-Preview ${SnapshotName} for VM ${VmName} was initiated by ${UserName}.
47	USER_RESTORE_FROM_SNAPSHOT	Info	VM ${VmName} restored from Snapshot by ${UserName}.

代码	名称	严重性	信息
48	USER_ADD_VM_TEMPLATE	Info	Creation of Template ${VmTemplateName} from VM ${VmName} was initiated by ${UserName}.
49	USER_UPDATE_VM_TEMPLATE	Info	Template ${VmTemplateName} configuration was updated by ${UserName}.
50	USER_REMOVE_VM_TEMPLATE	Info	Removal of Template ${VmTemplateName} was initiated by ${UserName}.
51	USER_ADD_VM_TEMPLATE_FINISHED_SUCCESS	Info	Creation of Template ${VmTemplateName} from VM ${VmName} has been completed.
52	USER_ADD_VM_TEMPLATE_FINISHED_FAILURE	Error	Failed to complete creation of Template ${VmTemplateName} from VM ${VmName}.
53	USER_ADD_VM_FINISHED_SUCCESS	Info	VM ${VmName} creation has been completed.
54	USER_FAILED_RUN_VM	Error	Failed to run VM ${VmName} (User: ${UserName}).
55	USER_FAILED_PAUSE_VM	Error	Failed to suspend VM ${VmName} (Host: ${VdsName}, User: ${UserName}).
56	USER_FAILED_STOP_VM	Error	Failed to power off VM ${VmName} (Host: ${VdsName}, User: ${UserName}).
57	USER_FAILED_ADD_VM	Error	Failed to create VM ${VmName} (User: ${UserName}).
58	USER_FAILED_UPDATE_VM	Error	Failed to update VM ${VmName} (User: ${UserName}).
59	USER_FAILED_REMOVE_VM	Error	

代码	名称	严重性	信息
60	USER_ADD_VM_FINISHED_FAILURE	Error	Failed to complete VM ${VmName} creation.
61	VM_DOWN	Info	VM ${VmName} is down. ${ExitMessage}
62	VM_MIGRATION_START	Info	Migration started (VM: ${VmName}, Source: ${VdsName}, Destination: ${DestinationVdsName}, User: ${UserName}).
63	VM_MIGRATION_DONE	Info	Migration completed (VM: ${VmName}, Source: ${VdsName}, Destination: ${DestinationVdsName}, Duration: ${Duration}, Total: ${TotalDuration}, Actual downtime: ${ActualDowntime})
64	VM_MIGRATION_ABORT	Error	Migration failed: ${MigrationError} (VM: ${VmName}, Source: ${VdsName}, Destination: ${DestinationVdsName}).
65	VM_MIGRATION_FAILED	Error	Migration failed${DueToMigrationError} (VM: ${VmName}, Source: ${VdsName}).
66	VM_FAILURE	Error	VM ${VmName} cannot be found on Host ${VdsName}.
67	VM_MIGRATION_START_SYSTEM_INITIATED	Info	Migration initiated by system (VM: ${VmName}, Source: ${VdsName}, Destination: ${DestinationVdsName}).
68	USER_CREATE_SNAPSHOT_FINISHED_SUCCESS	Info	Snapshot '${SnapshotName}' creation for VM '${VmName}' has been completed.
69	USER_CREATE_SNAPSHOT_FINISHED_FAILURE	Error	Failed to complete snapshot '${SnapshotName}' creation for VM '${VmName}'.

代码	名称	严重性	信息
70	USER_RUN_VM_AS_STATELESS_FINISHED_FAILURE	Error	Failed to complete starting of VM ${VmName}.
71	USER_TRY_BACK_TO_SNAPSHOT_FINISH_SUCCESS	Info	Snapshot-Preview ${SnapshotName} for VM ${VmName} has been completed.
72	USER_CHANGE_FLOPPY_VM	Info	Floppy ${DiskName} was inserted in VM ${VmName} by ${UserName}
73	USER_INITIATED_SHUTDOWN_VM	Info	VM shutdown initiated by ${UserName} on VM ${VmName} (Host: ${VdsName}) (Reason: ${Reason}).
74	USER_FAILED_SHUTDOWN_VM	Error	Failed to initiate shutdown on VM ${VmName} (Host: ${VdsName}, User: ${UserName}).
75	USER_FAILED_CHANGE_FLOPPY_VM	Error	Failed to change floppy ${DiskName} (User: ${UserName}).
76	USER_STOPPED_VM_INSTEAD_OF_SHUTDOWN	Info	VM ${VmName} was powered off ungracefully by ${UserName} (Host: ${VdsName}) (Reason: ${Reason}).
77	USER_FAILED_STOPPING_VM_INSTEAD_OF_SHUTDOWN	Error	Failed to power off VM ${VmName} (Host: ${VdsName}, User: ${UserName}).
78	USER_ADD_DISK_TO_VM	Info	Add-Disk operation of ${DiskAlias} was initiated on VM ${VmName} by ${UserName}.
79	USER_FAILED_ADD_DISK_TO_VM	Error	Add-Disk operation failed on VM ${VmName} (User: ${UserName}).
80	USER_REMOVE_DISK_FROM_VM	Info	Disk was removed from VM ${VmName} by ${UserName}.

代码	名称	严重性	信息
81	USER_FAILED_REMOVE_DISK_FROM_VM	Error	Failed to remove Disk from VM ${VmName} (User: ${UserName}).
82	USER_MOVED_VM	Info	VM ${VmName} moving to Domain ${StorageDomainName} was initiated by ${UserName}.
83	USER_FAILED_MOVE_VM	Error	Failed to initiate moving of VM ${VmName} to Domain ${StorageDomainName} (User: ${UserName}).
84	USER_MOVED_TEMPLATE	Info	Template ${VmTemplateName} moving to Domain ${StorageDomainName} was initiated by ${UserName}.
85	USER_FAILED_MOVE_TEMPLATE	Error	Failed to initiate moving Template ${VmTemplateName} to Domain ${StorageDomainName} (User: ${UserName}).
86	USER_COPIED_TEMPLATE	Info	Template ${VmTemplateName} copy to Domain ${StorageDomainName} was initiated by ${UserName}.
87	USER_FAILED_COPY_TEMPLATE	Error	Failed to initiate copy of Template ${VmTemplateName} to Domain ${StorageDomainName} (User: ${UserName}).
88	USER_UPDATE_VM_DISK	Info	VM ${VmName} ${DiskAlias} disk was updated by ${UserName}.
89	USER_FAILED_UPDATE_VM_DISK	Error	Failed to update VM ${VmName} disk ${DiskAlias} (User: ${UserName}).
90	VDS_FAILED_TO_GET_HOST_HARDWARE_INFO	Warning	Could not get hardware information for host ${VdsName}

代码	名称	严重性	信息
91	USER_MOVED_VM_FINISHED_SUCCESS	Info	Moving VM ${VmName} to Domain ${StorageDomainName} has been completed.
92	USER_MOVED_VM_FINISHED_FAILURE	Error	Failed to complete moving of VM ${VmName} to Domain ${StorageDomainName}.
93	USER_MOVED_TEMPLATE_FINISHED_SUCCESS	Info	Template ${VmTemplateName} moving to Domain ${StorageDomainName} has been completed.
94	USER_MOVED_TEMPLATE_FINISHED_FAILURE	Error	Failed to complete moving of Template ${VmTemplateName} to Domain ${StorageDomainName}.
95	USER_COPIED_TEMPLATE_FINISHED_SUCCESS	Info	Template ${VmTemplateName} copy to Domain ${StorageDomainName} has been completed.
96	USER_COPIED_TEMPLATE_FINISHED_FAILURE	Error	Failed to complete copy of Template ${VmTemplateName} to Domain ${StorageDomainName}.
97	USER_ADD_DISK_TO_VM_FINISHED_SUCCESS	Info	The disk ${DiskAlias} was successfully added to VM ${VmName}.
98	USER_ADD_DISK_TO_VM_FINISHED_FAILURE	Error	Add-Disk operation failed to complete on VM ${VmName}.
99	USER_TRY_BACK_TO_SNAPSHOT_FINISH_FAILURE	Error	Failed to complete Snapshot-Preview ${SnapshotName} for VM ${VmName}.
100	USER_RESTORE_FROM_SNAPSHOT_FINISH_SUCCESS	Info	VM ${VmName} restoring from Snapshot has been completed.
101	USER_RESTORE_FROM_SNAPSHOT_FINISH_FAILURE	Error	Failed to complete restoring from Snapshot of VM ${VmName}.

代码	名称	严重性	信息
102	USER_FAILED_CHANGE_DISK_VM	Error	Failed to change disk in VM ${VmName} (Host: ${VdsName}, User: ${UserName}).
103	USER_FAILED_RESUME_VM	Error	Failed to resume VM ${VmName} (Host: ${VdsName}, User: ${UserName}).
104	USER_FAILED_ADD_VDS	Error	Failed to add Host ${VdsName} (User: ${UserName}).
105	USER_FAILED_UPDATE_VDS	Error	Failed to update Host ${VdsName} (User: ${UserName}).
106	USER_FAILED_REMOVE_VDS	Error	Failed to remove Host ${VdsName} (User: ${UserName}).
107	USER_FAILED_VDS_RESTART	Error	Failed to restart Host ${VdsName}, (User: ${UserName}).
108	USER_FAILED_ADD_VM_TEMPLATE	Error	Failed to initiate creation of Template ${VmTemplateName} from VM ${VmName} (User: ${UserName}).
109	USER_FAILED_UPDATE_VM_TEMPLATE	Error	Failed to update Template ${VmTemplateName} (User: ${UserName}).
110	USER_FAILED_REMOVE_VM_TEMPLATE	Error	Failed to initiate removal of Template ${VmTemplateName} (User: ${UserName}).
111	USER_STOP_SUSPENDED_VM	Info	Suspended VM ${VmName} has had its save state cleared by ${UserName} (Reason: ${Reason}).
112	USER_STOP_SUSPENDED_VM_FAILED	Error	Failed to power off suspended VM ${VmName} (User: ${UserName}).

代码	名称	严重性	信息
113	USER_REMOVE_VM_FINISHED	Info	VM ${VmName} was successfully removed.
114	USER_VDC_LOGIN_FAILED	Error	User ${UserName} failed to log in.
115	USER_FAILED_TRY_BACK_TO_SNAPSHOT	Error	Failed to preview Snapshot ${SnapshotName} for VM ${VmName} (User: ${UserName}).
116	USER_FAILED_RESTORE_FROM_SNAPSHOT	Error	Failed to restore VM ${VmName} from Snapshot (User: ${UserName}).
117	USER_FAILED_CREATE_SNAPSHOT	Error	Failed to create Snapshot ${SnapshotName} for VM ${VmName} (User: ${UserName}).
118	USER_FAILED_VDS_START	Error	Failed to start Host ${VdsName}, (User: ${UserName}).
119	VM_DOWN_ERROR	Error	VM ${VmName} is down with error. ${ExitMessage}.
120	VM_MIGRATION_TO_SERVER_FAILED	Error	Migration failed${DueToMigrationError} (VM: ${VmName}, Source: ${VdsName}, Destination: ${DestinationVdsName}).
121	SYSTEM_VDS_RESTART	Info	Host ${VdsName} was restarted by the engine.
122	SYSTEM_FAILED_VDS_RESTART	Error	A restart initiated by the engine to Host ${VdsName} has failed.
123	VDS_SLOW_STORAGE_RESPONSE_TIME	Warning	Slow storage response time on Host ${VdsName}.
124	VM_IMPORT	Info	Started VM import of ${ImportedVmName} (User: ${UserName})

代码	名称	严重性	信息
125	VM_IMPORT_FAILED	Error	Failed to import VM ${ImportedVmName} (User: ${UserName})
126	VM_NOT_RESPONDING	Warning	VM ${VmName} is not responding.
127	VDS_RUN_IN_NO_KVM_MODE	Error	Host ${VdsName} running without virtualization hardware acceleration
128	VM_MIGRATION_TRYING_RERUN	Warning	Failed to migrate VM ${VmName} to Host ${DestinationVdsName}${DueToMigrationError}. Trying to migrate to another Host.
129	VM_CLEARED	Info	Unused
130	USER_SUSPEND_VM_FINISH_FAILURE_WILL_TRY_AGAIN	Error	Failed to complete suspending of VM ${VmName}, will try again.
131	USER_EXPORT_VM	Info	VM ${VmName} exported to ${ExportPath} by ${UserName}
132	USER_EXPORT_VM_FAILED	Error	Failed to export VM ${VmName} to ${ExportPath} (User: ${UserName})
133	USER_EXPORT_TEMPLATE	Info	Template ${VmTemplateName} exported to ${ExportPath} by ${UserName}
134	USER_EXPORT_TEMPLATE_FAILED	Error	Failed to export Template ${VmTemplateName} to ${ExportPath} (User: ${UserName})
135	TEMPLATE_IMPORT	Info	Started Template import of ${ImportedVmTemplateName} (User: ${UserName})
136	TEMPLATE_IMPORT_FAILED	Error	Failed to import Template ${ImportedVmTemplateName} (User: ${UserName})

代码	名称	严重性	信息
137	USER_FAILED_VDS_STOP	Error	Failed to stop Host ${VdsName}, (User: ${UserName}).
138	VM_PAUSED_ENOSPC	Error	VM ${VmName} has been paused due to no Storage space error.
139	VM_PAUSED_ERROR	Error	VM ${VmName} has been paused due to unknown storage error.
140	VM_MIGRATION_FAILED_DURING_MOVE_TO_MAINTENANCE	Error	Migration failed${DueToMigrationError} while Host is in 'preparing for maintenance' state.\n Consider manual intervention\: stopping/migrating Vms as Host's state will not\n turn to maintenance while VMs are still running on it.(VM: ${VmName}, Source: ${VdsName}, Destination: ${DestinationVdsName}).
141	VDS_VERSION_NOT_SUPPORTED_FOR_CLUSTER	Error	Host ${VdsName} is installed with VDSM version (${VdsSupportedVersions}) and cannot join cluster ${VdsGroupName} which is compatible with VDSM versions ${CompatibilityVersion}.
142	VM_SET_TO_UNKNOWN_STATUS	Warning	VM ${VmName} was set to the Unknown status.
143	VM_WAS_SET_DOWN_DUE_TO_HOST_REBOOT_OR_MANUAL_FENCE	Info	Vm ${VmName} was shut down due to ${VdsName} host reboot or manual fence
144	VM_IMPORT_INFO	Info	Value of field ${FieldName} of imported VM ${VmName} is ${FieldValue}. The field is reset to the default value
145	VM_PAUSED_EIO	Error	VM ${VmName} has been paused due to storage I/O problem.

代码	名称	严重性	信息
146	VM_PAUSED_EPERM	Error	VM ${VmName} has been paused due to storage permissions problem.
147	VM_POWER_DOWN_FAILED	Warning	Shutdown of VM ${VmName} failed.
148	VM_MEMORY_UNDER_GUARANTEED_VALUE	Error	VM ${VmName} on host ${VdsName} was guaranteed ${MemGuaranteed} MB but currently has ${MemActual} MB
149	USER_ADD	Info	User '${NewUserName}' was added successfully to the system.
150	USER_INITIATED_RUN_VM	Info	Starting VM ${VmName} was initiated by ${UserName}.
151	USER_INITIATED_RUN_VM_FAILED	Warning	Failed to run VM ${VmName} on Host ${VdsName}.
152	USER_RUN_VM_ON_NON_DEFAULT_VDS	Warning	Guest ${VmName} started on Host ${VdsName}. (Default Host parameter was ignored - assigned Host was not available).
153	USER_STARTED_VM	Info	VM ${VmName} was started by ${UserName} (Host: ${VdsName}).
154	VDS_CLUSTER_VERSION_NOT_SUPPORTED	Error	Host ${VdsName} is compatible with versions (${VdsSupportedVersions}) and cannot join Cluster ${VdsGroupName} which is set to version ${CompatibilityVersion}.
155	VDS_ARCHITECTURE_NOT_SUPPORTED_FOR_CLUSTER	Error	Host ${VdsName} has architecture ${VdsArchitecture} and cannot join Cluster ${VdsGroupName} which has architecture ${VdsGroupArchitecture}.

代码	名称	严重性	信息
156	CPU_TYPE_UNSUPPORTED_IN_THIS_CLUSTER_VERSION	Error	Host ${VdsName} moved to Non-Operational state as host CPU type is not supported in this cluster compatibility version or is not supported at all
157	USER_REBOOT_VM	Info	User ${UserName} initiated reboot of VM ${VmName}.
158	USER_FAILED_REBOOT_VM	Error	Failed to reboot VM ${VmName} (User: ${UserName}).
159	USER_FORCE_SELECTED_SPM	Info	Host ${VdsName} was force selected by ${UserName}
160	USER_ACCOUNT_DISABLED_OR_LOCKED	Error	User ${UserName} cannot login, as it got disabled or locked. Please contact the system administrator.
161	VM_CANCEL_MIGRATION	Info	Migration cancelled (VM: ${VmName}, Source: ${VdsName}, User: ${UserName}).
162	VM_CANCEL_MIGRATION_FAILED	Error	Failed to cancel migration for VM: ${VmName}
163	VM_STATUS_RESTORED	Info	VM ${VmName} status was restored to ${VmStatus}.
164	VM_SET_TICKET	Info	User ${UserName} initiated console session for VM ${VmName}
165	VM_SET_TICKET_FAILED	Error	User ${UserName} failed to initiate a console session for VM ${VmName}
166	VM_MIGRATION_NO_VDS_TO_MIGRATE_TO	Warning	No available host was found to migrate VM ${VmName} to.
167	VM_CONSOLE_CONNECTED	Info	User ${UserName} is connected to VM ${VmName}.

代码	名称	严重性	信息
168	VM_CONSOLE_DISCONNECTED	Info	User ${UserName} got disconnected from VM ${VmName}.
169	VM_FAILED_TO_PRESTART_IN_POOL	Warning	Cannot pre-start VM in pool '${VmPoolName}'. The system will continue trying.
170	USER_CREATE_LIVE_SNAPSHOT_FINISHED_FAILURE	Warning	Failed to create live snapshot '${SnapshotName}' for VM '${VmName}'. VM restart is recommended. Note that using the created snapshot might cause data inconsistency.
171	USER_RUN_VM_AS_STATELESS_WITH_DISKS_NOT_ALLOWING_SNAPSHOT	Warning	VM ${VmName} was run as stateless with one or more of disks that do not allow snapshots (User:${UserName}).
172	USER_REMOVE_VM_FINISHED_WITH_ILLEGAL_DISKS	Warning	VM ${VmName} has been removed, but the following disks could not be removed: ${DisksNames}. These disks will appear in the main disks tab in illegal state, please remove manually when possible.
173	USER_CREATE_LIVE_SNAPSHOT_NO_MEMORY_FAILURE	Error	Failed to save memory as part of Snapshot ${SnapshotName} for VM ${VmName} (User: ${UserName}).
174	VM_IMPORT_FROM_CONFIGURATION_EXECUTED_SUCCESSFULLY	Info	VM ${VmName} has been successfully imported from the given configuration.
175	VM_IMPORT_FROM_CONFIGURATION_ATTACH_DISKS_FAILED	Warning	VM ${VmName} has been imported from the given configuration but the following disk(s) failed to attach: ${DiskAliases}.

代码	名称	严重性	信息
176	VM_BALLOON_DRIVER_ERROR	Error	The Balloon driver on VM ${VmName} on host ${VdsName} is requested but unavailable.
177	VM_BALLOON_DRIVER_UNCONTROLLED	Error	The Balloon device on VM ${VmName} on host ${VdsName} is inflated but the device cannot be controlled (guest agent is down).
178	VM_MEMORY_NOT_IN_RECOMMENDED_RANGE	Warning	VM ${VmName} was configured with ${VmMemInMb}mb of memory while the recommended value range is ${VmMinMemInMb}mb - ${VmMaxMemInMb}mb
179	USER_INITIATED_RUN_VM_AND_PAUSE	Info	Starting in paused mode VM ${VmName} was initiated by ${UserName}.
180	TEMPLATE_IMPORT_FROM_CONFIGURATION_SUCCESS	Info	Template ${VmTemplateName} has been successfully imported from the given configuration.
181	TEMPLATE_IMPORT_FROM_CONFIGURATION_FAILED	Error	Failed to import Template ${VmTemplateName} from the given configuration.
182	USER_FAILED_ATTACH_USER_TO_VM	Error	Failed to attach User ${AdUserName} to VM ${VmName} (User: ${UserName}).
183	USER_ATTACH_TAG_TO_TEMPLATE	Info	Tag ${TagName} was attached to Templates(s) ${TemplatesNames} by ${UserName}.
184	USER_ATTACH_TAG_TO_TEMPLATE_FAILED	Error	Failed to attach Tag ${TagName} to Templates(s) ${TemplatesNames} (User: ${UserName}).

代码	名称	严重性	信息
185	USER_DETACH_TEMPLATE_FROM_TAG	Info	Tag ${TagName} was detached from Template(s) ${TemplatesNames} by ${UserName}.
186	USER_DETACH_TEMPLATE_FROM_TAG_FAILED	Error	Failed to detach Tag ${TagName} from TEMPLATE(s) ${TemplatesNames} (User: ${UserName}).
187	VDS_STORAGE_CONNECTION_FAILED_BUT_LAST_VDS	Error	Failed to connect Host ${VdsName} to Data Center, due to connectivity errors with the Storage. Host ${VdsName} will remain in Up state (but inactive), as it is the last Host in the Data Center, to enable manual intervention by the Administrator.
188	VDS_STORAGES_CONNECTION_FAILED	Error	Failed to connect Host ${VdsName} to the Storage Domains ${failedStorageDomains}.
189	VDS_STORAGE_VDS_STATS_FAILED	Error	Host ${VdsName} reports about one of the Active Storage Domains as Problematic.
190	UPDATE_OVF_FOR_STORAGE_DOMAIN_FAILED	Warning	Failed to update VMs/Templates OVF data for Storage Domain ${StorageDomainName} in Data Center ${StoragePoolName}.
191	CREATE_OVF_STORE_FOR_STORAGE_DOMAIN_FAILED	Warning	Failed to create OVF store disk for Storage Domain ${StorageDomainName}.\n The Disk with the id ${DiskId} might be removed manually for automatic attempt to create new one. \n OVF updates won't be attempted on the created disk.

代码	名称	严重性	信息
192	CREATE_OVF_STORE_FOR_STORAGE_DOMAIN_INITIATE_FAILED	Warning	Failed to create OVF store disk for Storage Domain ${StorageDomainName}. \n OVF data won't be updated meanwhile for that domain.
193	DELETE_OVF_STORE_FOR_STORAGE_DOMAIN_FAILED	Warning	Failed to delete the OVF store disk for Storage Domain ${StorageDomainName}.\n In order to detach the domain please remove it manually or try to detach the domain again for another attempt.
194	VM_CANCEL_CONVERSION	Info	Conversion cancelled (VM: ${VmName}, Source: ${VdsName}, User: ${UserName}).
195	VM_CANCEL_CONVERSION_FAILED	Error	Failed to cancel conversion for VM: ${VmName}
196	VM_RECOVERED_FROM_PAUSE_ERROR	Normal	VM ${VmName} has recovered from paused back to up.
200	IMPORTEXPORT_GET_VMS_INFO_FAILED	Error	Failed to retrieve VM/Templates information from export domain ${StorageDomainName}
201	IRS_DISK_SPACE_LOW_ERROR	Error	Critical, Low disk space. ${StorageDomainName} domain has ${DiskSpace} GB of free space.
202	IMPORTEXPORT_GET_EXTERNAL_VMS_INFO_FAILED	Error	Failed to retrieve VMs information from external server ${URL}
204	IRS_HOSTED_ON_VDS	Info	Storage Pool Manager runs on Host ${VdsName} (Address: ${ServerIp}).
205	PROVIDER_ADDED	Info	Provider ${ProviderName} was added. (User: ${UserName})

代码	名称	严重性	信息
206	PROVIDER_ADDITION_FAILED	Error	Failed to add provider ${ProviderName}. (User: ${UserName})
207	PROVIDER_UPDATED	Info	Provider ${ProviderName} was updated. (User: ${UserName})
208	PROVIDER_UPDATE_FAILED	Error	Failed to update provider ${ProviderName}. (User: ${UserName})
209	PROVIDER_REMOVED	Info	Provider ${ProviderName} was removed. (User: ${UserName})
210	PROVIDER_REMOVAL_FAILED	Error	Failed to remove provider ${ProviderName}. (User: ${UserName})
213	PROVIDER_CERTIFICATE_IMPORTED	Info	Certificate for provider ${ProviderName} was imported. (User: ${UserName})
214	PROVIDER_CERTIFICATE_IMPORT_FAILED	Error	Failed importing Certificate for provider ${ProviderName}. (User: ${UserName})
250	USER_UPDATE_VM_CLUSTER_DEFAULT_HOST_CLEARED	Info	${VmName} cluster was updated by ${UserName}, Default host was reset to auto assign.
251	USER_REMOVE_VM_TEMPLATE_FINISHED	Info	Removal of Template ${VmTemplateName} has been completed.
252	SYSTEM_FAILED_UPDATE_VM	Error	Failed to Update VM ${VmName} that was initiated by system.
253	SYSTEM_UPDATE_VM	Info	VM ${VmName} configuration was updated by system.
254	VM_ALREADY_IN_REQUESTED_STATUS	Info	VM ${VmName} is already ${VmStatus}, ${Action} was skipped. User: ${UserName}.

代码	名称	严重性	信息
302	USER_ADD_VM_POOL_WITH_VMS	Info	VM Pool ${VmPoolName} (containing ${VmsCount} VMs) was created by ${UserName}.
303	USER_ADD_VM_POOL_WITH_VMS_FAILED	Error	Failed to create VM Pool ${VmPoolName} (User: ${UserName}).
304	USER_REMOVE_VM_POOL	Info	VM Pool ${VmPoolName} was removed by ${UserName}.
305	USER_REMOVE_VM_POOL_FAILED	Error	Failed to remove VM Pool ${VmPoolName} (User: ${UserName}).
306	USER_ADD_VM_TO_POOL	Info	VM ${VmName} was added to VM Pool ${VmPoolName} by ${UserName}.
307	USER_ADD_VM_TO_POOL_FAILED	Error	Failed to add VM ${VmName} to VM Pool ${VmPoolName} (User: ${UserName}).
308	USER_REMOVE_VM_FROM_POOL	Info	VM ${VmName} was removed from VM Pool ${VmPoolName} by ${UserName}.
309	USER_REMOVE_VM_FROM_POOL_FAILED	Error	Failed to remove VM ${VmName} from VM Pool ${VmPoolName} (User: ${UserName}).
310	USER_ATTACH_USER_TO_POOL	Info	User ${AdUserName} was attached to VM Pool ${VmPoolName} by ${UserName}.
311	USER_ATTACH_USER_TO_POOL_FAILED	Error	Failed to attach User ${AdUserName} to VM Pool ${VmPoolName} (User: ${UserName}).
312	USER_DETACH_USER_FROM_POOL	Info	User ${AdUserName} was detached from VM Pool ${VmPoolName} by ${UserName}.

代码	名称	严重性	信息
313	USER_DETACH_USER_FROM_POOL_FAILED	Error	Failed to detach User ${AdUserName} from VM Pool ${VmPoolName} (User: ${UserName}).
314	USER_UPDATE_VM_POOL	Info	VM Pool ${VmPoolName} configuration was updated by ${UserName}.
315	USER_UPDATE_VM_POOL_FAILED	Error	Failed to update VM Pool ${VmPoolName} configuration (User: ${UserName}).
316	USER_ATTACH_USER_TO_VM_FROM_POOL	Info	Attaching User ${AdUserName} to VM ${VmName} in VM Pool ${VmPoolName} was initiated by ${UserName}.
317	USER_ATTACH_USER_TO_VM_FROM_POOL_FAILED	Error	Failed to attach User ${AdUserName} to VM from VM Pool ${VmPoolName} (User: ${UserName}).
318	USER_ATTACH_USER_TO_VM_FROM_POOL_FINISHED_SUCCESS	Info	User ${AdUserName} successfully attached to VM ${VmName} in VM Pool ${VmPoolName}.
319	USER_ATTACH_USER_TO_VM_FROM_POOL_FINISHED_FAILURE	Error	Failed to attach user ${AdUserName} to VM ${VmName} in VM Pool ${VmPoolName}.
320	USER_ADD_VM_POOL_WITH_VMS_ADD_VDS_FAILED	Error	Pool ${VmPoolName} Created, but some Vms failed to create (User: ${UserName}).
325	USER_REMOVE_ADUSER	Info	User ${AdUserName} was removed by ${UserName}.
326	USER_FAILED_REMOVE_ADUSER	Error	Failed to remove User ${AdUserName} (User: ${UserName}).
327	USER_FAILED_ADD_ADUSER	Warning	Failed to add User '${NewUserName}' to the system.

代码	名称	严重性	信息
342	USER_REMOVE_SNAPSHOT	Info	Snapshot '${SnapshotName}' deletion for VM '${VmName}' was initiated by ${UserName}.
343	USER_FAILED_REMOVE_SNAPSHOT	Error	Failed to remove Snapshot ${SnapshotName} for VM ${VmName} (User: ${UserName}).
344	USER_UPDATE_VM_POOL_WITH_VMS	Info	VM Pool ${VmPoolName} was updated by ${UserName}, ${VmsCount} VMs were added.
345	USER_UPDATE_VM_POOL_WITH_VMS_FAILED	Error	Failed to update VM Pool ${VmPoolName}(User: ${UserName}).
346	USER_PASSWORD_CHANGED	Info	Password changed successfully for ${UserName}
347	USER_PASSWORD_CHANGE_FAILED	Error	Failed to change password. (User: ${UserName})
348	USER_CLEAR_UNKNOWN_VMS	Info	All VMs' status on Non Responsive Host ${VdsName} were changed to 'Down' by ${UserName}
349	USER_FAILED_CLEAR_UNKNOWN_VMS	Error	Failed to clear VMs' status on Non Responsive Host ${VdsName}. (User: ${UserName}).
350	USER_ADD_BOOKMARK	Info	Bookmark ${BookmarkName} was added by ${UserName}.
351	USER_ADD_BOOKMARK_FAILED	Error	Failed to add bookmark: ${BookmarkName} (User: ${UserName}).
352	USER_UPDATE_BOOKMARK	Info	Bookmark ${BookmarkName} was updated by ${UserName}.
353	USER_UPDATE_BOOKMARK_FAILED	Error	Failed to update bookmark: ${BookmarkName} (User: ${UserName})

代码	名称	严重性	信息
354	USER_REMOVE_BOOKMARK	Info	Bookmark ${BookmarkName} was removed by ${UserName}.
355	USER_REMOVE_BOOKMARK_FAILED	Error	Failed to remove bookmark ${BookmarkName} (User: ${UserName})
356	USER_REMOVE_SNAPSHOT_FINISHED_SUCCESS	Info	Snapshot '${SnapshotName}' deletion for VM '${VmName}' has been completed.
357	USER_REMOVE_SNAPSHOT_FINISHED_FAILURE	Error	Failed to delete snapshot '${SnapshotName}' for VM '${VmName}'.
358	USER_VM_POOL_MAX_SUBSEQUENT_FAILURES_REACHED	Warning	Not all VMs where successfully created in VM Pool ${VmPoolName}.
359	USER_REMOVE_SNAPSHOT_FINISHED_FAILURE_PARTIAL_SNAPSHOT	Warning	Due to partial snapshot removal, Snapshot '${SnapshotName}' of VM '${VmName}' now contains only the following disks: '${DiskAliases}'.
360	USER_DETACH_USER_FROM_VM	Info	User ${AdUserName} was detached from VM ${VmName} by ${UserName}.
361	USER_FAILED_DETACH_USER_FROM_VM	Error	Failed to detach User ${AdUserName} from VM ${VmName} (User: ${UserName}).
370	USER_EXTEND_DISK_SIZE_FAILURE	Error	Failed to extend size of the disk '${DiskAlias}' to ${NewSize} GB, User: ${UserName}.
371	USER_EXTEND_DISK_SIZE_SUCCESS	Info	Size of the disk '${DiskAlias}' was successfully updated to ${NewSize} GB by ${UserName}.

代码	名称	严重性	信息
372	USER_EXTEND_DISK_SIZE_UPDATE_VM_FAILURE	Warning	Failed to update VM '${VmName}' with the new volume size. VM restart is recommended.
373	USER_REMOVE_DISK_SNAPSHOT	Info	Disk '${DiskAlias}' from Snapshot(s) '${Snapshots}' of VM '${VmName}' deletion was initiated by ${UserName}.
374	USER_FAILED_REMOVE_DISK_SNAPSHOT	Error	Failed to delete Disk '${DiskAlias}' from Snapshot(s) ${Snapshots} of VM ${VmName} (User: ${UserName}).
375	USER_REMOVE_DISK_SNAPSHOT_FINISHED_SUCCESS	Info	Disk '${DiskAlias}' from Snapshot(s) '${Snapshots}' of VM '${VmName}' deletion has been completed (User: ${UserName}).
376	USER_REMOVE_DISK_SNAPSHOT_FINISHED_FAILURE	Error	Failed to complete deletion of Disk '${DiskAlias}' from snapshot(s) '${Snapshots}' of VM '${VmName}' (User: ${UserName}).
377	USER_EXTENDED_DISK_SIZE	Info	Extending disk '${DiskAlias}' to ${NewSize} GB was initiated by ${UserName}.
378	USER_REGISTER_DISK_FINISHED_SUCCESS	Info	Disk '${DiskAlias}' has been successfully registered as a floating disk.
379	USER_REGISTER_DISK_FINISHED_FAILURE	Error	Failed to register Disk '${DiskAlias}'.
380	USER_EXTEND_DISK_SIZE_UPDATE_HOST_FAILURE	Warning	Failed to refresh volume size on host '${VdsName}'. Please try the operation again.
400	USER_ATTACH_VM_TO_AD_GROUP	Info	Group ${GroupName} was attached to VM ${VmName} by ${UserName}.

代码	名称	严重性	信息
401	USER_ATTACH_VM_TO_AD_GROUP_FAILED	Error	Failed to attach Group ${GroupName} to VM ${VmName} (User: ${UserName}).
402	USER_DETACH_VM_TO_AD_GROUP	Info	Group ${GroupName} was detached from VM ${VmName} by ${UserName}.
403	USER_DETACH_VM_TO_AD_GROUP_FAILED	Error	Failed to detach Group ${GroupName} from VM ${VmName} (User: ${UserName}).
404	USER_ATTACH_VM_POOL_TO_AD_GROUP	Info	Group ${GroupName} was attached to VM Pool ${VmPoolName} by ${UserName}.
405	USER_ATTACH_VM_POOL_TO_AD_GROUP_FAILED	Error	Failed to attach Group ${GroupName} to VM Pool ${VmPoolName} (User: ${UserName}).
406	USER_DETACH_VM_POOL_TO_AD_GROUP	Info	Group ${GroupName} was detached from VM Pool ${VmPoolName} by ${UserName}.
407	USER_DETACH_VM_POOL_TO_AD_GROUP_FAILED	Error	Failed to detach Group ${GroupName} from VM Pool ${VmPoolName} (User: ${UserName}).
408	USER_REMOVE_AD_GROUP	Info	Group ${GroupName} was removed by ${UserName}.
409	USER_REMOVE_AD_GROUP_FAILED	Error	Failed to remove group ${GroupName} (User: ${UserName}).
430	USER_UPDATE_TAG	Info	Tag ${TagName} configuration was updated by ${UserName}.
431	USER_UPDATE_TAG_FAILED	Error	Failed to update Tag ${TagName} (User: ${UserName}).
432	USER_ADD_TAG	Info	New Tag ${TagName} was created by ${UserName}.

代码	名称	严重性	信息
433	USER_ADD_TAG_FAILED	Error	Failed to create Tag named ${TagName} (User: ${UserName}).
434	USER_REMOVE_TAG	Info	Tag ${TagName} was removed by ${UserName}.
435	USER_REMOVE_TAG_FAILED	Error	Failed to remove Tag ${TagName} (User: ${UserName}).
436	USER_ATTACH_TAG_TO_USER	Info	Tag ${TagName} was attached to User(s) ${AttachUsersNames} by ${UserName}.
437	USER_ATTACH_TAG_TO_USER_FAILED	Error	Failed to attach Tag ${TagName} to User(s) ${AttachUsersNames} (User: ${UserName}).
438	USER_ATTACH_TAG_TO_USER_GROUP	Info	Tag ${TagName} was attached to Group(s) ${AttachGroupsNames} by ${UserName}.
439	USER_ATTACH_TAG_TO_USER_GROUP_FAILED	Error	Failed to attach Group(s) ${AttachGroupsNames} to Tag ${TagName} (User: ${UserName}).
440	USER_ATTACH_TAG_TO_VM	Info	Tag ${TagName} was attached to VM(s) ${VmsNames} by ${UserName}.
441	USER_ATTACH_TAG_TO_VM_FAILED	Error	Failed to attach Tag ${TagName} to VM(s) ${VmsNames} (User: ${UserName}).
442	USER_ATTACH_TAG_TO_VDS	Info	Tag ${TagName} was attached to Host(s) ${VdsNames} by ${UserName}.
443	USER_ATTACH_TAG_TO_VDS_FAILED	Error	Failed to attach Tag ${TagName} to Host(s) ${VdsNames} (User: ${UserName}).

代码	名称	严重性	信息
444	USER_DETACH_VDS_FROM_TAG	Info	Tag ${TagName} was detached from Host(s) ${VdsNames} by ${UserName}.
445	USER_DETACH_VDS_FROM_TAG_FAILED	Error	Failed to detach Tag ${TagName} from Host(s) ${VdsNames} (User: ${UserName}).
446	USER_DETACH_VM_FROM_TAG	Info	Tag ${TagName} was detached from VM(s) ${VmsNames} by ${UserName}.
447	USER_DETACH_VM_FROM_TAG_FAILED	Error	Failed to detach Tag ${TagName} from VM(s) ${VmsNames} (User: ${UserName}).
448	USER_DETACH_USER_FROM_TAG	Info	Tag ${TagName} detached from User(s) ${DetachUsersNames} by ${UserName}.
449	USER_DETACH_USER_FROM_TAG_FAILED	Error	Failed to detach Tag ${TagName} from User(s) ${DetachUsersNames} (User: ${UserName}).
450	USER_DETACH_USER_GROUP_FROM_TAG	Info	Tag ${TagName} was detached from Group(s) ${DetachGroupsNames} by ${UserName}.
451	USER_DETACH_USER_GROUP_FROM_TAG_FAILED	Error	Failed to detach Tag ${TagName} from Group(s) ${DetachGroupsNames} (User: ${UserName}).
452	USER_ATTACH_TAG_TO_USER_EXISTS	Warning	Tag ${TagName} already attached to User(s) ${AttachUsersNamesExists}.
453	USER_ATTACH_TAG_TO_USER_GROUP_EXISTS	Warning	Tag ${TagName} already attached to Group(s) ${AttachGroupsNamesExists}.
454	USER_ATTACH_TAG_TO_VM_EXISTS	Warning	Tag ${TagName} already attached to VM(s) ${VmsNamesExists}.

代码	名称	严重性	信息
455	USER_ATTACH_TAG_TO_VDS_EXISTS	Warning	Tag ${TagName} already attached to Host(s) ${VdsNamesExists}.
456	USER_LOGGED_IN_VM	Info	User ${GuestUser} logged in to VM ${VmName}.
457	USER_LOGGED_OUT_VM	Info	User ${GuestUser} logged out from VM ${VmName}.
458	USER_LOCKED_VM	Info	User ${GuestUser} locked VM ${VmName}.
459	USER_UNLOCKED_VM	Info	User ${GuestUser} unlocked VM ${VmName}.
460	USER_ATTACH_TAG_TO_TEMPLATE_EXISTS	Warning	Tag ${TagName} already attached to Template(s) ${TemplatesNamesExists}.
467	UPDATE_TAGS_VM_DEFAULT_DISPLAY_TYPE	Info	Vm ${VmName} tag default display type was updated
468	UPDATE_TAGS_VM_DEFAULT_DISPLAY_TYPE_FAILED	Info	Failed to update Vm ${VmName} tag default display type
470	USER_ATTACH_VM_POOL_TO_AD_GROUP_INTERNAL	Info	Group ${GroupName} was attached to VM Pool ${VmPoolName}.
471	USER_ATTACH_VM_POOL_TO_AD_GROUP_FAILED_INTERNAL	Error	Failed to attach Group ${GroupName} to VM Pool ${VmPoolName}.
472	USER_ATTACH_USER_TO_POOL_INTERNAL	Info	User ${AdUserName} was attached to VM Pool ${VmPoolName}.
473	USER_ATTACH_USER_TO_POOL_FAILED_INTERNAL	Error	Failed to attach User ${AdUserName} to VM Pool ${VmPoolName} (User: ${UserName}).
493	VDS_ALREADY_IN_REQUESTED_STATUS	Warning	Host ${HostName} is already ${AgentStatus}, Power Management ${Operation} operation skipped.

代码	名称	严重性	信息
494	VDS_MANUAL_FENCE_STATUS	Info	Manual fence for host ${VdsName} was started.
495	VDS_MANUAL_FENCE_STATUS_FAILED	Error	Manual fence for host ${VdsName} failed.
496	VDS_FENCE_STATUS	Info	Host ${VdsName} power management was verified successfully.
497	VDS_FENCE_STATUS_FAILED	Error	Failed to verify Host ${VdsName} power management.
498	VDS_APPROVE	Info	Host ${VdsName} was successfully approved by user ${UserName}.
499	VDS_APPROVE_FAILED	Error	Failed to approve Host ${VdsName}.
500	VDS_FAILED_TO_RUN_VMS	Error	Host ${VdsName} will be switched to Error status for ${Time} minutes because it failed to run a VM.
501	USER_SUSPEND_VM	Info	Suspending VM ${VmName} was initiated by User ${UserName} (Host: ${VdsName}).
502	USER_FAILED_SUSPEND_VM	Error	Failed to suspend VM ${VmName} (Host: ${VdsName}).
503	USER_SUSPEND_VM_OK	Info	VM ${VmName} on Host ${VdsName} is suspended.
504	VDS_INSTALL	Info	Host ${VdsName} installed
505	VDS_INSTALL_FAILED	Error	Host ${VdsName} installation failed. ${FailedInstallMessage}.
506	VDS_INITIATED_RUN_VM	Info	VM ${VmName} was restarted on Host ${VdsName}
509	VDS_INSTALL_IN_PROGRESS	Info	Installing Host ${VdsName}. ${Message}.

代码	名称	严重性	信息
510	VDS_INSTALL_IN_PROGRESS_WARNING	Warning	Host ${VdsName} installation in progress . ${Message}.
511	VDS_INSTALL_IN_PROGRESS_ERROR	Error	Failed to install Host ${VdsName}. ${Message}.
512	USER_SUSPEND_VM_FINISH_SUCCESS	Info	Suspending VM ${VmName} has been completed.
513	VDS_RECOVER_FAILED_VMS_UNKNOWN	Error	Host ${VdsName} cannot be reached, VMs state on this host are marked as Unknown.
514	VDS_INITIALIZING	Warning	Host ${VdsName} is initializing. Message: ${ErrorMessage}
515	VDS_CPU_LOWER_THAN_CLUSTER	Warning	Host ${VdsName} moved to Non-Operational state as host does not meet the cluster's minimum CPU level. Missing CPU features : ${CpuFlags}
516	VDS_CPU_RETRIEVE_FAILED	Warning	Failed to determine Host ${VdsName} CPU level - could not retrieve CPU flags.
517	VDS_SET_NONOPERATIONAL	Info	Host ${VdsName} moved to Non-Operational state.
518	VDS_SET_NONOPERATIONAL_FAILED	Error	Failed to move Host ${VdsName} to Non-Operational state.
519	VDS_SET_NONOPERATIONAL_NETWORK	Warning	Host ${VdsName} does not comply with the cluster ${VdsGroupName} networks, the following networks are missing on host: '${Networks}'
520	USER_ATTACH_USER_TO_VM	Info	User ${AdUserName} was attached to VM ${VmName} by ${UserName}.
521	USER_SUSPEND_VM_FINISH_FAILURE	Error	Failed to complete suspending of VM ${VmName}.

代码	名称	严重性	信息
522	VDS_SET_NONOPERATIONAL_DOMAIN	Warning	Host ${VdsName} cannot access the Storage Domain(s) ${StorageDomainNames} attached to the Data Center ${StoragePoolName}. Setting Host state to Non-Operational.
523	VDS_SET_NONOPERATIONAL_DOMAIN_FAILED	Error	Host ${VdsName} cannot access the Storage Domain(s) ${StorageDomainNames} attached to the Data Center ${StoragePoolName}. Failed to set Host state to Non-Operational.
524	VDS_DOMAIN_DELAY_INTERVAL	Warning	Storage domain ${StorageDomainName} experienced a high latency of ${Delay} seconds from host ${VdsName}. This may cause performance and functional issues. Please consult your Storage Administrator.
525	VDS_INITIATED_RUN_AS_STATELESS_VM_NOT_YET_RUNNING	Info	Starting VM ${VmName} as stateless was initiated.
528	USER_EJECT_VM_DISK	Info	CD was ejected from VM ${VmName} by ${UserName}.
529	USER_EJECT_VM_FLOPPY	Info	Floppy was ejected from VM ${VmName} by ${UserName}
530	VDS_MANUAL_FENCE_FAILED_CALL_FENCE_SPM	Warning	Manual fence did not revoke the selected SPM (${VdsName}) since the master storage domain\n was not active or could not use another host for the fence operation.
531	VDS_LOW_MEM	Warning	Available memory of host ${HostName} [${AvailableMemory} MB] is under defined threshold [${Threshold} MB].

代码	名称	严重性	信息
532	VDS_HIGH_MEM_USE	Warning	Used memory of host ${HostName} [${UsedMemory}%] exceeded defined threshold [${Threshold}%].
533	VDS_HIGH_NETWORK_USE	Warning	
534	VDS_HIGH_CPU_USE	Warning	Used CPU of host ${HostName} [${UsedCpu}%] exceeded defined threshold [${Threshold}%].
535	VDS_HIGH_SWAP_USE	Warning	Used swap memory of host ${HostName} [${UsedSwap}%] exceeded defined threshold [${Threshold}%].
536	VDS_LOW_SWAP	Warning	Available swap memory of host ${HostName} [${AvailableSwapMemory} MB] is under defined threshold [${Threshold} MB].
537	VDS_INITIATED_RUN_VM_AS_STATELESS	Info	VM ${VmName} was restarted on Host ${VdsName} as stateless
538	USER_RUN_VM_AS_STATELESS	Info	VM ${VmName} started on Host ${VdsName} as stateless
539	VDS_AUTO_FENCE_STATUS	Info	Auto fence for host ${VdsName} was started.
540	VDS_AUTO_FENCE_STATUS_FAILED	Error	Auto fence for host ${VdsName} failed.
541	VDS_AUTO_FENCE_FAILED_CALL_FENCE_SPM	Warning	Auto fence did not revoke the selected SPM (${VdsName}) since the master storage domain\n was not active or could not use another host for the fence operation.
550	VDS_PACKAGES_IN_PROGRESS	Info	Package update Host ${VdsName}. ${Message}.

代码	名称	严重性	信息
551	VDS_PACKAGES_IN_PROGRESS_WARNING	Warning	Host ${VdsName} update packages in progress . ${Message}.
552	VDS_PACKAGES_IN_PROGRESS_ERROR	Error	Failed to update packages Host ${VdsName}. ${Message}.
555	USER_MOVE_TAG	Info	Tag ${TagName} was moved from ${OldParnetTagName} to ${NewParentTagName} by ${UserName}.
556	USER_MOVE_TAG_FAILED	Error	Failed to move Tag ${TagName} from ${OldParnetTagName} to ${NewParentTagName} (User: ${UserName}).
600	USER_VDS_MAINTENANCE	Info	Host ${VdsName} was switched to Maintenance mode by ${UserName} (Reason: ${Reason}).
601	CPU_FLAGS_NX_IS_MISSING	Warning	Host ${VdsName} is missing the NX cpu flag. This flag can be enabled via the host BIOS. Please set Disable Execute (XD) for an Intel host, or No Execute (NX) for AMD. Please make sure to completely power off the host for this change to take effect.
602	USER_VDS_MAINTENANCE_MIGRATION_FAILED	Warning	Host ${VdsName} cannot change into maintenance mode - not all Vms have been migrated successfully. Consider manual intervention: stopping/migrating Vms: ${failedVms} (User: ${UserName}).

代码	名称	严重性	信息
603	VDS_SET_NONOPERATIONAL_IFACE_DOWN	Warning	Host ${VdsName} moved to Non-Operational state because interfaces which are down are needed by required networks in the current cluster: '${NicsWithNetworks}'.
604	VDS_TIME_DRIFT_ALERT	Warning	Host ${VdsName} has time-drift of ${Actual} seconds while maximum configured value is ${Max} seconds.
605	PROXY_HOST_SELECTION	Info	Host ${Proxy} from ${Origin} was chosen as a proxy to execute fencing on Host ${VdsName}.
606	HOST_REFRESHED_CAPABILITIES	Info	Successfully refreshed the capabilities of host ${VdsName}.
607	HOST_REFRESH_CAPABILITIES_FAILED	Error	Failed to refresh the capabilities of host ${VdsName}.
608	HOST_INTERFACE_HIGH_NETWORK_USE	Warning	Host ${HostName} has network interface which exceeded the defined threshold [${Threshold}%] (${InterfaceName}: transmit rate[${TransmitRate}%], receive rate [${ReceiveRate}%])
609	HOST_INTERFACE_STATE_UP	Normal	Interface ${InterfaceName} on host ${VdsName}, changed state to up
610	HOST_INTERFACE_STATE_DOWN	Warning	Interface ${InterfaceName} on host ${VdsName}, changed state to down
611	HOST_BOND_SLAVE_STATE_UP	Normal	Slave ${SlaveName} of bond ${BondName} on host ${VdsName}, changed state to up

代码	名称	严重性	信息
612	HOST_BOND_SLAVE_STATE_DOWN	Warning	Slave ${SlaveName} of bond ${BondName} on host ${VdsName}, changed state to down
613	FENCE_KDUMP_LISTENER_IS_NOT_ALIVE	Error	Unable to determine if Kdump is in progress on host ${VdsName}, because fence_kdump listener is not running.
614	KDUMP_FLOW_DETECTED_ON_VDS	Info	Kdump flow is in progress on host ${VdsName}.
615	KDUMP_FLOW_NOT_DETECTED_ON_VDS	Info	Kdump flow is not in progress on host ${VdsName}.
616	KDUMP_FLOW_FINISHED_ON_VDS	Info	Kdump flow finished on host ${VdsName}.
617	KDUMP_DETECTION_NOT_CONFIGURED_ON_VDS	Warning	Kdump integration is enabled for host ${VdsName}, but kdump is not configured properly on host.
618	HOST_REGISTRATION_FAILED_INVALID_CLUSTER	Info	No default or valid cluster was found, Host ${VdsName} registration failed
700	DISK_ALIGNMENT_SCAN_START	Info	Starting alignment scan of disk '${DiskAlias}'.
701	DISK_ALIGNMENT_SCAN_FAILURE	Warning	Alignment scan of disk '${DiskAlias}' failed.
702	DISK_ALIGNMENT_SCAN_SUCCESS	Info	Alignment scan of disk '${DiskAlias}' is complete.
809	USER_ADD_VDS_GROUP	Info	Cluster ${VdsGroupName} was added by ${UserName}
810	USER_ADD_VDS_GROUP_FAILED	Error	Failed to add Host cluster (User: ${UserName})
811	USER_UPDATE_VDS_GROUP	Info	Host cluster ${VdsGroupName} was updated by ${UserName}

代码	名称	严重性	信息
812	USER_UPDATE_VDS_GROUP_FAILED	Error	Failed to update Host cluster (User: ${UserName})
813	USER_REMOVE_VDS_GROUP	Info	Host cluster ${VdsGroupName} was removed by ${UserName}
814	USER_REMOVE_VDS_GROUP_FAILED	Error	Failed to remove Host cluster (User: ${UserName})
815	USER_VDC_LOGOUT_FAILED	Error	Failed to log User ${UserName} out.
816	MAC_POOL_EMPTY	Warning	No MAC addresses left in the MAC Address Pool.
817	CERTIFICATE_FILE_NOT_FOUND	Error	Could not find oVirt Engine Certificate file.
818	RUN_VM_FAILED	Error	Cannot run VM ${VmName} on Host ${VdsName}. Error: ${ErrMsg}
819	VDS_REGISTER_ERROR_UPDATING_HOST	Error	Host registration failed - cannot update Host Name for Host ${VdsName2}. (Host: ${VdsName1})
820	VDS_REGISTER_ERROR_UPDATING_HOST_ALL_TAKEN	Error	Host registration failed - all available Host Names are taken. (Host: ${VdsName1})
821	VDS_REGISTER_HOST_IS_ACTIVE	Error	Host registration failed - cannot change Host Name of active Host ${VdsName2}. (Host: ${VdsName1})
822	VDS_REGISTER_ERROR_UPDATING_NAME	Error	Host registration failed - cannot update Host Name for Host ${VdsName2}. (Host: ${VdsName1})
823	VDS_REGISTER_ERROR_UPDATING_NAMES_ALL_TAKEN	Error	Host registration failed - all available Host Names are taken. (Host: ${VdsName1})

代码	名称	严重性	信息
824	VDS_REGISTER_NAME_IS_ACTIVE	Error	Host registration failed - cannot change Host Name of active Host ${VdsName2}. (Host: ${VdsName1})
825	VDS_REGISTER_AUTO_APPROVE_PATTERN	Error	Host registration failed - auto approve pattern error. (Host: ${VdsName1})
826	VDS_REGISTER_FAILED	Error	Host registration failed. (Host: ${VdsName1})
827	VDS_REGISTER_EXISTING_VDS_UPDATE_FAILED	Error	Host registration failed - cannot update existing Host. (Host: ${VdsName1})
828	VDS_REGISTER_SUCCEEDED	Info	Host ${VdsName1} registered.
829	VM_MIGRATION_ON_CONNECT_CHECK_FAILED	Error	VM migration logic failed. (VM name: ${VmName})
830	VM_MIGRATION_ON_CONNECT_CHECK_SUCCEEDED	Info	Migration check failed to execute.
831	USER_VDC_SESSION_TERMINATED	Info	User ${UserName} forcibly logout user ${TerminatedSessionUsername}.
832	USER_VDC_SESSION_TERMINATION_FAILED	Error	User ${UserName} failed to forcibly logout user ${TerminatedSessionUsername}.
833	MAC_ADDRESS_IS_IN_USE	Warning	Network Interface ${IfaceName} has MAC address ${MACAddr} which is in use.
834	VDS_REGISTER_EMPTY_ID	Warning	Host registration failed, empty host id (Host: ${VdsHostName})
835	SYSTEM_UPDATE_VDS_GROUP	Info	Host cluster ${VdsGroupName} was updated by system
836	SYSTEM_UPDATE_VDS_GROUP_FAILED	Info	Failed to update Host cluster by system

代码	名称	严重性	信息
837	MAC_ADDRESSES_POOL_NOT_INITIALIZED	Warning	Mac Address Pool is not initialized. ${Message}
838	MAC_ADDRESS_IS_IN_USE_UNPLUG	Warning	Network Interface ${IfaceName} has MAC address ${MACAddr} which is in use, therefore it is being unplugged from VM ${VmName}.
840	HOST_UPGRADE_STARTED	Info	Host ${VdsName} upgrade was started (User: ${UserName}).
841	HOST_UPGRADE_FAILED	Error	Failed to upgrade Host ${VdsName} (User: ${UserName}).
842	HOST_UPGRADE_FINISHED	Info	Host ${VdsName} upgrade was completed successfully.
845	HOST_CERTIFICATION_IS_ABOUT_TO_EXPIRE	Warning	Host ${VdsName} certification is about to expire at ${ExpirationDate}. Please renew the host's certification.
846	ENGINE_CERTIFICATION_HAS_EXPIRED	Info	Engine's certification has expired at ${ExpirationDate}. Please renew the engine's certification.
847	ENGINE_CERTIFICATION_IS_ABOUT_TO_EXPIRE	Warning	Engine's certification is about to expire at ${ExpirationDate}. Please renew the engine's certification.
848	ENGINE_CA_CERTIFICATION_HAS_EXPIRED	Info	Engine's CA certification has expired at ${ExpirationDate}.
849	ENGINE_CA_CERTIFICATION_IS_ABOUT_TO_EXPIRE	Warning	Engine's CA certification is about to expire at ${ExpirationDate}.

代码	名称	严重性	信息
850	USER_ADD_PERMISSION	Info	User/Group ${SubjectName}, Namespace ${Namespace}, Authorization provider: ${Authz} was granted permission for Role ${RoleName} on ${VdcObjectType} ${VdcObjectName}, by ${UserName}.
851	USER_ADD_PERMISSION_FAILED	Error	User ${UserName} failed to grant permission for Role ${RoleName} on ${VdcObjectType} ${VdcObjectName} to User/Group ${SubjectName}.
852	USER_REMOVE_PERMISSION	Info	User/Group ${SubjectName} Role ${RoleName} permission was removed from ${VdcObjectType} ${VdcObjectName} by ${UserName}
853	USER_REMOVE_PERMISSION_FAILED	Error	User ${UserName} failed to remove permission for Role ${RoleName} from ${VdcObjectType} ${VdcObjectName} to User/Group ${SubjectName}
854	USER_ADD_ROLE	Info	Role ${RoleName} granted to ${UserName}
855	USER_ADD_ROLE_FAILED	Error	Failed to grant role ${RoleName} (User ${UserName})
856	USER_UPDATE_ROLE	Info	${UserName} Role was updated to the ${RoleName} Role
857	USER_UPDATE_ROLE_FAILED	Error	Failed to update role ${RoleName} to ${UserName}
858	USER_REMOVE_ROLE	Info	Role ${RoleName} removed from ${UserName}

代码	名称	严重性	信息
859	USER_REMOVE_ROLE_FAILED	Error	Failed to remove role ${RoleName} (User ${UserName})
860	USER_ATTACHED_ACTION_GROUP_TO_ROLE	Info	Action group ${ActionGroup} was attached to Role ${RoleName} by ${UserName}
861	USER_ATTACHED_ACTION_GROUP_TO_ROLE_FAILED	Error	Failed to attach Action group ${ActionGroup} to Role ${RoleName} (User: ${UserName})
862	USER_DETACHED_ACTION_GROUP_FROM_ROLE	Info	Action group ${ActionGroup} was detached from Role ${RoleName} by ${UserName}
863	USER_DETACHED_ACTION_GROUP_FROM_ROLE_FAILED	Error	Failed to attach Action group ${ActionGroup} to Role ${RoleName} by ${UserName}
864	USER_ADD_ROLE_WITH_ACTION_GROUP	Info	Role ${RoleName} was added by ${UserName}
865	USER_ADD_ROLE_WITH_ACTION_GROUP_FAILED	Error	Failed to add role ${RoleName}
866	USER_ADD_SYSTEM_PERMISSION	Info	User/Group ${SubjectName} was granted permission for Role ${RoleName} on ${VdcObjectType} by ${UserName}.
867	USER_ADD_SYSTEM_PERMISSION_FAILED	Error	User ${UserName} failed to grant permission for Role ${RoleName} on ${VdcObjectType} to User/Group ${SubjectName}.
868	USER_REMOVE_SYSTEM_PERMISSION	Info	User/Group ${SubjectName} Role ${RoleName} permission was removed from ${VdcObjectType} by ${UserName}

代码	名称	严重性	信息
869	USER_REMOVE_SYSTEM_PERMISSION_FAILED	Error	User ${UserName} failed to remove permission for Role ${RoleName} from ${VdcObjectType} to User/Group ${SubjectName}
870	USER_ADD_PROFILE	Info	Profile created for ${UserName}
871	USER_ADD_PROFILE_FAILED	Error	Failed to create profile for ${UserName}
872	USER_UPDATE_PROFILE	Info	Updated profile for ${UserName}
873	USER_UPDATE_PROFILE_FAILED	Error	Failed to update profile for ${UserName}
874	USER_REMOVE_PROFILE	Info	Removed profile for ${UserName}
875	USER_REMOVE_PROFILE_FAILED	Error	Failed to remove profile for ${UserName}
876	HOST_CERTIFICATION_IS_INVALID	Error	Host ${VdsName} certification is invalid. The certification has no peer certificates.
877	HOST_CERTIFICATION_HAS_EXPIRED	Info	
878	ENGINE_CERTIFICATION_IS_ABOUT_TO_EXPIRE_ALERT	Info	Engine's certification is about to expire at ${ExpirationDate}. Please renew the engine's certification.
879	HOST_CERTIFICATION_IS_ABOUT_TO_EXPIRE_ALERT	Info	Host ${VdsName} certification is about to expire at ${ExpirationDate}. Please renew the host's certification.
880	HOST_CERTIFICATION_ENROLLMENT_STARTED	Normal	Enrolling certificate for host ${VdsName} was started (User: ${UserName}).

代码	名称	严重性	信息
881	HOST_CERTIFICATION_ENROLLMENT_FINISHED	Normal	Enrolling certificate for host ${VdsName} was completed successfully (User: ${UserName}).
882	HOST_CERTIFICATION_ENROLLMENT_FAILED	Error	Failed to enroll certificate for host ${VdsName} (User: ${UserName}).
883	ENGINE_CA_CERTIFICATION_IS_ABOUT_TO_EXPIRE_ALERT	Info	Engine's CA certification is about to expire at ${ExpirationDate}.
900	AD_COMPUTER_ACCOUNT_SUCCEEDED	Info	Account creation successful.
901	AD_COMPUTER_ACCOUNT_FAILED	Error	Account creation failed.
918	USER_FORCE_REMOVE_STORAGE_POOL	Info	Data Center ${StoragePoolName} was forcibly removed by ${UserName}
919	USER_FORCE_REMOVE_STORAGE_POOL_FAILED	Error	Failed to forcibly remove Data Center ${StoragePoolName}. (User: ${UserName})
920	NETWORK_ATTACH_NETWORK_TO_VDS	Info	Attach network: ${NetworkName} to Host: ${VdsName} by ${UserName}.
921	NETWORK_ATTACH_NETWORK_TO_VDS_FAILED	Error	Failed to attach network: ${NetworkName} to Host: ${VdsName} (User: ${UserName}).
922	NETWORK_DETACH_NETWORK_FROM_VDS	Info	Detach network: ${NetworkName} from Host: ${VdsName} by ${UserName}.
923	NETWORK_DETACH_NETWORK_FROM_VDS_FAILED	Error	Failed to detach network: ${NetworkName} from Host: ${VdsName} (User: ${UserName}).

代码	名称	严重性	信息
924	NETWORK_ADD_BOND	Info	Add bond: ${BondName} with interfaces: ${Interfaces} for Host: ${VdsName} by ${UserName}.
925	NETWORK_ADD_BOND_FAILED	Error	Failed to add bond: ${BondName} with interfaces: ${Interfaces} for Host: ${VdsName} (User:${UserName}).
926	NETWORK_REMOVE_BOND	Info	Remove bond: ${BondName} for Host: ${VdsName} (User:${UserName}).
927	NETWORK_REMOVE_BOND_FAILED	Error	Failed to remove bond: ${BondName} for Host: ${VdsName} (User:${UserName}).
928	NETWORK_VDS_NETWORK_MATCH_CLUSTER	Info	Vds ${VdsName} network match to cluster ${VdsGroupName}
929	NETWORK_VDS_NETWORK_NOT_MATCH_CLUSTER	Error	Vds ${VdsName} network does not match to cluster ${VdsGroupName}
930	NETWORK_REMOVE_VM_INTERFACE	Info	Interface ${InterfaceName} (${InterfaceType}) was removed from VM ${VmName}. (User: ${UserName})
931	NETWORK_REMOVE_VM_INTERFACE_FAILED	Error	Failed to remove Interface ${InterfaceName} (${InterfaceType}) from VM ${VmName}. (User: ${UserName})
932	NETWORK_ADD_VM_INTERFACE	Info	Interface ${InterfaceName} (${InterfaceType}) was added to VM ${VmName}. (User: ${UserName})
933	NETWORK_ADD_VM_INTERFACE_FAILED	Error	Failed to add Interface ${InterfaceName} (${InterfaceType}) to VM ${VmName}. (User: ${UserName})

代码	名称	严重性	信息
934	NETWORK_UPDATE_VM_INTERFACE	Info	Interface ${InterfaceName} (${InterfaceType}) was updated for VM ${VmName}. ${LinkState} (User: ${UserName})
935	NETWORK_UPDATE_VM_INTERFACE_FAILED	Error	Failed to update Interface ${InterfaceName} (${InterfaceType}) for VM ${VmName}. (User: ${UserName})
936	NETWORK_ADD_TEMPLATE_INTERFACE	Info	Interface ${InterfaceName} (${InterfaceType}) was added to Template ${VmTemplateName}. (User: ${UserName})
937	NETWORK_ADD_TEMPLATE_INTERFACE_FAILED	Error	Failed to add Interface ${InterfaceName} (${InterfaceType}) to Template ${VmTemplateName}. (User: ${UserName})
938	NETWORK_REMOVE_TEMPLATE_INTERFACE	Info	Interface ${InterfaceName} (${InterfaceType}) was removed from Template ${VmTemplateName}. (User: ${UserName})
939	NETWORK_REMOVE_TEMPLATE_INTERFACE_FAILED	Error	Failed to remove Interface ${InterfaceName} (${InterfaceType}) from Template ${VmTemplateName}. (User: ${UserName})
940	NETWORK_UPDATE_TEMPLATE_INTERFACE	Info	Interface ${InterfaceName} (${InterfaceType}) was updated for Template ${VmTemplateName}. (User: ${UserName})
941	NETWORK_UPDATE_TEMPLATE_INTERFACE_FAILED	Error	Failed to update Interface ${InterfaceName} (${InterfaceType}) for Template ${VmTemplateName}. (User: ${UserName})

代码	名称	严重性	信息
942	NETWORK_ADD_NETWORK	Info	Network ${NetworkName} was added to Data Center: ${StoragePoolName}
943	NETWORK_ADD_NETWORK_FAILED	Error	Failed to add Network ${NetworkName} to Data Center: ${StoragePoolName}
944	NETWORK_REMOVE_NETWORK	Info	Network ${NetworkName} was removed from Data Center: ${StoragePoolName}
945	NETWORK_REMOVE_NETWORK_FAILED	Error	Failed to remove Network ${NetworkName} from Data Center: ${StoragePoolName}
946	NETWORK_ATTACH_NETWORK_TO_VDS_GROUP	Info	Network ${NetworkName} attached to Cluster ${VdsGroupName}
947	NETWORK_ATTACH_NETWORK_TO_VDS_GROUP_FAILED	Error	Failed to attach Network ${NetworkName} to Cluster ${VdsGroupName}
948	NETWORK_DETACH_NETWORK_TO_VDS_GROUP	Info	Network ${NetworkName} detached from Cluster ${VdsGroupName}
949	NETWORK_DETACH_NETWORK_TO_VDS_GROUP_FAILED	Error	Failed to detach Network ${NetworkName} from Cluster ${VdsGroupName}
950	USER_ADD_STORAGE_POOL	Info	Data Center ${StoragePoolName}, Compatibility Version ${CompatibilityVersion} and Quota Type ${QuotaEnforcementType} was added by ${UserName}
951	USER_ADD_STORAGE_POOL_FAILED	Error	Failed to add Data Center ${StoragePoolName}. (User: ${UserName})
952	USER_UPDATE_STORAGE_POOL	Info	Data Center ${StoragePoolName} was updated by ${UserName}

代码	名称	严重性	信息
953	USER_UPDATE_STORAGE_POOL_FAILED	Error	Failed to update Data Center ${StoragePoolName}. (User: ${UserName})
954	USER_REMOVE_STORAGE_POOL	Info	Data Center ${StoragePoolName} was removed by ${UserName}
955	USER_REMOVE_STORAGE_POOL_FAILED	Error	Failed to remove Data Center ${StoragePoolName}. (User: ${UserName})
956	USER_ADD_STORAGE_DOMAIN	Info	Storage Domain ${StorageDomainName} was added by ${UserName}
957	USER_ADD_STORAGE_DOMAIN_FAILED	Error	Failed to add Storage Domain ${StorageDomainName}. (User: ${UserName})
958	USER_UPDATE_STORAGE_DOMAIN	Info	Storage Domain ${StorageDomainName} was updated by ${UserName}
959	USER_UPDATE_STORAGE_DOMAIN_FAILED	Error	Failed to update Storage Domain ${StorageDomainName}. (User: ${UserName})
960	USER_REMOVE_STORAGE_DOMAIN	Info	Storage Domain ${StorageDomainName} was removed by ${UserName}
961	USER_REMOVE_STORAGE_DOMAIN_FAILED	Error	Failed to remove Storage Domain ${StorageDomainName}. (User: ${UserName})
962	USER_ATTACH_STORAGE_DOMAIN_TO_POOL	Info	Storage Domain ${StorageDomainName} was attached to Data Center ${StoragePoolName} by ${UserName}
963	USER_ATTACH_STORAGE_DOMAIN_TO_POOL_FAILED	Error	Failed to attach Storage Domain ${StorageDomainName} to Data Center ${StoragePoolName}. (User: ${UserName})

代码	名称	严重性	信息
964	USER_DETACH_STORAGE_DOMAIN_FROM_POOL	Info	Storage Domain ${StorageDomainName} was detached from Data Center ${StoragePoolName} by ${UserName}
965	USER_DETACH_STORAGE_DOMAIN_FROM_POOL_FAILED	Error	Failed to detach Storage Domain ${StorageDomainName} to Data Center ${StoragePoolName}. (User: ${UserName})
966	USER_ACTIVATED_STORAGE_DOMAIN	Info	Storage Domain ${StorageDomainName} (Data Center ${StoragePoolName}) was activated by ${UserName}
967	USER_ACTIVATE_STORAGE_DOMAIN_FAILED	Error	Failed to activate Storage Domain ${StorageDomainName} (Data Center ${StoragePoolName}) by ${UserName}
968	USER_DEACTIVATED_STORAGE_DOMAIN	Info	Storage Domain ${StorageDomainName} (Data Center ${StoragePoolName}) was deactivated and has moved to 'Preparing for maintenance' until it will no longer be accessed by any Host of the Data Center.
969	USER_DEACTIVATE_STORAGE_DOMAIN_FAILED	Error	Failed to deactivate Storage Domain ${StorageDomainName} (Data Center ${StoragePoolName}).
970	SYSTEM_DEACTIVATED_STORAGE_DOMAIN	Warning	Storage Domain ${StorageDomainName} (Data Center ${StoragePoolName}) was deactivated by system because it's not visible by any of the hosts.
971	SYSTEM_DEACTIVATE_STORAGE_DOMAIN_FAILED	Error	Failed to deactivate Storage Domain ${StorageDomainName} (Data Center ${StoragePoolName}).

代码	名称	严重性	信息
972	USER_EXTENDED_STORAGE_DOMAIN	Info	Storage ${StorageDomainName} has been extended by ${UserName}. Please wait for refresh.
973	USER_EXTENDED_STORAGE_DOMAIN_FAILED	Error	Failed to extend Storage Domain ${StorageDomainName}. (User: ${UserName})
974	USER_REMOVE_VG	Info	Volume group ${VgId} was removed by ${UserName}.
975	USER_REMOVE_VG_FAILED	Error	Failed to remove Volume group ${VgId}. (User: UserName)
976	USER_ACTIVATE_STORAGE_POOL	Info	Data Center ${StoragePoolName} was activated. (User: ${UserName})
977	USER_ACTIVATE_STORAGE_POOL_FAILED	Error	Failed to activate Data Center ${StoragePoolName}. (User: ${UserName})
978	SYSTEM_FAILED_CHANGE_STORAGE_POOL_STATUS	Error	Failed to change Data Center ${StoragePoolName} status.
979	SYSTEM_CHANGE_STORAGE_POOL_STATUS_NO_HOST_FOR_SPM	Error	Fencing failed on Storage Pool Manager ${VdsName} for Data Center ${StoragePoolName}. Setting status to Non-Operational.
980	SYSTEM_CHANGE_STORAGE_POOL_STATUS_PROBLEMATIC	Warning	Invalid status on Data Center ${StoragePoolName}. Setting status to Non Responsive.
981	USER_FORCE_REMOVE_STORAGE_DOMAIN	Info	Storage Domain ${StorageDomainName} was forcibly removed by ${UserName}
982	USER_FORCE_REMOVE_STORAGE_DOMAIN_FAILED	Error	Failed to forcibly remove Storage Domain ${StorageDomainName}. (User: ${UserName})

代码	名称	严重性	信息
983	RECONSTRUCT_MASTER_FAILED_NO_MASTER	Warning	No valid Data Storage Domains are available in Data Center ${StoragePoolName} (please check your storage infrastructure).
984	RECONSTRUCT_MASTER_DONE	Info	Reconstruct Master Domain for Data Center ${StoragePoolName} completed.
985	RECONSTRUCT_MASTER_FAILED	Error	Failed to Reconstruct Master Domain for Data Center ${StoragePoolName}.
986	SYSTEM_CHANGE_STORAGE_POOL_STATUS_PROBLEMATIC_SEARCHING_NEW_SPM	Warning	Data Center is being initialized, please wait for initialization to complete.
987	SYSTEM_CHANGE_STORAGE_POOL_STATUS_PROBLEMATIC_WITH_ERROR	Warning	Invalid status on Data Center ${StoragePoolName}. Setting Data Center status to Non Responsive (On host ${VdsName}, Error: ${Error}).
988	USER_CONNECT_HOSTS_TO_LUN_FAILED	Error	Failed to connect Host ${VdsName} to device. (User: ${UserName})
989	SYSTEM_CHANGE_STORAGE_POOL_STATUS_PROBLEMATIC_FROM_NON_OPERATIONAL	Info	Try to recover Data Center ${StoragePoolName}. Setting status to Non Responsive.
990	SYSTEM_MASTER_DOMAIN_NOT_IN_SYNC	Warning	Sync Error on Master Domain between Host ${VdsName} and oVirt Engine. Domain: ${StorageDomainName} is marked as Master in oVirt Engine database but not on the Storage side. Please consult with Support on how to fix this issue.
991	RECOVERY_STORAGE_POOL	Info	Data Center ${StoragePoolName} was recovered by ${UserName}

代码	名称	严重性	信息
992	RECOVERY_STORAGE_POOL_FAILED	Error	Failed to recover Data Center ${StoragePoolName} (User:${UserName})
993	SYSTEM_CHANGE_STORAGE_POOL_STATUS_RESET_IRS	Info	Data Center ${StoragePoolName} was reset. Setting status to Non Responsive (Elect new Storage Pool Manager).
994	CONNECT_STORAGE_SERVERS_FAILED	Warning	Failed to connect Host ${VdsName} to Storage Servers
995	CONNECT_STORAGE_POOL_FAILED	Warning	Failed to connect Host ${VdsName} to Storage Pool ${StoragePoolName}
996	STORAGE_DOMAIN_ERROR	Error	The error message for connection ${Connection} returned by VDSM was: ${ErrorMessage}
997	REFRESH_REPOSITORY_IMAGE_LIST_FAILED	Error	Refresh image list failed for domain(s): ${imageDomains}. Please check domain activity.
998	REFRESH_REPOSITORY_IMAGE_LIST_SUCCEEDED	Info	Refresh image list succeeded for domain(s): ${imageDomains}
999	STORAGE_ALERT_VG_METADATA_CRITICALLY_FULL	Error	The system has reached the 80% watermark on the VG metadata area size on ${StorageDomainName}.\nThis is due to a high number of Vdisks or large Vdisks size allocated on this specific VG.
1000	STORAGE_ALERT_SMALL_VG_METADATA	Warning	The allocated VG metadata area size is smaller than 50MB on ${StorageDomainName},\nwhich might limit its capacity (the number of Vdisks and/or their size).

代码	名称	严重性	信息
1001	USER_RUN_VM_FAILURE_STATELESS_SNAPSHOT_LEFT	Error	Failed to start VM ${VmName}, because exist snapshot for stateless state. Snapshot will be deleted.
1002	USER_ATTACH_STORAGE_DOMAINS_TO_POOL	Info	Storage Domains were attached to Data Center ${StoragePoolName} by ${UserName}
1003	USER_ATTACH_STORAGE_DOMAINS_TO_POOL_FAILED	Error	Failed to attach Storage Domains to Data Center ${StoragePoolName}. (User: ${UserName})
1004	STORAGE_DOMAIN_TASKS_ERROR	Warning	Storage Domain ${StorageDomainName} is down while there are tasks running on it. These tasks may fail.
1005	UPDATE_OVF_FOR_STORAGE_POOL_FAILED	Warning	Failed to update VMs/Templates OVF data in Data Center ${StoragePoolName}.
1006	UPGRADE_STORAGE_POOL_ENCOUNTERED_PROBLEMS	Warning	Data Center ${StoragePoolName} has encountered problems during upgrade process.
1007	REFRESH_REPOSITORY_IMAGE_LIST_INCOMPLETE	Warning	Refresh image list probably incomplete for domain ${imageDomain}, only ${imageListSize} images discovered.
1008	NUNBER_OF_LVS_ON_STORAGE_DOMAIN_EXCEEDED_THRESHOLD	Warning	The number of LVs on the domain ${storageDomainName} exceeded ${maxNumOfLVs}, you are approaching the limit where performance may degrade.
1010	RELOAD_CONFIGURATIONS_SUCCESS	Info	System Configurations reloaded successfully.

代码	名称	严重性	信息
1011	RELOAD_CONFIGURATIONS_FAILURE	Error	System Configurations failed to reload.
1012	NETWORK_ACTIVATE_VM_INTERFACE_SUCCESS	Info	Network Interface ${InterfaceName} (${InterfaceType}) was plugged to VM ${VmName}. (User: ${UserName})
1013	NETWORK_ACTIVATE_VM_INTERFACE_FAILURE	Error	Failed to plug Network Interface ${InterfaceName} (${InterfaceType}) to VM ${VmName}. (User: ${UserName})
1014	NETWORK_DEACTIVATE_VM_INTERFACE_SUCCESS	Info	Network Interface ${InterfaceName} (${InterfaceType}) was unplugged from VM ${VmName}. (User: ${UserName})
1015	NETWORK_DEACTIVATE_VM_INTERFACE_FAILURE	Error	Failed to unplug Network Interface ${InterfaceName} (${InterfaceType}) from VM ${VmName}. (User: ${UserName})
1016	UPDATE_FOR_OVF_STORES_FAILED	Warning	Failed to update OVF disks ${DisksIds}, OVF data isn't updated on those OVF stores (Data Center ${DataCenterName}, Storage Domain ${StorageDomainName}).
1017	RETRIEVE_OVF_STORE_FAILED	Warning	Failed to retrieve VMs and Templates from the OVF disk of Storage Domain ${StorageDomainName}.
1018	OVF_STORE_DOES_NOT_EXISTS	Warning	This Data center compatibility version does not support importing a data domain with its entities (VMs and Templates). The imported domain will be imported without them.

代码	名称	严重性	信息
1019	UPDATE_DESCRIPTION_FOR_DISK_FAILED	Error	Failed to update the meta data description of disk ${DiskName} (Data Center ${DataCenterName}, Storage Domain ${StorageDomainName}).
1020	UPDATE_DESCRIPTION_FOR_DISK_SKIPPED_SINCE_STORAGE_DOMAIN_NOT_ACTIVE	Warning	Not updating the metadata of Disk ${DiskName} (Data Center ${DataCenterName}. Since the Storage Domain ${StorageDomainName} is not in active.
1021	RETRIEVE_UNREGISTERED_ENTITIES_NOT_SUPPORTED_IN_DC_VERSION	Warning	Skipping retrieval attempt of VMs and Templates from the OVF_STORE disk of Storage Domain ${StorageDomainName} since it is not supported by the Data Center version.
1022	USER_REFRESH_LUN_STORAGE_DOMAIN	Info	Resize LUNs operation succeeded.
1023	USER_REFRESH_LUN_STORAGE_DOMAIN_FAILED	Error	Failed to resize LUNs.
1024	USER_REFRESH_LUN_STORAGE_DIFFERENT_SIZE_DOMAIN_FAILED	Error	Failed to resize LUNs.\n Not all the hosts are seeing the same LUN size.
1025	VM_PAUSED	Info	VM ${VmName} has been paused.
1026	FAILED_TO_STORE_ENTIRE_DISK_FIELD_IN_DISK_DESCRIPTION_METADATA	Warning	Failed to store field ${DiskFieldName} as a part of ${DiskAlias}'s description metadata due to storage space limitations. The field ${DiskFieldName} will be truncated.

代码	名称	严重性	信息
1027	FAILED_TO_STORE_ENTIRE_DISK_FIELD_AND_REST_OF_FIELDS_IN_DISK_DESCRIPTION_METADATA	Warning	Failed to store field ${DiskFieldName} as a part of ${DiskAlias}'s description metadata due to storage space limitations. The value will be truncated and the following fields will not be stored at all: ${DiskFieldsNames}.
1028	FAILED_TO_STORE_DISK_FIELDS_IN_DISK_DESCRIPTION_METADATA	Warning	Failed to store the following fields in the description metadata of disk ${DiskAlias} due to storage space limitations: ${DiskFieldsNames}.
1029	STORAGE_DOMAIN_MOVED_TO_MAINTENANCE	Info	Storage Domain ${StorageDomainName} (Data Center ${StoragePoolName}) successfully moved to Maintenance as it's no longer accessed by any Host of the Data Center.
1030	USER_DEACTIVATED_LAST_MASTER_STORAGE_DOMAIN	Info	Storage Domain ${StorageDomainName} (Data Center ${StoragePoolName}) was deactivated.
1098	NETWORK_UPDATE_DISPLAY_FOR_HOST_WITH_ACTIVE_VM	Warning	Display Network was updated on Host ${VdsName} with active VMs attached. The change will be applied to those VMs after their next reboot. Running VMs might loose display connectivity until then.
1099	NETWORK_UPDATE_DISPLAY_FOR_CLUSTER_WITH_ACTIVE_VM	Warning	Display Network (${NetworkName}) was updated for Cluster ${VdsGroupName} with active VMs attached. The change will be applied to those VMs after their next reboot.

代码	名称	严重性	信息
1100	NETWORK_UPDATE_DISPLAY_TO_VDS_GROUP	Info	Update Display Network (${NetworkName}) for Cluster ${VdsGroupName}. (User: ${UserName})
1101	NETWORK_UPDATE_DISPLAY_TO_VDS_GROUP_FAILED	Error	Failed to update Display Network (${NetworkName}) for Cluster ${VdsGroupName}. (User: ${UserName})
1102	NETWORK_UPDATE_NETWORK_TO_VDS_INTERFACE	Info	Update Network ${NetworkName} in Host ${VdsName}. (User: ${UserName})
1103	NETWORK_UPDATE_NETWORK_TO_VDS_INTERFACE_FAILED	Error	Failed to update Network ${NetworkName} in Host ${VdsName}. (User: ${UserName})
1104	NETWORK_COMMINT_NETWORK_CHANGES	Info	Network changes were saved on host ${VdsName}
1105	NETWORK_COMMINT_NETWORK_CHANGES_FAILED	Error	Failed to commit network changes on ${VdsName}
1106	NETWORK_HOST_USING_WRONG_CLUSER_VLAN	Warning	${VdsName} is having wrong vlan id: ${VlanIdHost}, expected vlan id: ${VlanIdCluster}
1107	NETWORK_HOST_MISSING_CLUSER_VLAN	Warning	${VdsName} is missing vlan id: ${VlanIdCluster} that is expected by the cluster
1108	VDS_NETWORK_MTU_DIFFER_FROM_LOGICAL_NETWORK	Info	
1109	BRIDGED_NETWORK_OVER_MULTIPLE_INTERFACES	Warning	Bridged network ${NetworkName} is attached to multiple interfaces: ${Interfaces} on Host ${VdsName}.

代码	名称	严重性	信息
1110	VDS_NETWORKS_OUT_OF_SYNC	Warning	Host ${VdsName}'s following network(s) are not synchronized with their Logical Network configuration: ${Networks}.
1112	NETWORK_UPDTAE_NETWORK_ON_CLUSTER	Info	Network ${NetworkName} on Cluster ${VdsGroupName} updated.
1113	NETWORK_UPDTAE_NETWORK_ON_CLUSTER_FAILED	Error	Failed to update Network ${NetworkName} on Cluster ${VdsGroupName}.
1114	NETWORK_UPDATE_NETWORK	Info	Network ${NetworkName} was updated on Data Center: ${StoragePoolName}
1115	NETWORK_UPDATE_NETWORK_FAILED	Error	Failed to update Network ${NetworkName} on Data Center: ${StoragePoolName}
1116	NETWORK_UPDATE_VM_INTERFACE_LINK_UP	Info	Link State is UP.
1117	NETWORK_UPDATE_VM_INTERFACE_LINK_DOWN	Info	Link State is DOWN.
1118	INVALID_INTERFACE_FOR_MANAGEMENT_NETWORK_CONFIGURATION	Error	Failed to configure management network on host ${VdsName}. Host ${VdsName} has an invalid interface ${InterfaceName} for the management network configuration.
1119	VLAN_ID_MISMATCH_FOR_MANAGEMENT_NETWORK_CONFIGURATION	Error	Failed to configure management network on host ${VdsName}. Host ${VdsName} has an interface ${InterfaceName} for the management network configuration with VLAN-ID (${VlanId}), which is different from data-center definition (${MgmtVlanId}).

代码	名称	严重性	信息
1120	SETUP_NETWORK_FAILED_FOR_MANAGEMENT_NETWORK_CONFIGURATION	Error	Failed to configure management network on host ${VdsName} due to setup networks failure.
1121	PERSIST_NETWORK_FAILED_FOR_MANAGEMENT_NETWORK	Warning	Failed to activate host ${VdsName} due to failure in persisting the management network configuration.
1122	ADD_VNIC_PROFILE	Info	VM network interface profile ${VnicProfileName} was added to network ${NetworkName} in Data Center: ${DataCenterName}. (User: ${UserName})
1123	ADD_VNIC_PROFILE_FAILED	Error	Failed to add VM network interface profile ${VnicProfileName} to network ${NetworkName} in Data Center: ${DataCenterName} (User: ${UserName})
1124	UPDATE_VNIC_PROFILE	Info	VM network interface profile ${VnicProfileName} was updated for network ${NetworkName} in Data Center: ${DataCenterName}. (User: ${UserName})
1125	UPDATE_VNIC_PROFILE_FAILED	Error	Failed to update VM network interface profile ${VnicProfileName} for network ${NetworkName} in Data Center: ${DataCenterName}. (User: ${UserName})
1126	REMOVE_VNIC_PROFILE	Info	VM network interface profile ${VnicProfileName} was removed from network ${NetworkName} in Data Center: ${DataCenterName}. (User: ${UserName})

代码	名称	严重性	信息
1127	REMOVE_VNIC_PROFILE_FAILED	Error	Failed to remove VM network interface profile ${VnicProfileName} from network ${NetworkName} in Data Center: ${DataCenterName}. (User: ${UserName})
1128	NETWORK_WITHOUT_INTERFACES	Warning	Network ${NetworkName} is not attached to any interface on host ${VdsName}.
1129	VNIC_PROFILE_UNSUPPORTED_FEATURES	Warning	VM ${VmName} has network interface ${NicName} which is using profile ${VnicProfile} with unsupported feature(s) '${UnsupportedFeatures}' by VM cluster ${VdsGroupName} (version ${CompatibilityVersion}).
1131	REMOVE_NETWORK_BY_LABEL_FAILED	Error	Network ${Network} cannot be removed from the following hosts: ${HostNames} in data-center ${StoragePoolName}.
1132	LABEL_NETWORK	Info	Network ${NetworkName} was labeled ${Label} in data-center ${StoragePoolName}.
1133	LABEL_NETWORK_FAILED	Error	Failed to label network ${NetworkName} with label ${Label} in data-center ${StoragePoolName}.
1134	UNLABEL_NETWORK	Info	Network ${NetworkName} was unlabeled in data-center ${StoragePoolName}.
1135	UNLABEL_NETWORK_FAILED	Error	Failed to unlabel network ${NetworkName} in data-center ${StoragePoolName}.
1136	LABEL_NIC	Info	Network interface card ${NicName} was labeled ${Label} on host ${VdsName}.

代码	名称	严重性	信息
1137	LABEL_NIC_FAILED	Error	Failed to label network interface card ${NicName} with label ${Label} on host ${VdsName}.
1138	UNLABEL_NIC	Info	Label ${Label} was removed from network interface card ${NicName} on host ${VdsName}.
1139	UNLABEL_NIC_FAILED	Error	Failed to remove label ${Label} from network interface card ${NicName} on host ${VdsName}.
1140	SUBNET_REMOVED	Info	Subnet ${SubnetName} was removed from provider ${ProviderName}. (User: ${UserName})
1141	SUBNET_REMOVAL_FAILED	Error	Failed to remove subnet ${SubnetName} from provider ${ProviderName}. (User: ${UserName})
1142	SUBNET_ADDED	Info	Subnet ${SubnetName} was added on provider ${ProviderName}. (User: ${UserName})
1143	SUBNET_ADDITION_FAILED	Error	Failed to add subnet ${SubnetName} on provider ${ProviderName}. (User: ${UserName})
1144	CONFIGURE_NETWORK_BY_LABELS_WHEN_CHANGING_CLUSTER_FAILED	Error	Failed to configure networks on host ${VdsName} while changing its cluster.
1145	PERSIST_NETWORK_ON_HOST	Info	(${Sequence}/${Total}): Applying changes for network(s) ${NetworkNames} on host ${VdsName}. (User: ${UserName})

代码	名称	严重性	信息
1146	PERSIST_NETWORK_ON_HOST_FINISHED	Info	(${Sequence}/${Total}): Successfully applied changes for network(s) ${NetworkNames} on host ${VdsName}. (User: ${UserName})
1147	PERSIST_NETWORK_ON_HOST_FAILED	Error	(${Sequence}/${Total}): Failed to apply changes for network(s) ${NetworkNames} on host ${VdsName}. (User: ${UserName})
1148	MULTI_UPDATE_NETWORK_NOT_POSSIBLE	Warning	Cannot apply network ${NetworkName} changes to hosts on unsupported data center ${StoragePoolName}. (User: ${UserName})
1149	REMOVE_PORT_FROM_EXTERNAL_PROVIDER_FAILED	Warning	Failed to remove vNIC ${NicName} from external network provider ${ProviderName}. The vNIC can be identified on the provider by device id ${NicId}.
1150	IMPORTEXPORT_EXPORT_VM	Info	Vm ${VmName} was exported successfully to ${StorageDomainName}
1151	IMPORTEXPORT_EXPORT_VM_FAILED	Error	Failed to export Vm ${VmName} to ${StorageDomainName}
1152	IMPORTEXPORT_IMPORT_VM	Info	Vm ${VmName} was imported successfully to Data Center ${StoragePoolName}, Cluster ${VdsGroupName}
1153	IMPORTEXPORT_IMPORT_VM_FAILED	Error	Failed to import Vm ${VmName} to Data Center ${StoragePoolName}, Cluster ${VdsGroupName}
1154	IMPORTEXPORT_REMOVE_TEMPLATE	Info	Template ${VmTemplateName} was removed from ${StorageDomainName}

代码	名称	严重性	信息
1155	IMPORTEXPORT_REMOVE_TEMPLATE_FAILED	Error	Failed to remove Template ${VmTemplateName} from ${StorageDomainName}
1156	IMPORTEXPORT_EXPORT_TEMPLATE	Info	Template ${VmTemplateName} was exported successfully to ${StorageDomainName}
1157	IMPORTEXPORT_EXPORT_TEMPLATE_FAILED	Error	Failed to export Template ${VmTemplateName} to ${StorageDomainName}
1158	IMPORTEXPORT_IMPORT_TEMPLATE	Info	Template ${VmTemplateName} was imported successfully to Data Center ${StoragePoolName}, Cluster ${VdsGroupName}
1159	IMPORTEXPORT_IMPORT_TEMPLATE_FAILED	Error	Failed to import Template ${VmTemplateName} to Data Center ${StoragePoolName}, Cluster ${VdsGroupName}
1160	IMPORTEXPORT_REMOVE_VM	Info	Vm ${VmName} was removed from ${StorageDomainName}
1161	IMPORTEXPORT_REMOVE_VM_FAILED	Error	Failed to remove Vm ${VmName} remove from ${StorageDomainName}
1162	IMPORTEXPORT_STARTING_EXPORT_VM	Info	Starting export Vm ${VmName} to ${StorageDomainName}
1163	IMPORTEXPORT_STARTING_IMPORT_TEMPLATE	Info	Starting to import Template ${VmTemplateName} to Data Center ${StoragePoolName}, Cluster ${VdsGroupName}
1164	IMPORTEXPORT_STARTING_EXPORT_TEMPLATE	Info	Starting to export Template ${VmTemplateName} to ${StorageDomainName}
1165	IMPORTEXPORT_STARTING_IMPORT_VM	Info	Starting to import Vm ${VmName} to Data Center ${StoragePoolName}, Cluster ${VdsGroupName}

代码	名称	严重性	信息
1166	IMPORTEXPORT_STARTING_REMOVE_TEMPLATE	Info	Starting to remove Template ${VmTemplateName} remove ${StorageDomainName}
1167	IMPORTEXPORT_STARTING_REMOVE_VM	Info	Starting to remove Vm ${VmName} remove from ${StorageDomainName}
1168	IMPORTEXPORT_FAILED_TO_IMPORT_VM	Warning	Failed to read VM '${ImportedVmName}' OVF, it may be corrupted. Underlying error message: ${ErrorMessage}
1169	IMPORTEXPORT_FAILED_TO_IMPORT_TEMPLATE	Warning	Failed to read Template '${Template}' OVF, it may be corrupted. Underlying error message: ${ErrorMessage}
1170	IMPORTEXPORT_IMPORT_TEMPLATE_INVALID_INTERFACES	Normal	While importing Template ${EntityName}, the Network/s ${Networks} were found to be Non-VM Networks or do not exist in Cluster. Network Name was not set in the Interface/s ${Interfaces}.
1171	USER_ACCOUNT_PASSWORD_EXPIRED	Error	User ${UserName} cannot login, as the user account password has expired. Please contact the system administrator.
1172	AUTH_FAILED_INVALID_CREDENTIALS	Error	User ${UserName} cannot login, please verify the username and password.
1173	AUTH_FAILED_CLOCK_SKEW_TOO_GREAT	Error	User ${UserName} cannot login, the engine clock is not synchronized with directory services. Please contact the system administrator.
1174	AUTH_FAILED_NO_KDCS_FOUND	Error	User ${UserName} cannot login, authentication domain cannot be found. Please contact the system administrator.

代码	名称	严重性	信息
1175	AUTH_FAILED_DNS_ERROR	Error	User ${UserName} cannot login, there's an error in DNS configuration. Please contact the system administrator.
1176	AUTH_FAILED_OTHER	Error	User ${UserName} cannot login, unknown kerberos error. Please contact the system administrator.
1177	AUTH_FAILED_DNS_COMMUNICATION_ERROR	Error	User ${UserName} cannot login, cannot lookup DNS for SRV records. Please contact the system administrator.
1178	AUTH_FAILED_CONNECTION_TIMED_OUT	Error	User ${UserName} cannot login, connection to LDAP server has timed out. Please contact the system administrator.
1179	AUTH_FAILED_WRONG_REALM	Error	User ${UserName} cannot login, please verify your domain name.
1180	AUTH_FAILED_CONNECTION_ERROR	Error	User ${UserName} cannot login, connection refused or some configuration problems exist. Possible DNS error. Please contact the system administrator.
1181	AUTH_FAILED_CANNOT_FIND_LDAP_SERVER_FOR_DOMAIN	Error	User ${UserName} cannot login, cannot find valid LDAP server for domain. Please contact the system administrator.
1182	AUTH_FAILED_NO_USER_INFORMATION_WAS_FOUND	Error	User ${UserName} cannot login, no user information was found. Please contact the system administrator.
1183	AUTH_FAILED_CLIENT_NOT_FOUND_IN_KERBEROS_DATABASE	Error	User ${UserName} cannot login, user was not found in domain. Please contact the system administrator.

代码	名称	严重性	信息
1184	AUTH_FAILED_INTERNAL_KERBEROS_ERROR	Error	User ${UserName} cannot login, an internal error has ocurred in the Kerberos implementation of the JVM. Please contact the system administrator.
1185	USER_ACCOUNT_EXPIRED	Error	The account for ${UserName} got expired. Please contact the system administrator.
1186	IMPORTEXPORT_NO_PROXY_HOST_AVAILABLE_IN_DC	Error	No Host in Data Center '${StoragePoolName}' can serve as a proxy to retrieve remote VMs information (User: ${UserName}).
1187	IMPORTEXPORT_HOST_CANNOT_SERVE_AS_PROXY	Error	Host ${VdsName} cannot be used as a proxy to retrieve remote VMs information since it is not up (User: ${UserName}).
1189	IMPORTEXPORT_IMPORT_VM_FAILED_UPDATING_OVF	Error	Failed to import Vm ${VmName} to Data Center ${StoragePoolName}, Cluster ${VdsGroupName}, could not update VM data in export.
1190	USER_RESTORE_FROM_SNAPSHOT_START	Info	Restoring VM ${VmName} from snapshot started by user ${UserName}.
1191	VM_DISK_ALREADY_CHANGED	Info	CD ${DiskName} is already inserted to VM ${VmName}, disk change action was skipped. User: ${UserName}.
1192	VM_DISK_ALREADY_EJECTED	Info	CD is already ejected from VM ${VmName}, disk change action was skipped. User: ${UserName}.
1193	IMPORTEXPORT_STARTING_CONVERT_VM	Info	Starting to convert Vm ${VmName}
1194	IMPORTEXPORT_CONVERT_FAILED	Info	Failed to convert Vm ${VmName}

代码	名称	严重性	信息
1195	IMPORTEXPORT_CANNOT_GET_OVF	Info	Failed to get the configuration of converted Vm ${VmName}
1196	IMPORTEXPORT_INVALID_OVF	Info	Failed to process the configuration of converted Vm ${VmName}
1200	ENTITY_RENAMED	Info	${EntityType} ${OldEntityName} was renamed from ${OldEntityName} to ${NewEntityName} by ${UserName}.
1201	UPDATE_HOST_NIC_VFS_CONFIG	Info	The VFs configuration of network interface card ${NicName} on host ${VdsName} was updated.
1202	UPDATE_HOST_NIC_VFS_CONFIG_FAILED	Error	Failed to update the VFs configuration of network interface card ${NicName} on host ${VdsName}.
1203	ADD_VFS_CONFIG_NETWORK	Info	Network ${NetworkName} was added to the VFs configuration of network interface card ${NicName} on host ${VdsName}.
1204	ADD_VFS_CONFIG_NETWORK_FAILED	Info	Failed to add ${NetworkName} to the VFs configuration of network interface card ${NicName} on host ${VdsName}.
1205	REMOVE_VFS_CONFIG_NETWORK	Info	Network ${NetworkName} was removed from the VFs configuration of network interface card ${NicName} on host ${VdsName}.
1206	REMOVE_VFS_CONFIG_NETWORK_FAILED	Info	Failed to remove ${NetworkName} from the VFs configuration of network interface card ${NicName} on host ${VdsName}.

代码	名称	严重性	信息
1207	ADD_VFS_CONFIG_LABEL	Info	Label ${Label} was added to the VFs configuration of network interface card ${NicName} on host ${VdsName}.
1208	ADD_VFS_CONFIG_LABEL_FAILED	Info	Failed to add ${Label} to the VFs configuration of network interface card ${NicName} on host ${VdsName}.
1209	REMOVE_VFS_CONFIG_LABEL	Info	Label ${Label} was removed from the VFs configuration of network interface card ${NicName} on host ${VdsName}.
1210	REMOVE_VFS_CONFIG_LABEL_FAILED	Info	Failed to remove ${Label} from the VFs configuration of network interface card ${NicName} on host ${VdsName}.
1300	NUMA_ADD_VM_NUMA_NODE_SUCCESS	Info	Add VM NUMA node successfully.
1301	NUMA_ADD_VM_NUMA_NODE_FAILED	Error	Add VM NUMA node failed.
1310	NUMA_UPDATE_VM_NUMA_NODE_SUCCESS	Info	Update VM NUMA node successfully.
1311	NUMA_UPDATE_VM_NUMA_NODE_FAILED	Error	Update VM NUMA node failed.
1320	NUMA_REMOVE_VM_NUMA_NODE_SUCCESS	Info	Remove VM NUMA node successfully.
1321	NUMA_REMOVE_VM_NUMA_NODE_FAILED	Error	Remove VM NUMA node failed.
1402	USER_LOGIN_ON_BEHALF_FAILED	Error	Failed to execute login on behalf - ${LoginOnBehalfLogInfo}.
2000	USER_HOTPLUG_DISK	Info	VM ${VmName} disk ${DiskAlias} was plugged by ${UserName}.

代码	名称	严重性	信息
2001	USER_FAILED_HOTPLUG_DISK	Error	Failed to plug disk ${DiskAlias} to VM ${VmName} (User: ${UserName}).
2002	USER_HOTUNPLUG_DISK	Info	VM ${VmName} disk ${DiskAlias} was unplugged by ${UserName}.
2003	USER_FAILED_HOTUNPLUG_DISK	Error	Failed to unplug disk ${DiskAlias} from VM ${VmName} (User: ${UserName}).
2004	USER_COPIED_TEMPLATE_DISK	Info	User ${UserName} is copying template disk ${DiskAlias} to domain ${StorageDomainName}.
2005	USER_FAILED_COPY_TEMPLATE_DISK	Error	User ${UserName} failed to copy template disk ${DiskAlias} to domain ${StorageDomainName}.
2006	USER_COPIED_TEMPLATE_DISK_FINISHED_SUCCESS	Info	User ${UserName} finished copying template disk ${DiskAlias} to domain ${StorageDomainName}.
2007	USER_COPIED_TEMPLATE_DISK_FINISHED_FAILURE	Error	User ${UserName} finished with error copying template disk ${DiskAlias} to domain ${StorageDomainName}.
2008	USER_MOVED_VM_DISK	Info	User ${UserName} moving disk ${DiskAlias} to domain ${StorageDomainName}.
2009	USER_FAILED_MOVED_VM_DISK	Error	User ${UserName} failed to move disk ${DiskAlias} to domain ${StorageDomainName}.
2010	USER_MOVED_VM_DISK_FINISHED_SUCCESS	Info	User ${UserName} finished moving disk ${DiskAlias} to domain ${StorageDomainName}.
2011	USER_MOVED_VM_DISK_FINISHED_FAILURE	Error	User ${UserName} have failed to move disk ${DiskAlias} to domain ${StorageDomainName}.

代码	名称	严重性	信息
2012	USER_FINISHED_REMOVE_DISK_NO_DOMAIN	Info	Disk ${DiskAlias} was successfully removed (User ${UserName}).
2013	USER_FINISHED_FAILED_REMOVE_DISK_NO_DOMAIN	Warning	Failed to remove disk ${DiskAlias} (User ${UserName}).
2014	USER_FINISHED_REMOVE_DISK	Info	Disk ${DiskAlias} was successfully removed from domain ${StorageDomainName} (User ${UserName}).
2015	USER_FINISHED_FAILED_REMOVE_DISK	Warning	Failed to remove disk ${DiskAlias} from storage domain ${StorageDomainName} (User: ${UserName}).
2016	USER_ATTACH_DISK_TO_VM	Info	Disk ${DiskAlias} was successfully attached to VM ${VmName} by ${UserName}.
2017	USER_FAILED_ATTACH_DISK_TO_VM	Error	Failed to attach Disk ${DiskAlias} to VM ${VmName} (User: ${UserName}).
2018	USER_DETACH_DISK_FROM_VM	Info	Disk ${DiskAlias} was successfully detached from VM ${VmName} by ${UserName}.
2019	USER_FAILED_DETACH_DISK_FROM_VM	Error	Failed to detach Disk ${DiskAlias} from VM ${VmName} (User: ${UserName}).
2020	USER_ADD_DISK	Info	Add-Disk operation of '${DiskAlias}' was initiated by ${UserName}.
2021	USER_ADD_DISK_FINISHED_SUCCESS	Info	The disk '${DiskAlias}' was successfully added.
2022	USER_ADD_DISK_FINISHED_FAILURE	Error	Add-Disk operation failed to complete.
2023	USER_FAILED_ADD_DISK	Error	Add-Disk operation failed (User: ${UserName}).

代码	名称	严重性	信息
2024	USER_RUN_UNLOCK_ENTITY_SCRIPT	Info	
2025	USER_MOVE_IMAGE_GROUP_FAILED_TO_DELETE_SRC_IMAGE	Warning	Possible failure while deleting ${DiskAlias} from the source Storage Domain ${StorageDomainName} during the move operation. The Storage Domain may be manually cleaned-up from possible leftovers (User:${UserName}).
2026	USER_MOVE_IMAGE_GROUP_FAILED_TO_DELETE_DST_IMAGE	Warning	Possible failure while clearing possible leftovers of ${DiskAlias} from the target Storage Domain ${StorageDomainName} after the move operation failed to copy the image to it properly. The Storage Domain may be manually cleaned-up from possible leftovers (User:${UserName}).
2027	USER_IMPORT_IMAGE	Info	User ${UserName} importing image ${RepoImageName} to domain ${StorageDomainName}.
2028	USER_IMPORT_IMAGE_FINISHED_SUCCESS	Info	User ${UserName} successfully imported image ${RepoImageName} to domain ${StorageDomainName}.
2029	USER_IMPORT_IMAGE_FINISHED_FAILURE	Error	User ${UserName} failed to import image ${RepoImageName} to domain ${StorageDomainName}.
2030	USER_EXPORT_IMAGE	Info	User ${UserName} exporting image ${RepoImageName} to domain ${DestinationStorageDomainName}.
2031	USER_EXPORT_IMAGE_FINISHED_SUCCESS	Info	User ${UserName} successfully exported image ${RepoImageName} to domain ${DestinationStorageDomainName}.

代码	名称	严重性	信息
2032	USER_EXPORT_IMAGE_FINISHED_FAILURE	Error	User ${UserName} failed to export image ${RepoImageName} to domain ${DestinationStorageDomainName}.
2033	HOT_SET_NUMBER_OF_CPUS	Info	Hotplug CPU: changed the number of CPUs on VM ${vmName} from ${previousNumberOfCpus} to ${numberOfCpus}
2034	FAILED_HOT_SET_NUMBER_OF_CPUS	Error	Failed to hot set number of CPUS to VM ${vmName}. Underlying error message: ${ErrorMessage}
2035	USER_ISCSI_BOND_HOST_RESTART_WARNING	Warning	The following Networks has been removed from the iSCSI bond ${IscsiBondName}: ${NetworkNames}. for those changes to take affect, the hosts must be moved to maintenance and activated again.
2036	ADD_DISK_INTERNAL	Info	Add-Disk operation of '${DiskAlias}' was initiated by the system.
2037	ADD_DISK_INTERNAL_FAILURE	Info	Add-Disk operation of '${DiskAlias}' failed to complete.
2038	USER_REMOVE_DISK_INITIATED	Info	Removal of Disk ${DiskAlias} from domain ${StorageDomainName} was initiated by ${UserName}.
2039	HOT_SET_MEMORY	Info	Hotset memory: changed the amount of memory on VM ${vmName} from ${previousMem} to ${newMem}
2040	FAILED_HOT_SET_MEMORY	Error	Failed to hot set memory to VM ${vmName}. Underlying error message: ${ErrorMessage}

代码	名称	严重性	信息
3000	USER_ADD_QUOTA	Info	Quota ${QuotaName} has been added by ${UserName}.
3001	USER_FAILED_ADD_QUOTA	Error	Failed to add Quota ${QuotaName}. The operation was initiated by ${UserName}.
3002	USER_UPDATE_QUOTA	Info	Quota ${QuotaName} has been updated by ${UserName}.
3003	USER_FAILED_UPDATE_QUOTA	Error	Failed to update Quota ${QuotaName}. The operation was initiated by ${UserName}..
3004	USER_DELETE_QUOTA	Info	Quota ${QuotaName} has been deleted by ${UserName}.
3005	USER_FAILED_DELETE_QUOTA	Error	Failed to delete Quota ${QuotaName}. The operation was initiated by ${UserName}..
3006	USER_EXCEEDED_QUOTA_VDS_GROUP_GRACE_LIMIT	Error	Cluster-Quota ${QuotaName} limit exceeded and operation was blocked. Utilization: ${Utilization}, Requested: ${Requested} - Please select a different quota or contact your administrator to extend the quota.
3007	USER_EXCEEDED_QUOTA_VDS_GROUP_LIMIT	Warning	Cluster-Quota ${QuotaName} limit exceeded and entered the grace zone. Utilization: ${Utilization} (It is advised to select a different quota or contact your administrator to extend the quota).
3008	USER_EXCEEDED_QUOTA_VDS_GROUP_THRESHOLD	Warning	Cluster-Quota ${QuotaName} is about to exceed. Utilization: ${Utilization}

代码	名称	严重性	信息
3009	USER_EXCEEDED_QUOTA_STORAGE_GRACE_LIMIT	Error	Storage-Quota ${QuotaName} limit exceeded and operation was blocked. Utilization(used/requested): ${CurrentStorage}%/${Requested}% - Please select a different quota or contact your administrator to extend the quota.
3010	USER_EXCEEDED_QUOTA_STORAGE_LIMIT	Warning	Storage-Quota ${QuotaName} limit exceeded and entered the grace zone. Utilization: ${CurrentStorage}% (It is advised to select a different quota or contact your administrator to extend the quota).
3011	USER_EXCEEDED_QUOTA_STORAGE_THRESHOLD	Warning	Storage-Quota ${QuotaName} is about to exceed. Utilization: ${CurrentStorage}%
3012	QUOTA_STORAGE_RESIZE_LOWER_THEN_CONSUMPTION	Warning	Storage-Quota ${QuotaName}: the new size set for this quota is less than current disk utilization.
3013	MISSING_QUOTA_STORAGE_PARAMETERS_PERMISSIVE_MODE	Warning	Missing Quota for Disk, proceeding since in Permissive (Audit) mode.
3014	MISSING_QUOTA_CLUSTER_PARAMETERS_PERMISSIVE_MODE	Warning	Missing Quota for VM ${VmName}, proceeding since in Permissive (Audit) mode.
3015	USER_EXCEEDED_QUOTA_VDS_GROUP_GRACE_LIMIT_PERMISSIVE_MODE	Warning	Cluster-Quota ${QuotaName} limit exceeded, proceeding since in Permissive (Audit) mode. Utilization: ${Utilization}, Requested: ${Requested} - Please select a different quota or contact your administrator to extend the quota.

代码	名称	严重性	信息
3016	USER_EXCEEDED_QUOTA_STORAGE_GRACE_LIMIT_PERMISSIVE_MODE	Warning	Storage-Quota ${QuotaName} limit exceeded, proceeding since in Permissive (Audit) mode. Utilization(used/requested): ${CurrentStorage}%/${Requested}% - Please select a different quota or contact your administrator to extend the quota.
4000	GLUSTER_VOLUME_CREATE	Info	Gluster Volume ${glusterVolumeName} created on cluster ${vdsGroupName}.
4001	GLUSTER_VOLUME_CREATE_FAILED	Error	Creation of Gluster Volume ${glusterVolumeName} failed on cluster ${vdsGroupName}.
4002	GLUSTER_VOLUME_OPTION_ADDED	Info	Volume Option ${Key}
4003	GLUSTER_VOLUME_OPTION_SET_FAILED	Error	Volume Option ${Key}
4004	GLUSTER_VOLUME_START	Info	Gluster Volume ${glusterVolumeName} of cluster ${vdsGroupName} started.
4005	GLUSTER_VOLUME_START_FAILED	Error	Could not start Gluster Volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4006	GLUSTER_VOLUME_STOP	Info	Gluster Volume ${glusterVolumeName} stopped on cluster ${vdsGroupName}.
4007	GLUSTER_VOLUME_STOP_FAILED	Error	Could not stop Gluster Volume ${glusterVolumeName} on cluster ${vdsGroupName}.
4008	GLUSTER_VOLUME_OPTIONS_RESET	Info	Volume Option ${Key}
4009	GLUSTER_VOLUME_OPTIONS_RESET_FAILED	Error	Could not reset Gluster Volume ${glusterVolumeName} Options on cluster ${vdsGroupName}.

代码	名称	严重性	信息
4010	GLUSTER_VOLUME_DELETE	Info	Gluster Volume ${glusterVolumeName} deleted on cluster ${vdsGroupName}.
4011	GLUSTER_VOLUME_DELETE_FAILED	Error	Could not delete Gluster Volume ${glusterVolumeName} on cluster ${vdsGroupName}.
4012	GLUSTER_VOLUME_REBALANCE_START	Info	Gluster Volume ${glusterVolumeName} rebalance started on cluster ${vdsGroupName}.
4013	GLUSTER_VOLUME_REBALANCE_START_FAILED	Error	Could not start Gluster Volume ${glusterVolumeName} rebalance on cluster ${vdsGroupName}.
4014	GLUSTER_VOLUME_REMOVE_BRICKS	Info	Bricks removed from Gluster Volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4015	GLUSTER_VOLUME_REMOVE_BRICKS_FAILED	Error	Could not remove bricks from Gluster Volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4016	GLUSTER_VOLUME_REPLACE_BRICK_FAILED	Error	Replace Gluster Volume ${glusterVolumeName} Brick failed on cluster ${vdsGroupName}
4017	GLUSTER_VOLUME_REPLACE_BRICK_START	Info	Gluster Volume ${glusterVolumeName} Replace Brick started on cluster ${vdsGroupName}.
4018	GLUSTER_VOLUME_REPLACE_BRICK_START_FAILED	Error	Could not start Gluster Volume ${glusterVolumeName} Replace Brick on cluster ${vdsGroupName}.
4019	GLUSTER_VOLUME_ADD_BRICK	Info	${NoOfBricks} brick(s) added to volume ${glusterVolumeName} of cluster ${vdsGroupName}.

代码	名称	严重性	信息
4020	GLUSTER_VOLUME_ADD_BRICK_FAILED	Error	Failed to add bricks to the Gluster Volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4021	GLUSTER_SERVER_REMOVE_FAILED	Error	Failed to remove host ${VdsName} from Cluster ${VdsGroupName}.
4022	GLUSTER_VOLUME_PROFILE_START	Info	Gluster Volume ${glusterVolumeName} profiling started on cluster ${vdsGroupName}.
4023	GLUSTER_VOLUME_PROFILE_START_FAILED	Error	Could not start profiling on gluster volume ${glusterVolumeName} of cluster ${vdsGroupName}
4024	GLUSTER_VOLUME_PROFILE_STOP	Info	Gluster Volume ${glusterVolumeName} profiling stopped on cluster ${vdsGroupName}.
4025	GLUSTER_VOLUME_PROFILE_STOP_FAILED	Error	Could not stop Profiling on gluster volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4026	GLUSTER_VOLUME_CREATED_FROM_CLI	Warning	Detected new volume ${glusterVolumeName} on cluster ${VdsGroupName}, and added it to engine DB.
4027	GLUSTER_VOLUME_DELETED_FROM_CLI	Info	Detected deletion of volume ${glusterVolumeName} on cluster ${VdsGroupName}, and deleted it from engine DB.
4028	GLUSTER_VOLUME_OPTION_SET_FROM_CLI	Warning	Detected new option ${key}
4029	GLUSTER_VOLUME_OPTION_RESET_FROM_CLI	Warning	Detected option ${key}

代码	名称	严重性	信息
4030	GLUSTER_VOLUME_PROPERTIES_CHANGED_FROM_CLI	Warning	Detected changes in properties of volume ${glusterVolumeName} of cluster ${VdsGroupName}, and updated the same in engine DB.
4031	GLUSTER_VOLUME_BRICK_ADDED_FROM_CLI	Warning	Detected new brick ${brick} on volume ${glusterVolumeName} of cluster ${VdsGroupName}, and added it to engine DB.
4032	GLUSTER_VOLUME_BRICK_REMOVED_FROM_CLI	Info	Detected brick ${brick} removed from Volume ${glusterVolumeName} of cluster ${VdsGroupName}, and removed it from engine DB.
4033	GLUSTER_SERVER_REMOVED_FROM_CLI	Info	Detected server ${VdsName} removed from Cluster ${VdsGroupName}, and removed it from engine DB.
4034	GLUSTER_VOLUME_INFO_FAILED	Error	Failed to fetch gluster volume list from server ${VdsName}.
4035	GLUSTER_COMMAND_FAILED	Error	Gluster command [${Command}] failed on server ${VdsName}.
4038	GLUSTER_SERVER_REMOVE	Info	Host ${VdsName} removed from Cluster ${VdsGroupName}.
4039	GLUSTER_VOLUME_STARTED_FROM_CLI	Warning	Detected that Volume ${glusterVolumeName} of Cluster ${VdsGroupName} was started, and updated engine DB with it's new status.
4040	GLUSTER_VOLUME_STOPPED_FROM_CLI	Warning	Detected that Volume ${glusterVolumeName} of Cluster ${VdsGroupName} was stopped, and updated engine DB with it's new status.

代码	名称	严重性	信息
4041	GLUSTER_VOLUME_OPTION_CHANGED_FROM_CLI	Info	Detected change in value of option ${key} from ${oldValue} to ${newValue} on volume ${glusterVolumeName} of cluster ${VdsGroupName}, and updated it to engine DB.
4042	GLUSTER_HOOK_ENABLE	Info	Gluster Hook ${GlusterHookName} enabled on cluster ${VdsGroupName}.
4043	GLUSTER_HOOK_ENABLE_FAILED	Error	Failed to enable Gluster Hook ${GlusterHookName} on cluster ${VdsGroupName}. ${FailureMessage}
4044	GLUSTER_HOOK_ENABLE_PARTIAL	Warning	Gluster Hook ${GlusterHookName} enabled on some of the servers on cluster ${VdsGroupName}. ${FailureMessage}
4045	GLUSTER_HOOK_DISABLE	Info	Gluster Hook ${GlusterHookName} disabled on cluster ${VdsGroupName}.
4046	GLUSTER_HOOK_DISABLE_FAILED	Error	Failed to disable Gluster Hook ${GlusterHookName} on cluster ${VdsGroupName}. ${FailureMessage}
4047	GLUSTER_HOOK_DISABLE_PARTIAL	Warning	Gluster Hook ${GlusterHookName} disabled on some of the servers on cluster ${VdsGroupName}. ${FailureMessage}
4048	GLUSTER_HOOK_LIST_FAILED	Error	Failed to retrieve hook list from ${VdsName} of Cluster ${VdsGroupName}.
4049	GLUSTER_HOOK_CONFLICT_DETECTED	Warning	Detected conflict in hook ${HookName} of Cluster ${VdsGroupName}.
4050	GLUSTER_HOOK_DETECTED_NEW	Info	Detected new hook ${HookName} in Cluster ${VdsGroupName}.

代码	名称	严重性	信息
4051	GLUSTER_HOOK_DETECTED_DELETE	Info	Detected removal of hook ${HookName} in Cluster ${VdsGroupName}.
4052	GLUSTER_VOLUME_OPTION_MODIFIED	Info	Volume Option ${Key} changed to ${Value} from ${oldvalue} on ${glusterVolumeName} of cluster ${vdsGroupName}.
4053	GLUSTER_HOOK_GETCONTENT_FAILED	Error	Failed to read content of hook ${HookName} in Cluster ${VdsGroupName}.
4054	GLUSTER_SERVICES_LIST_FAILED	Error	Could not fetch statuses of services from server ${VdsName}. Updating statuses of all services on this server to UNKNOWN.
4055	GLUSTER_SERVICE_TYPE_ADDED_TO_CLUSTER	Info	Service type ${ServiceType} was not mapped to cluster ${VdsGroupName}. Mapped it now.
4056	GLUSTER_CLUSTER_SERVICE_STATUS_CHANGED	Info	Status of service type ${ServiceType} changed from ${OldStatus} to ${NewStatus} on cluster ${VdsGroupName}
4057	GLUSTER_SERVICE_ADDED_TO_SERVER	Info	Service ${ServiceName} was not mapped to server ${VdsName}. Mapped it now.
4058	GLUSTER_SERVER_SERVICE_STATUS_CHANGED	Info	Status of service ${ServiceName} on server ${VdsName} changed from ${OldStatus} to ${NewStatus}. Updating in engine now.
4059	GLUSTER_HOOK_UPDATED	Info	Gluster Hook ${GlusterHookName} updated on conflicting servers.
4060	GLUSTER_HOOK_UPDATE_FAILED	Error	Failed to update Gluster Hook ${GlusterHookName} on conflicting servers. ${FailureMessage}

代码	名称	严重性	信息
4061	GLUSTER_HOOK_ADDED	Info	Gluster Hook ${GlusterHookName} added on conflicting servers.
4062	GLUSTER_HOOK_ADD_FAILED	Error	Failed to add Gluster Hook ${GlusterHookName} on conflicting servers. ${FailureMessage}
4063	GLUSTER_HOOK_REMOVED	Info	Gluster Hook ${GlusterHookName} removed from all servers in cluster ${VdsGroupName}.
4064	GLUSTER_HOOK_REMOVE_FAILED	Error	Failed to remove Gluster Hook ${GlusterHookName} from cluster ${VdsGroupName}. ${FailureMessage}
4065	GLUSTER_HOOK_REFRESH	Info	Refreshed gluster hooks in Cluster ${VdsGroupName}.
4066	GLUSTER_HOOK_REFRESH_FAILED	Error	Failed to refresh gluster hooks in Cluster ${VdsGroupName}.
4067	GLUSTER_SERVICE_STARTED	Info	${servicetype} service started on host ${VdsName} of cluster ${VdsGroupName}.
4068	GLUSTER_SERVICE_START_FAILED	Error	Could not start ${servicetype} service on host ${VdsName} of cluster ${VdsGroupName}.
4069	GLUSTER_SERVICE_STOPPED	Info	${servicetype} services stopped on host ${VdsName} of cluster ${VdsGroupName}.
4070	GLUSTER_SERVICE_STOP_FAILED	Error	Could not stop ${servicetype} service on host ${VdsName} of cluster ${VdsGroupName}.
4071	GLUSTER_SERVICES_LIST_NOT_FETCHED	Info	Could not fetch list of services from ${ServiceGroupType} named ${ServiceGroupName}.

代码	名称	严重性	信息
4072	GLUSTER_SERVICE_RESTARTED	Info	${servicetype} service re-started on host ${VdsName} on cluster ${VdsGroupName}.
4073	GLUSTER_SERVICE_RESTART_FAILED	Error	Could not re-start ${servicetype} service on host ${VdsName} on cluster ${VdsGroupName}.
4074	GLUSTER_VOLUME_OPTIONS_RESET_ALL	Info	All Volume Options reset on ${glusterVolumeName} of cluster ${vdsGroupName}.
4075	GLUSTER_HOST_UUID_NOT_FOUND	Error	Could not find gluster uuid of server ${VdsName} on Cluster ${VdsGroupName}.
4076	GLUSTER_VOLUME_BRICK_ADDED	Info	Brick [${brickpath}] on host [${servername}] added to volume [${glusterVolumeName}]
4077	GLUSTER_CLUSTER_SERVICE_STATUS_ADDED	Info	Status of service type ${ServiceType} set to ${NewStatus} on cluster ${VdsGroupName}
4078	GLUSTER_VOLUME_REBALANCE_STOP	Info	Gluster Volume ${glusterVolumeName} rebalance stopped of cluster ${vdsGroupName}.
4079	GLUSTER_VOLUME_REBALANCE_STOP_FAILED	Error	Could not stop rebalance of gluster volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4080	START_REMOVING_GLUSTER_VOLUME_BRICKS	Info	Started removing bricks from Volume ${glusterVolumeName} of cluster ${vdsGroupName}
4081	START_REMOVING_GLUSTER_VOLUME_BRICKS_FAILED	Error	Could not start remove bricks from Volume ${glusterVolumeName} of cluster ${vdsGroupName}
4082	GLUSTER_VOLUME_REMOVE_BRICKS_STOP	Info	Stopped removing bricks from Volume ${glusterVolumeName} of cluster ${vdsGroupName}

代码	名称	严重性	信息
4083	GLUSTER_VOLUME_REMOVE_BRICKS_STOP_FAILED	Error	Failed to stop remove bricks from Volume ${glusterVolumeName} of cluster ${vdsGroupName}
4084	GLUSTER_VOLUME_REMOVE_BRICKS_COMMIT	Info	Gluster volume ${glusterVolumeName} remove bricks committed on cluster ${vdsGroupName}. ${NoOfBricks} brick(s) removed from volume ${glusterVolumeName}.
4085	GLUSTER_VOLUME_REMOVE_BRICKS_COMMIT_FAILED	Error	Gluster volume ${glusterVolumeName} remove bricks could not be commited on cluster ${vdsGroupName}
4086	GLUSTER_BRICK_STATUS_CHANGED	Warning	Detected change in status of brick ${brickpath} of volume ${glusterVolumeName} from ${oldValue} to ${newValue}.
4087	GLUSTER_VOLUME_REBALANCE_FINISHED	Info	${action} ${status} on volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4088	GLUSTER_VOLUME_MIGRATE_BRICK_DATA_FINISHED	Info	${action} ${status} for brick(s) on volume ${glusterVolumeName} of cluster ${vdsGroupName}. Please review to abort or commit.
4089	GLUSTER_VOLUME_REBALANCE_START_DETECTED_FROM_CLI	Info	Detected start of rebalance on volume ${glusterVolumeName} of Cluster ${VdsGroupName} from CLI.
4090	START_REMOVING_GLUSTER_VOLUME_BRICKS_DETECTED_FROM_CLI	Info	Detected start of brick removal for bricks ${brick} on volume ${glusterVolumeName} of Cluster ${VdsGroupName} from CLI.

代码	名称	严重性	信息
4091	GLUSTER_VOLUME_REBALANCE_NOT_FOUND_FROM_CLI	Warning	Could not find information for rebalance on volume ${glusterVolumeName} of Cluster ${VdsGroupName} from CLI. Marking it as unknown.
4092	REMOVE_GLUSTER_VOLUME_BRICKS_NOT_FOUND_FROM_CLI	Warning	Could not find information for remove brick on volume ${glusterVolumeName} of Cluster ${VdsGroupName} from CLI. Marking it as unknown.
4093	GLUSTER_VOLUME_DETAILS_REFRESH	Info	Refreshed details of the volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4094	GLUSTER_VOLUME_DETAILS_REFRESH_FAILED	Error	Failed to refresh the details of volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4095	GLUSTER_HOST_UUID_ALREADY_EXISTS	Error	Gluster UUID of host ${VdsName} on Cluster ${VdsGroupName} already exists.
4096	USER_FORCE_SELECTED_SPM_STOP_FAILED	Error	Failed to force select ${VdsName} as the SPM due to a failure to stop the current SPM.
4097	GLUSTER_GEOREP_SESSION_DELETED_FROM_CLI	Warning	Detected deletion of geo-replication session ${geoRepSessionKey} from volume ${glusterVolumeName}
4098	GLUSTER_GEOREP_SESSION_DETECTED_FROM_CLI	Warning	Detected new geo-replication session ${geoRepSessionKey} for volume ${glusterVolumeName}. Adding it to engine.
4099	GLUSTER_GEOREP_SESSION_REFRESH	Info	Refreshed geo-replication sessions for volume ${glusterVolumeName}.
4100	GLUSTER_GEOREP_SESSION_REFRESH_FAILED	Error	Failed to refresh geo-replication sessions for volume ${glusterVolumeName}.

代码	名称	严重性	信息
4101	GEOREP_SESSION_STOP	Info	Geo-replication session on volume ${glusterVolumeName} has been stopped.
4102	GEOREP_SESSION_STOP_FAILED	Error	Failed to stop geo-replication session on volume ${glusterVolumeName}
4103	GEOREP_SESSION_DELETED	Info	Geo-replication session deleted on volume ${glusterVolumeName}
4104	GEOREP_SESSION_DELETE_FAILED	Error	Failed to delete geo-replication session on volume ${glusterVolumeName}
4105	GLUSTER_GEOREP_CONFIG_SET	Info	Configuration ${key} has been set to ${value} on the geo-rep session ${geoRepSessionKey}.
4106	GLUSTER_GEOREP_CONFIG_SET_FAILED	Error	Failed to set the configuration ${key} to ${value} on geo-rep session ${geoRepSessionKey}.
4107	GLUSTER_GEOREP_CONFIG_LIST	Info	Refreshed configuration options for geo-replication session ${geoRepSessionKey}
4108	GLUSTER_GEOREP_CONFIG_LIST_FAILED	Error	Failed to refresh configuration options for geo-replication session ${geoRepSessionKey}
4109	GLUSTER_GEOREP_CONFIG_SET_DEFAULT	Info	Configuration of ${key} of session ${geoRepSessionKey} reset to its default value .
4110	GLUSTER_GEOREP_CONFIG_SET_DEFAULT_FAILED	Error	Failed to set ${key} of session ${geoRepSessionKey} to its default value.
4111	GLUSTER_VOLUME_SNAPSHOT_DELETED	Info	Gluster volume snapshot ${snapname} deleted.
4112	GLUSTER_VOLUME_SNAPSHOT_DELETE_FAILED	Error	Failed to delete gluster volume snapshot ${snapname}.

代码	名称	严重性	信息
4113	GLUSTER_VOLUME_ALL_SNAPSHOTS_DELETED	Info	Deleted all the gluster volume snapshots for the volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4114	GLUSTER_VOLUME_ALL_SNAPSHOTS_DELETE_FAILED	Error	Failed to delete all the gluster volume snapshots for the volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4115	GLUSTER_VOLUME_SNAPSHOT_ACTIVATED	Info	Activated the gluster volume snapshot ${snapname} on volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4116	GLUSTER_VOLUME_SNAPSHOT_ACTIVATE_FAILED	Error	Failed to activate the gluster volume snapshot ${snapname} on volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4117	GLUSTER_VOLUME_SNAPSHOT_DEACTIVATED	Info	De-activated the gluster volume snapshot ${snapname} on volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4118	GLUSTER_VOLUME_SNAPSHOT_DEACTIVATE_FAILED	Error	Failed to de-activate gluster volume snapshot ${snapname} on volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4119	GLUSTER_VOLUME_SNAPSHOT_RESTORED	Info	Restored the volume ${glusterVolumeName} of cluster ${vdsGroupName} to the state of gluster volume snapshot ${snapname}.
4120	GLUSTER_VOLUME_SNAPSHOT_RESTORE_FAILED	Error	Failed to restore the volume ${glusterVolumeName} of cluster ${vdsGroupName} to the state of gluster volume snapshot ${snapname}.
4121	GLUSTER_VOLUME_SNAPSHOT_CONFIG_UPDATED	Info	Updated Gluster volume snapshot configuration(s).

代码	名称	严重性	信息
4122	GLUSTER_VOLUME_SNAPSHOT_CONFIG_UPDATE_FAILED	Error	Failed to update gluster volume snapshot configuration(s).
4123	GLUSTER_VOLUME_SNAPSHOT_CONFIG_UPDATE_FAILED_PARTIALLY	Error	Failed to update gluster volume snapshot configuration(s) ${failedSnapshotConfigs}.
4124	NEW_STORAGE_DEVICE_DETECTED	Info	Found new storage device ${storageDevice} on host ${VdsName}, and added it to engine DB."
4125	STORAGE_DEVICE_REMOVED_FROM_THE_HOST	Info	Detected deletion of storage device ${storageDevice} on host ${VdsName}, and deleting it from engine DB."
4126	SYNC_STORAGE_DEVICES_IN_HOST	Info	Manually synced the storage devices from host ${VdsName}
4127	SYNC_STORAGE_DEVICES_IN_HOST_FAILED	Error	Failed to sync storage devices from host ${VdsName}
4128	GEOREP_OPTION_SET_FROM_CLI	Warning	Detected new option ${key}
4129	GEOREP_OPTION_CHANGED_FROM_CLI	Warning	Detected change in value of option ${key} from ${oldValue} to ${value} for geo-replication session on volume ${glusterVolumeName} of cluster ${VdsGroupName}, and updated it to engine.
4130	GLUSTER_MASTER_VOLUME_STOP_FAILED_DURING_SNAPSHOT_RESTORE	Error	Could not stop master volume ${glusterVolumeName} of cluster ${vdsGroupName} during snapshot restore.
4131	GLUSTER_MASTER_VOLUME_SNAPSHOT_RESTORE_FAILED	Error	Could not restore master volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4132	GLUSTER_VOLUME_SNAPSHOT_CREATED	Info	Snapshot ${snapname} created for volume ${glusterVolumeName} of cluster ${vdsGroupName}.

代码	名称	严重性	信息
4133	GLUSTER_VOLUME_SNAPSHOT_CREATE_FAILED	Error	Could not create snapshot for volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4134	GLUSTER_VOLUME_SNAPSHOT_SCHEDULED	Info	Snapshots scheduled on volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4135	GLUSTER_VOLUME_SNAPSHOT_SCHEDULE_FAILED	Error	Failed to schedule snapshots on the volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4136	GLUSTER_VOLUME_SNAPSHOT_RESCHEDULED	Info	Rescheduled snapshots on volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4137	GLUSTER_VOLUME_SNAPSHOT_RESCHEDULE_FAILED	Error	Failed to reschedule snapshots on volume ${glusterVolumeName} of cluster ${vdsGroupName}.
4138	CREATE_GLUSTER_BRICK	Info	Brick ${brickName} created successfully on host ${vdsName} of cluster ${vdsGroupName}.
4139	CREATE_GLUSTER_BRICK_FAILED	Error	Failed to create brick ${brickName} on host ${vdsName} of cluster ${vdsGroupName}.
4140	GLUSTER_GEO_REP_PUB_KEY_FETCH_FAILED	Error	Failed to fetch public keys.
4141	GLUSTER_GET_PUB_KEY	Info	Public key fetched.
4142	GLUSTER_GEOREP_PUBLIC_KEY_WRITE_FAILED	Error	Failed to write public keys to ${VdsName}
4143	GLUSTER_WRITE_PUB_KEYS	Info	Public keys written to ${VdsName}
4144	GLUSTER_GEOREP_SETUP_MOUNT_BROKER_FAILED	Error	Failed to setup geo-replication mount broker for user ${geoRepUserName} on the slave volume ${geoRepSlaveVolumeName}.

代码	名称	严重性	信息
4145	GLUSTER_SETUP_GEOREP_MOUNT_BROKER	Info	Geo-replication mount broker has been setup for user ${geoRepUserName} on the slave volume ${geoRepSlaveVolumeName}.
4146	GLUSTER_GEOREP_SESSION_CREATE_FAILED	Error	Failed to create geo-replication session between master volume : ${glusterVolumeName} of cluster ${vdsGroupName} and slave volume : ${geoRepSlaveVolumeName} for the user ${geoRepUserName}.
4147	CREATE_GLUSTER_VOLUME_GEOREP_SESSION	Info	Created geo-replication session between master volume : ${glusterVolumeName} of cluster ${vdsGroupName} and slave volume : ${geoRepSlaveVolumeName} for the user ${geoRepUserName}.
4148	GLUSTER_VOLUME_SNAPSHOT_SOFT_LIMIT_REACHED	Info	Gluster Volume Snapshot soft limit reached for the volume ${glusterVolumeName} on cluster ${vdsGroupName}.
4149	HOST_FEATURES_INCOMPATIBILE_WITH_CLUSTER	Error	Host ${VdsName} does not comply with the list of features supported by cluster ${VdsGroupName}. ${UnSupportedFeature} is not supported by the Host
4150	GLUSTER_VOLUME_SNAPSHOT_SCHEDULE_DELETED	Info	Snapshot schedule deleted for volume ${glusterVolumeName} of ${vdsGroupName}.
4151	GLUSTER_BRICK_STATUS_DOWN	Info	Status of brick ${brickpath} of volume ${glusterVolumeName} on cluster ${VdsGroupName} is down.

代码	名称	严重性	信息
4152	GLUSTER_VOLUME_SNAPSHOT_DETECTED_NEW	Info	Found new gluster volume snapshot ${snapname} for volume ${glusterVolumeName} on cluster ${VdsGroupName}, and added it to engine DB."
4153	GLUSTER_VOLUME_SNAPSHOT_DELETED_FROM_CLI	Info	Detected deletion of gluster volume snapshot ${snapname} for volume ${glusterVolumeName} on cluster ${VdsGroupName}, and deleting it from engine DB."
4154	GLUSTER_VOLUME_SNAPSHOT_CLUSTER_CONFIG_DETECTED_NEW	Info	Found new gluster volume snapshot configuration ${snapConfigName} with value ${snapConfigValue} on cluster ${VdsGroupName}, and added it to engine DB."
4155	GLUSTER_VOLUME_SNAPSHOT_VOLUME_CONFIG_DETECTED_NEW	Info	Found new gluster volume snapshot configuration ${snapConfigName} with value ${snapConfigValue} for volume ${glusterVolumeName} on cluster ${VdsGroupName}, and added it to engine DB."
4156	GLUSTER_VOLUME_SNAPSHOT_HARD_LIMIT_REACHED	Info	Gluster Volume Snapshot hard limit reached for the volume ${glusterVolumeName} on cluster ${vdsGroupName}.
4157	GLUSTER_CLI_SNAPSHOT_SCHEDULE_DISABLE_FAILED	Error	Failed to disable gluster CLI based snapshot schedule on cluster ${vdsGroupName}.
4158	GLUSTER_CLI_SNAPSHOT_SCHEDULE_DISABLED	Info	Disabled gluster CLI based scheduling successfully on cluster ${vdsGroupName}.
4159	SET_UP_PASSWORDLESS_SSH	Info	Password-less SSH has been setup for user ${geoRepUserName} on the nodes of remote volume ${geoRepSlaveVolumeName} from the nodes of the volume ${glusterVolumeName}.

代码	名称	严重性	信息
4160	SET_UP_PASSWORDLESS_SSH_FAILED	Error	Failed to setup Passwordless ssh for user ${geoRepUserName} on the nodes of remote volume ${geoRepSlaveVolumeName} from the nodes of the volume ${glusterVolumeName}.
4436	GLUSTER_SERVER_ADD_FAILED	Error	Failed to add host ${VdsName} into Cluster ${VdsGroupName}.
4437	GLUSTER_SERVERS_LIST_FAILED	Error	Failed to fetch gluster peer list from server ${VdsName} on Cluster ${VdsGroupName}.
4595	GLUSTER_VOLUME_GEO_REP_START_FAILED_EXCEPTION	Error	Failed to start geo-replication session on volume ${glusterVolumeName}
4596	GLUSTER_VOLUME_GEO_REP_START	Info	Geo-replication session on volume ${glusterVolumeName} has been started.
4597	GLUSTER_VOLUME_GEO_REP_PAUSE_FAILED	Error	Failed to pause geo-replication session on volume ${glusterVolumeName} of cluster ${vdsGroupName}
4598	GLUSTER_VOLUME_GEO_REP_RESUME_FAILED	Error	Failed to resume geo-replication session on volume ${glusterVolumeName} of cluster ${vdsGroupName}
4599	GLUSTER_VOLUME_GEO_REP_RESUME	Info	Geo-replication session on volume ${glusterVolumeName} of cluster ${vdsGroupName} has been resumed.
4600	GLUSTER_VOLUME_GEO_REP_PAUSE	Info	Geo-replication session on volume ${glusterVolumeName} of cluster ${vdsGroupName} has been paused.
9000	VDS_ALERT_FENCE_IS_NOT_CONFIGURED	Info	Failed to verify Power Management configuration for Host ${VdsName}.

代码	名称	严重性	信息
9001	VDS_ALERT_FENCE_TEST_FAILED	Info	Power Management test failed for Host ${VdsName}.${Reason}
9002	VDS_ALERT_FENCE_OPERATION_FAILED	Info	Failed to power fence host ${VdsName}. Please check the host status and it's power management settings, and then manually reboot it and click "Confirm Host Has Been Rebooted"
9003	VDS_ALERT_FENCE_OPERATION_SKIPPED	Info	Host ${VdsName} became non responsive. It has no power management configured. Please check the host status, manually reboot it, and click "Confirm Host Has Been Rebooted"
9004	VDS_ALERT_FENCE_NO_PROXY_HOST	Info	There is no other host in the data center that can be used to test the power management settings.
9005	VDS_ALERT_FENCE_STATUS_VERIFICATION_FAILED	Info	Failed to verify Host ${Host} ${Status} status, Please ${Status} Host ${Host} manually.
9006	CANNOT_HIBERNATE_RUNNING_VMS_AFTER_CLUSTER_CPU_UPGRADE	Warning	Hibernation of VMs after CPU upgrade of Cluster ${VdsGroup} is not supported. Please stop and restart those VMs in case you wish to hibernate them
9007	VDS_ALERT_SECONDARY_AGENT_USED_FOR_FENCE_OPERATION	Info	Secondary fence agent was used to ${Operation} Host ${VdsName}
9008	VDS_HOST_NOT_RESPONDING_CONNECTING	Warning	Host ${VdsName} is not responding. It will stay in Connecting state for a grace period of ${Seconds} seconds and after that an attempt to fence the host will be issued.

代码	名称	严重性	信息
9009	VDS_ALERT_PM_HEALTH_CHECK_FENCE_AGENT_NON_RESPONSIVE	Info	Health check on Host ${VdsName} indicates that Fence-Agent ${AgentId} is non-responsive.
9010	VDS_ALERT_PM_HEALTH_CHECK_START_MIGHT_FAIL	Info	Health check on Host ${VdsName} indicates that future attempts to Start this host using Power-Management are expected to fail.
9011	VDS_ALERT_PM_HEALTH_CHECK_STOP_MIGHT_FAIL	Info	Health check on Host ${VdsName} indicates that future attempts to Stop this host using Power-Management are expected to fail.
9012	VDS_ALERT_PM_HEALTH_CHECK_RESTART_MIGHT_FAIL	Info	Health check on Host ${VdsName} indicates that future attempts to Restart this host using Power-Management are expected to fail.
9013	VDS_ALERT_FENCE_OPERATION_SKIPPED_BROKEN_CONNECTIVITY	Info	Host ${VdsName} became non responsive and was not restarted due to Fencing Policy: ${Percents} percents of the Hosts in the Cluster have connectivity issues.
9014	VDS_ALERT_NOT_RESTARTED_DUE_TO_POLICY	Info	Host ${VdsName} became non responsive and was not restarted due to the Cluster Fencing Policy.
9015	VDS_ALERT_FENCE_DISABLED_BY_CLUSTER_POLICY	Info	Host ${VdsName} became Non Responsive and was not restarted due to disabled fencing in the Cluster Fencing Policy.
9016	FENCE_DISABLED_IN_CLUSTER_POLICY	Info	Fencing is disabled in Fencing Policy of the Cluster ${VdsGroupName}, so HA VMs running on a non-responsive host will not be restarted elsewhere.

代码	名称	严重性	信息
9017	FENCE_OPERATION_STARTED	Info	Power management ${Action} of Host ${VdsName} initiated.
9018	FENCE_OPERATION_SUCCEEDED	Info	Power management ${Action} of Host ${VdsName} succeeded.
9019	FENCE_OPERATION_FAILED	Error	Power management ${Action} of Host ${VdsName} failed.
9020	FENCE_OPERATION_USING_AGENT_AND_PROXY_STARTED	Info	Executing power management ${Action} on Host ${Host} using Proxy Host ${ProxyHost} and Fence Agent ${AgentType}:${AgentIp}.
9021	FENCE_OPERATION_USING_AGENT_AND_PROXY_FAILED	Warning	Execution of power management ${Action} on Host ${Host} using Proxy Host ${ProxyHost} and Fence Agent ${AgentType}:${AgentIp} failed.
9022	ENGINE_NO_FULL_BACKUP	Info	There is no full backup available, please run engine-backup to prevent data loss in case of corruption.
9023	ENGINE_NO_WARM_BACKUP	Info	Full backup was created on ${Date} and it's too old. Please run engine-backup to prevent data loss in case of corruption.
9024	ENGINE_BACKUP_STARTED	Normal	Engine backup started.
9025	ENGINE_BACKUP_COMPLETED	Normal	Engine backup completed successfully.
9026	ENGINE_BACKUP_FAILED	Error	Engine backup failed.
9500	TASK_STOPPING_ASYNC_TASK	Info	Stopping async task ${CommandName} that started at ${Date}

代码	名称	严重性	信息
9501	TASK_CLEARING_ASYNC_TASK	Info	Clearing asynchronous task ${CommandName} that started at ${Date}
9506	USER_ACTIVATE_STORAGE_DOMAIN_FAILED_ASYNC	Warning	Failed to autorecover Storage Domain ${StorageDomainName} (Data Center ${StoragePoolName}).
9600	IMPORTEXPORT_IMPORT_VM_INVALID_INTERFACES	Warning	While importing VM ${EntityName}, the Network/s ${Networks} were found to be Non-VM Networks or do not exist in Cluster or are missing a suitable VM network interface profile. Network Name was not set in the Interface/s ${Interfaces}.
9601	VDS_SET_NON_OPERATIONAL_VM_NETWORK_IS_BRIDGELESS	Warning	Host ${VdsName} does not comply with the cluster ${VdsGroupName} networks, the following VM networks are non-VM networks: '${Networks}'
9602	HA_VM_FAILED	Error	Highly Available VM ${VmName} failed. It will be restarted automatically.
9603	HA_VM_RESTART_FAILED	Error	Restart of the Highly Available VM ${VmName} failed.
9604	EMULATED_MACHINES_INCOMPATIBLE_WITH_CLUSTER	Warning	Host ${VdsName} does not comply with the cluster ${VdsGroupName} emulated machine. The cluster emulated machine is ${clusterEmulatedMachines} and the host emulated machines are ${hostSupportedEmulatedMachines}.

代码	名称	严重性	信息
9605	EXCEEDED_MAXIMUM_NUM_OF_RESTART_HA_VM_ATTEMPTS	Error	Highly Available VM ${VmName} could not be restarted automatically, exceeded the maximum number of attempts.
9606	IMPORTEXPORT_SNAPSHOT_VM_INVALID_INTERFACES	Warning	While previewing a snapshot of VM ${EntityName}, the Network/s ${Networks} were found to be Non-VM Networks or do not exist in Cluster. Network Name was not set in the Interface/s ${Interfaces}.
9607	ADD_VM_FROM_SNAPSHOT_INVALID_INTERFACES	Warning	While adding vm ${EntityName} from snapshot, the Network/s ${Networks} were found to be Non-VM Networks or do not exist in Cluster. Network Name was not set in the Interface/s ${Interfaces}.
9608	RNG_SOURCES_INCOMPATIBLE_WITH_CLUSTER	Warning	Host ${VdsName} does not comply with the cluster ${VdsGroupName} Random Number Generator sources. The Hosts supported sources are: ${hostSupportedRngSources}; and the cluster requirements are: ${clusterRequiredRngSources}.
9609	EMULATED_MACHINES_INCOMPATIBLE_WITH_CLUSTER_LEVEL	Warning	Host ${VdsName} does not comply with the cluster ${VdsGroupName} emulated machines. The current cluster compatibility level supports ${clusterEmulatedMachines} and the host emulated machines are ${hostSupportedEmulatedMachines}.

代码	名称	严重性	信息
9610	MIXING_RHEL_VERSIONS_IN_CLUSTER	Warning	Not possible to mix RHEL 6.x and 7.x hosts in one cluster. Tried adding ${addingRhel} host to a cluster with ${previousRhel} hosts.
9700	DWH_STARTED	Info	ETL Service started.
9701	DWH_STOPPED	Info	ETL Service stopped.
9704	DWH_ERROR	Error	Error in ETL Service.
9801	EXTERNAL_EVENT_NORMAL	Info	An external event with NORMAL severity has been added.
9802	EXTERNAL_EVENT_WARNING	Warning	An external event with WARNING severity has been added.
9803	EXTERNAL_EVENT_ERROR	Error	An external event with ERROR severity has been added.
9804	EXTERNAL_ALERT	Info	An external event with ALERT severity has been added.
9901	WATCHDOG_EVENT	Warning	Watchdog event (${wdaction}) triggered on ${VmName} at ${wdevent} (host time).
9910	USER_ADD_CLUSTER_POLICY	Info	Scheduling Policy ${ClusterPolicy} was added. (User: ${UserName})
9911	USER_FAILED_TO_ADD_CLUSTER_POLICY	Error	Failed to add Scheduling Policy: ${ClusterPolicy}. (User: ${UserName})
9912	USER_UPDATE_CLUSTER_POLICY	Info	Scheduling Policy ${ClusterPolicy} was updated. (User: ${UserName})
9913	USER_FAILED_TO_UPDATE_CLUSTER_POLICY	Error	Failed to update Scheduling Policy: ${ClusterPolicy}. (User: ${UserName})

代码	名称	严重性	信息
9914	USER_REMOVE_CLUSTER_POLICY	Info	Scheduling Policy ${ClusterPolicy} was removed. (User: ${UserName})
9915	USER_FAILED_TO_REMOVE_CLUSTER_POLICY	Error	Failed to remove Scheduling Policy: ${ClusterPolicy}. (User: ${UserName})
9920	FAILED_TO_CONNECT_TO_SCHEDULER_PROXY	Error	Failed to connect to external scheduler proxy. External filters, scoring functions and load balancing will not be performed.
10000	VDS_UNTRUSTED	Error	Host ${VdsName} was set to non-operational. Host is not trusted by the attestation service.
10001	USER_UPDATE_VM_FROM_TRUSTED_TO_UNTRUSTED	Warning	The VM ${VmName} was updated from trusted cluster to non-trusted cluster.
10002	USER_UPDATE_VM_FROM_UNTRUSTED_TO_TRUSTED	Warning	The VM ${VmName} was updated from non-trusted cluster to trusted cluster.
10003	IMPORTEXPORT_IMPORT_VM_FROM_TRUSTED_TO_UNTRUSTED	Warning	The VM ${VmName} was created in trusted cluster and imported into a non-trusted cluster
10004	IMPORTEXPORT_IMPORT_VM_FROM_UNTRUSTED_TO_TRUSTED	Warning	The VM ${VmName} was created in non-trusted cluster and imported into a trusted cluster
10005	USER_ADD_VM_FROM_TRUSTED_TO_UNTRUSTED	Warning	The VM ${VmName} was created in an untrusted cluster. It was originated from the Template ${VmTemplateName} which was created in a trusted cluster.

代码	名称	严重性	信息
10006	USER_ADD_VM_FROM_UNTRUSTED_TO_TRUSTED	Warning	The VM ${VmName} was created in a trusted cluster. It was originated from the Template ${VmTemplateName} which was created in an untrusted cluster.
10007	IMPORTEXPORT_IMPORT_TEMPLATE_FROM_TRUSTED_TO_UNTRUSTED	Warning	The Template ${VmTemplateName} was created in trusted cluster and imported into a non-trusted cluster
10008	IMPORTEXPORT_IMPORT_TEMPLATE_FROM_UNTRUSTED_TO_TRUSTED	Warning	The Template ${VmTemplateName} was created in non-trusted cluster and imported into a trusted cluster
10009	USER_ADD_VM_TEMPLATE_FROM_TRUSTED_TO_UNTRUSTED	Warning	The non-trusted Template ${VmTemplateName} was created from trusted Vm ${VmName}.
10010	USER_ADD_VM_TEMPLATE_FROM_UNTRUSTED_TO_TRUSTED	Warning	The trusted template ${VmTemplateName} was created from non-trusted Vm ${VmName}.
10011	USER_UPDATE_VM_TEMPLATE_FROM_TRUSTED_TO_UNTRUSTED	Warning	The Template ${VmTemplateName} was updated from trusted cluster to non-trusted cluster.
10012	USER_UPDATE_VM_TEMPLATE_FROM_UNTRUSTED_TO_TRUSTED	Warning	The Template ${VmTemplateName} was updated from non-trusted cluster to trusted cluster.
10100	USER_ADDED_NETWORK_QOS	Info	Network QoS ${QosName} was added. (User: ${UserName})
10101	USER_FAILED_TO_ADD_NETWORK_QOS	Error	Failed to add Network QoS ${QosName}. (User: ${UserName})
10102	USER_REMOVED_NETWORK_QOS	Info	Network QoS ${QosName} was removed. (User: ${UserName})

代码	名称	严重性	信息
10103	USER_FAILED_TO_REMOVE_NETWORK_QOS	Error	Failed to remove Network QoS ${QosName}. (User: ${UserName})
10104	USER_UPDATED_NETWORK_QOS	Info	Network QoS ${QosName} was updated. (User: ${UserName})
10105	USER_FAILED_TO_UPDATE_NETWORK_QOS	Error	Failed to update Network QoS ${QosName}. (User: ${UserName})
10110	USER_ADDED_QOS	Info	QoS ${QoSName} was added. (User: ${UserName})
10111	USER_FAILED_TO_ADD_QOS	Error	Failed to add QoS ${QoSName}. (User: ${UserName})
10112	USER_REMOVED_QOS	Info	QoS ${QoSName} was removed. (User: ${UserName})
10113	USER_FAILED_TO_REMOVE_QOS	Error	Failed to remove QoS ${QoSName}. (User: ${UserName})
10114	USER_UPDATED_QOS	Info	QoS ${QoSName} was updated. (User: ${UserName})
10115	USER_FAILED_TO_UPDATE_QOS	Error	Failed to update QoS ${QoSName}. (User: ${UserName})
10120	USER_ADDED_DISK_PROFILE	Info	Disk Profile ${ProfileName} was successfully added (User: ${UserName}).
10121	USER_FAILED_TO_ADD_DISK_PROFILE	Error	Failed to add Disk Profile (User: ${UserName}).
10122	USER_REMOVED_DISK_PROFILE	Info	Disk Profile ${ProfileName} was successfully removed (User: ${UserName}).
10123	USER_FAILED_TO_REMOVE_DISK_PROFILE	Error	Failed to remove Disk Profile ${ProfileName} (User: ${UserName}).

代码	名称	严重性	信息
10124	USER_UPDATED_DISK_PROFILE	Info	Disk Profile ${ProfileName} was successfully updated (User: ${UserName}).
10125	USER_FAILED_TO_UPDATE_DISK_PROFILE	Error	Failed to update Disk Profile ${ProfileName} (User: ${UserName}).
10130	USER_ADDED_CPU_PROFILE	Info	CPU Profile ${ProfileName} was successfully added (User: ${UserName}).
10131	USER_FAILED_TO_ADD_CPU_PROFILE	Error	Failed to add CPU Profile (User: ${UserName}).
10132	USER_REMOVED_CPU_PROFILE	Info	CPU Profile ${ProfileName} was successfully removed (User: ${UserName}).
10133	USER_FAILED_TO_REMOVE_CPU_PROFILE	Error	Failed to remove CPU Profile ${ProfileName} (User: ${UserName}).
10134	USER_UPDATED_CPU_PROFILE	Info	CPU Profile ${ProfileName} was successfully updated (User: ${UserName}).
10135	USER_FAILED_TO_UPDATE_CPU_PROFILE	Error	Failed to update CPU Profile ${ProfileName} (User: ${UserName}).
10200	USER_UPDATED_MOM_POLICIES	Info	Mom policy was updated on host ${VdsName}.
10201	USER_FAILED_TO_UPDATE_MOM_POLICIES	Warning	Mom policy could not be updated on host ${VdsName}.
10250	PM_POLICY_UP_TO_MAINTENANCE	Info	Host ${Host} is not currently needed, activating maintenance mode in preparation for shutdown.
10251	PM_POLICY_MAINTENANCE_TO_DOWN	Info	Host ${Host} is not currently needed, shutting down.

代码	名称	严重性	信息
10252	PM_POLICY_TO_UP	Info	Reactivating host ${Host} according to the current power management policy.
10300	CLUSTER_ALERT_HA_RESERVATION	Info	Cluster ${ClusterName} failed the HA Reservation check, HA VMs on host(s): ${Hosts} will fail to migrate in case of a failover, consider adding resources or shutting down unused VMs.
10301	CLUSTER_ALERT_HA_RESERVATION_DOWN	Info	Cluster ${ClusterName} passed the HA Reservation check.
10350	USER_ADDED_AFFINITY_GROUP	Info	Affinity Group ${affinityGroupName} was added. (User: ${UserName})
10351	USER_FAILED_TO_ADD_AFFINITY_GROUP	Error	Failed to add Affinity Group ${affinityGroupName}. (User: ${UserName})
10352	USER_UPDATED_AFFINITY_GROUP	Info	Affinity Group ${affinityGroupName} was updated. (User: ${UserName})
10353	USER_FAILED_TO_UPDATE_AFFINITY_GROUP	Error	Failed to update Affinity Group ${affinityGroupName}. (User: ${UserName})
10354	USER_REMOVED_AFFINITY_GROUP	Info	Affinity Group ${affinityGroupName} was removed. (User: ${UserName})
10355	USER_FAILED_TO_REMOVE_AFFINITY_GROUP	Error	Failed to remove Affinity Group ${affinityGroupName}. (User: ${UserName})
10400	ISCSI_BOND_ADD_SUCCESS	Info	iSCSI bond '${IscsiBondName}' was successfully created in Data Center '${StoragePoolName}'.
10401	ISCSI_BOND_ADD_FAILED	Error	Failed to create iSCSI bond '${IscsiBondName}' in Data Center '${StoragePoolName}'.

代码	名称	严重性	信息
10402	ISCSI_BOND_EDIT_SUCCESS	Info	iSCSI bond '${IscsiBondName}' was successfully updated.
10403	ISCSI_BOND_EDIT_FAILED	Error	Failed to update iSCSI bond '${IscsiBondName}'.
10404	ISCSI_BOND_REMOVE_SUCCESS	Info	iSCSI bond '${IscsiBondName}' was removed from Data Center '${StoragePoolName}'
10405	ISCSI_BOND_REMOVE_FAILED	Error	Failed to remove iSCSI bond '${IscsiBondName}' from Data Center '${StoragePoolName}'
10406	ISCSI_BOND_EDIT_SUCCESS_WITH_WARNING	Warning	iSCSI bond '${IscsiBondName}' was successfully updated but some of the hosts encountered connection issues.
10407	ISCSI_BOND_ADD_SUCCESS_WITH_WARNING	Warning	iSCSI bond '${IscsiBondName}' was successfully created in Data Center '${StoragePoolName}' but some of the hosts encountered connection issues.
10450	USER_SET_HOSTED_ENGINE_MAINTENANCE	Info	Hosted Engine HA maintenance mode was updated on host ${VdsName}.
10451	USER_FAILED_TO_SET_HOSTED_ENGINE_MAINTENANCE	Error	Hosted Engine HA maintenance mode could not be updated on host ${VdsName}.
10452	VDS_MAINTENANCE_MANUAL_HA	Warning	Host ${VdsName} was switched to Maintenance mode, but Hosted Engine HA maintenance could not be enabled. Please enable it manually.

代码	名称	严重性	信息
10453	USER_VDS_MAINTENANCE_MANUAL_HA	Warning	Host ${VdsName} was switched to Maintenance mode by ${UserName}, but Hosted Engine HA maintenance could not be enabled. Please enable it manually.
10454	VDS_ACTIVATE_MANUAL_HA	Warning	Host ${VdsName} was activated by ${UserName}, but the Hosted Engine HA service may still be in maintenance mode. If necessary, please correct this manually.
10455	VDS_ACTIVATE_MANUAL_HA_ASYNC	Warning	Host ${VdsName} was autorecovered, but the Hosted Engine HA service may still be in maintenance mode. If necessary, please correct this manually.
10456	HOSTED_ENGINE_VM_IMPORT_SUCCEEDED	Normal	Hosted Engine VM was imported successfully
10460	HOSTED_ENGINE_DOMAIN_IMPORT_SUCCEEDED	Normal	Hosted Engine storage domain imported successfully
10461	HOSTED_ENGINE_DOMAIN_IMPORT_FAILED	Error	Failed to import the Hosted Engine Storage Domain
10500	EXTERNAL_SCHEDULER_PLUGIN_ERROR	Error	Running the external scheduler plugin '${PluginName}' failed: '${ErrorMessage}'
10501	EXTERNAL_SCHEDULER_ERROR	Error	Running the external scheduler failed: '${ErrorMessage}'
10550	VM_SLA_POLICY	Info	VM ${VmName} SLA Policy was set. CPU limit is set to ${cpuLimit}
10551	FAILED_VM_SLA_POLICY	Error	Failed to set SLA Policy to VM ${VmName}. Underlying error message: ${ErrorMessage}

代码	名称	严重性	信息
10600	USER_REMOVE_AUDIT_LOG	Info	Event list message ${AuditLogId} was removed by User ${UserName}.
10601	USER_REMOVE_AUDIT_LOG_FAILED	Error	User ${UserName} failed to remove event list message ${AuditLogId}.
10602	USER_CLEAR_ALL_AUDIT_LOG	Info	
10603	USER_CLEAR_ALL_AUDIT_LOG_FAILED	Error	
10604	USER_DISPLAY_ALL_AUDIT_LOG	Info	
10605	USER_DISPLAY_ALL_AUDIT_LOG_FAILED	Error	
10700	MAC_POOL_ADD_SUCCESS	Info	MAC Pool '${MacPoolName}' (id
10701	MAC_POOL_ADD_FAILED	Error	Failed to create MAC Pool '${MacPoolName}'. (User: ${UserName})
10702	MAC_POOL_EDIT_SUCCESS	Info	MAC Pool '${MacPoolName}' (id
10703	MAC_POOL_EDIT_FAILED	Error	Failed to update MAC Pool '${MacPoolName}' (id
10704	MAC_POOL_REMOVE_SUCCESS	Info	MAC Pool '${MacPoolName}' (id
10705	MAC_POOL_REMOVE_FAILED	Error	Failed to remove MAC Pool '${MacPoolName}' (id
10750	CINDER_PROVIDER_ERROR	Error	An error occurred on Cinder provider: '${CinderException}'
10751	CINDER_DISK_CONNECTION_FAILURE	Error	Failed to retrieve connection information for Cinder Disk '${DiskAlias}'.

代码	名称	严重性	信息
10752	CINDER_DISK_CONNECTION_VOLUME_DRIVER_UNSUPPORTED	Error	Unsupported volume driver for Cinder Disk '${DiskAlias}'.
10753	USER_FINISHED_FAILED_REMOVE_CINDER_DISK	Error	Failed to remove disk ${DiskAlias} from storage domain ${StorageDomainName}. The following entity id could not be deleted from the Cinder provider '${imageId}'. (User: ${UserName}).
10754	USER_ADDED_LIBVIRT_SECRET	Info	Authentication Key ${LibvirtSecretUUID} was added. (User: ${UserName}).
10755	USER_FAILED_TO_ADD_LIBVIRT_SECRET	Error	Failed to add Authentication Key ${LibvirtSecretUUID}. (User: ${UserName}).
10756	USER_UPDATE_LIBVIRT_SECRET	Info	Authentication Key ${LibvirtSecretUUID} was updated. (User: ${UserName}).
10757	USER_FAILED_TO_UPDATE_LIBVIRT_SECRET	Error	Failed to update Authentication Key ${LibvirtSecretUUID}. (User: ${UserName}).
10758	USER_REMOVED_LIBVIRT_SECRET	Info	Authentication Key ${LibvirtSecretUUID} was removed. (User: ${UserName}).
10759	USER_FAILED_TO_REMOVE_LIBVIRT_SECRET	Error	Failed to remove Authentication Key ${LibvirtSecretUUID}. (User: ${UserName}).
10760	FAILED_TO_REGISTER_LIBVIRT_SECRET	Error	Failed to register Authentication Keys for storage domain ${StorageDomainName} on host ${VdsName}.

代码	名称	严重性	信息
10761	FAILED_TO_UNREGISTER_LIBVIRT_SECRET	Error	Failed to unregister Authentication Keys for storage domain ${StorageDomainName} on host ${VdsName}.
10762	FAILED_TO_REGISTER_LIBVIRT_SECRET_ON_VDS	Error	Failed to register Authentication Keys on host ${VdsName}.
10763	NO_LIBRBD_PACKAGE_AVAILABLE_ON_VDS	Error	Librbd1 package is not available on host ${VdsName}, which is mandatory for using Cinder storage domains.
10764	FAILED_TO_FREEZE_VM	Warning	Failed to freeze guest filesystems on VM ${VmName}. Note that using the created snapshot might cause data inconsistency.
10765	FAILED_TO_THAW_VM	Warning	Failed to thaw guest filesystems on VM ${VmName}. The filesystems might be unresponsive until the VM is restarted.
10766	FREEZE_VM_INITIATED	Normal	Freeze of guest filesystems on VM ${VmName} was initiated.
10767	FREEZE_VM_SUCCESS	Normal	Guest filesystems on VM ${VmName} have been frozen successfully.
10768	THAW_VM_SUCCESS	Normal	Guest filesystems on VM ${VmName} have been thawed successfully.
10769	USER_FAILED_TO_FREEZE_VM	Warning	Failed to freeze guest filesystems on ${VmName} (Host: ${VdsName}, User: ${UserName}).
10770	USER_FAILED_TO_THAW_VM	Warning	Failed to thaw guest filesystems on ${VmName} (Host: ${VdsName}, User: ${UserName}).

代码	名称	严重性	信息
10771	VDS_CANNOT_CONNECT_TO_GLUSTERFS	Error	Host ${VdsName} cannot connect to Glusterfs. Verify that glusterfs-cli package is installed on the host.
10780	AFFINITY_RULES_ENFORCEMENT_MANAGER_START	Normal	Affinity Rules Enforcement Manager started.
10781	AFFINITY_RULES_ENFORCEMENT_MANAGER_INTERVAL_REACHED	Normal	
10800	VM_ADD_HOST_DEVICES	Info	Host devices ${NamesAdded} were attached to Vm ${VmName} by User ${UserName}.
10801	VM_REMOVE_HOST_DEVICES	Info	Host devices ${NamesRemoved} were detached from Vm ${VmName} by User ${UserName}.
10802	VDS_BROKER_COMMAND_FAILURE	Error	VDSM ${VdsName} command failed: ${message}
10803	IRS_BROKER_COMMAND_FAILURE	Error	VDSM command failed: ${message}
10804	VDS_UNKNOWN_HOST	Error	The address of host ${VdsName} could not be determined
10810	SYSTEM_CHANGE_STORAGE_POOL_STATUS_UP_REPORTING_HOSTS	Normal	Data Center ${StoragePoolName} status was changed to UP as some of its hosts are in status UP.
10811	SYSTEM_CHANGE_STORAGE_POOL_STATUS_NON_RESPONSIVE_NO_REPORTING_HOSTS	Info	Data Center ${StoragePoolName} status was changed to Non Responsive as none of its hosts are in status UP.
10900	HOST_SYNC_ALL_NETWORKS_FAILED	Error	Failed to sync all host ${VdsName} networks
10901	HOST_SYNC_ALL_NETWORKS_FINISHED	Info	Managed to sync all host ${VdsName} networks.

代码	名称	严重性	信息
10902	PERSIST_HOST_SETUP_NETWORK_ON_HOST	Info	(${Sequence}/${Total}): Applying network's changes on host ${VdsName}. (User: ${UserName})
10903	PERSIST_SETUP_NETWORK_ON_HOST_FINISHED	Info	(${Sequence}/${Total}): Successfully applied changes on host ${VdsName}. (User: ${UserName})
10904	PERSIST_SETUP_NETWORK_ON_HOST_FAILED	Error	(${Sequence}/${Total}): Failed to apply changes on host ${VdsName}. (User: ${UserName})
11000	USER_ADD_EXTERNAL_JOB	Info	New external Job ${description} was added by user ${UserName}
11001	USER_ADD_EXTERNAL_JOB_FAILED	Error	Failed to add new external Job ${description}

附录 D. 时区

D.1. 时区

在为一个虚拟机或虚拟机模板指定时区时，API 会把 Windows 标准格式时区名映射为 tz 数据库格式。这意味着 API 只接受特定 tz 数据库代码。如下表所示：

表 D.1. 接受的 tz 数据库代码

tz 数据库格式	Windows 标准格式
Africa/Cairo	Egypt Standard Time
Africa/Casablanca	Morocco Standard Time
Africa/Johannesburg	South Africa Standard Time
Africa/Lagos	W. Central Africa Standard Time
Africa/Nairobi	E. Africa Standard Time
Africa/Reykjavik	Greenwich Standard Time
Africa/Windhoek	Namibia Standard Time
America/Anchorage	Alaskan Standard Time
America/Bogota	SA Pacific Standard Time
America/Buenos_Aires	Argentina Standard Time
America/Caracas	Venezuela Standard Time
America/Chicago	Central Standard Time
America/Chihuahua	Mexico Standard Time
America/Chihuahua	Mountain Standard Time
America/Denver	Mountain Standard Time
America/Godthab	Greenland Standard Time
America/Guatemala	Central America Standard Time
America/Halifax	Atlantic Standard Time
America/La_Paz	SA Western Standard Time

tz 数据库格式	Windows 标准格式
America/Los_Angeles	Pacific Standard Time
America/Manaus	Central Brazilian Standard Time
America/Mexico_City	Central Standard Time
America/Mexico_City	Mexico Standard Time
America/Montevideo	Montevideo Standard Time
America/New_York	Eastern Standard Time
America/Phoenix	US Mountain Standard Time
America/Regina	Canada Central Standard Time
America/Santiago	Pacific SA Standard Time
America/Sao_Paulo	E. South America Standard Time
America/St_Johns	Newfoundland Standard Time
America/Tijuana	Pacific Standard Time
Asia/Amman	Jordan Standard Time
Asia/Baghdad	Arabic Standard Time
Asia/Baku	Azerbaijan Standard Time
Asia/Bangkok	SE Asia Standard Time
Asia/Beirut	Middle East Standard Time
Asia/Calcutta	India Standard Time
Asia/Colombo	Sri Lanka Standard Time
Asia/Dhaka	Central Asia Standard Time
Asia/Dubai	Arabian Standard Time
Asia/Irkutsk	North Asia East Standard Time

tz 数据库格式	Windows 标准格式
Asia/Jerusalem	Israel Standard Time
Asia/Kabul	Afghanistan Standard Time
Asia/Karachi	Pakistan Standard Time
Asia/Katmandu	Nepal Standard Time
Asia/Krasnoyarsk	North Asia Standard Time
Asia/Novosibirsk	N. Central Asia Standard Time
Asia/Rangoon	Myanmar Standard Time
Asia/Riyadh	Arab Standard Time
Asia/Seoul	Korea Standard Time
Asia/Shanghai	China Standard Time
Asia/Singapore	Singapore Standard Time
Asia/Taipei	Taipei Standard Time
Asia/Tashkent	West Asia Standard Time
Asia/Tehran	Iran Standard Time
Asia/Tokyo	Tokyo Standard Time
Asia/Vladivostok	Vladivostok Standard Time
Asia/Yakutsk	Yakutsk Standard Time
Asia/Yekaterinburg	Ekaterinburg Standard Time
Asia/Yerevan	Armenian Standard Time
Asia/Yerevan	Caucasus Standard Time
Atlantic/Azores	Azores Standard Time
Atlantic/Cape_Verde	Cape Verde Standard Time
Atlantic/South_Georgia	Mid-Atlantic Standard Time

tz 数据库格式	Windows 标准格式
Australia/Adelaide	Cen. Australia Standard Time
Australia/Brisbane	E. Australia Standard Time
Australia/Darwin	AUS Central Standard Time
Australia/Hobart	Tasmania Standard Time
Australia/Perth	W. Australia Standard Time
Australia/Sydney	AUS Eastern Standard Time
Etc/GMT-3	Georgian Standard Time
Etc/GMT+12	Dateline Standard Time
Etc/GMT+3	SA Eastern Standard Time
Etc/GMT+5	US Eastern Standard Time
Europe/Berlin	W. Europe Standard Time
Europe/Budapest	Central Europe Standard Time
Europe/Istanbul	GTB Standard Time
Europe/Kiev	FLE Standard Time
Europe/London	GMT Standard Time
Europe/Minsk	E. Europe Standard Time
Europe/Moscow	Russian Standard Time
Europe/Paris	Romance Standard Time
Europe/Warsaw	Central European Standard Time
Indian/Mauritius	Mauritius Standard Time
Pacific/Apia	Samoa Standard Time
Pacific/Auckland	New Zealand Standard Time
Pacific/Fiji	Fiji Standard Time

tz 数据库格式	Windows 标准格式
Pacific/Guadalcanal	Central Pacific Standard Time
Pacific/Honolulu	Hawaiian Standard Time
Pacific/Port_Moresby	West Pacific Standard Time
Pacific/Tongatapu	Tonga Standard Time

附录 E. 修订历史

修订 3.6-6.1 **Sat Jan 23 2016** **Red Hat Localization Services**

与 XML 源 3.6-6 版本同步的翻译文件

修订 3.6-6 **Fri 15 Jan 2016** **Red Hat Enterprise Virtualization 文档团队**

BZ#1281667 - 在包括事件代码的表格中添加了严重性级别和错误信息。

修订 3.6-5 **Thu 14 Jan 2016** **Red Hat Enterprise Virtualization 文档团队**

BZ#1284288 - 将 'rhevm' 管理网络改为 'ovirtmgmt'。

修订 3.6-4 **Wed 23 Dec 2015** **Red Hat Enterprise Virtualization 文档团队**

BZ#1194228 - 增加了与 All-Content 头相关的信息。

修订 3.6-3 **Wed 18 Nov 2015** **Red Hat Enterprise Virtualization 文档团队**

Red Hat Enterprise Virtualization 3.6 beta 的最终版。

修订 3.6-2 **Wed 18 Nov 2015** **Red Hat Enterprise Virtualization 文档团队**

BZ#1255281 - 增加了如何在虚拟机中进行内存热插的信息。
BZ#1255229 - 更新了电源管理 REST API 示例。
BZ#1250784 - 增加了在存储域中指定默认 'wipe after delete' 设置的方法。
BZ#1278783 - 更新了创建和更新虚拟机池的示例。

修订 3.6-1 **Mon 10 Aug 2015** **Red Hat Enterprise Virtualization 文档团队**

初始创建 Red Hat Enterprise Virtualization 3.6 发行本。