



Red Hat Virtualization 4.4

REST API 指南

使用红帽虚拟化 REST 应用程序编程接口

Red Hat Virtualization 4.4 REST API 指南

使用红帽虚拟化 REST 应用程序编程接口

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/REST_API_Guide.ent file |.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

摘要

本指南描述了 Red Hat Virtualization Manager 表述性状态转移应用程序编程接口。本指南从 ovirt-engine-api-model 代码中的文档注释生成，目前已完成。随着新内容可用，此文档的更新版本将会发布。

目录

第1章 简介	49
1.1. 表述状态传输	49
1.2. API 先决条件	49
第2章 身份验证和安全	50
2.1. TLS/SSL 认证	50
2.1.1. 获取 CA 证书	50
2.1.2. 将证书导入到客户端	52
2.2. AUTHENTICATION	52
2.2.1. OAuth 身份验证	52
2.2.2. 基本身份验证	53
2.2.3. 身份验证会话	54
2.2.3.1. 请求经过身份验证的会话	54
第3章 常见概念	56
3.1. 类型	56
3.2. 识别的类型	56
3.3. 对象 (OBJECT)	56
3.4. 集合	56
3.5. 表示	56
3.5.1. XML 表述	57
3.5.2. JSON 表示	57
3.6. 服务	58
3.7. 搜索	61
3.7.1. 最大结果参数	61
3.7.2. 问题单敏感度	61
3.7.3. 搜索语法	61
3.7.4. 通配符	62
3.7.5. 分页	62
3.8. 以下链接	63
3.9. 权限	65
3.10. 处理错误	65
第4章 快速入门示例	66
4.1. 访问 API 入口点	66
4.2. 列出数据中心	68
4.3. 列出主机集群	69
4.4. 列出逻辑网络	69
4.5. 列出主机	70
4.6. 创建 NFS 数据存储	71
4.7. 创建 NFS ISO 存储	73
4.8. 将存储域附加到数据中心	74
4.9. 创建虚拟机	75
4.10. 创建虚拟机 NIC	77
4.11. 创建虚拟机磁盘	78
4.12. 将 ISO 镜像附加到虚拟机	79
4.13. 启动虚拟机	81
第5章 REQUESTS	83
第6章 服务	149
6.1. AFFINITYGROUP	149

6.1.1. 获取 GET	149
6.1.1.1. follow	149
6.1.2. 删除 DELETE	150
6.1.3. 更新 PUT	150
6.2. AFFINITYGROUPOHOST	150
6.2.1. 删除 DELETE	150
6.3. AFFINITYGROUPOHOSTLABEL	151
6.3.1. 删除 DELETE	151
6.4. AFFINITYGROUPOHOSTLABELS	151
6.4.1. 添加 POST	151
6.4.2. 列出 GET	152
6.4.2.1. follow	152
6.4.2.2. Max	152
6.5. AFFINITYGROUPOHOSTS	153
6.5.1. 添加 POST	153
6.5.2. 列出 GET	153
6.5.2.1. follow	154
6.5.2.2. Max	154
6.6. AFFINITYGROUPOVM	154
6.6.1. 删除 DELETE	154
6.7. AFFINITYGROUPOVMLABEL	154
6.7.1. 删除 DELETE	155
6.8. AFFINITYGROUPOVMLABELS	155
6.8.1. 添加 POST	155
6.8.2. 列出 GET	156
6.8.2.1. follow	156
6.8.2.2. Max	156
6.9. AFFINITYGROUPOVMS	156
6.9.1. 添加 POST	157
6.9.2. 列出 GET	157
6.9.2.1. follow	157
6.9.2.2. Max	158
6.10. AFFINITYGROUPS	158
6.10.1. 添加 POST	158
6.10.2. 列出 GET	159
6.10.2.1. follow	159
6.10.2.2. Max	159
6.11. AFFINITYLABEL	159
6.11.1. 获取 GET	160
6.11.1.1. follow	160
6.11.2. 删除 DELETE	160
6.11.3. 更新 PUT	160
6.12. AFFINITYLABELHOST	160
6.12.1. 获取 GET	161
6.12.1.1. follow	161
6.12.2. 删除 DELETE	161
6.13. AFFINITYLABELHOSTS	161
6.13.1. 添加 POST	162
6.13.2. 列出 GET	162
6.13.2.1. follow	162
6.14. AFFINITYLABELVM	162
6.14.1. 获取 GET	163
6.14.1.1. follow	163

6.14.2. 删除 DELETE	163
6.15. AFFINITYLABELVMS	163
6.15.1. 添加 POST	163
6.15.2. 列出 GET	164
6.15.2.1. follow	164
6.16. AFFINITYLABELS	164
6.16.1. 添加 POST	164
6.16.2. 列出 GET	165
6.16.2.1. follow	165
6.16.2.2. Max	165
6.17. 区域	165
6.18. ASSIGNEDAFFINITYLABEL	166
6.18.1. 获取 GET	166
6.18.1.1. follow	167
6.18.2. 删除 DELETE	167
6.19. ASSIGNEDAFFINITYLABELS	167
6.19.1. 添加 POST	167
6.19.2. 列出 GET	167
6.19.2.1. follow	168
6.20. ASSIGNEDCPUPROFILE	168
6.20.1. 获取 GET	168
6.20.1.1. follow	168
6.20.2. 删除 DELETE	168
6.21. ASSIGNEDCPUPROFILES	169
6.21.1. 添加 POST	169
6.21.2. 列出 GET	169
6.21.2.1. follow	169
6.21.2.2. Max	170
6.22. ASSIGNEDDISKPROFILE	170
6.22.1. 获取 GET	170
6.22.1.1. follow	170
6.22.2. 删除 DELETE	170
6.23. ASSIGNEDDISKPROFILES	170
6.23.1. 添加 POST	171
6.23.2. 列出 GET	171
6.23.2.1. follow	171
6.23.2.2. Max	171
6.24. ASSIGNEDPERMISSIONS	172
6.24.1. 添加 POST	172
6.24.2. 列出 GET	173
6.24.2.1. follow	174
6.25. ASSIGNEDROLES	174
6.25.1. 列出 GET	174
6.25.1.1. follow	175
6.25.1.2. Max	175
6.26. ASSIGNEDTAG	175
6.26.1. 获取 GET	175
6.26.1.1. follow	176
6.26.2. 删除 DELETE	176
6.27. ASSIGNEDTAGS	176
6.27.1. 添加 POST	176
6.27.2. 列出 GET	177
6.27.2.1. follow	177

6.27.2.2. Max	178
6.28. ASSIGNEDVNICPROFILE	178
6.28.1. 获取 GET	178
6.28.1.1. follow	178
6.28.2. 删除 DELETE	178
6.29. ASSIGNEDVNICPROFILES	178
6.29.1. 添加 POST	179
6.29.2. 列出 GET	179
6.29.2.1. follow	179
6.29.2.2. Max	179
6.30. ATTACHEDSTORAGEDOMAIN	180
6.30.1. 激活 POST	180
6.30.2. 取消激活 POST	180
6.30.2.1. force	181
6.30.3. 获取 GET	181
6.30.3.1. follow	181
6.30.4. 删除 DELETE	182
6.31. ATTACHEDSTORAGEDOMAINDISK	182
6.31.1. 复制 POST	182
6.31.2. 导出 POST	183
6.31.3. 获取 GET	183
6.31.3.1. follow	184
6.31.4. 移动 POST	184
6.31.5. 注册 POST	184
6.31.6. 删除 DELETE	184
6.31.7. 稀疏 POST	184
6.31.8. 更新 PUT	185
6.32. ATTACHEDSTORAGEDOMAINDISKS	185
6.32.1. 添加 POST	185
6.32.1.1. unregistered	186
6.32.2. 列出 GET	187
6.32.2.1. 磁盘	187
6.32.2.2. follow	187
6.32.2.3. Max	187
6.33. ATTACHEDSTORAGEDOMAINS	187
6.33.1. 添加 POST	188
6.33.2. 列出 GET	188
6.33.2.1. follow	188
6.33.2.2. Max	188
6.34. BALANCE	188
6.34.1. 获取 GET	189
6.34.1.1. follow	189
6.34.2. 删除 DELETE	189
6.35. 平衡	189
6.35.1. 添加 POST	189
6.35.2. 列出 GET	190
6.35.2.1. follow	190
6.35.2.2. Max	190
6.36. 书签	190
6.36.1. 获取 GET	191
6.36.1.1. follow	191
6.36.2. 删除 DELETE	191
6.36.3. 更新 PUT	192

6.37. 书签	192
6.37.1. 添加 POST	193
6.37.2. 列出 GET	193
6.37.2.1. follow	194
6.37.2.2. Max	194
6.38. CLUSTER	194
6.38.1. 获取 GET	194
6.38.1.1. follow	196
6.38.2. refreshglusterhealstatus POST	196
6.38.3. 删除 DELETE	196
6.38.4. resetemulatedmachine POST	197
6.38.5. syncallnetworks POST	197
6.38.6. 更新 PUT	197
6.38.7. 升级 POST	198
6.38.7.1. correlation_id	199
6.39. CLUSTERENABLEDFEATURE	199
6.39.1. 获取 GET	199
6.39.1.1. follow	200
6.39.2. 删除 DELETE	200
6.40. CLUSTERENABLEDFEATURES	200
6.40.1. 添加 POST	201
6.40.2. 列出 GET	201
6.40.2.1. follow	202
6.41. CLUSTEREXTERNALPROVIDERS	202
6.41.1. 列出 GET	202
6.41.1.1. follow	202
6.42. CLUSTERFEATURE	203
6.42.1. 获取 GET	203
6.42.1.1. follow	203
6.43. CLUSTERFEATURES	203
6.43.1. 列出 GET	204
6.43.1.1. follow	204
6.44. CLUSTERLEVEL	204
6.44.1. 获取 GET	205
6.44.1.1. follow	205
6.45. CLUSTERLEVELS	206
6.45.1. 列出 GET	206
6.45.1.1. follow	206
6.46. CLUSTERNETWORK	207
6.46.1. 获取 GET	207
6.46.1.1. follow	207
6.46.2. 删除 DELETE	207
6.46.3. 更新 PUT	207
6.47. CLUSTERNETWORKS	208
6.47.1. 添加 POST	208
6.47.2. 列出 GET	208
6.47.2.1. follow	209
6.47.2.2. Max	209
6.48. 集群	209
6.48.1. 添加 POST	209
6.48.2. 列出 GET	210
6.48.2.1. case_sensitive	211
6.48.2.2. follow	211

6.48.2.3. Max	211
6.49. COPYABLE	211
6.49.1. 复制 POST	211
6.50. CPUPROFILE	212
6.50.1. 获取 GET	212
6.50.1.1. follow	212
6.50.2. 删除 DELETE	212
6.50.3. 更新 PUT	212
6.51. CPUPROFILES	213
6.51.1. 添加 POST	213
6.51.2. 列出 GET	213
6.51.2.1. follow	214
6.51.2.2. Max	214
6.52. DATACENTER	214
6.52.1. 清理finishedtasks POST	214
6.52.2. 获取 GET	215
6.52.2.1. follow	216
6.52.3. 删除 DELETE	216
6.52.3.1. force	216
6.52.4. setmaster POST	216
6.52.5. 更新 PUT	217
6.53. DATACENTERNETWORK	218
6.53.1. 获取 GET	218
6.53.1.1. follow	218
6.53.2. 删除 DELETE	218
6.53.3. 更新 PUT	219
6.54. DATACENTERNETWORKS	219
6.54.1. 添加 POST	219
6.54.2. 列出 GET	220
6.54.2.1. follow	220
6.54.2.2. Max	220
6.55. DATACENTERS	220
6.55.1. 添加 POST	221
6.55.2. 列出 GET	221
6.55.2.1. case_sensitive	223
6.55.2.2. follow	223
6.55.2.3. Max	223
6.56. DISK	223
6.56.1. 转换 POST	224
6.56.1.1. follow	224
6.56.2. 复制 POST	224
6.56.2.1. disk_profile	225
6.56.2.2. quota	226
6.56.2.3. storage_domain	226
6.56.3. 导出 POST	226
6.56.4. 获取 GET	227
6.56.4.1. all_content	227
6.56.4.2. follow	227
6.56.5. 移动 POST	227
6.56.5.1. disk_profile	228
6.56.5.2. quota	229
6.56.6. 减少 POST	229
6.56.7. refreshlun POST	229

6.56.8. 删除 DELETE	230
6.56.9. 稀疏 POST	230
6.56.10. 更新 PUT	230
6.57. DISKATTACHMENT	232
6.57.1. 获取 GET	232
6.57.1.1. follow	232
6.57.2. 删除 DELETE	233
6.57.2.1. detach_only	233
6.57.3. 更新 PUT	233
6.58. DISKATTACHMENTS	234
6.58.1. 添加 POST	234
6.58.2. 列出 GET	235
6.58.2.1. follow	235
6.59. DISKPROFILE	235
6.59.1. 获取 GET	236
6.59.1.1. follow	236
6.59.2. 删除 DELETE	236
6.59.3. 更新 PUT	236
6.60. DISKPROFILES	237
6.60.1. 添加 POST	237
6.60.2. 列出 GET	237
6.60.2.1. follow	237
6.60.2.2. Max	237
6.61. DISKSNAPSHOT	238
6.61.1. 获取 GET	238
6.61.1.1. follow	238
6.61.2. 删除 DELETE	238
6.62. DISKSNAPSHOTS	238
6.62.1. 列出 GET	239
6.62.1.1. follow	239
6.62.1.2. include_active	239
6.62.1.3. include_template	239
6.62.1.4. Max	239
6.63. 磁盘	240
6.63.1. 添加 POST	240
6.63.2. 列出 GET	243
6.63.2.1. case_sensitive	244
6.63.2.2. follow	244
6.63.2.3. Max	244
6.64. 域	244
6.64.1. 获取 GET	244
6.64.1.1. follow	245
6.65. DOMAINGROUP	245
6.65.1. 获取 GET	245
6.65.1.1. follow	246
6.66. DOMAINGROUPS	246
6.66.1. 列出 GET	246
6.66.1.1. case_sensitive	246
6.66.1.2. follow	247
6.66.1.3. Max	247
6.67. DOMAINUSER	247
6.67.1. 获取 GET	247
6.67.1.1. follow	248

6.68. DOMAINUSERGROUPS	248
6.68.1. 列出 GET	248
6.68.1.1. follow	248
6.69. DOMAINUSERS	249
6.69.1. 列出 GET	249
6.69.1.1. case_sensitive	250
6.69.1.2. follow	250
6.69.1.3. Max	250
6.70. 域	250
6.70.1. 列出 GET	250
6.70.1.1. follow	251
6.70.1.2. Max	251
6.71. ENGINEKATELLOERRATA	251
6.71.1. 列出 GET	252
6.71.1.1. follow	253
6.71.1.2. Max	253
6.72. EVENT	253
6.72.1. 获取 GET	253
6.72.1.1. follow	254
6.72.2. 删除 DELETE	254
6.73. EVENTSUBSCRIPTION	254
6.73.1. 获取 GET	255
6.73.2. 删除 DELETE	255
6.74. EVENTSUBSCRIPTIONS	255
6.74.1. 添加 POST	256
6.74.2. 列出 GET	256
6.74.2.1. follow	257
6.74.2.2. Max	257
6.75. 事件	257
6.75.1. 添加 POST	258
6.75.2. 列出 GET	259
6.75.2.1. case_sensitive	260
6.75.2.2. follow	260
6.75.2.3. 来自	260
6.75.2.4. Max	261
6.75.2.5. search	261
6.75.3. 取消删除 POST	262
6.76. EXTERNALCOMPUTERESOURCE	262
6.76.1. 获取 GET	262
6.76.1.1. follow	263
6.77. EXTERNALCOMPUTERESOURCES	263
6.77.1. 列出 GET	263
6.77.1.1. follow	264
6.77.1.2. Max	264
6.78. EXTERNALDISCOVEREDHOST	264
6.78.1. 获取 GET	265
6.78.1.1. follow	265
6.79. EXTERNALDISCOVEREDHOSTS	266
6.79.1. 列出 GET	266
6.79.1.1. follow	267
6.79.1.2. Max	267
6.80. EXTERNALHOST	267
6.80.1. 获取 GET	267

6.80.1.1. follow	267
6.81. EXTERNALHOSTGROUP	268
6.81.1. 获取 GET	268
6.81.1.1. follow	269
6.82. EXTERNALHOSTGROUPS	269
6.82.1. 列出 GET	269
6.82.1.1. follow	270
6.82.1.2. Max	270
6.83. EXTERNALHOSTPROVIDER	270
6.83.1. 获取 GET	271
6.83.1.1. follow	271
6.83.2. importcertificates POST	272
6.83.3. 删除 DELETE	272
6.83.4. 测试连接 POST	272
6.83.5. 更新 PUT	272
6.84. EXTERNALHOSTPROVIDERS	273
6.84.1. 添加 POST	273
6.84.2. 列出 GET	273
6.84.2.1. follow	274
6.84.2.2. Max	274
6.85. EXTERNALHOSTS	274
6.85.1. 列出 GET	274
6.85.1.1. follow	274
6.85.1.2. Max	275
6.86. EXTERNALNETWORKPROVIDERCONFIGURATION	275
6.86.1. 获取 GET	275
6.86.1.1. follow	275
6.87. EXTERNALNETWORKPROVIDERCONFIGURATIONS	275
6.87.1. 列出 GET	276
6.87.1.1. follow	276
6.88. EXTERNALPROVIDER	276
6.88.1. importcertificates POST	276
6.88.2. 测试连接 POST	277
6.89. EXTERNALPROVIDERCERTIFICATE	277
6.89.1. 获取 GET	277
6.89.1.1. follow	278
6.90. EXTERNALPROVIDERCERTIFICATES	278
6.90.1. 列出 GET	278
6.90.1.1. follow	279
6.90.1.2. Max	279
6.91. EXTERNALTEMPLATEIMPORTS	279
6.91.1. 添加 POST	279
6.92. EXTERNALVMIMPORTS	280
6.92.1. 添加 POST	280
6.93. FENCEAGENT	281
6.93.1. 获取 GET	281
6.93.1.1. follow	282
6.93.2. 删除 DELETE	282
6.93.3. 更新 PUT	282
6.94. FENCEAGENTS	282
6.94.1. 添加 POST	283
6.94.2. 列出 GET	284
6.94.2.1. follow	285

6.94.2.2. Max	285
6.95. FILE	285
6.95.1. 获取 GET	285
6.95.1.1. follow	285
6.96. 文件	286
6.96.1. 列出 GET	286
6.96.1.1. case_sensitive	287
6.96.1.2. follow	287
6.96.1.3. Max	287
6.97. FILTER	287
6.97.1. 获取 GET	287
6.97.1.1. follow	288
6.97.2. 删除 DELETE	288
6.98. 过滤器	288
6.98.1. 添加 POST	288
6.98.2. 列出 GET	289
6.98.2.1. follow	289
6.98.2.2. Max	289
6.99. 关注	289
6.100. GLUSTERBRICK	289
6.100.1. 获取 GET	290
6.100.1.1. follow	291
6.100.2. 删除 DELETE	291
6.100.3. 替换 POST	291
6.101. GLUSTERBRICKS	292
6.101.1. 激活 POST	292
6.101.2. 添加 POST	293
6.101.3. 列出 GET	294
6.101.3.1. follow	295
6.101.3.2. Max	295
6.101.4. 迁移 POST	295
6.101.5. 删除 DELETE	296
6.101.6. stopmigrate POST	297
6.101.6.1. brick	297
6.102. GLUSTERHOOK	297
6.102.1. 禁用 POST	298
6.102.2. 启用 POST	298
6.102.3. 获取 GET	298
6.102.3.1. follow	299
6.102.4. 删除 DELETE	299
6.102.5. 解析 POST	299
6.103. GLUSTERHOOKS	299
6.103.1. 列出 GET	300
6.103.1.1. follow	300
6.103.1.2. Max	300
6.104. GLUSTERVOLUME	300
6.104.1. 获取 GET	301
6.104.1.1. follow	302
6.104.2. getprofilestatistics POST	302
6.104.3. 重新平衡 POST	303
6.104.3.1. fix_layout	303
6.104.3.2. force	303
6.104.4. 删除 DELETE	304

6.104.5. resetalloptions POST	304
6.104.6. resetoption POST	304
6.104.7. setoption POST	305
6.104.8. 启动 POST	305
6.104.8.1. force	306
6.104.9. startprofile POST	306
6.104.10. 停止 POST	306
6.104.11. stopprofile POST	307
6.104.12. 停止平衡 POST	307
6.105. GLUSTERVOLUMES	308
6.105.1. 添加 POST	308
6.105.2. 列出 GET	309
6.105.2.1. case_sensitive	309
6.105.2.2. follow	310
6.105.2.3. Max	310
6.106. 组	310
6.106.1. 获取 GET	310
6.106.1.1. follow	311
6.106.2. 删除 DELETE	311
6.107. 组	311
6.107.1. 添加 POST	312
6.107.2. 列出 GET	312
6.107.2.1. case_sensitive	313
6.107.2.2. follow	313
6.107.2.3. Max	313
6.108. 主机	314
6.108.1. 激活 POST	315
6.108.2. 批准 POST	315
6.108.2.1. 激活	316
6.108.2.2. reboot	316
6.108.3. commitnetconfig POST	316
6.108.4. copyhostnetworks POST	317
6.108.5. 取消激活 POST	318
6.108.5.1. stop_gluster_service	318
6.108.6. 发现iscsi POST	318
6.108.7. 注册certificate POST	319
6.108.8. 隔离 POST	319
6.108.9. forceelectspm POST	320
6.108.10. 获取 GET	321
6.108.10.1. all_content	321
6.108.10.2. follow	321
6.108.11. 安装 POST	322
6.108.11.1. 激活	323
6.108.11.2. deploy_hosted_engine	323
6.108.11.3. reboot	323
6.108.11.4. undeploy_hosted_engine	324
6.108.12. iscsidiscover POST	324
6.108.12.1. iscsi_targets	324
6.108.13. iscsilogin POST	325
6.108.14. 刷新 POST	325
6.108.15. 删除 DELETE	325
6.108.16. setupnetworks POST	326
6.108.16.1. commit_on_success	330

6.108.17. syncallnetworks POST	330
6.108.18. unregisteredstoragedomainsdiscover POST	331
6.108.19. 更新 PUT	331
6.108.20. 升级 POST	332
6.108.20.1. reboot	332
6.108.20.2. timeout	332
6.108.21. upgradecheck POST	332
6.109. HOSTCPUUNITS	333
6.109.1. 列出 GET	333
6.109.1.1. follow	333
6.110. HOSTDEVICE	333
6.110.1. 获取 GET	333
6.110.1.1. follow	334
6.111. HOSTDEVICES	334
6.111.1. 列出 GET	334
6.111.1.1. follow	335
6.111.1.2. Max	335
6.112. HOSTHOOK	335
6.112.1. 获取 GET	335
6.112.1.1. follow	335
6.113. HOSTHOOKS	336
6.113.1. 列出 GET	336
6.113.1.1. follow	336
6.113.1.2. Max	336
6.114. HOSTNIC	336
6.114.1. 获取 GET	337
6.114.1.1. all_content	337
6.114.1.2. follow	337
6.114.2. 更新虚拟功能配置 POST	338
6.115. HOSTNICS	338
6.115.1. 列出 GET	338
6.115.1.1. all_content	339
6.115.1.2. follow	339
6.115.1.3. Max	339
6.116. HOSTNUMANODE	340
6.116.1. 获取 GET	340
6.116.1.1. follow	340
6.117. HOSTNUMANODES	340
6.117.1. 列出 GET	340
6.117.1.1. follow	341
6.117.1.2. Max	341
6.118. HOSTSTORAGE	341
6.118.1. 列出 GET	341
6.118.1.1. follow	342
6.118.1.2. report_status	342
6.119. 主机	343
6.119.1. 添加 POST	343
6.119.1.1. 激活	345
6.119.1.2. deploy_hosted_engine	345
6.119.1.3. reboot	345
6.119.1.4. undeploy_hosted_engine	345
6.119.2. 列出 GET	345
6.119.2.1. all_content	346

6.119.2.2. case_sensitive	347
6.119.2.3. check_vms_in_affinity_closure	347
6.119.2.4. follow	347
6.119.2.5. Max	348
6.119.2.6. migration_target_of	348
6.120. ICON	348
6.120.1. 获取 GET	348
6.120.1.1. follow	349
6.121. ICONS	349
6.121.1. 列出 GET	349
6.121.1.1. follow	350
6.121.1.2. Max	350
6.122. IMAGE	350
6.122.1. 获取 GET	350
6.122.1.1. follow	350
6.122.2. 导入 POST	350
6.123. IMAGETRANSFER	351
6.123.1. 取消 POST	356
6.123.2. 扩展 POST	356
6.123.3. 最终 POST	356
6.123.4. 获取 GET	356
6.123.4.1. follow	357
6.123.5. 暂停 POST	357
6.123.6. 恢复 POST	357
6.124. IMAGETRANSFERS	357
6.124.1. 添加 POST	358
6.124.2. 列出 GET	359
6.124.2.1. follow	359
6.125. 镜像	359
6.125.1. 列出 GET	359
6.125.1.1. follow	360
6.125.1.2. Max	360
6.126. INSTANCETYPE	360
6.126.1. 获取 GET	360
6.126.1.1. follow	361
6.126.2. 删除 DELETE	361
6.126.3. 更新 PUT	361
6.127. INSTANCETYPEGRAPHICSCONSOLE	362
6.127.1. 获取 GET	362
6.127.1.1. follow	362
6.127.2. 删除 DELETE	363
6.128. INSTANCETYPEGRAPHICSCONSOLES	363
6.128.1. 添加 POST	363
6.128.2. 列出 GET	363
6.128.2.1. follow	364
6.128.2.2. Max	364
6.129. INSTANCETYPENIC	364
6.129.1. 获取 GET	364
6.129.1.1. follow	364
6.129.2. 删除 DELETE	365
6.129.3. 更新 PUT	365
6.130. INSTANCETYPENICS	365
6.130.1. 添加 POST	365

6.130.2. 列出 GET	366
6.130.2.1. follow	366
6.130.2.2. Max	366
6.131. INSTANCETYPEWATCHDOG	366
6.131.1. 获取 GET	367
6.131.1.1. follow	367
6.131.2. 删除 DELETE	367
6.131.3. 更新 PUT	367
6.132. INSTANCETYPEWATCHDOGS	368
6.132.1. 添加 POST	368
6.132.2. 列出 GET	368
6.132.2.1. follow	369
6.132.2.2. Max	369
6.133. INSTANCETYPES	369
6.133.1. 添加 POST	369
6.133.2. 列出 GET	371
6.133.2.1. case_sensitive	371
6.133.2.2. follow	371
6.133.2.3. Max	371
6.134. ISCSIBOND	372
6.134.1. 获取 GET	372
6.134.1.1. follow	372
6.134.2. 删除 DELETE	372
6.134.3. 更新 PUT	372
6.135. ISCSIBONDS	373
6.135.1. 添加 POST	373
6.135.2. 列出 GET	374
6.135.2.1. follow	374
6.135.2.2. Max	374
6.136. JOB	375
6.136.1. 清除 POST	375
6.136.2. 最终 POST	375
6.136.2.1. succeeded	376
6.136.3. 获取 GET	376
6.136.3.1. follow	377
6.137. JOBS	377
6.137.1. 添加 POST	377
6.137.2. 列出 GET	378
6.137.2.1. case_sensitive	379
6.137.2.2. follow	379
6.137.2.3. Max	379
6.138. KATELLOERRATA	379
6.138.1. 列出 GET	380
6.138.1.1. follow	381
6.138.1.2. Max	381
6.139. KATELLOERRATUM	381
6.139.1. 获取 GET	381
6.139.1.1. follow	382
6.140. LINKLAYERDISCOVERYPROTOCOL	382
6.140.1. 列出 GET	382
6.140.1.1. 元素	383
6.140.1.2. follow	383
6.141. MACPOOL	383

6.141.1. 获取 GET	383
6.141.1.1. follow	384
6.141.2. 删除 DELETE	384
6.141.3. 更新 PUT	384
6.142. MACPOOLS	385
6.142.1. 添加 POST	385
6.142.2. 列出 GET	386
6.142.2.1. follow	386
6.142.2.2. Max	386
6.143. 可测量	387
6.144. 可移动	387
6.144.1. 移动 POST	387
6.145. NETWORK	387
6.145.1. 获取 GET	387
6.145.1.1. follow	388
6.145.2. 删除 DELETE	388
6.145.3. 更新 PUT	389
6.146. NETWORKATTACHMENT	390
6.146.1. 获取 GET	390
6.146.1.1. follow	391
6.146.2. 删除 DELETE	391
6.146.3. 更新 PUT	391
6.147. NETWORKATTACHMENTS	391
6.147.1. 添加 POST	391
6.147.2. 列出 GET	392
6.147.2.1. follow	392
6.147.2.2. Max	392
6.148. NETWORKFILTER	392
6.148.1. 获取 GET	393
6.148.1.1. follow	393
6.149. NETWORKFILTERS	393
6.149.1. 列出 GET	395
6.149.1.1. follow	395
6.150. NETWORKLABEL	395
6.150.1. 获取 GET	395
6.150.1.1. follow	396
6.150.2. 删除 DELETE	396
6.151. NETWORKLABELS	396
6.151.1. 添加 POST	396
6.151.2. 列出 GET	397
6.151.2.1. follow	397
6.151.2.2. Max	398
6.152. 网络	398
6.152.1. 添加 POST	398
6.152.2. 列出 GET	399
6.152.2.1. case_sensitive	400
6.152.2.2. follow	400
6.152.2.3. Max	400
6.153. NICNETWORKFILTERPARAMETER	400
6.153.1. 获取 GET	401
6.153.1.1. follow	401
6.153.2. 删除 DELETE	401
6.153.3. 更新 PUT	401

6.154. NICNETWORKFILTERPARAMETERS	402
6.154.1. 添加 POST	402
6.154.2. 列出 GET	403
6.154.2.1. follow	403
6.155. OPENSTACKIMAGE	403
6.155.1. 获取 GET	404
6.155.1.1. follow	404
6.155.2. 导入 POST	404
6.156. OPENSTACKIMAGEPROVIDER	405
6.156.1. 获取 GET	405
6.156.1.1. follow	405
6.156.2. importcertificates POST	406
6.156.3. 删除 DELETE	406
6.156.4. 测试连接 POST	406
6.156.5. 更新 PUT	406
6.157. OPENSTACKIMAGEPROVIDERS	407
6.157.1. 添加 POST	407
6.157.2. 列出 GET	407
6.157.2.1. follow	408
6.157.2.2. Max	408
6.158. OPENSTACKIMAGES	408
6.158.1. 列出 GET	408
6.158.1.1. follow	408
6.158.1.2. Max	408
6.159. OPENSTACKNETWORK	409
6.159.1. 获取 GET	409
6.159.1.1. follow	409
6.159.2. 导入 POST	409
6.159.2.1. data_center	409
6.160. OPENSTACKNETWORKPROVIDER	410
6.160.1. 获取 GET	410
6.160.1.1. follow	411
6.160.2. importcertificates POST	411
6.160.3. 删除 DELETE	411
6.160.4. 测试连接 POST	411
6.160.5. 更新 PUT	412
6.161. OPENSTACKNETWORKPROVIDERS	412
6.161.1. 添加 POST	413
6.161.2. 列出 GET	413
6.161.2.1. follow	413
6.161.2.2. Max	414
6.162. OPENSTACKNETWORKS	414
6.162.1. 列出 GET	414
6.162.1.1. follow	414
6.162.1.2. Max	414
6.163. OPENSTACKSUBNET	415
6.163.1. 获取 GET	415
6.163.1.1. follow	415
6.163.2. 删除 DELETE	415
6.164. OPENSTACKSUBNETS	415
6.164.1. 添加 POST	415
6.164.2. 列出 GET	416
6.164.2.1. follow	416

6.164.2.2. Max	416
6.165. OPENSTACKVOLUMEAUTHENTICATIONKEY	416
6.165.1. 获取 GET	417
6.165.1.1. follow	417
6.165.2. 删除 DELETE	417
6.165.3. 更新 PUT	417
6.166. OPENSTACKVOLUMEAUTHENTICATIONKEYS	418
6.166.1. 添加 POST	418
6.166.2. 列出 GET	418
6.166.2.1. follow	419
6.166.2.2. Max	419
6.167. OPENSTACKVOLUMEPROVIDER	419
6.167.1. 获取 GET	419
6.167.1.1. follow	419
6.167.2. importcertificates POST	420
6.167.3. 删除 DELETE	420
6.167.3.1. force	420
6.167.4. 测试连接 POST	420
6.167.5. 更新 PUT	420
6.168. OPENSTACKVOLUMEPROVIDERS	421
6.168.1. 添加 POST	421
6.168.2. 列出 GET	422
6.168.2.1. follow	422
6.168.2.2. Max	422
6.169. OPENSTACKVOLUMETYPE	422
6.169.1. 获取 GET	423
6.169.1.1. follow	423
6.170. OPENSTACKVOLUMETYPES	423
6.170.1. 列出 GET	423
6.170.1.1. follow	424
6.170.1.2. Max	424
6.171. OPERATINGSYSTEM	424
6.171.1. 获取 GET	424
6.171.1.1. follow	424
6.172. OPERATINGSYSTEMS	424
6.172.1. 列出 GET	425
6.172.1.1. follow	425
6.172.1.2. Max	425
6.173. 权限	425
6.173.1. 获取 GET	426
6.173.1.1. follow	426
6.173.2. 删除 DELETE	426
6.174. ALLOW	426
6.174.1. 获取 GET	426
6.174.1.1. follow	427
6.174.2. 删除 DELETE	427
6.175. ALLOW	427
6.175.1. 添加 POST	428
6.175.2. 列出 GET	428
6.175.2.1. follow	429
6.175.2.2. Max	429
6.176. QOS	429
6.176.1. 获取 GET	429

6.176.1.1. follow	430
6.176.2. 删除 DELETE	430
6.176.3. 更新 PUT	430
6.177. QOSS	431
6.177.1. 添加 POST	431
6.177.2. 列出 GET	432
6.177.2.1. follow	433
6.177.2.2. Max	433
6.178. QUOTA	433
6.178.1. 获取 GET	433
6.178.1.1. follow	434
6.178.2. 删除 DELETE	434
6.178.3. 更新 PUT	434
6.179. QUOTA_CLUSTERLIMIT	435
6.179.1. 获取 GET	435
6.179.1.1. follow	435
6.179.2. 删除 DELETE	435
6.180. QUOTA_CLUSTERLIMITS	436
6.180.1. 添加 POST	436
6.180.2. 列出 GET	436
6.180.2.1. follow	437
6.180.2.2. Max	437
6.181. QUOTA_STORAGELIMIT	437
6.181.1. 获取 GET	437
6.181.1.1. follow	437
6.181.2. 删除 DELETE	437
6.182. QUOTA_STORAGELIMITS	438
6.182.1. 添加 POST	438
6.182.2. 列出 GET	439
6.182.2.1. follow	439
6.182.2.2. Max	439
6.183. 配额	439
6.183.1. 添加 POST	440
6.183.2. 列出 GET	440
6.183.2.1. follow	441
6.183.2.2. Max	441
6.184. 角色	441
6.184.1. 获取 GET	441
6.184.1.1. follow	442
6.184.2. 删除 DELETE	442
6.184.3. 更新 PUT	442
6.185. 角色	443
6.185.1. 添加 POST	443
6.185.2. 列出 GET	444
6.185.2.1. follow	444
6.185.2.2. Max	444
6.186. SCHEDULINGPOLICIES	445
6.186.1. 添加 POST	445
6.186.2. 列出 GET	445
6.186.2.1. follow	446
6.186.2.2. Max	446
6.187. SCHEDULINGPOLICY	446
6.187.1. 获取 GET	446

6.187.1.1. follow	446
6.187.2. 删除 DELETE	447
6.187.3. 更新 PUT	447
6.188. SCHEDULINGPOLICYUNIT	447
6.188.1. 获取 GET	447
6.188.1.1. follow	447
6.188.2. 删除 DELETE	448
6.189. SCHEDULINGPOLICYUNITS	448
6.189.1. 列出 GET	448
6.189.1.1. follow	448
6.189.1.2. Max	449
6.190. SNAPSHOT	449
6.190.1. 获取 GET	449
6.190.1.1. follow	449
6.190.2. 删除 DELETE	449
6.190.2.1. all_content	450
6.190.3. 恢复 POST	450
6.190.3.1. 磁盘	451
6.191. SNAPSHOTCDROM	451
6.191.1. 获取 GET	451
6.191.1.1. follow	451
6.192. SNAPSHOTCDROMS	452
6.192.1. 列出 GET	452
6.192.1.1. follow	452
6.192.1.2. Max	452
6.193. SNAPSHOTDISK	452
6.193.1. 获取 GET	453
6.193.1.1. follow	453
6.194. SNAPSHOTDISKS	453
6.194.1. 列出 GET	453
6.194.1.1. follow	454
6.194.1.2. Max	454
6.195. SNAPSHOTNIC	454
6.195.1. 获取 GET	454
6.195.1.1. follow	454
6.196. SNAPSHOTNICS	454
6.196.1. 列出 GET	455
6.196.1.1. follow	455
6.196.1.2. Max	455
6.197. 快照	455
6.197.1. 添加 POST	456
6.197.2. 列出 GET	457
6.197.2.1. all_content	457
6.197.2.2. follow	458
6.197.2.3. Max	458
6.198. SSHPUBLICKEY	458
6.198.1. 获取 GET	458
6.198.1.1. follow	458
6.198.2. 删除 DELETE	458
6.198.3. 更新 PUT	459
6.199. SSHPUBLICKEYS	459
6.199.1. 添加 POST	459
6.199.2. 列出 GET	460

6.199.2.1. follow	461
6.199.2.2. Max	461
6.200. 统计	461
6.200.1. 获取 GET	461
6.200.1.1. follow	461
6.201. STATISTICS	461
6.201.1. 列出 GET	462
6.201.1.1. follow	463
6.201.1.2. Max	463
6.202. 步骤	463
6.202.1. 最终 POST	463
6.202.1.1. succeeded	464
6.202.2. 获取 GET	464
6.202.2.1. follow	465
6.203. 步骤	465
6.203.1. 添加 POST	465
6.203.2. 列出 GET	466
6.203.2.1. follow	467
6.203.2.2. Max	467
6.204. 存储	467
6.204.1. 获取 GET	467
6.204.1.1. follow	468
6.204.1.2. report_status	468
6.205. STORAGEDOMAIN	469
6.205.1. 获取 GET	469
6.205.1.1. follow	470
6.205.2. 附加 POST	470
6.205.3. 减少luns POST	470
6.205.4. refreshluns POST	471
6.205.5. 删除 DELETE	472
6.205.5.1. destroy	472
6.205.5.2. 主机	473
6.205.6. 更新 PUT	473
6.205.7. updateovfstore POST	474
6.206. STORAGEDOMAINCONTENTDISK	474
6.206.1. 获取 GET	474
6.206.1.1. follow	475
6.207. STORAGEDOMAINCONTENTDISKS	475
6.207.1. 列出 GET	475
6.207.1.1. case_sensitive	476
6.207.1.2. follow	476
6.207.1.3. Max	476
6.208. STORAGEDOMAINDISK	476
6.208.1. 复制 POST	477
6.208.2. 导出 POST	477
6.208.3. 获取 GET	477
6.208.3.1. follow	478
6.208.4. 移动 POST	478
6.208.5. 减少 POST	478
6.208.6. 删除 DELETE	479
6.208.7. 稀疏 POST	479
6.208.8. 更新 PUT	479
6.209. STORAGEDOMAINDISKS	480

6.209.1. 添加 POST	480
6.209.1.1. unregistered	480
6.209.2. 列出 GET	481
6.209.2.1. follow	482
6.209.2.2. Max	482
6.209.2.3. unregistered	482
6.210. STORAGEDOMAINSERVERCONNECTION	482
6.210.1. 获取 GET	482
6.210.1.1. follow	483
6.210.2. 删除 DELETE	483
6.211. STORAGEDOMAINSERVERCONNECTIONS	483
6.211.1. 添加 POST	483
6.211.2. 列出 GET	483
6.211.2.1. follow	484
6.211.2.2. Max	484
6.212. STORAGEDOMAINTEMPLATE	484
6.212.1. 获取 GET	484
6.212.1.1. follow	485
6.212.2. 导入 POST	485
6.212.2.1. clone	486
6.212.3. 注册 POST	486
6.212.3.1. allow_partial_import	487
6.212.3.2. registration_configuration	487
6.212.3.3. vnic_profile_mappings	487
6.212.4. 删除 DELETE	487
6.213. STORAGEDOMAINTEMPLATES	488
6.213.1. 列出 GET	488
6.213.1.1. follow	488
6.213.1.2. Max	488
6.213.1.3. unregistered	489
6.214. STORAGEDOMAINVM	489
6.214.1. 获取 GET	489
6.214.1.1. follow	489
6.214.2. 导入 POST	489
6.214.2.1. clone	491
6.214.2.2. collapse_snapshots	491
6.214.3. 注册 POST	491
6.214.3.1. allow_partial_import	492
6.214.3.2. reassign_bad_macs	492
6.214.3.3. registration_configuration	493
6.214.3.4. vnic_profile_mappings	493
6.214.4. 删除 DELETE	493
6.215. STORAGEDOMAINVMDISKATTACHMENT	493
6.215.1. 获取 GET	494
6.215.1.1. follow	494
6.216. STORAGEDOMAINVMDISKATTACHMENTS	494
6.216.1. 列出 GET	494
6.216.1.1. follow	495
6.217. STORAGEDOMAINVMS	495
6.217.1. 列出 GET	496
6.217.1.1. follow	496
6.217.1.2. Max	496
6.217.1.3. unregistered	496

6.218. STORAGEDOMAINS	497
6.218.1. 添加 POST	497
6.218.2. 列出 GET	498
6.218.2.1. case_sensitive	499
6.218.2.2. follow	499
6.218.2.3. Max	499
6.219. STORAGESERVERCONNECTION	499
6.219.1. 获取 GET	500
6.219.1.1. follow	500
6.219.2. 删除 DELETE	500
6.219.2.1. 主机	500
6.219.3. 更新 PUT	501
6.219.3.1. force	501
6.220. STORAGESERVERCONNECTIONEXTENSION	502
6.220.1. 获取 GET	502
6.220.1.1. follow	502
6.220.2. 删除 DELETE	502
6.220.3. 更新 PUT	502
6.221. STORAGESERVERCONNECTIONEXTENSIONS	503
6.221.1. 添加 POST	503
6.221.2. 列出 GET	504
6.221.2.1. follow	504
6.221.2.2. Max	504
6.222. STORAGESERVERCONNECTIONS	505
6.222.1. 添加 POST	505
6.222.2. 列出 GET	505
6.222.2.1. follow	506
6.222.2.2. Max	506
6.223. 系统	506
6.223.1. 获取 GET	506
6.223.1.1. follow	508
6.223.2. 重新载入配置 POST	508
6.224. SYSTEMOPTION	508
6.224.1. 获取 GET	509
6.224.1.1. version	510
6.225. SYSTEMOPTIONS	510
6.226. SYSTEMPERMISSIONS	511
6.226.1. 添加 POST	511
6.226.2. 列出 GET	512
6.226.2.1. follow	513
6.227. TAG	513
6.227.1. 获取 GET	513
6.227.1.1. follow	514
6.227.2. 删除 DELETE	514
6.227.3. 更新 PUT	514
6.228. TAGS	515
6.228.1. 添加 POST	515
6.228.2. 列出 GET	516
6.228.2.1. follow	517
6.228.2.2. Max	517
6.229. TEMPLATE	517
6.229.1. 导出 POST	518
6.229.1.1. 专用	518

6.229.2. 获取 GET	519
6.229.2.1. follow	519
6.229.3. 删除 DELETE	519
6.229.4. 更新 PUT	519
6.230. TEMPLATECDROM	520
6.230.1. 获取 GET	520
6.230.1.1. cdrom	521
6.230.1.2. follow	521
6.231. TEMPLATECDROMS	521
6.231.1. 列出 GET	522
6.231.1.1. follow	522
6.231.1.2. Max	522
6.232. TEMPLATEDISK	522
6.232.1. 复制 POST	523
6.232.2. 导出 POST	523
6.232.3. 获取 GET	523
6.232.3.1. follow	523
6.232.4. 删除 DELETE	524
6.233. TEMPLATEDISKATTACHMENT	524
6.233.1. 获取 GET	524
6.233.1.1. follow	524
6.233.2. 删除 DELETE	524
6.234. TEMPLATEDISKATTACHMENTS	525
6.234.1. 列出 GET	525
6.234.1.1. follow	526
6.235. TEMPLATEDISKS	526
6.235.1. 列出 GET	526
6.235.1.1. follow	526
6.235.1.2. Max	526
6.236. TEMPLATEGRAPHICSCONSOLE	526
6.236.1. 获取 GET	527
6.236.1.1. follow	527
6.236.2. 删除 DELETE	527
6.237. TEMPLATEGRAPHICSCONSOLES	527
6.237.1. 添加 POST	528
6.237.2. 列出 GET	528
6.237.2.1. follow	528
6.237.2.2. Max	528
6.238. TEMPLATEMEDIATEDDEVICE	528
6.238.1. 获取 GET	529
6.238.1.1. follow	529
6.238.2. 删除 DELETE	529
6.238.3. 更新 PUT	529
6.238.3.1. devices	530
6.239. TEMPLATEMEDIATEDDEVICES	530
6.239.1. 添加 POST	531
6.239.2. 列出 GET	531
6.239.2.1. follow	531
6.239.2.2. Max	532
6.240. TEMPLATENIC	532
6.240.1. 获取 GET	532
6.240.1.1. follow	532
6.240.2. 删除 DELETE	532

6.240.3. 更新 PUT	532
6.241. TEMPLATENICS	533
6.241.1. 添加 POST	533
6.241.2. 列出 GET	533
6.241.2.1. follow	534
6.241.2.2. Max	534
6.242. TEMPLATEWATCHDOG	534
6.242.1. 获取 GET	534
6.242.1.1. follow	534
6.242.2. 删除 DELETE	534
6.242.3. 更新 PUT	535
6.243. TEMPLATEWATCHDOGS	535
6.243.1. 添加 POST	535
6.243.2. 列出 GET	535
6.243.2.1. follow	536
6.243.2.2. Max	536
6.244. 模板	536
6.244.1. 添加 POST	536
6.244.1.1. clone_permissions	539
6.244.1.2. seal	539
6.244.2. 列出 GET	540
6.244.2.1. case_sensitive	540
6.244.2.2. follow	540
6.244.2.3. Max	540
6.245. UNMANAGEDNETWORK	541
6.245.1. 获取 GET	541
6.245.1.1. follow	541
6.245.2. 删除 DELETE	541
6.246. UNMANAGEDNETWORKS	541
6.246.1. 列出 GET	542
6.246.1.1. follow	542
6.246.1.2. Max	542
6.247. 用户	542
6.247.1. 获取 GET	543
6.247.1.1. follow	543
6.247.2. 删除 DELETE	543
6.247.3. 更新 PUT	544
6.248. USEROPTION	545
6.248.1. 获取 GET	545
6.248.2. 删除 DELETE	545
6.249. USEROPTIONS	546
6.249.1. 添加 POST	546
6.249.2. 列出 GET	546
6.250. 用户	547
6.250.1. 添加 POST	547
6.250.2. 列出 GET	548
6.250.2.1. case_sensitive	549
6.250.2.2. follow	549
6.250.2.3. Max	549
6.251. VIRTUALFUNCTIONALLOWEDNETWORK	550
6.251.1. 获取 GET	550
6.251.1.1. follow	550
6.251.2. 删除 DELETE	550

6.252. VIRTUALFUNCTIONALLOWEDNETWORKS	550
6.252.1. 添加 POST	550
6.252.2. 列出 GET	551
6.252.2.1. follow	551
6.252.2.2. Max	551
6.253. VM	551
6.253.1. autopincpuandnumanodes POST	553
6.253.1.1. optimize_cpu_settings	553
6.253.2. 取消迁移 POST	554
6.253.3. 克隆 POST	554
6.253.3.1. discard_snapshots	554
6.253.4. commitsnapshot POST	554
6.253.5. 分离 POST	555
6.253.6. 导出 POST	555
6.253.7. freezefilesystems POST	556
6.253.8. 获取 GET	557
6.253.8.1. all_content	557
6.253.8.2. follow	558
6.253.8.3. next_run	558
6.253.8.4. ovf_as_ova	559
6.253.9. 登录 POST	559
6.253.10. 维护 POST	560
6.253.11. 迁移 POST	560
6.253.11.1. cluster	561
6.253.11.2. force	561
6.253.11.3. 主机	561
6.253.11.4. migrate_vms_in_affinity_closure	561
6.253.12. previewsnapshot POST	562
6.253.12.1. 磁盘	562
6.253.12.2. lease	563
6.253.13. 重启 POST	563
6.253.14. 删除 DELETE	564
6.253.14.1. force	564
6.253.15. reordermacaddresses POST	564
6.253.16. 重置 POST	565
6.253.17. 截屏 POST	565
6.253.18. 关闭 POST	565
6.253.18.1. reason	566
6.253.19. 启动 POST	566
6.253.19.1. pause	567
6.253.19.2. use_cloud_init	567
6.253.19.3. use_ignition	568
6.253.19.4. use_initialization	568
6.253.19.5. use_sysprep	568
6.253.19.6. vm	568
6.253.19.7. 易失性	568
6.253.20. 停止 POST	568
6.253.20.1. reason	569
6.253.21. 挂起 POST	569
6.253.22. thawfilesystems POST	570
6.253.23. ticket POST	570
6.253.24. undosnapshot POST	571
6.253.25. 更新 PUT	572

6.253.25.1. next_run	572
6.254. VMAPPLICATION	572
6.254.1. 获取 GET	572
6.254.1.1. 应用	573
6.254.1.2. follow	573
6.255. VMAPPLICATIONS	573
6.255.1. 列出 GET	574
6.255.1.1. 应用程序	574
6.255.1.2. follow	575
6.255.1.3. Max	575
6.256. VMBACKUP	575
6.256.1. 最终 POST	575
6.256.2. 获取 GET	575
6.256.2.1. backup	576
6.256.2.2. follow	576
6.257. VMBACKUPDISK	576
6.257.1. 获取 GET	576
6.257.1.1. follow	577
6.258. VMBACKUPDISKS	577
6.258.1. 列出 GET	577
6.258.1.1. follow	577
6.258.1.2. Max	577
6.259. VMBACKUPS	578
6.259.1. 添加 POST	578
6.259.1.1. require_consistency	579
6.259.1.2. use_active	580
6.259.2. 列出 GET	580
6.259.2.1. Backups	580
6.259.2.2. follow	581
6.259.2.3. Max	581
6.260. VMCDROM	581
6.260.1. 获取 GET	581
6.260.1.1. current	582
6.260.1.2. follow	582
6.260.2. 更新 PUT	582
6.260.2.1. current	583
6.261. VMCDROMS	584
6.261.1. 添加 POST	584
6.261.2. 列出 GET	584
6.261.2.1. follow	585
6.261.2.2. Max	585
6.262. VMCHECKPOINT	585
6.262.1. 获取 GET	585
6.262.1.1. checkpoint	585
6.262.1.2. follow	586
6.262.2. 删除 DELETE	586
6.263. VMCHECKPOINTDISK	586
6.263.1. 获取 GET	586
6.263.1.1. follow	586
6.264. VMCHECKPOINTDISKS	587
6.264.1. 列出 GET	587
6.264.1.1. follow	587
6.264.1.2. Max	587

6.265. VMCHECKPOINTS	587
6.265.1. 列出 GET	588
6.265.1.1. checkpoints	588
6.265.1.2. follow	588
6.265.1.3. Max	588
6.266. VMDISK	589
6.266.1. 激活 POST	589
6.266.2. 取消激活 POST	589
6.266.3. 导出 POST	589
6.266.4. 获取 GET	590
6.266.4.1. follow	590
6.266.5. 移动 POST	590
6.266.6. 减少 POST	590
6.266.7. 删除 DELETE	591
6.266.8. 更新 PUT	591
6.267. VMDISKS	591
6.267.1. 添加 POST	591
6.267.2. 列出 GET	592
6.267.2.1. follow	592
6.267.2.2. Max	592
6.268. VMGRAPHICSCONSOLE	592
6.268.1. 获取 GET	593
6.268.1.1. current	593
6.268.1.2. follow	593
6.268.2. proxyticket POST	594
6.268.3. remoteviewerconnectionfile POST	594
6.268.3.1. remote_viewer_connection_file	595
6.268.4. 删除 DELETE	596
6.268.5. ticket POST	596
6.269. VMGRAPHICSCONSOLES	596
6.269.1. 添加 POST	597
6.269.2. 列出 GET	597
6.269.2.1. current	598
6.269.2.2. follow	598
6.269.2.3. Max	598
6.270. VMHOSTDEVICE	598
6.270.1. 获取 GET	599
6.270.1.1. follow	599
6.270.2. 删除 DELETE	600
6.271. VMHOSTDEVICES	600
6.271.1. 添加 POST	600
6.271.2. 列出 GET	601
6.271.2.1. follow	602
6.271.2.2. Max	602
6.272. VMMEDIAEDDEVICE	602
6.272.1. 获取 GET	602
6.272.1.1. follow	602
6.272.2. 删除 DELETE	602
6.272.3. 更新 PUT	603
6.272.3.1. device	604
6.273. VMMEDIAEDDEVICES	604
6.273.1. 添加 POST	604
6.273.2. 列出 GET	604

6.273.2.1. follow	605
6.273.2.2. Max	605
6.274. VMNIC	605
6.274.1. 激活 POST	605
6.274.2. 取消激活 POST	606
6.274.3. 获取 GET	606
6.274.3.1. follow	606
6.274.4. 删除 DELETE	606
6.274.5. 更新 PUT	607
6.275. VMNICS	608
6.275.1. 添加 POST	608
6.275.2. 列出 GET	610
6.275.2.1. follow	610
6.275.2.2. Max	610
6.276. VMNUMANODE	610
6.276.1. 获取 GET	610
6.276.1.1. follow	611
6.276.2. 删除 DELETE	611
6.276.3. 更新 PUT	611
6.277. VMNUMANODES	612
6.277.1. 添加 POST	612
6.277.2. 列出 GET	613
6.277.2.1. follow	613
6.277.2.2. Max	613
6.278. VMPOOL	614
6.278.1. assignvm POST	614
6.278.2. 获取 GET	614
6.278.2.1. follow	615
6.278.3. 删除 DELETE	615
6.278.4. 更新 PUT	615
6.278.4.1. seal	616
6.279. VMPOOLS	617
6.279.1. 添加 POST	617
6.279.1.1. seal	617
6.279.2. 列出 GET	618
6.279.2.1. case_sensitive	619
6.279.2.2. follow	619
6.279.2.3. Max	619
6.280. VMREPORTEDDEVICE	619
6.280.1. 获取 GET	619
6.280.1.1. follow	620
6.281. VMREPORTEDDEVICES	620
6.281.1. 列出 GET	620
6.281.1.1. follow	620
6.281.1.2. Max	621
6.282. VMSESSION	621
6.282.1. 获取 GET	621
6.282.1.1. follow	621
6.283. VMSESSIONS	621
6.283.1. 列出 GET	621
6.283.1.1. follow	622
6.283.1.2. Max	622
6.284. VMWATCHDOG	622

6.284.1. 获取 GET	623
6.284.1.1. follow	623
6.284.1.2. watchdog	623
6.284.2. 删除 DELETE	624
6.284.3. 更新 PUT	624
6.284.3.1. watchdog	625
6.285. VMWATCHDOGS	625
6.285.1. 添加 POST	625
6.285.1.1. watchdog	626
6.285.2. 列出 GET	626
6.285.2.1. follow	626
6.285.2.2. Max	627
6.285.2.3. watchdogs	627
6.286. VM	627
6.286.1. 添加 POST	627
6.286.1.1. auto_pinning_policy	630
6.286.1.2. clone	631
6.286.1.3. clone_permissions	632
6.286.1.4. filter	632
6.286.1.5. seal	633
6.286.2. 列出 GET	633
6.286.2.1. all_content	634
6.286.2.2. case_sensitive	635
6.286.2.3. follow	635
6.286.2.4. ovf_as_ova	635
6.287. VNICPROFILE	635
6.287.1. 获取 GET	636
6.287.1.1. follow	636
6.287.2. 删除 DELETE	636
6.287.3. 更新 PUT	636
6.288. VNICPROFILES	637
6.288.1. 添加 POST	637
6.288.2. 列出 GET	638
6.288.2.1. follow	639
6.288.2.2. Max	639
6.289. WEIGHT	639
6.289.1. 获取 GET	639
6.289.1.1. follow	640
6.289.2. 删除 DELETE	640
6.290. WEIGHT	640
6.290.1. 添加 POST	640
6.290.2. 列出 GET	640
6.290.2.1. follow	641
6.290.2.2. Max	641
第 7 章 类型	642
7.1. ACCESSPROTOCOL 枚举	642
7.2. ACTION STRUCT	642
7.3. AFFINITYGROUP STRUCT	647
7.3.1. broken	648
7.3.2. Enforcing	648
7.3.3. 正	648
7.4. AFFINITYLABELSTRUCT	649

7.4.1. has_implicit_affinity_group	650
7.4.2. read_only	650
7.5. AFFINITYRULE STRUCT	650
7.5.1. enabled	650
7.5.2. Enforcing	651
7.5.3. 正	651
7.6. AGENT STRUCT	651
7.6.1. 主机	652
7.7. AGENTCONFIGURATION STRUCT	652
7.7.1. network_mappings	652
7.8. API STRUCT	652
7.8.1. authenticated_user	654
7.8.2. effective_user	654
7.9. APISUMMARY STRUCT	654
7.10. APISUMMARYITEM STRUCT	655
7.11. APPLICATION STRUCT	655
7.12. 架构 枚举	656
7.12.1. s390x	656
7.13. AUTHORIZEDKEY STRUCT	657
7.14. AUTONUMASTATUS ENUM	657
7.15. AUTOPINNINGPOLICY ENUM	657
7.15.1. 调整	658
7.15.2. disabled	658
7.15.3. existing	658
7.16. BACKUP STRUCT	658
7.16.1. to_checkpoint_id	659
7.17. BACKUPPHASE ENUM	659
7.17.1. 初始化	660
7.18. BALANCE STRUCT	660
7.19. BIOS STRUCT	660
7.20. BIOSTYPE ENUM	661
7.20.1. cluster_default	661
7.20.2. i440fx_sea_bios	661
7.21. BLOCKSTATISTICSTRUCT	661
7.22. BONDING STRUCT	662
7.22.1. ad_partner_mac	662
7.22.2. 选项	662
7.22.3. slaves	662
7.22.4. active_slave	663
7.23. BOOKMARK STRUCT	663
7.24. BOOT STRUCT	663
7.24.1. devices	664
7.25. BOOTDEVICE ENUM	664
7.25.1. cdrom	664
7.25.2. network	664
7.26. BOOTMENU STRUCT	664
7.27. BOOTPROTOCOL ENUM	665
7.27.1. autoconf	665
7.27.2. dhcp	665
7.27.3. poly_dhcp_autoconf	666
7.28. BRICKPROFILEDETAIL STRUCT	666
7.29. CDROM STRUCT	666
7.29.1. vms	667

7.30. 证书 结构	667
7.31. CHECKPOINT STRUCT	667
7.32. CHECKPOINTSTATE ENUM	668
7.32.1. Created	668
7.33. CLOUDINIT STRUCT	668
7.34. CLOUDINITNETWORKPROTOCOL ENUM	669
7.34.1. Eni	669
7.34.2. openstack_metadata	670
7.35. CLUSTER STRUCT	670
7.35.1. bios_type	674
7.35.2. custom_scheduling_policy_properties	674
7.35.3. fencing_policy	675
7.35.4. fips_mode	675
7.35.5. gluster_tuned_profile	676
7.35.6. log_max_memory_used_threshold	676
7.35.7. log_max_memory_used_threshold_type	676
7.35.8. maintenance_reason_required	676
7.35.9. optional_reason	676
7.35.10. required_rng_sources	676
7.35.11. upgrade_correlation_id	677
7.35.12. version	677
7.35.13. vnc_encryption	678
7.35.14. external_network_providers	679
7.35.15. scheduling_policy	679
7.36. CLUSTERFEATURE STRUCT	679
7.37. CLUSTERLEVEL STRUCT	680
7.38. CLUSTERUPGRADEACTION ENUM	680
7.38.1. finish	681
7.38.2. start	681
7.38.3. update_progress	681
7.39. CONFIGURATION STRUCT	681
7.39.1. data	681
7.40. CONFIGURATIONTYPE ENUM	683
7.40.1. OVA	683
7.40.2. obf	683
7.41. CONSOLE STRUCT	684
7.42. CORE STRUCT	684
7.43. CPU STRUCT	684
7.44. CPUMODE ENUM	685
7.45. CPUPINNINGPOLICY ENUM	685
7.45.1. 专用	685
7.45.2. Manual	686
7.45.3. none	686
7.45.4. resize_and_pin_numa	686
7.46. CUPROFILE STRUCT	686
7.47. CPUTOPOLOGY STRUCT	687
7.48. CPUTUNE STRUCT	687
7.49. CPUTYPE STRUCT	687
7.50. CREATIONSTATUS ENUM	687
7.51. CUSTOMPROPERTY STRUCT	688
7.52. DATACENTER STRUCT	688
7.52.1. version	689
7.53. DATACENTERSTATUS ENUM	690

7.54. DEVICE STRUCT	690
7.54.1. vms	691
7.55. DISK STRUCT	691
7.55.1. Active	693
7.55.2. actual_size	693
7.55.3. Bootable	693
7.55.4. external_disk	694
7.55.5. initial_size	694
7.55.6. Interface	694
7.55.7. provisioned_size	695
7.55.8. qcow_version	695
7.55.9. read_only	695
7.55.10. sgio	695
7.55.11. 共享	696
7.55.12. total_size	696
7.55.13. wipe_after_delete	696
7.55.14. statistics	697
7.55.15. storage_domains	698
7.55.16. vms	698
7.56. DISKATTACHMENT STRUCT	698
7.56.1. Active	699
7.56.2. logical_name	699
7.56.3. read_only	699
7.56.4. uses_scsi_reservation	700
7.57. DISKBACKUP 枚举	700
7.58. DISKBACKUPMODE ENUM	700
7.58.1. full	701
7.58.2. incremental	701
7.59. DISKCONTENTTYPE ENUM	701
7.60. DISKFORMAT ENUM	702
7.61. DISKINTERFACE ENUM	702
7.61.1. IDE	702
7.61.2. VirtIO	703
7.61.3. virtio_scsi	703
7.62. DISKPROFILE STRUCT	703
7.63. DISKSNAPSHOT STRUCT	703
7.63.1. Active	705
7.63.2. actual_size	705
7.63.3. Bootable	705
7.63.4. external_disk	706
7.63.5. initial_size	706
7.63.6. Interface	707
7.63.7. provisioned_size	707
7.63.8. qcow_version	707
7.63.9. read_only	707
7.63.10. sgio	708
7.63.11. 共享	708
7.63.12. total_size	708
7.63.13. wipe_after_delete	708
7.63.14. statistics	709
7.63.15. storage_domains	710
7.63.16. vms	710
7.64. DISKSTATUS ENUM	710

7.64.1. Locked	710
7.65. DISKSTORAGETYPE ENUM	711
7.66. DISKTYPE ENUM	711
7.67. 显示 结构	711
7.67.1. allow_override	712
7.67.2. certificate	712
7.67.3. copy_paste_enabled	712
7.67.4. disconnect_action	713
7.67.5. disconnect_action_delay	713
7.67.6. file_transfer_enabled	713
7.67.7. keyboard_layout	713
7.67.8. monitor	713
7.67.9. proxy	714
7.67.10. secure_port	714
7.67.11. single_qxl_pci	714
7.67.12. smartcard_enabled	714
7.68. DISPLAYTYPE ENUM	714
7.68.1. spice	714
7.68.2. vnc	715
7.69. DNS STRUCT	715
7.70. DNSRESOLVERCONFIGURATION STRUCT	715
7.70.1. name_servers	715
7.71. DOMAIN STRUCT	715
7.71.1. 用户	716
7.72. DYNAMICCPU STRUCT	716
7.73. ENTITYEXTERNALSTATUS ENUM	716
7.73.1. 错误	717
7.73.2. 失败	717
7.74. ENTITYPROFILEDETAIL STRUCT	717
7.75. ERRORHANDLING STRUCT	717
7.76. EVENT STRUCT	717
7.76.1. correlation_id	718
7.76.2. flood_rate	718
7.76.3. index	718
7.76.4. log_on_host	719
7.76.5. cluster	719
7.76.6. data_center	719
7.76.7. 主机	719
7.76.8. storage_domain	720
7.76.9. 模板	720
7.76.10. user	720
7.76.11. vm	720
7.77. EVENTSUBSCRIPTION STRUCT	720
7.77.1. 地址	720
7.77.2. event	721
7.77.3. notification_method	721
7.77.4. user	721
7.78. EXTERNALCOMPUTERESOURCE STRUCT	721
7.79. EXTERNALDISCOVEREDHOST STRUCT	722
7.80. EXTERNALHOST STRUCT	723
7.81. EXTERNALHOSTGROUP STRUCT	723
7.82. EXTERNALHOSTPROVIDER STRUCT	724
7.82.1. required_authentication	725

7.82.2. compute_resources	725
7.82.3. discovered_hosts	726
7.82.4. host_groups	726
7.83. EXTERNALNETWORKPROVIDERCONFIGURATION STRUCT	726
7.84. EXTERNALPROVIDER STRUCT	726
7.84.1. required_authentication	727
7.85. EXTERNALSTATUS ENUM	727
7.85.1. 错误	728
7.85.2. 失败	728
7.85.3. info	728
7.85.4. 确定	728
7.85.5. warning	728
7.86. EXTERNALSYSTEMTYPE ENUM	728
7.87. EXTERNALTEMPLATEIMPORT STRUCT	729
7.87.1. clone	729
7.87.2. url	729
7.87.3. cpu_profile	730
7.87.4. quota	730
7.87.5. 模板	730
7.88. EXTERNALVMIMPORT STRUCT	730
7.88.1. sparse	731
7.88.2. url	731
7.88.3. cpu_profile	732
7.88.4. drivers_iso	732
7.88.5. 主机	732
7.88.6. quota	732
7.88.7. vm	732
7.89. EXTERNALVMPROVIDERTYPE ENUM	732
7.90. FAULT STRUCT	733
7.91. FENCETYPE ENUM	733
7.92. FENCINGPOLICY STRUCT	733
7.92.1. skip_if_connectivity_broken	734
7.92.2. skip_if_gluster_bricks_up	734
7.92.3. skip_if_gluster_quorum_not_met	734
7.92.4. skip_if_sd_active	734
7.93. FILE STRUCT	734
7.94. FILTER STRUCT	735
7.95. FIPSMODE ENUM	735
7.95.1. disabled	736
7.95.2. enabled	736
7.95.3. 未定义	736
7.96. FIREWALLTYPE ENUM	736
7.96.1. firewalld	737
7.96.2. iptables	737
7.97. FLOPPY STRUCT	737
7.97.1. vms	738
7.98. FOPSTATISTICSTRUCT	738
7.99. GLUSTERBRICK STRUCT	738
7.99.1. vms	739
7.100. GLUSTERBRICKADVANCEDDETAILS STRUCT	739
7.100.1. vms	740
7.101. GLUSTERBRICKMEMORYINFO STRUCT	740
7.102. GLUSTERBRICKSTATUS ENUM	741

7.103. GLUSTERCLIENT STRUCT	741
7.104. GLUSTERHOOK STRUCT	741
7.105. GLUSTERHOOKSTATUS ENUM	742
7.106. GLUSTERMEMORYPOOL STRUCT	742
7.107. GLUSTERSERVERHOOK STRUCT	743
7.108. GLUSTERSTATE ENUM	744
7.109. GLUSTERVOLUME STRUCT	744
7.110. GLUSTERVOLUMEPROFILEDETAILS STRUCT	745
7.111. GLUSTERVOLUMESTATUS ENUM	745
7.112. GLUSTERVOLUMETYPE ENUM	746
7.112.1. disperse	746
7.112.2. distribution	746
7.112.3. distributed_disperse	747
7.112.4. distributed_replicate	747
7.112.5. distributed_stripe	747
7.112.6. distributed_stripped_replicate	747
7.112.7. 复制	748
7.112.8. stripe	748
7.112.9. striped_replicate	748
7.113. GRACEPERIOD STRUCT	748
7.114. GRAPHICSCONSOLE STRUCT	748
7.115. GRAPHICSTYPE ENUM	749
7.115.1. spice	749
7.115.2. vnc	750
7.116. GROUP STRUCT	750
7.116.1. 角色	750
7.117. GUESTOPERATINGSYSTEM STRUCT	751
7.118. HARDWAREINFORMATION STRUCT	752
7.119. HIGHAVAILABILITY STRUCT	753
7.119.1. enabled	753
7.119.2. priority	753
7.120. HOOK STRUCT	754
7.121. HOOKCONTENTTYPE ENUM	755
7.122. HOOKSTAGE ENUM	755
7.123. HOOKSTATUS ENUM	755
7.124. HOST STRUCT	756
7.124.1. external_status	758
7.124.2. hosted_engine	758
7.124.3. kdump_status	758
7.124.4. ksm	758
7.124.5. libvirt_version	759
7.124.6. network_operation_in_progress	759
7.124.7. override_iptables	759
7.124.8. 协议	759
7.124.9. se_linux	760
7.124.10. spm	760
7.124.11. status_detail	760
7.124.12. transparent_huge_pages	760
7.124.13. version	761
7.124.14. cpu_units	762
7.124.15. external_network_provider_configurations	763
7.124.16. katello_errata	763
7.124.17. statistics	763

7.125. HOSTCPUUNIT STRUCT	765
7.126. HOSTDEVICE STRUCT	766
7.126.1. 驱动程序	766
7.127. HOSTDEVICEPASSTHROUGH STRUCT	767
7.128. HOSTNIC STRUCT	767
7.128.1. ad_aggregator_id	769
7.128.2. bridged	770
7.128.3. network	770
7.128.4. statistics	770
7.129. HOSTNICVIRTUALFUNCTIONSCONFIGURATION STRUCT	771
7.129.1. max_number_of_virtual_functions	771
7.129.2. number_of_virtual_functions	772
7.130. HOSTPROTOCOL 枚举	772
7.131. HOSTSTATUS ENUM	772
7.131.1. 错误	773
7.131.2. 初始化	773
7.131.3. install_failed	773
7.131.4. installing_os	773
7.131.5. 维护	774
7.131.6. non_operational	774
7.131.7. non_responsive	774
7.131.8. pending_approval	774
7.131.9. preparing_for_maintenance	774
7.132. HOSTSTORAGE STRUCT	774
7.132.1. driver_options	775
7.132.2. driver_sensitive_options	776
7.132.3. nfs_retrans	777
7.132.4. nfs_timeo	777
7.133. HOSTTYPE ENUM	778
7.133.1. ovirt_node	778
7.133.2. rhev_h	778
7.134. HOSTEDENGINE STRUCT	778
7.135. ICON STRUCT	779
7.135.1. media_type	779
7.136. IDENTIFIED STRUCT	780
7.137. 镜像 结构	780
7.138. IMAGEFILETYPE ENUM	781
7.138.1. iso	781
7.139. IMAGETRANSFER STRUCT	781
7.139.1. 方向	782
7.139.2. 格式	782
7.139.3. inactivity_timeout	782
7.139.4. phase	782
7.139.5. proxy_url	783
7.139.6. Kow	783
7.139.7. transfer_url	784
7.139.8. backup	784
7.139.9. 主机	784
7.139.10. image	784
7.140. IMAGETRANSFERDIRECTION ENUM	785
7.141. IMAGETRANSFERPHASE ENUM	785
7.141.1. cancelled	786
7.141.2. finalizing_success	786

7.141.3. finished_failure	787
7.141.4. finished_success	787
7.141.5. 初始化	787
7.141.6. paused_system	787
7.141.7. 恢复	787
7.141.8. Unknown	787
7.142. IMAGETRANSFERTIMEOUTPOLICY ENUM	787
7.142.1. 取消	788
7.142.2. legacy	788
7.142.3. pause	788
7.143. INHERITABLEBOOLEAN ENUM	788
7.144. INITIALIZATION STRUCT	789
7.144.1. cloud_init	790
7.144.2. cloud_init_network_protocol	790
7.145. INSTANCETYPE STRUCT	791
7.145.1. auto_pinning_policy	795
7.145.2. cpu	795
7.145.3. cpu_pinning_policy	796
7.145.4. custom_compatibility_version	796
7.145.5. high_availability	796
7.145.6. 初始化	796
7.145.7. large_icon	797
7.145.8. lease	797
7.145.9. 内存	797
7.145.10. migration_downtime	798
7.145.11. Origin	798
7.145.12. placement_policy	799
7.145.13. small_icon	800
7.145.14. sso	800
7.145.15. tpm_enabled	800
7.146. IO STRUCT	801
7.147. IP STRUCT	801
7.147.1. 地址	802
7.147.2. 子网掩码	802
7.147.3. version	802
7.148. IPADDRESSASSIGNMENT STRUCT	803
7.149. IPVERSION ENUM	803
7.150. ISCSIBOND STRUCT	803
7.151. ISCSIDetails STRUCT	804
7.152. JOB STRUCT	805
7.152.1. external	805
7.153. JOBSTATUS ENUM	806
7.153.1. 中止	806
7.153.2. 完成	806
7.153.3. Started	806
7.153.4. Unknown	806
7.154. KATELLOERRATUM STRUCT	807
7.154.1. 严重性	807
7.154.2. type	807
7.155. KDUMPSTATUS ENUM	808
7.156. 内核结构	808
7.157. KSM STRUCT	808
7.158. LINKLAYERDISCOVERYPROTOCOLELEMENT STRUCT	809

7.158.1. OUI	810
7.158.2. 子类型	810
7.159. LOGMAXMEMORYUSEDTHRESHOLDTYPE ENUM	810
7.159.1. absolute_value_in_mb	810
7.159.2. 百分比	810
7.160. LOGSEVERITY ENUM	811
7.160.1. alert	811
7.160.2. 错误	811
7.160.3. Normal	811
7.160.4. warning	811
7.161. LOGICALUNIT STRUCT	811
7.161.1. discard_max_size	812
7.161.2. discard_zeroes_data	813
7.162. LUNSTATUS ENUM	813
7.163. MDEVTYPE STRUCT	813
7.164. MAC STRUCT	814
7.165. MACPOOL STRUCT	814
7.165.1. allow_duplicates	815
7.165.2. default_pool	815
7.165.3. 范围	815
7.166. MEMORYOVERCOMMIT STRUCT	815
7.167. MEMORYPOLICY STRUCT	816
7.167.1. Guaranteed	816
7.167.2. Max	816
7.168. MESSAGEBROKERTYPE ENUM	817
7.169. METHOD STRUCT	817
7.170. MIGRATEONERROR ENUM	817
7.171. MIGRATIONBANDWIDTH STRUCT	817
7.171.1. custom_value	818
7.172. MIGRATIONBANDWIDTHASSIGNMENTMETHOD ENUM	818
7.172.1. auto	818
7.173. MIGRATIONOPTIONS STRUCT	818
7.174. MIGRATIONPOLICY STRUCT	819
7.175. NETWORK STRUCT	819
7.175.1. dns_resolver_configuration	821
7.175.2. port_isolation	821
7.175.3. 必需	821
7.175.4. status	822
7.175.5. 使用方法	822
7.175.6. vdsm_name	822
7.175.7. cluster	822
7.175.8. external_provider	823
7.175.9. external_provider_physical_network	823
7.176. NETWORKATTACHMENT STRUCT	823
7.176.1. dns_resolver_configuration	826
7.176.2. 属性	826
7.177. NETWORKCONFIGURATION STRUCT	827
7.178. NETWORKFILTER STRUCT	828
7.178.1. version	829
7.179. NETWORKFILTERPARAMETER STRUCT	829
7.180. NETWORKLABELSTRUCT	830
7.181. NETWORKPLUGINTYPE ENUM	830
7.181.1. open_vswitch	830

7.182. NETWORKSTATUS ENUM	831
7.183. NETWORKUSAGE 枚举	831
7.183.1. default_route	831
7.183.2. 管理	832
7.184. NFSPROFILEDETAIL STRUCT	832
7.185. NFSVERSION ENUM	832
7.185.1. v4_0	832
7.185.2. v4_2	833
7.186. NIC STRUCT	833
7.186.1. network	834
7.186.2. vms	835
7.187. NICCONFIGURATION STRUCT	835
7.188. NICINTERFACE ENUM	835
7.189. NICSTATUS ENUM	836
7.190. NOTABLEEVENT ENUM	836
7.190.1. gluster_volume_rebalance_not_found_from_cli	848
7.190.2. host_untrusted	848
7.190.3. remove_gluster_volume_bricks_not_found_from_cli	848
7.191. NOTIFICATIONMETHOD ENUM	848
7.191.1. smtp	849
7.191.2. SNMP	849
7.192. NUMANO DESTRUCT	849
7.192.1. statistics	850
7.193. NUMANODEPIN STRUCT	851
7.193.1. host_numa_node	851
7.193.2. pinned	852
7.194. NUMATUNEMODE ENUM	852
7.195. OPENSTACKIMAGE STRUCT	852
7.196. OPENSTACKIMAGEPROVIDER STRUCT	852
7.196.1. required_authentication	853
7.196.2. tenant_name	853
7.197. OPENSTACKNETWORK STRUCT	854
7.198. OPENSTACKNETWORKPROVIDER STRUCT	854
7.198.1. agent_configuration	855
7.198.2. auto_sync	855
7.198.3. external_plugin_type	856
7.198.4. plugin_type	856
7.198.5. read_only	857
7.198.6. required_authentication	857
7.198.7. tenant_name	857
7.198.8. 非受管	857
7.199. OPENSTACKNETWORKPROVIDERTYPE ENUM	858
7.199.1. external	858
7.199.2. Neutron	858
7.200. OPENSTACKPROVIDER STRUCT	858
7.200.1. required_authentication	859
7.200.2. tenant_name	859
7.201. OPENSTACKSUBNET STRUCT	859
7.201.1. ip_version	860
7.202. OPENSTACKVOLUMEPROVIDER STRUCT	860
7.202.1. required_authentication	861
7.202.2. tenant_name	861
7.203. OPENSTACKVOLUMETYPE STRUCT	862

7.204. OPENSTACKVOLUMEAUTHENTICATIONKEY STRUCT	862
7.205. OPENSTACKVOLUMEAUTHENTICATIONKEYUSAGETYPE ENUM	863
7.206. OPERATINGSYSTEM STRUCT	863
7.206.1. 引导	864
7.206.2. cmdline	864
7.206.3. custom_kernel_cmdline	864
7.206.4. initrd	865
7.206.5. kernel	865
7.206.6. reported_kernel_cmdline	865
7.206.7. type	866
7.207. OPERATINGSYSTEMINFO STRUCT	866
7.207.1. large_icon	866
7.207.2. small_icon	867
7.208. OPTION STRUCT	867
7.209. OSTYPE ENUM	867
7.210. 软件包 结构	868
7.211. PAYLOAD STRUCT	869
7.212. PAYLOADENCODING 枚举	869
7.213. PERMISSION STRUCT	869
7.214. PERMIT STRUCT	870
7.215. PMPROXY STRUCT	871
7.216. PMPROXYTYPE ENUM	871
7.217. POLICYUNITTYPE ENUM	871
7.218. PORTMIRRORING STRUCT	872
7.219. POWERMANAGEMENT STRUCT	872
7.219.1. 代理	872
7.219.2. automatic_pm_enabled	873
7.219.3. kdump_detection	873
7.219.4. type	873
7.220. POWERMANAGEMENTSTATUS ENUM	873
7.221. 产品 结构	873
7.222. PRODUCTINFO STRUCT	874
7.222.1. vendor	875
7.223. PROFILEDETAIL STRUCT	875
7.224. 属性 结构	875
7.225. PROXYTICKET STRUCT	875
7.226. QCOWVERSION ENUM	876
7.226.1. qcow2_v3	876
7.227. QOS STRUCT	876
7.227.1. cpu_limit	878
7.227.2. inbound_average	878
7.227.3. inbound_burst	878
7.227.4. inbound_peak	878
7.227.5. max_iops	879
7.227.6. max_read_iops	879
7.227.7. max_read_throughput	879
7.227.8. max_throughput	879
7.227.9. max_write_iops	879
7.227.10. max_write_throughput	880
7.227.11. outbound_average	880
7.227.12. outbound_average_linkshare	880
7.227.13. outbound_average_realtime	880
7.227.14. outbound_average_upperlimit	880

7.227.15. outbound_burst	881
7.227.16. outbound_peak	881
7.228. QOSTYPE ENUM	881
7.229. QUOTA STRUCT	882
7.230. QUOTA_CLUSTERLIMIT STRUCT	883
7.231. QUOTA_MODETYPE ENUM	884
7.232. QUOTA_STORAGELIMIT STRUCT	884
7.233. RANGE STRUCT	885
7.234. RATE STRUCT	885
7.235. REGISTRATION_AFFINITY_GROUP_MAPPING STRUCT	885
7.235.1. 来自	886
7.236. REGISTRATION_AFFINITY_LABEL_MAPPING STRUCT	886
7.236.1. 来自	887
7.237. REGISTRATION_CLUSTER_MAPPING STRUCT	887
7.237.1. 来自	887
7.237.2. 为	888
7.238. REGISTRATION_CONFIGURATION STRUCT	888
7.239. REGISTRATION_DOMAIN_MAPPING STRUCT	891
7.239.1. 来自	892
7.240. REGISTRATION_LUN_MAPPING STRUCT	892
7.240.1. 来自	893
7.241. REGISTRATION_ROLE_MAPPING STRUCT	893
7.241.1. 来自	893
7.242. REGISTRATION_VNIC_PROFILE_MAPPING STRUCT	894
7.242.1. 来自	895
7.242.2. 为	895
7.243. REPORTED_CONFIGURATION STRUCT	896
7.244. REPORTED_DEVICE STRUCT	896
7.245. REPORTED_DEVICE_TYPE ENUM	896
7.246. RESOLUTION_TYPE 枚举	897
7.247. RNG_DEVICE STRUCT	897
7.248. RNG_SOURCE ENUM	897
7.248.1. urandom	897
7.249. ROLE STRUCT	898
7.249.1. 可变	898
7.250. ROLE_TYPE ENUM	898
7.251. SCHEDULING_POLICY STRUCT	899
7.252. SCHEDULING_POLICY_UNIT STRUCT	899
7.253. SCSI_GENERIC_IO ENUM	900
7.254. SELINUX STRUCT	900
7.255. SELINUX_MODE ENUM	900
7.256. SERIAL_NUMBER STRUCT	901
7.257. SERIAL_NUMBER_POLICY ENUM	901
7.257.1. 主机	901
7.258. SESSION STRUCT	902
7.258.1. console_user	902
7.258.2. ip	902
7.258.3. 协议	902
7.258.4. user	903
7.259. SKIP_IF_CONNECTIVITY_BROKEN STRUCT	903
7.259.1. enabled	903
7.259.2. threshold	903
7.260. SKIP_IF_SD_ACTIVE STRUCT	904

7.260.1. enabled	904
7.261. SNAPSHOT STRUCT	904
7.261.1. auto_pinning_policy	908
7.261.2. cpu	908
7.261.3. cpu_pinning_policy	909
7.261.4. custom_compatibility_version	909
7.261.5. high_availability	909
7.261.6. 初始化	909
7.261.7. large_icon	910
7.261.8. lease	910
7.261.9. 内存	910
7.261.10. migration_downtime	911
7.261.11. next_run_configuration_exists	911
7.261.12. numa_tune_mode	911
7.261.13. Origin	912
7.261.14. persist_memorystate	912
7.261.15. placement_policy	913
7.261.16. small_icon	913
7.261.17. sso	914
7.261.18. stop_reason	914
7.261.19. tpm_enabled	914
7.261.20. affinity_labels	916
7.261.21. katello_errata	916
7.261.22. original_template	916
7.261.23. statistics	917
7.262. SNAPSHOTSTATUS ENUM	917
7.262.1. Locked	917
7.263. SNAPSHOTTYPE ENUM	917
7.263.1. 预览	918
7.263.2. 无状态	918
7.264. SPECIALOBJECTS STRUCT	918
7.265. SPM STRUCT	918
7.266. SPMSTATUS ENUM	919
7.267. SSH STRUCT	919
7.267.1. 指纹	919
7.267.2. public_key	920
7.268. SSHAUTHENTICATIONMETHOD ENUM	920
7.269. SSHPUBLICKEY STRUCT	920
7.270. SSO STRUCT	920
7.271. SSOMETHOD ENUM	921
7.272. STATISTICSTRUCT	921
7.273. STATISTICKIND ENUM	923
7.274. STATISTICUNIT ENUM	923
7.275. STEP STRUCT	924
7.275.1. external	925
7.276. STEPENUM ENUM	925
7.276.1. 执行	926
7.276.2. 挪威	926
7.276.3. rebalancing_volume	926
7.276.4. removing_bricks	926
7.276.5. Unknown	926
7.276.6. 验证	926
7.277. STEPSTATUS ENUM	926

7.277.1. 中止	927
7.277.2. 完成	927
7.277.3. Started	927
7.277.4. Unknown	927
7.278. STORAGECONNECTION STRUCT	927
7.279. STORAGECONNECTIONEXTENSION STRUCT	928
7.280. STORAGEDOMAIN STRUCT	929
7.280.1. backup	931
7.280.2. block_size	931
7.280.3. discard_after_delete	931
7.280.4. supports_discard	932
7.280.5. supports_discard_zeroes_data	932
7.280.6. wipe_after_delete	932
7.280.7. data_center	933
7.281. STORAGEDOMAINLEASE STRUCT	933
7.282. STORAGEDOMAINSTATUS ENUM	933
7.283. STORAGEDOMAINTYPE ENUM	934
7.283.1. data	935
7.283.2. export	935
7.283.3. image	935
7.283.4. iso	935
7.283.5. managed_block_storage	935
7.283.6. 卷	935
7.284. STORAGEFORMAT ENUM	935
7.284.1. v1	936
7.284.2. v2	936
7.284.3. v3	936
7.284.4. v5	936
7.285. STORAGETYPE ENUM	937
7.285.1. cinder	937
7.285.2. Glance	937
7.285.3. glusterfs	937
7.285.4. managed_block_storage	938
7.286. SWITCHTYPE ENUM	938
7.287. SYSTEMOPTIONSTRUCT	938
7.288. SYSTEMOPTIONVALUE STRUCT	938
7.289. TAG STRUCT	939
7.290. TEMPLATE STRUCT	939
7.290.1. auto_pinning_policy	942
7.290.2. cpu	942
7.290.3. cpu_pinning_policy	943
7.290.4. custom_compatibility_version	943
7.290.5. high_availability	943
7.290.6. 初始化	943
7.290.7. large_icon	944
7.290.8. lease	944
7.290.9. 内存	944
7.290.10. migration_downtime	945
7.290.11. Origin	945
7.290.12. placement_policy	946
7.290.13. small_icon	947
7.290.14. sso	947
7.290.15. tpm_enabled	947

7.291. TEMPLATESTATUS ENUM	948
7.292. TEMPLATEVERSION STRUCT	949
7.292.1. version_number	949
7.293. TICKETSTRUCT	949
7.294. TIMEZONE STRUCT	949
7.294.1. utc_offset	950
7.295. TRANSPARENTHUGEPAGES STRUCT	950
7.296. TRANSPORTTYPE ENUM	950
7.297. UNMANAGEDNETWORK STRUCT	950
7.298. USB STRUCT	951
7.299. USBTYPE ENUM	951
7.299.1. legacy	952
7.299.2. native	952
7.300. 用户 结构	952
7.300.1. namespace	953
7.300.2. principal	953
7.300.3. user_name	953
7.300.4. user_options	953
7.301. USEROPTIONSTRUCT	954
7.301.1. 内容	954
7.302. VALUE STRUCT	955
7.303. VALUETYPE ENUM	955
7.304. VCPUPIN STRUCT	955
7.305. VENDOR STRUCT	955
7.306. 版本 结构	956
7.307. VGPUPLACEMENT ENUM	956
7.307.1. 合并	956
7.307.2. separated	957
7.308. VIRTIOSCSI STRUCT	957
7.309. VIRTUALNUMANODE STRUCT	957
7.309.1. statistics	958
7.310. VLAN STRUCT	959
7.311. VM STRUCT	960
7.311.1. auto_pinning_policy	963
7.311.2. cpu	963
7.311.3. cpu_pinning_policy	964
7.311.4. custom_compatibility_version	964
7.311.5. high_availability	964
7.311.6. 初始化	964
7.311.7. large_icon	965
7.311.8. lease	965
7.311.9. 内存	965
7.311.10. migration_downtime	966
7.311.11. next_run_configuration_exists	966
7.311.12. numa_tune_mode	966
7.311.13. Origin	967
7.311.14. placement_policy	967
7.311.15. small_icon	968
7.311.16. sso	968
7.311.17. stop_reason	968
7.311.18. tpm_enabled	969
7.311.19. affinity_labels	970
7.311.20. katello_errata	970

7.311.21. original_template	971
7.311.22. statistics	971
7.312. VMAFFINITY ENUM	971
7.313. VMBASE STRUCT	972
7.313.1. auto_pinning_policy	974
7.313.2. cpu	975
7.313.3. cpu_pinning_policy	975
7.313.4. custom_compatibility_version	975
7.313.5. high_availability	976
7.313.6. 初始化	976
7.313.7. large_icon	976
7.313.8. lease	976
7.313.9. 内存	977
7.313.10. migration_downtime	977
7.313.11. Origin	978
7.313.12. placement_policy	979
7.313.13. small_icon	979
7.313.14. sso	980
7.313.15. tpm_enabled	980
7.314. VMDEVICETYPE ENUM	980
7.315. VMMEDATEDDEVICE STRUCT	980
7.315.1. vms	981
7.316. VMPLACEMENTPOLICY STRUCT	981
7.317. VMPOOL STRUCT	981
7.317.1. auto_storage_select	982
7.317.2. prestarted_vms	983
7.317.3. 有状态	983
7.317.4. tpm_enabled	983
7.317.5. instance_type	983
7.317.6. vm	984
7.318. VMPOOLTYPE ENUM	984
7.318.1. 自动	984
7.318.2. Manual	984
7.319. VMSTATUS ENUM	984
7.319.1. paused	985
7.319.2. powering_up	985
7.319.3. restoring_state	986
7.319.4. saving_state	986
7.319.5. 暂停	986
7.319.6. Unknown	986
7.319.7. up	986
7.319.8. wait_for_launch	986
7.320. VMSTORAGEERRORRESUMEBEHAVIOUR ENUM	986
7.320.1. auto_resume	987
7.320.2. kill	987
7.320.3. leave_paused	987
7.321. VMSUMMARY STRUCT	987
7.322. VMATYPE ENUM	988
7.322.1. desktop	988
7.322.2. high_performance	988
7.322.3. server	990
7.323. VNICPASSTHROUGH STRUCT	990
7.324. VNICPASSTHROUGHMODE ENUM	990

7.325. VNICPROFILE STRUCT	991
7.325.1. migratable	991
7.325.2. pass_through	991
7.325.3. port_mirroring	992
7.325.4. network_filter	993
7.325.5. qos	993
7.326. VNICPROFILEMAPPING STRUCT	993
7.326.1. source_network_name	994
7.326.2. source_network_profile_name	994
7.326.3. target_vnic_profile	995
7.327. VOLUMEGROUP STRUCT	995
7.328. WATCHDOG STRUCT	995
7.328.1. model	996
7.328.2. vms	996
7.329. WATCHDOGACTION ENUM	996
7.329.1. none	997
7.330. WATCHDOGMODEL ENUM	997
7.330.1. diag288	997
7.330.2. i6300esb	997
7.331. WEIGHT STRUCT	998
附录 A. 原语类型	999
A.1. 字符串 原语	999
A.2. 布尔值 原语	999
A.3. 整数 原语	999
A.4. 十进制 原语	1001
A.5. 日期 原语	1001
附录 B. API 版本 4 的更改	1003
B.1. API 版本 4.0 的更改 (显示版本 3.6)	1003
B.1.1. 删除了 YAML 支持	1003
B.1.2. 重新命名复杂的类型	1003
B.1.3. 使用 enum 类型替换 Status 类型	1005
B.1.4. 删除 NIC 网络和 port_mirroring 属性	1005
B.1.5. 删除 NIC active 属性	1006
B.1.6. 删除磁盘 类型 属性	1006
B.1.7. 删除 磁盘大小 属性	1006
B.1.8. 删除了对将虚拟机固定到单一主机的支持	1006
B.1.9. 删除了 capabilities.permits 元素	1007
B.1.10. 删除了 storage_manager 元素	1007
B.1.11. 删除了数据中心 storage_type 元素	1007
B.1.12. 删除 timezone 元素	1008
B.1.13. 删除了 guest_info 元素	1008
B.1.14. 使用 type 元素替换 CPU id 属性	1009
B.1.15. 在 CPU 拓扑中使用元素而不是属性	1010
B.1.16. 在 VCPU pin 中使用元素而不是属性	1010
B.1.17. 在 VCPU pin 中使用元素而不是属性	1010
B.1.18. 在内存过量使用中元素而不是属性	1011
B.1.19. 在控制台中使用元素而不是属性	1011
B.1.20. 在 VIRTIO SCSI 中使用元素而不是属性	1012
B.1.21. 对于电源管理代理 类型, 使用 element 而不是 属性	1012
B.1.22. 在电源管理代理选项中使用元素而不是属性	1012
B.1.23. 在 IP 地址中使用元素而不是属性 :	1013

B.1.24. 在 MAC 地址中使用元素而不是属性 :	1013
B.1.25. 在引导设备中使用元素而不是属性 :	1013
B.1.26. 对于操作系统 类型, 使用 element 而不是 属性	1014
B.1.27. 从请求中删除了 force 参数, 以检索主机	1014
B.1.28. 删除了已弃用的主机电源管理配置	1015
B.1.29. 使用多个 boot.devices.device 而不是多次 引导	1015
B.1.30. 删除了 disk.clone 和 disk.detach_only 元素	1016
B.1.31. 将元素 vmpool 重命名为 vm_pool	1017
B.1.32. 使用 logical_units 而不是多个 logical_units	1017
B.1.33. 删除了 snapshot.collapse_snapshots 元素	1018
B.1.34. 重命名 存储和 host_storage 元素	1018
B.1.35. 删除了 permissions.clone 元素	1019
B.1.36. 重新命名随机数生成器 源 元素	1020
B.1.37. 删除了中间 标签.parent 元素	1021
B.1.38. 删除调度内置名称和阈值	1022
B.1.39. 删除了 bricks.replica_count 和 bricks.stripe_count 元素	1023
B.1.40. 将统计 类型 属性重命名为 kind	1023
B.1.41. 使用多个 vcpu_pins.vcpu_pin, 而不是多个 vcpu_pin	1023
B.1.42. 使用 force 参数强制删除数据中心	1024
B.1.43. 使用 force 参数强制删除主机	1024
B.1.44. 使用参数强制删除存储域	1025
B.1.45. 使用 主机 参数删除存储服务器连接	1025
B.1.46. 使用 force 和 storage_domain 参数来删除模板磁盘	1026
B.1.47. 不要通过 VM 磁盘 API 删除磁盘	1026
B.1.48. 使用 强制 查询参数强制删除虚拟机	1026
B.1.49. 使用 POST 而不是 DELETE 来删除多个 brick	1027
B.1.50. 删除 scheduling_policy.policy 元素	1027
B.1.51. 添加了 snapshot.snapshot_type	1028
B.1.52. 从 虚拟机中删除了 移动 操作	1028
B.1.53. 将 report_configurations.in_sync 移到 network_attachment	1028
B.1.54. 使用集群级别替换 的功能	1029
B.1.55. 使用 磁盘 附加替换的磁盘	1030
B.1.56. 使用 iscsi_targets 元素发现未注册的存储	1031
B.2. ENGINE 版本 4.5 中的更改	1032
B.2.1. 由 Managed Block Storage 取代的 OpenStack 卷(Cinder)集成。	1032
附录 C. 法律通知	1033

第 1 章 简介

Red Hat Virtualization Manager 提供**表述性状态转移(REST)API**。API 为软件开发人员和系统管理员提供控制标准 Web 界面之外的红帽虚拟化环境。API 有助于开发人员和管理员将红帽虚拟化环境的功能与通过标准 Hypertext 传输协议(HTTP)访问 API 的自定义脚本或外部应用程序集成。

API 的优点是：

- 广泛的客户端支持 - 任何支持 HTTP 协议的编程语言、框架或系统都可以使用 API。
- 自描述性 - 客户端应用程序需要最少了解虚拟化基础架构，因为运行时会发现许多细节。
- 基于资源的模式 - 基于资源的 REST 模式提供了管理虚拟化平台的自然方法。

这使开发人员和管理员能够：

- 与企业 IT 系统集成。
- 与第三方虚拟化软件集成。
- 执行自动维护或错误检查任务。
- 使用脚本在红帽虚拟化环境中自动执行重复任务。

本文档可用作 Red Hat Virtualization API 的参考。它旨在为开发人员和管理员提供相关说明和示例，以帮助通过 API（直接或使用提供的 SDK）利用其红帽虚拟化环境的功能。

1.1. 表述状态传输

表述性状态转移(REST) 是一种设计架构，侧重于特定服务及其表示法的资源。资源表示是对应于服务器上一个特定受管元素的重要信息抽象。客户端向位于统一资源标识符(URI)的服务器元素发送请求，并使用标准 HTTP 方法（如 **GET**、**POST**、**PUT** 和 **DELETE**）执行操作。这在客户端和服务器之间提供无状态通信，每个请求都独立于任何其他请求，并且包含完成请求所需的所有信息。

1.2. API 先决条件

使用 Red Hat Virtualization API 的先决条件：

- Red Hat Virtualization Manager 联网安装，其中包括 API。
- 启动和接收来自 API 服务器的 HTTP 请求的客户端或编程库。例如：
 - [oVirt Python SDK](#)。
 - [oVirt Java SDK](#)。
 - [cURL 命令行工具](#)。
 - [RESTClient](#)，RESTful Web 服务的调试器。
- Hypertext 传输协议(HTTP)知识，这是用于 REST API 交互的协议。Internet 工程任务组提供注释请求(RFC)，解释 Hypertext 传输协议(<http://www.ietf.org/rfc/rfc2616.txt>)。
- 具备可扩展标记语言(XML)或 JavaScript 对象表示法(JSON)的知识，API 用于构建资源表示。W3C 提供了 XML 的完整规范，地址为 <http://www.w3.org/TR/xml>。ECMA 国际提供 JSON 的免费出版物，地址为 <http://www.ecma-international.org>。

第 2 章 身份验证和安全

2.1. TLS/SSL 认证

红帽虚拟化 API 需要 Hypertext 传输协议安全(HTTPS)^[1]用于与客户端软件（如 SDK 和 CLI 组件）进行安全交互。这包括获取服务器使用的 **CA 证书**，并将其导入到客户端的证书存储中。

2.1.1. 获取 CA 证书

您可以从 Red Hat Virtualization Manager 获取 CA 证书，并使用以下方法之一将其传送到客户端机器：

方法 1

获取 CA 证书的首选方法是使用 **openssl s_client** 命令行工具与服务器执行真正的 TLS 握手，然后提取它所提供的证书。运行类似如下的命令：

```
$ openssl s_client \
-connect myengine.example.com:443 \
-showcerts \
< /dev/null
```

这个命令将连接到服务器并显示类似如下的输出：

```
CONNECTED(00000003)
depth=1 C = US, O = Example Inc., CN = myengine.example.com.23416
verify error:num=19:self signed certificate in certificate chain
---
Certificate chain
 0 s:/C=US/O=Example Inc./CN=myengine.example.com
  i:/C=US/O=Example Inc./CN=myengine.example.com.23416
-----BEGIN CERTIFICATE-----
MIIEaTCCA1GgAwIBAgICEAQwDQYJKoZIhvcNAQEFBQAwSTELMAkGA1UEBhMCMVVMx
FTATBgNVBAoTDEV4YW1wbGUgSW5jLjEjMCEGA1UEAxMaZW5naW5lINDEuZXhhbXBs
SVlJe7e5FTEtHJGTAEWWM6dGbsFhip5VXM0gfqg=
-----END CERTIFICATE-----
 1 s:/C=US/O=Example Inc./CN=myengine.example.com.23416
  i:/C=US/O=Example Inc./CN=myengine.example.com.23416
-----BEGIN CERTIFICATE-----
MIIDxjCCAq6gAwIBAgICEAAwDQYJKoZIhvcNAQEFBQAwSTELMAkGA1UEBhMCMVVMx
FTATBgNVBAoTDEV4YW1wbGUgSW5jLjEjMCEGA1UEAxMaZW5naW5lINDEuZXhhbXBs
Pkyg1rQHR6ebGQ==
-----END CERTIFICATE-----
```

-----BEGIN CERTIFICATE----- 和 -----END CERTIFICATE----- 之间的文本显示服务器提供的证书。第一个是服务器本身的证书，最后一个是 CA 的证书。将 CA 证书（包括标记）复制到 **ca.crt** 文件中。结果应如下所示：

```
-----BEGIN CERTIFICATE-----
MIIDxjCCAq6gAwIBAgICEAAwDQYJKoZIhvcNAQEFBQAwSTELMAkGA1UEBhMCMVVMx
FTATBgNVBAoTDEV4YW1wbGUgSW5jLjEjMCEGA1UEAxMaZW5naW5lINDEuZXhhbXBs
Pkyg1rQHR6ebGQ==
-----END CERTIFICATE-----
```



重要

这是获取服务器使用的 CA 证书的最可靠方法。此处描述的其余方法将在大多数情形中可用，但是如果被服务器管理员手动替换，它们将无法获得正确的 CA 证书。

方法 2

如果无法使用上述的 `openssl s_client` 方法，您可以使用命令行工具从 Red Hat Virtualization Manager 下载 CA 证书。

命令行工具示例包括 `curl` 和 `wget`，它们都可在多个平台上使用。

如果使用 `curl`：

```
$ curl \
--output ca.crt \
'http://myengine.example.com/ovirt-engine/services/pki-resource?resource=ca-
certificate&format=X509-PEM-CA'
```

如果使用 `wget`：

```
$ wget \
--output-document ca.crt \
'http://myengine.example.com/ovirt-engine/services/pki-resource?resource=ca-
certificate&format=X509-PEM-CA'
```

方法 3

使用 Web 浏览器导航到位于以下位置的证书：

```
https://myengine.example.com/ovirt-engine/services/pki-resource?resource=ca-
certificate&format=X509-PEM-CA
```

根据所选的浏览器，证书下载或导入浏览器的密钥存储。

1. **如果浏览器下载证书**：将文件保存为 **ca.crt**。
2. **如果浏览器导入证书**：从浏览器的认证选项导出它，并将其保存为 **ca.crt**。

方法 4

登录 Red Hat Virtualization Manager，从信任存储导出证书，并将其复制到您的客户端计算机。

1. 以 root 用户身份登录 Red Hat Virtualization Manager 机器。
2. 使用 Java **密钥工具管理** 实用程序从信任存储导出证书：

```
# keytool \
-keystore /etc/pki/ovirt-engine/.truststore \
-storepass mypass \
-exportcert \
-alias cacert \
-rfc \
-file ca.crt
```

这会创建一个名为 **ca.crt** 的证书文件。

3. 使用 `scp` 命令将证书复制到客户端机器：

```
$ scp ca.crt myuser@myclient.example.com:/home/myuser/.
```

每个方法都会在您的客户端机器上产生名为 `ca.crt` 的证书文件。然后，您必须将此文件导入到客户端的证书存储中。

2.1.2. 将证书导入到客户端

将证书导入到客户端取决于客户端如何存储和解释证书。有关导入证书的更多信息，请参阅您的客户端文档。

2.2. AUTHENTICATION

具有 Red Hat Virtualization Manager 帐户的任何用户都可以访问该 API。所有请求都必须使用 **OAuth** 或基本身份验证进行身份验证，如下所述。

2.2.1. OAuth 身份验证

自 Red Hat Virtualization 版本 4.0 起，首选身份验证机制为 **OAuth 2.0**，如 [RFC 6749](#) 所述。

OAuth 是一种复杂的协议，具有多种获取授权和访问令牌的机制。与 Red Hat Virtualization API 一起使用，唯一支持的是 *Resource Owner Password Credentials Grant*，如 [RFC 6749 的 4.3 节](#) 所述。

您必须首先获取令牌，将用户名和密码发送到 Red Hat Virtualization Manager 单点登录服务：

```
POST /ovirt-engine/sso/oauth/token HTTP/1.1
Host: myengine.example.com
Content-Type: application/x-www-form-urlencoded
Accept: application/json
```

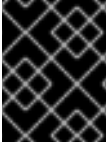
请求正文必须包含 `grant_type`、`scope`、`username` 和 `password` 参数：

表 2.1. OAuth 令牌请求参数

名称	值
<code>grant_type</code>	<code>password</code>
<code>scope</code>	<code>ovirt-app-api</code>
<code>username</code>	<code>admin@internal</code>
<code>password</code>	<code>mypassword</code>

这些参数必须经过 [URL 编码](#)。例如，用户名中的 `@` 字符需要编码为 `%40`。生成的请求正文将类似如下：

```
grant_type=password&scope=ovirt-app-api&username=admin%40internal&password=mypassword
```

重要

在 OAuth RFC 中，**scope** 参数被描述为可选，但在将它与 Red Hat Virtualization API 搭配使用时，它的值必须是 **ovirt-app-api**。

如果用户名和密码有效，Red Hat Virtualization Manager 单点登录服务将使用类似本文档的 JSON 文档进行响应：

```
{
  "access_token": "fqBR1ftzh8wBCviLxJcYuV5oSDI=",
  "token_type": "bearer",
  "scope": "...",
  ...
}
```

出于 API 身份验证目的，唯一相关的名称/值对是 **access_token**。不要以任何方式操作；使用它与 SSO 服务提供的完全相同。

获取令牌后，它可用于通过在 HTTP **授权** 标头中包含和使用 **Bearer** 方案来对 API 执行请求。例如，要获取虚拟机列表，请发送如下请求：

```
GET /ovirt-engine/api/vms HTTP/1.1
Host: myengine.example.com
Accept: application/xml
Authorization: Bearer fqBR1ftzh8wBCviLxJcYuV5oSDI=
```

该令牌可以多次用于多个请求，但它最终会过期。当它过期时，服务器将使用 401 HTTP 响应代码拒绝请求：

```
HTTP/1.1 401 Unauthorized
```

当发生这种情况时，需要一个新的令牌，因为 Red Hat Virtualization Manager 单点登录服务目前不支持刷新令牌。可以使用上述相同方法请求新的令牌。

2.2.2. 基本身份验证



重要

基本身份验证只支持向后兼容，自 Red Hat Virtualization 版本 4.0 起已弃用，并将在以后的中删除。

每个请求使用 HTTP 基本身份验证^[2]对凭据进行编码：如果请求没有包含适当的 **授权** 标头，服务器会发送 **401 Authorization Required** 响应：

```
HEAD /ovirt-engine/api HTTP/1.1
Host: myengine.example.com

HTTP/1.1 401 Authorization Required
```

请求通过指定域的**授权**标头发出。使用 **username@domain:password** 约定在提供的凭证中对适当的 Red Hat Virtualization Manager 域和用户进行编码。

下表显示了 Base64 中编码凭据的流程。

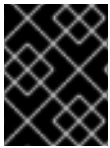
表 2.2. 编码凭证用于 API 访问

项	值
用户名	admin
域	internal
密码	mypassword
未编码凭证	admin@internal:mypassword
Base64 编码的凭证	YWRtaW5AaW50ZXJuYWw6bXlwYXNzd29yZA==

提供 Base64 编码的凭证，如下所示：

```
HEAD /ovirt-engine/api HTTP/1.1
Host: myengine.example.com
Authorization: Basic YWRtaW5AaW50ZXJuYWw6bXlwYXNzd29yZA==

HTTP/1.1 200 OK
```



重要

基本身份验证涉及潜在的敏感信息，如密码，以纯文本形式发送。API 需要 Hypertext 传输协议安全(HTTPS)进行纯文本请求的传输级别的加密。



重要

有些 Base64 库将结果分为多行，并使用换行字符终止每行。这会破坏标头，并导致出现错误的请求。**Authorization** 标头需要在标头内一行上编码的凭证。

2.2.3. 身份验证会话

API 也提供身份验证会话支持。使用身份验证详细信息发送初始请求，然后使用会话 Cookie 发送所有后续请求进行身份验证。

2.2.3.1. 请求经过身份验证的会话

1. 使用 **Authorization** 和 **Prefer: persistent-auth** 标头发送请求：

```
HEAD /ovirt-engine/api HTTP/1.1
Host: myengine.example.com
Authorization: Basic YWRtaW5AaW50ZXJuYWw6bXlwYXNzd29yZA==
Prefer: persistent-auth

HTTP/1.1 200 OK
...
```

这会返回一个带有以下标头的响应：

```
Set-Cookie: JSESSIONID=5dQja5ubr4yvl2MM2z+LZxrK; Path=/ovirt-engine/api; Secure
```

记录 **JSESSIONID=** 值。在本例中，值为 **5dQja5ubr4yvl2MM2z+LZxrK**。

2. 使用 **JSESSIONID=** 值的 **Prefer: persistent-auth** 和 **Cookie** 标头发送随后的所有请求。使用经过身份验证的会话时，不再需要 **Authorization** 标头。

```
HEAD /ovirt-engine/api HTTP/1.1
Host: myengine.example.com
Prefer: persistent-auth
Cookie: JSESSIONID=5dQja5ubr4yvl2MM2z+LZxrK
```

```
HTTP/1.1 200 OK
```

```
...
```

3. 当不再需要会话时，请在没有 **Prefer: persistent-auth** 标头的情况下对服务器执行请求。

```
HEAD /ovirt-engine/api HTTP/1.1
Host: myengine.example.com
Authorization: Basic YWRtaW5AaW50ZXJuYWw6bXlwYXNzd29yZA==
```

```
HTTP/1.1 200 OK
```

```
...
```

[1] RFC 2818 HTTP Over TLS 中描述了 HTTPS。

[2] RFC 2617 HTTP 身份验证中介绍了基本身份验证：基本和摘要访问身份验证。

第 3 章 常见概念

3.1. 类型

API 使用类型概念来描述接受和返回的不同对象类型。

有三种相关的类型：

原语类型

描述简单对象种类，如字符串或整数。

枚举类型

描述有效值列表，如 `VmStatus` 或 `DiskFormat`。

结构化类型

描述结构化对象，具有多个属性和链接，如 `Vm` 或 `Disk`。

3.2. 识别的类型

API 使用的许多类型都代表有标识的对象，它们具有唯一标识符并且独立于其他对象。用于描述这些对象的类型扩展了 `Identified` 类型，其中包含以下常用属性集合：

属性	类型	描述
<code>id</code>	字符串	虚拟化基础架构中的每个对象包含一个 <code>id</code> ，它充当唯一标识符。
<code>href</code>	字符串	对象的规范位置作为绝对路径。
<code>name</code>	字符串	用户提供的对象的人类可读名称。 <code>名称</code> 在同一类型的所有对象中都是唯一的。
<code>description</code>	字符串	免费的用户提供对象的人类可读描述。



重要

目前，对于大多数类型的对象，`id` 属性实际上是一个随机生成的 `UUID`，但这是一个实施细节，用户不应依赖它，因为它将来可能会有所变化。相反，用户应该假定这些标识符只是字符串。

3.3. 对象 (OBJECT)

对象是 API 支持的类型的单独实例。例如，标识符为 123 的虚拟机是 `Vm` 类型的对象。

3.4. 集合

集合是一组同类型的对象。

3.5. 表示

在客户端和服务端之间传输对象时，需要表示对象的状态。该 API 支持 XML 和 JSON 作为对象状态的一种表示，包括用于输入和输出的对象。

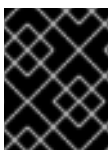
3.5.1. XML 表述

对象的 XML 表示法包含与对象类型对应的 XML 元素、**id** 和 **href** 属性的 XML 属性，以及其余属性的嵌套 XML 元素。例如，虚拟机的 XML 表述如下：

```
<vm id="123" href="/ovirt-engine/api/vms/123">
  <name>myvm</name>
  <description>My VM</description>
  <memory>1073741824</memory>
  ...
</vm>
```

对象集合的 XML 表示由 XML 元素组成，以对象类型命名，以复数表示。这包含集合对象的表示。例如，虚拟机集合的 XML 表述如下：

```
<vms>
  <vm id="123" href="/ovirt-engine/api/vms/123">
    <name>yourvm</name>
    <description>Your VM</description>
    <memory>1073741824</memory>
    ...
  </vm>
  <vm id="456" href="/ovirt-engine/api/vms/456">
    <name>myname</name>
    <description>My description</description>
    <memory>2147483648</memory>
    ...
  </vm>
  ...
</vms>
```



重要

在对象 XML 表示中，**id** 和 **href** 属性是唯一以 XML 属性表示的属性，其余则表示为嵌套的 XML 元素。

3.5.2. JSON 表示

对象的 JSON 表示形式包含含有各个属性（包括 **id** 和 **href**）的名称/值对的 JSON 文档。例如，虚拟机的 JSON 表示形式如下：

```
{
  "id": "123",
  "href": "/ovirt-engine/api/vms/123",
  "name": "myvm",
  "description": "My VM",
  "memory": 1073741824,
  ...
}
```

对象集合的 JSON 表示形式包含一个 JSON 文档，包含一个名称/值对（以单数形式命名对象类型），后者包含一个数组，其中包含集合对象的表示形式。例如，虚拟机集合的 JSON 代表形式如下：

```
{
  "vm": [
    {
      "id": "123",
      "href": "/ovirt-engine/api/vms/123",
      "name": "myvm",
      "description": "My VM",
      "memory": 1073741824,
      ...
    },
    {
      "id": "456",
      "href": "/ovirt-engine/api/vms/456",
      "name": "yourvm",
      "description": "Your VM",
      "memory": 2147483648,
      ...
    },
  ],
}
```

3.6. 服务

服务是负责检索、添加更新、删除和执行 API 支持的对象操作的服务器的一部分。

有两种相关的服务：

管理一组对象的服务

这些服务可用于列出现有对象和添加新对象。例如，[Vms](#) 服务负责管理系统中可用的虚拟机的集合。

管理特定对象的服务

这些服务负责在特定对象中检索、更新、删除和执行操作。例如，[Vm](#) 服务负责管理特定的虚拟机。

每一服务都可通过服务器中的特定路径访问。例如，可以通过路径 `/vms` 在中管理系统中可用的虚拟机集合的服务，以及管理虚拟机 123 的服务可通过路径 `/vms/123` 获取。

所有类型的服务都具有一组方法，表示它们可以执行的操作。管理对象集合的服务通常具有 **列表** 和 **添加** 方法。管理特定对象的服务通常具有 **get**、**update** 和 **remove** 方法。此外，服务也可能具有代表较少常见操作的操作方法。例如，[Vm](#) 服务具有一个用于启动虚拟机的 `start` 方法。

对于更常用的方法，方法名称和 HTTP 方法名称之间存在直接映射：

方法名称	HTTP 方法
add	POST
get	GET
list	GET

方法名称	HTTP 方法
update	PUT
remove	删除

HTTP 请求中使用的路径是服务的路径，其前缀为 `/ovirt-engine/api`。

例如，列出虚拟机的请求应如下所示，使用 HTTP **GET** 方法和路径 `/vms`：

```
GET /ovirt-engine/api/vms
```

对于操作方法，HTTP 方法始终为 **POST**，方法的名称作为后缀添加到路径中。例如，启动虚拟机 123 的请求应如下所示，使用 HTTP **POST** 方法和路径 `/vms/123/start`：

```
POST /ovirt-engine/api/vms/123/start
```

每种方法具有一组参数。

参数分为两类：

main 参数

main 参数对应检索、添加或更新的对象或集合。这只适用于 **add**、**get**、**列表** 和 **更新** 方法，每个方法正好有一个这样的主参数。

辅助参数

其余的参数。

例如，添加虚拟机的操作（请参阅 [此处](#)）具有三个参数：`vm`、**克隆** 和 `clone_permissions`。主要参数是 `vm`，因为它描述了要添加的对象。`clone` 和 `clone_permissions` 参数是辅助参数。

用于输入的主要参数必须包含在 HTTP 请求的正文中。例如，在添加虚拟机时，类型为 `Vm` 的 `vm` 参数必须包含在请求正文中。因此，添加虚拟机的完整请求（包括所有 HTTP 详情）必须如下所示：

```
POST /ovirt-engine/api/vms HTTP/1.1
Host: myengine.example.com
Authorization: Bearer fqBR1ftzh8wBCviLxJcYuV5oSDI=
Content-Type: application/xml
Accept: application/xml
```

```
<vm>
  <name>myvm</name>
  <description>My VM</description>
  <cluster>
    <name>Default</name>
  </cluster>
  <template>
    <name>Blank</name>
  </template>
</vm>
```

当用于输出时，主要参数包含在响应正文中。例如，添加虚拟机时，`vm` 参数将包含在响应正文中。因此，完整的响应正文如下所示：

```

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

<vm href="/ovirt-engine/api/vms/123" id="123">
  <name>myvm</name>
  <description>My VM</description>
  ...
</vm>

```

辅助参数仅允许进行输入（操作方法除外，稍后介绍），且必须作为查询参数包含。例如，当添加将 **clone** 参数设置为 **true** 的虚拟机时，完整的请求必须如下所示：

```

POST /ovirt-engine/api/vms?clone=true HTTP/1.1
Host: myengine.example.com
Authorization: Bearer fqBR1ftzh8wBCviLxJcYuV5oSDI=
Content-Type: application/xml
Accept: application/xml

<vm>
  <name>myvm</name>
  <description>My VM</description>
  <cluster>
    <name>Default</name>
  </cluster>
  <template>
    <name>Blank</name>
  </template>
</vm>

```

操作方法仅具有次要参数。它们可用于输入和输出，并且它们应包含在请求正文中，并用 **action** 元素打包。例如，用于启动虚拟机的操作方法（请参阅 [此处](#)）有一个 **vm** 参数，用于描述应当如何启动虚拟机，以及用于指定是否应使用 **cloud-init** 参数来配置客户机操作系统的 **use_cloud_init** 参数。因此，在使用 XML 时，使用 *cloud-init* 启动虚拟机 **123** 的完整请求将如下所示：

```

POST /ovirt-engine/api/vms/123/start HTTP/1.1
Host: myengine.example.com
Authorization: Bearer fqBR1ftzh8wBCviLxJcYuV5oSDI=
Content-Type: application/xml
Accept: application/xml

<action>
  <use_cloud_init>true</use_cloud_init>
  <vm>
    <initialization>
      <nic_configurations>
        <nic_configuration>
          <name>eth0</name>
          <on_boot>true</on_boot>
          <boot_protocol>static</boot_protocol>
          <ip>
            <address>192.168.0.100</address>
            <netmask>255.255.255.0</netmask>
            <gateway>192.168.0.1</netmask>
          </ip>
        </nic_configuration>

```



```

    </nic_configurations>
    <dns_servers>192.168.0.1</dns_servers>
  </initialization>
</vm>
</action>

```

3.7. 搜索

某些服务的列表方法有一个 **搜索** 参数，可用于指定搜索条件。使用时，服务器将仅返回满足这些条件的集合中的对象。例如，以下请求将仅返回名为 **myvm** 的虚拟机：

```
GET /ovirt-engine/api/vms?search=name%3Dmyvm
```

3.7.1. 最大结果参数

使用 **max** 参数限制返回的对象数量。例如，无论系统中有多少可用的虚拟机，以下请求将仅返回一个虚拟机：

```
GET /ovirt-engine/api/vms?max=1
```

没有 **max** 参数的搜索请求将返回所有对象。建议指定 **max** 参数，以减少请求对系统整体性能的影响。

3.7.2. 问题单敏感度

默认情况下，查询不区分大小写。例如，以下请求将返回名为 **myvm**、**MyVM** 和 **MYVM** 的虚拟机：

```
GET /ovirt-engine/api/vms?search=name%3Dmyvm
```

可选的 **区分大小写** 的布尔值参数可用于更改此冲突。例如，要完全获取名为 **myhost** 的虚拟机，而不是 **MyHost** 或 **MYHOST**，请发送如下请求：

```
GET /ovirt-engine/api/vms?search=name%3D=myvm&case_sensitive=true
```

3.7.3. 搜索语法

搜索 参数使用与 Red Hat Virtualization 查询语言相同的语法：

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

sortby 子句是可选的，仅当对结果进行排序时才需要。

搜索查询示例：

集合(Collection)	标准	结果
主机	vms.status=up	返回正在运行虚拟机的所有主机的列表。
vms	domain=example.com	返回在指定域上运行的所有虚拟机的列表。

集合(Collection)	标准	结果
vms	users.name=mary	返回属于用户名为 name mary 的所有虚拟机的列表。
事件	severity > 正常 排序	返回所有严重级别高于 正常 并按照 time 属性值排序的事件的列表。
事件	severity > normal sortby time desc	返回所有严重级别高于 正常 的事件列表，并按其 time 属性值降序排列。

搜索参数的值必须经过 [URL 编码](#)，才能翻译保留的字符，如 operators 和空格。例如，等号应编码为 **%3D**：

```
GET /ovirt-engine/api/vms?search=name%3Dmyvm
```

3.7.4. 通配符

星号可用作值的一部分，以指示任何字符串匹配，包括 empty 字符串。例如，以下请求将返回名称以 myvm 开头的所有虚拟机，如 **myvm**、**myvm 2**、**myvm a** 或 **myvm-webserver**：

```
GET /ovirt-engine/api/vms?search=name%3Dmyvm*
```

3.7.5. 分页

有些 Red Hat Virtualization 环境包含大量对象。通过一个请求检索所有这些请求并不实际，会损害性能。为了允许通过页面检索它们，**搜索** 参数支持可选 **page** 子句。这与 **max** 参数相结合，是分页的基础。例如，要获得虚拟机的第一页，页面大小为 10 个虚拟机，请按如下所示发送请求：

```
GET /ovirt-engine/api/vms?search=page%201&max=10
```



注意

搜索参数是 URL 编码的，**搜索** 参数的实际值在编码前是第 **1** 页，因此这实际上正在请求第一页。

增加 **页面** 值以检索下一页：

```
GET /ovirt-engine/api/vms?search=page%202&max=10
```

page 子句可以和 **search** 参数内的其他子句搭配使用。例如，以下请求将返回虚拟机的第二页，但按名称排序：

```
GET /ovirt-engine/api/vms?search=sortby%20name%20page%202&max=10
```



重要

API 是无状态的；不能在不同请求之间保留状态，因为所有请求都是相互独立的。因此，如果您的请求之间发生状态更改，则页面结果可能会不一致。

例如，如果您从虚拟机列表中请求特定页面，并且在请求下一页之前创建或删除虚拟机，您的结果可能缺失，或者包含重复项。

3.8. 以下链接

API 返回对相关对象的引用作为链接。例如，当检索虚拟机时，它包含到其磁盘附加和网络接口卡的链接：

```
<vm id="123" href="/ovirt-engine/api/vms/123">
...
<link rel="diskattachments" href="/ovirt-engine/api/vms/123/diskattachments"/>
<link rel="nics" href="/ovirt-engine/api/vms/123/nics"/>
...
</vm>
```

这些链接对象的完整描述可通过发送单独的请求来检索：

```
GET /ovirt-engine/api/vms/123/diskattachments
GET /ovirt-engine/api/vms/123/nics
```

但在某些情况下，使用 API 检索同一请求中链接的信息更为方便。例如，当额外的网络往返会带来不便的开销时，或者当多个请求以不接受的方式使应用代码变得复杂时，这很有用。对于这些用例，API 提供了一个 **以下参数**，它允许应用仅使用一个请求来检索链接的信息。

以下 **参数的值** 是由逗号分隔的字符串列表。这些字符串各自是链接的对象的路径。例如，要检索磁盘附件和上例中的 NIC，请求应类似如下：

```
GET /ovirt-engine/api/vms/123?follow=disk_attachments,nics
```

这将返回类似如下的响应：

```
<vm id="123" href="/ovirt-engine/api/vms/123">
...
<disk_attachments>
  <disk_attachment id="456" href="/ovirt-engine/api/vms/123/diskattachments/456">
    <active>true</active>
    <bootable>true</bootable>
    <interface>virtio_scsi</interface>
    <pass_discard>>false</pass_discard>
    <read_only>>false</read_only>
    <uses_scsi_reservation>>false</uses_scsi_reservation>
    <disk id="789" href="/ovirt-engine/api/disks/789"/>
  </disk_attachment>
  ...
</disk_attachments>
<nics>
  <nic id="234" href="/ovirt-engine/api/vms/123/nics/234">
    <name>eth0</name>
    <interface>virtio</interface>
```

```

<linked>true</linked>
<mac>
  <address>00:1a:4a:16:01:00</address>
</mac>
<plugged>true</plugged>
</nic>
...
</nics>
...
</vm>

```

链接对象的路径可以是单个词语，如上例中所示，也可以是一系列词语（用点分隔）来请求嵌套数据。例如，上例使用 **disk_attachments** 来检索磁盘附加的完整描述，但每个磁盘附加包含一个指向磁盘的链接，而这并未遵循。要追踪到磁盘的链接，可以使用以下请求：

```
GET /ovirt-engine/api/vms/123?follow=disk_attachments.disk
```

这将产生以下响应：

```

<vm id="123" href="/ovirt-engine/api/vms/123">
  <disk_attachments>
    <disk_attachment id="456" href="/ovirt-engine/api/vms/123/diskattachments/456">
      <active>true</active>
      <bootable>true</bootable>
      <interface>virtio_scsi</interface>
      <pass_discard>false</pass_discard>
      <read_only>false</read_only>
      <uses_scsi_reservation>false</uses_scsi_reservation>
      <disk id="789" href="/ovirt-engine/api/disks/789">
        <name>mydisk</name>
        <description>My disk</description>
        <actual_size>0</actual_size>
        <format>raw</format>
        <sparse>true</sparse>
        <status>ok</status>
        <storage_type>image</storage_type>
        <total_size>0</total_size>
        ...
      </disk>
    </disk_attachment>
    ...
  </disk_attachments>
  ...
</vm>

```

路径可根据需要深层进行。例如，也要获取磁盘统计信息：

```
GET /ovirt-engine/api/vms/123?follow=disk_attachments.disk.statistics
```

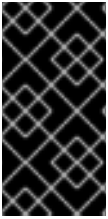
可以组合多个路径元素和多个路径。例如，要获取磁盘附件和网络接口卡，以及它们的统计信息：

```
GET /ovirt-engine/api/vms/123?follow=disk_attachments.disk.statistics,nics.statistics
```



重要

几乎所有检索对象的操作都支持 **以下参数**，但确保明确检查参考文档，因为某些操作可能不支持它，或者可能会提供有关如何使用它来获取最佳性能的建议。



重要

使用以下 **参数**，将开销从客户端移到服务器端。当您请求其他数据时，服务器必须获取并将其与基本数据合并。这在服务器端消耗 CPU 和内存，并且大部分情况下需要额外的数据库查询。这可能会对服务器的性能造成负面影响，特别是在大规模环境中。确保在真实的环境中测试您的应用程序，并且仅在合理时才使用以下参数。

3.9. 权限

管理单个对象的许多服务都提供对 **权限** 服务的引用，该服务管理分配给该对象的权限。每个权限包含指向用户或组、角色和对象的链接。例如，可以检索分配给特定虚拟机的权限来发送如下请求：

```
GET /ovirt-engine/api/vms/123/permissions
```

响应正文如下所示：

```
<permissions>
  <permission id="456" href="/ovirt-engine/api/vms/123/permissions/456">
    <user id="789" href="/ovirt-engine/api/users/789"/>
    <role id="abc" href="/ovirt-engine/api/roles/abc"/>
    <vm id="123" href="/ovirt-engine/api/vms/123"/>
  </permission>
  ...
</permissions>
```

向发送 **POST** 请求的对象添加权限，并将权限表示给此服务。每个新权限都需要一个角色和用户。

3.10. 处理错误

某些错误需要在标准 HTTP 状态代码之外做进一步的说明。例如，API 报告 **响应正文** 中失败的对象状态更新或操作失败。此错误包含 **原因和 详细信息** 属性。例如，当服务器收到在没有强制 **name** 属性的情况下创建虚拟机的请求时，它将使用以下 HTTP 响应行响应：

```
HTTP/1.1 400 Bad Request
```

和以下响应正文：

```
<fault>
  <reason>Incomplete parameters</reason>
  <detail>Vm [name] required for add</detail>
</fault>
```

第 4 章 快速入门示例

本节中的示例演示了如何使用 REST API 设置基本红帽虚拟化环境并创建虚拟机。除了标准先决条件外，这些示例还需要以下内容：

- 联网和配置红帽虚拟化安装。
- 包含您要安装的虚拟机操作系统的 ISO 文件。本章使用 [CentOS 7](#) 作为安装 ISO 示例。

API 示例使用 `curl` 来演示客户端应用的 API 请求。您可以使用任何发送 HTTP 请求的应用。



重要

本例中的 HTTP 请求标头省略 **Host** 和 **Authorization** 标头。但是，这些字段是必需的，需要特定于 Red Hat Virtualization 安装的数据。

`curl` 示例使用 `admin@internal` 作为用户名，`my password` 作为密码，`/etc/pki/ovirt-engine/ca.pem` 用作证书位置，`myengine.example.com` 作为主机名。您必须将它们替换为您的环境的正确值。

Red Hat Virtualization 会为每个资源生成唯一标识符，用于 `id` 属性。本例中的标识符代码将与 Red Hat Virtualization 环境中的标识符代码不同。

在许多示例中，为了简洁起见，API 返回的结果的一些属性已被省略。例如，请参阅[集群引用](#)来获取属性的完整列表。

4.1. 访问 API 入口点

以下请求检索 API 版本 4 的主要入口点表示：

```
GET /ovirt-engine/api HTTP/1.1
Version: 4
Accept: application/xml
```

同一请求，但使用 `/v4` URL 前缀而不是 **Version** 标头：

```
GET /ovirt-engine/api/v4 HTTP/1.1
Accept: application/xml
```

同一个请求，使用 `curl` 命令：

```
curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api
```

结果是一个类型为 `Api` 的对象：

```
<api>
<link href="/ovirt-engine/api/clusters" rel="clusters"/>
```

```

<link href="/ovirt-engine/api/datacenters" rel="datacenters"/>
...
<product_info>
  <name>oVirt Engine</name>
  <vendor>ovirt.org</vendor>
  <version>
    <build>0</build>
    <full_version>4.0.0-0.0.el7</full_version>
    <major>4</major>
    <minor>0</minor>
    <revision>0</revision>
  </version>
</product_info>
<special_objects>
  <blank_template href="..." id="..."/>
  <root_tag href="..." id="..."/>
</special_objects>
<summary>
  <hosts>
    <active>23</active>
    <total>30</total>
  </hosts>
  <storage_domains>
    <active>5</active>
    <total>6</total>
  </storage_domains>
  <users>
    <active>12</active>
    <total>102</total>
  </users>
  <vms>
    <active>253</active>
    <total>545</total>
  </vms>
</summary>
<time>2016-10-06T15:38:18.548+02:00</time>
</api>

```

重要

如果未使用标头和 URL 前缀，服务器将自动选择版本。默认为版本 **4**。您可以使用 **ENGINE_API_DEFAULT_VERSION** 配置参数更改默认版本：

```

# echo "ENGINE_API_DEFAULT_VERSION=3" > \
/etc/ovirt-engine/engine.conf.d/99-set-default-version.conf
# systemctl restart ovirt-engine

```

更改此参数会影响未明确指定版本的 API 的所有用户。

入口点为用户提供虚拟环境中集合的链接。每个集合链接的 **Therel 属性** 都会为每个链接提供一个参考点。本例中的下一步将检查数据中心集合，该集合可通过 **datacenters** 链接使用。

入口点还包含其他数据，如 **product_info**、**special_objects** 和 **summary**。本例之外的章节将对此数据进行阐述。

4.2. 列出数据中心

Red Hat Virtualization 在安装时创建一个默认数据中心。本例使用 **Default** 数据中心作为虚拟环境的基础。

以下请求检索数据中心的表示法：

```
GET /ovirt-engine/api/datacenters HTTP/1.1
Accept: application/xml
```

同一个请求，使用 **curl** 命令：

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api/datacenters
```

结果将是 **DataCenter** 类型的对象列表：

```
<data_centers>
<data_center href="/ovirt-engine/api/datacenters/001" id="001">
  <name>Default</name>
  <description>The default Data Center</description>
  <link href="/ovirt-engine/api/datacenters/001/clusters" rel="clusters"/>
  <link href="/ovirt-engine/api/datacenters/001/storagedomains" rel="storagedomains"/>
  ...
  <local>false</local>
  <quota_mode>disabled</quota_mode>
  <status>up</status>
  <supported_versions>
    <version>
      <major>4</major>
      <minor>0</minor>
    </version>
  </supported_versions>
  <version>
    <major>4</major>
    <minor>0</minor>
  </version>
</data_center>
...
</data_centers>
```

请注意您的 **Default** 数据中心的 **ID**。它相对于虚拟环境的其他资源标识此数据中心。

数据中心还包含指向管理附加到数据中心的存储域的服务的链接：

```
<link href="/ovirt-engine/api/datacenters/001/storagedomains" rel="storagedomains"/>
```

该服务用于从主 **storagedomains** 集合中附加存储域，本例稍后将对此进行阐述。

4.3. 列出主机集群

Red Hat Virtualization 在安装时创建一个默认主机集群。本例使用 **Default** 集群对 Red Hat Virtualization 环境中的资源进行分组。

以下请求检索集群集合的表示形式：

```
GET /ovirt-engine/api/clusters HTTP/1.1
Accept: application/xml
```

同一个请求，使用 **curl** 命令：

```
curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api/clusters
```

结果将是类型为 **Cluster** 的对象列表：

```
<clusters>
  <cluster href="/ovirt-engine/api/clusters/002" id="002">
    <name>Default</name>
    <description>The default server cluster</description>
    <link href="/ovirt-engine/api/clusters/002/networks" rel="networks"/>
    <link href="/ovirt-engine/api/clusters/002" rel="permissions"/>
    ...
    <cpu>
      <architecture>x86_64</architecture>
      <type>Intel Nehalem Family</type>
    </cpu>
    <version>
      <major>4</major>
      <minor>0</minor>
    </version>
    <data_center href="/ovirt-engine/api/datacenters/001" id="001"/>
  </cluster>
  ...
</clusters>
```

请注意 **默认** 主机集群的 **id**。它根据虚拟环境的其他资源标识此主机集群。

Default 集群通过使用 **data_center** 链接的 **id** 和 **href** 属性的关系，与 **Default** 数据中心 关联：

```
<data_center href="/ovirt-engine/api/datacenters/001" id="001"/>
```

网络 链接是对管理与此集群关联的网络的 **服务的** 引用。下一节将更详细地阐述网络集合。

4.4. 列出逻辑网络

Red Hat Virtualization 在安装时创建一个默认的 **ovirtmgmt** 网络。此网络充当红帽虚拟化管理器访问主机的管理网络。

此网络与 **Default** 集群关联，是 **Default** 数据中心的成员。本例使用 **ovirtmgmt** 网络来连接虚拟机。

以下请求检索逻辑网络列表：

```
GET /ovirt-engine/api/networks HTTP/1.1
Accept: application/xml
```

同一个请求，使用 **curl** 命令：

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api/networks
```

结果将是网络类型的对象列表：

```
<networks>
  <network href="/ovirt-engine/api/networks/003" id="003">
    <name>ovirtmgmt</name>
    <description>Management Network</description>
    <link href="/ovirt-engine/api/networks/003/permissions" rel="permissions"/>
    <link href="/ovirt-engine/api/networks/003/vnicprofiles" rel="vnicprofiles"/>
    <link href="/ovirt-engine/api/networks/003/networklabels" rel="networklabels"/>
    <mtu>0</mtu>
    <stp>false</stp>
    <usages>
      <usage>vm</usage>
    </usages>
    <data_center href="/ovirt-engine/api/datacenters/001" id="001"/>
  </network>
  ...
</networks>
```

ovirtmgmt 网络通过利用数据中心的 ID 的关系附加到 **Default** 数据中心。

ovirtmgmt 网络还通过集群网络子集合中的关系附加到 **Default** 集群。

4.5. 列出主机

本示例检索主机列表，并显示在虚拟环境中注册的名为 **myhost** 的主机：

```
GET /ovirt-engine/api/hosts HTTP/1.1
Accept: application/xml
```

同一个请求，使用 **curl** 命令：

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
```

```
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api/hosts
```

结果将是主机类型的对象列表：

```
<hosts>
<host href="/ovirt-engine/api/hosts/004" id="004">
  <name>myhost</name>
  <link href="/ovirt-engine/api/hosts/004/nics" rel="nics"/>
  ...
  <address>node40.example.com</address>
  <cpu>
    <name>Intel Core Processor (Haswell, no TSX)</name>
    <speed>3600</speed>
    <topology>
      <cores>1</cores>
      <sockets>2</sockets>
      <threads>1</threads>
    </topology>
  </cpu>
  <memory>8371830784</memory>
  <os>
    <type>RHEL</type>
    <version>
      <full_version>7 - 2.1511.el7.centos.2.10</full_version>
      <major>7</major>
    </version>
  </os>
  <port>54321</port>
  <status>up</status>
  <cluster href="/ovirt-engine/api/clusters/002" id="002"/>
</host>
...
</hosts>
```

请注意主机的 **id**。它根据虚拟环境的其他资源标识此主机。

此主机是 **Default** 集群的成员，访问 **nics** 子集合显示此主机已连接到 **ovirtmgmt** 网络。

4.6. 创建 NFS 数据存储

NFS 数据存储域是连接到数据中心的 NFS 共享，为虚拟 guest 映像提供存储。创建新存储域需要 **POST** 请求，其中包含存储域表示法，并发送到存储域集合的 URL。

在存储域上，您可以默认启用 **wipe** 后删除选项。要配置此功能，请在 **POST** 请求中指定 **wipe_after_delete**。可以在创建域后编辑此选项，但是这样做不会在删除已存在的磁盘属性后更改擦除。

请求应类似如下：

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

请求正文应类似如下：

```
<storage_domain>
  <name>mydata</name>
  <type>data</type>
  <description>My data</description>
  <storage>
    <type>nfs</type>
    <address>my nfs.example.com</address>
    <path>/exports/mydata</path>
  </storage>
  <host>
    <name>myhost</name>
  </host>
</storage_domain>
```

同一个请求，使用 **curl** 命令：

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '
<storage_domain>
  <name>mydata</name>
  <description>My data</description>
  <type>data</type>
  <storage>
    <type>nfs</type>
    <address>my nfs.example.com</address>
    <path>/exports/mydata</path>
  </storage>
  <host>
    <name>myhost</name>
  </host>
</storage_domain>
'\
https://myengine.example.com/ovirt-engine/api/storagedomains
```

服务器使用主机 **myhost** 创建名为 **my data** 的 NFS 数据存储域，其导出路径为 **my nfs.example.com:/exports/mydata**。该 API 还返回以下表示新创建的存储域资源（类型为 [StorageDomain](#)）：

```
<storage_domain href="/ovirt-engine/api/storagedomains/005" id="005">
  <name>mydata</name>
  <description>My data</description>
  <available>42949672960</available>
  <committed>0</committed>
  <master>false</master>
  <status>unattached</status>
  <storage>
    <address>my nfs.example.com</address>
```

```

<path>/exports/mydata</path>
<type>nfs</type>
</storage>
<storage_format>v3</storage_format>
<type>data</type>
<used>9663676416</used>
</storage_domain>

```

4.7. 创建 NFS ISO 存储

NFS ISO 存储域是连接到数据中心的已挂载 NFS 共享，为 DVD/CD-ROM ISO 和虚拟软盘(VFD)镜像文件提供存储。创建新存储域需要 **POST** 请求，其中包含存储域表示法，并将其发送到存储域集合的 URL：

请求应类似如下：

```

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

```

请求正文应类似如下：

```

<storage_domain>
  <name>myisos</name>
  <description>My ISOs</description>
  <type>iso</type>
  <storage>
    <type>nfs</type>
    <address>my nfs.example.com</address>
    <path>/exports/myisos</path>
  </storage>
  <host>
    <name>myhost</name>
  </host>
</storage_domain>

```

同一个请求，使用 **curl** 命令：

```

# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '
<storage_domain>
  <name>myisos</name>
  <description>My ISOs</description>
  <type>iso</type>
  <storage>
    <type>nfs</type>
    <address>my nfs.example.com</address>
    <path>/exports/myisos</path>

```

```

</storage>
<host>
  <name>myhost</name>
</host>
</storage_domain>
'\
https://myengine.example.com/ovirt-engine/api/storagedomains

```

服务器使用主机 **myhost** 创建名为 **myisos** 的 NFS ISO 存储域，其导出路径为 **mynfs.example.com:/exports/myisos**。该 API 还返回以下表示新创建的存储域资源（类型为 [StorageDomain](#)）：

```

<storage_domain href="/ovirt-engine/api/storagedomains/006" id="006">
  <name>myiso</name>
  <description>My ISOs</description>
  <available>42949672960</available>
  <committed>0</committed>
  <master>false</master>
  <status>unattached</status>
  <storage>
    <address>mynfs.example.com</address>
    <path>/exports/myisos</path>
    <type>nfs</type>
  </storage>
  <storage_format>v1</storage_format>
  <type>iso</type>
  <used>9663676416</used>
</storage_domain>

```

4.8. 将存储域附加到数据中心

以下示例将 **mydata** 和 **myisos** 存储域附加到 **Default** 数据中心。

要附加 **mydata** 存储域，请发送类似如下的请求：

```

POST /ovirt-engine/api/datacenters/001/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

```

这样的请求正文：

```

<storage_domain>
  <name>mydata</name>
</storage_domain>

```

同一个请求，使用 **curl** 命令：

```

# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \

```

```
--data '
<storage_domain>
  <name>mydata</name>
</storage_domain>
'\
https://myengine.example.com/ovirt-engine/api/datacenters/001/storagedomains
```

要附加 **myisos** 存储域，请发送类似如下的请求：

```
POST /ovirt-engine/api/datacenters/001/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

这样的请求正文：

```
<storage_domain>
  <name>myisos</name>
</storage_domain>
```

同一个请求，使用 **curl** 命令：

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '
<storage_domain>
  <name>myisos</name>
</storage_domain>
'\
https://myengine.example.com/ovirt-engine/api/datacenters/001/storagedomains
```

4.9. 创建虚拟机

以下示例使用虚拟环境的空白模板为基础，在 **Default** 集群上创建一个名为 **myvm** 的虚拟机。该请求还将虚拟机的内存定义为 **512 MiB**，并将启动设备设置为虚拟硬盘。

请求应包含类型为 **Vm** 的对象，用于描述要创建的虚拟机：

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

请求正文应类似如下：

```

<vm>
  <name>myvm</name>
  <description>My VM</description>
  <cluster>
    <name>Default</name>
  </cluster>
  <template>
    <name>Blank</name>
  </template>
  <memory>536870912</memory>
  <os>
    <boot>
      <devices>
        <device>hd</device>
      </devices>
    </boot>
  </os>
</vm>

```

同一个请求，使用 `curl` 命令：

```

# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '
<vm>
  <name>myvm</name>
  <description>My VM</description>
  <cluster>
    <name>Default</name>
  </cluster>
  <template>
    <name>Blank</name>
  </template>
  <memory>536870912</memory>
  <os>
    <boot>
      <devices>
        <device>hd</device>
      </devices>
    </boot>
  </os>
</vm>
'\
https://myengine.example.com/ovirt-engine/api/vms

```

响应正文将是 `Vm` 类型的对象：


```

<vm href="/ovirt-engine/api/vms/007" id="007">
  <name>myvm</name>
  <link href="/ovirt-engine/api/vms/007/diskattachments" rel="diskattachments"/>
  <link href="/ovirt-engine/api/vms/007/nics" rel="nics"/>
  ...
  <cpu>
    <architecture>x86_64</architecture>
    <topology>
      <cores>1</cores>
      <sockets>1</sockets>
      <threads>1</threads>
    </topology>
  </cpu>
  <memory>1073741824</memory>
  <os>
    <boot>
      <devices>
        <device>hd</device>
      </devices>
    </boot>
    <type>other</type>
  </os>
  <type>desktop</type>
  <cluster href="/ovirt-engine/api/clusters/002" id="002"/>
  <status>down</status>
  <original_template href="/ovirt-engine/api/templates/000" id="00"/>
  <template href="/ovirt-engine/api/templates/000" id="000"/>
</vm>

```

4.10. 创建虚拟机 NIC

以下示例创建一个虚拟网络接口，将示例虚拟机连接到 `ovirtmgmt` 网络。

请求应类似如下：

```

POST /ovirt-engine/api/vms/007/nics HTTP/1.1
Content-Type: application/xml
Accept: application/xml

```

请求正文应包含类型 `Nic` 描述要创建的 NIC 的对象：

```

<nic>
  <name>mynic</name>
  <description>My network interface card</description>
</nic>

```

同一个请求，使用 `curl` 命令：

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '
<nic>
  <name>mynic</name>
  <description>My network interface card</description>
</nic>
'\
https://myengine.example.com/ovirt-engine/api/vms/007/nics
```

4.11. 创建虚拟机磁盘

以下示例为示例虚拟机创建一个 8 GiB 写时复制磁盘。

请求应类似如下：

```
POST /ovirt-engine/api/vms/007/diskattachments HTTP/1.1
Content-Type: application/xml
Accept: application/xml
```

请求正文应该是类型为 [DiskAttachment](#) 的对象，描述磁盘以及如何将其附加到虚拟机：

```
<disk_attachment>
  <bootable>false</bootable>
  <interface>virtio</interface>
  <active>true</active>
  <disk>
    <description>My disk</description>
    <format>cow</format>
    <name>mydisk</name>
    <provisioned_size>8589934592</provisioned_size>
    <storage_domains>
      <storage_domain>
        <name>mydata</name>
      </storage_domain>
    </storage_domains>
  </disk>
</disk_attachment>
```

同一个请求，使用 curl 命令：

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '
<disk_attachment>
  <bootable>>false</bootable>
  <interface>virtio</interface>
  <active>>true</active>
  <disk>
    <description>My disk</description>
    <format>cow</format>
    <name>mydisk</name>
    <provisioned_size>8589934592</provisioned_size>
    <storage_domains>
      <storage_domain>
        <name>mydata</name>
      </storage_domain>
    </storage_domains>
  </disk>
</disk_attachment>
'\
https://myengine.example.com/ovirt-engine/api/vms/007/diskattachments
```

storage_domains 属性告知 API 将磁盘存储在 mydata 存储域中。

4.12. 将 ISO 镜像附加到虚拟机

下面的虚拟机的引导介质需要一个 CD-ROM 或者 DVD ISO 镜像进行操作系统安装。这个示例使用 CentOS 7 镜像。

ISO 映像必须在 myisos ISO 域中提供，供虚拟机使用。您可以使用 [第 6.124 节 “ImageTransfers”](#) 创建镜像传输，使用 [第 6.123 节 “ImageTransfer”](#) 上传 ISO 镜像。

上传 ISO 镜像后，API 可用于从 ISO 存储域请求文件列表：

```
GET /ovirt-engine/api/storagedomains/006/files HTTP/1.1
Accept: application/xml
```

同一个请求，使用 `curl` 命令：

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
https://myengine.example.com/ovirt-engine/api/storagedomains/006/files
```

服务器返回以下 **File** 类型的对象列表，每个可用 ISO（或软盘）镜像一个对象：

```
<files>
  <file href="..." id="CentOS-7-x86_64-Minimal.iso">
    <name>CentOS-7-x86_64-Minimal.iso</name>
  </file>
  ...
</files>
```

API 用户将 `CentOS-7-x86_64-Minimal.iso` 附加到示例虚拟机。连接 ISO 映像等同于在管理或用户门户应用程序中使用 *Change CD* 按钮。

请求应类似如下：

```
PUT /ovirt-engine/api/vms/007/cdroms/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

请求正文应该是类型为 **Cdrom** 的对象，它包含一个 `file` 属性，以指示 ISO（或软盘）镜像的标识符：

```
<cdrom>
  <file id="CentOS-7-x86_64-Minimal.iso"/>
</cdrom>
```

同一个请求，使用 `curl` 命令：

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request PUT \
--header 'Version: 4' \
```

```

--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '
<cdrom>
  <file id="CentOS-7-x86_64-Minimal.iso"/>
</cdrom>
'\
https://myengine.example.com/ovirt-engine/api/vms/007/cdroms/00000000-0000-0000-0000-000000000000

```

如需了解更多详细信息，请参阅管理虚拟机 CD-ROMS 的[服务文档](#)。

4.13. 启动虚拟机

虚拟环境已经完成，虚拟机包含运行所需的所有组件。本例使用 `start` 方法启动虚拟机。

请求应类似如下：

```

POST /ovirt-engine/api/vms/007/start HTTP/1.1
Accept: application/xml
Content-type: application/xml

```

请求正文应类似如下：

```

<action>
  <vm>
    <os>
      <boot>
        <devices>
          <device>cdrom</device>
        </devices>
      </boot>
    </os>
  </vm>
</action>

```

同一个请求，使用 `curl` 命令：

```

# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \

```

```
--header 'Accept: application/xml' \  
--data '  
<action>  
  <vm>  
    <os>  
      <boot>  
        <devices>  
          <device>cdrom</device>  
        </devices>  
      </boot>  
    </os>  
  </vm>  
</action>  
' \  
https://myengine.example.com/ovirt-engine/api/vms/007/start
```

额外请求正文将虚拟机的引导设备设置为 **CD-ROM** 仅用于此启动。这可让虚拟机从附加的 **ISO** 镜像安装操作系统。引导设备恢复为**磁盘**以进行以后的所有引导。

第 5 章 REQUESTS

本节枚举 API 中提供的所有请求。

- [POST /affinitylabels](#)
- [GET /affinitylabels](#)
- [GET /affinitylabels/{label:id}](#)
- [PUT /affinitylabels/{label:id}](#)
- [DELETE /affinitylabels/{label:id}](#)
- [POST /affinitylabels/{label:id}/hosts](#)
- [GET /affinitylabels/{label:id}/hosts](#)
- [DELETE /affinitylabels/{label:id}/hosts/{host:id}](#)
- [GET /affinitylabels/{label:id}/hosts/{host:id}](#)
- [POST /affinitylabels/{label:id}/vms](#)
- [GET /affinitylabels/{label:id}/vms](#)
- [DELETE /affinitylabels/{label:id}/vms/{vm:id}](#)

- **GET /affinitylabels/{label:id}/vms/{vm:id}**
- **POST/书签**
- **GET/书签**
- **GET /书签/{bookmark:id}**
- **PUT/书签 /{bookmark:id}**
- **DELETE /书签/{bookmark:id}**
- **GET /clusterlevels**
- **GET /clusterlevels/{level:id}**
- **GET /clusterlevels/{level:id}/clusterfeatures**
- **GET /clusterlevels/{level:id}/clusterfeatures/{feature:id}**
- **POST/集群**
- **GET /cluster**
- **GET /cluster/{cluster:id}**
- **PUT /集群/{cluster:id}**

- **DELETE** /集群/{cluster:id}
- **POST** /集群/{cluster:id}/affinitygroups
- **GET** /cluster/{cluster:id}/affinitygroups
- **GET** /cluster/{cluster:id}/affinitygroups/{group:id}
- **PUT** /集群/{cluster:id}/affinitygroups/{group:id}
- **DELETE** /集群/{cluster:id}/affinitygroups/{group:id}
- **POST** /集群/{cluster:id}/affinitygroups/{group:id}/hostlabels
- **GET** /cluster/{cluster:id}/affinitygroups/{group:id}/hostlabels
- **DELETE** /集群/{cluster:id}/affinitygroups/{group:id}/hostlabels/{label:id}
- **POST** /集群/{cluster:id}/affinitygroups/{group:id}/hosts
- **GET** /cluster/{cluster:id}/affinitygroups/{group:id}/hosts
- **DELETE** /集群/{cluster:id}/affinitygroups/{group:id}/hosts/{host:idorname}
- **POST** /集群/{cluster:id}/affinitygroups/{group:id}/vmlabels
- **GET** /cluster/{cluster:id}/affinitygroups/{group:id}/vmlabels

- **DELETE** /集群/{cluster:id}/affinitygroups/{group:id}/vmlabels/{label:id}
- **POST** /集群/{cluster:id}/affinitygroups/{group:id}/vms
- **GET** /cluster/{cluster:id}/affinitygroups/{group:id}/vms
- **DELETE** /集群/{cluster:id}/affinitygroups/{group:id}/vms/{vm:id}
- **POST** /cluster/{cluster:id}/cpuprofiles
- **GET** /cluster/{cluster:id}/cpuprofiles
- **GET** /cluster/{cluster:id}/cpuprofiles/{profile:id}
- **DELETE** /集群/{cluster:id}/cpuprofiles/{profile:id}
- **GET** /cluster/{cluster:id}/enabledfeatures
- **POST** /集群/{cluster:id}/enabledfeatures
- **GET** /cluster/{cluster:id}/enabledfeatures/{feature:id}
- **DELETE** /集群/{cluster:id}/enabledfeatures/{feature:id}
- **GET** /cluster/{cluster:id}/externalnetworkproviders
- **GET** /cluster/{cluster:id}/glusterhooks

- **GET** /cluster/{cluster:id}/glusterhooks/{hook:id}
- **DELETE** /集群/{cluster:id}/glusterhooks/{hook:id}
- **POST** /cluster/{cluster:id}/glusterhooks/{hook:id}/disable
- **POST** /cluster/{cluster:id}/glusterhooks/{hook:id}/enable
- **POST** /cluster/{cluster:id}/glusterhooks/{hook:id}/resolve
- **POST** /集群/{cluster:id}/glustervolumes
- **GET** /cluster/{cluster:id}/glustervolumes
- **GET** /cluster/{cluster:id}/glustervolumes/{volume:id}
- **DELETE** /集群/{cluster:id}/glustervolumes/{volume:id}
- **POST** /cluster/{cluster:id}/glustervolumes/{volume:id}/getprofilestatistics
- **POST** /集群/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
- **GET** /cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
- **DELETE** /集群/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
- **POST** /cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/activate

- **POST** /cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/migrate
- **POST** /cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/stopmigrate
- **GET** /cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}
- **DELETE** /集群/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}
- **POST** /集群/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}/replace
- **GET** /cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}/statistics
- **GET**
/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}/statistics/{statistic:id}
- **POST** /集群/{cluster:id}/glustervolumes/{volume:id}/rebalance
- **POST** /cluster/{cluster:id}/glustervolumes/{volume:id}/resetalloptions
- **POST** /cluster/{cluster:id}/glustervolumes/{volume:id}/resetoption
- **POST** /cluster/{cluster:id}/glustervolumes/{volume:id}/setoption
- **POST** /集群/{cluster:id}/glustervolumes/{volume:id}/start
- **POST** /cluster/{cluster:id}/glustervolumes/{volume:id}/startprofile
- **GET** /cluster/{cluster:id}/glustervolumes/{volume:id}/statistics

- **GET** /cluster/{cluster:id}/glustervolumes/{volume:id}/statistics/{statistic:id}
- **POST** /集群/{cluster:id}/glustervolumes/{volume:id}/stop
- **POST** /cluster/{cluster:id}/glustervolumes/{volume:id}/stopprofile
- **POST** /集群/{cluster:id}/glustervolumes/{volume:id}/stoprebalance
- **GET** /cluster/{cluster:id}/networkfilters
- **GET** /cluster/{cluster:id}/networkfilters/{networkfilter:id}
- **POST** /集群/{cluster:id}/网络
- **GET** /cluster/{cluster:id}/network
- **GET** /cluster/{cluster:id}/network/{network:id}
- **DELETE** /集群/{cluster:id}/network/{network:id}
- **PUT** /集群/{cluster:id}/network/{network:id}
- **POST** /集群/{cluster:id}/权限
- **GET** /cluster/{cluster:id}/permissions
- **GET** /cluster/{cluster:id}/permissions/{permission:id}

- **DELETE** /集群/{cluster:id}/permissions/{permission:id}
- **POST** /集群/{cluster:id}/refreshglusterhealstatus
- **POST** /集群/{cluster:id}/resetemulatedmachine
- **POST** /集群/{cluster:id}/syncallnetworks
- **POST** /cluster/{cluster:id}/upgrade
- **POST** /cpuprofiles
- **GET** /cpuprofiles
- **GET** /cpuprofiles/{profile:id}
- **PUT** /cpuprofiles/{profile:id}
- **DELETE** /cpuprofiles/{profile:id}
- **POST** /cpuprofiles/{profile:id}/permissions
- **GET** /cpuprofiles/{profile:id}/permissions
- **GET** /cpuprofiles/{profile:id}/permissions/{permission:id}
- **DELETE** /cpuprofiles/{profile:id}/permissions/{permission:id}

- **POST/数据中心**
- **GET /datacenters**
- **GET /datacenters/{datacenter:id}**
- **PUT /datacenters/{datacenter:id}**
- **DELETE /datacenters/{datacenter:id}**
- **POST /datacenters/{datacenter:id}/cleanfinishedtasks**
- **POST /datacenters/{datacenter:id}/cluster**
- **GET /datacenters/{datacenter:id}/cluster**
- **GET /datacenters/{datacenter:id}/cluster/{cluster:id}**
- **PUT /datacenters/{datacenter:id}/cluster/{cluster:id}**
- **DELETE /datacenters/{datacenter:id}/集群/{cluster:id}**
- **POST /datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups**
- **GET /datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups**
- **GET /datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}**

- **PUT** /datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}
- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/affinitygroups/{group:id}
- **POST**
/datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}/hostlabels
- **GET**
/datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}/hostlabels
- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/affinitygroups/{group:id}/hostlabels/{label:id}
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}/hosts
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}/hosts
- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/affinitygroups/{group:id}/hosts/{host:idname}
- **POST**
/datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}/vmlabels
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}/vmlabels
- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/affinitygroups/{group:id}/vmlabels/{label:id}
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}/vms
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/affinitygroups/{group:id}/vms

- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/affinitygroups/{group:id}/vms/{vm:id}
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/cpuprofiles
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/cpuprofiles
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/cpuprofiles/{profile:id}
- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/cpuprofiles/{profile:id}
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/enabledfeatures
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/enabledfeatures
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/enabledfeature{id}
- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/enabledfeature{id}
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/externalnetworkproviders
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/glusterhooks
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/glusterhooks/{hook:id}
- **DELETE** /datacenters/{datacenter:id}/cluster/{cluster:id}/glusterhooks/{hook:id}
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/glusterhooks/{hook:id}/disable

- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/glusterhooks/{hook:id}/enable
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/glusterhooks/{hook:id}/resolve
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}
- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/glustervolumes/{volume:id}
- **POST**
/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/getprofilestatistics
- **POST**
/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
- **GET**
/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
- **DELETE**
/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
- **POST**
/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/activate
- **POST**
/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/migrate

- **POST**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/stopmigrate`
- **GET**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}`
- **DELETE**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}`
- **POST**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}/replace`
- **GET**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}/statistics`
- **GET**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}/statistics/{statistic:id}`
- **POST**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/rebalance`
- **POST**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/resetallocations`
- **POST**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/resetoption`
- **POST**
`/datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/setoption`

- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/start
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/startprofile
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/statistics
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/statistics/{statistic:id}
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/stop
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/stopprofile
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/glustervolumes/{volume:id}/stoprebalance
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/networkfilters
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/networkfilters/{networkfilter:id}
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/network
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/network
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/network/{network:id}

- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/网络/{network:id}
- **PUT** /datacenters/{datacenter:id}/cluster/{cluster:id}/network/{network:id}
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/权限
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/权限
- **GET** /datacenters/{datacenter:id}/cluster/{cluster:id}/permissions/{permission:id}
- **DELETE** /datacenters/{datacenter:id}/集群/{cluster:id}/permissions/{permission:id}
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/refreshglusterhealstatus
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/resetemulatedmachine
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/syncallnetworks
- **POST** /datacenters/{datacenter:id}/cluster/{cluster:id}/upgrade
- **POST** /datacenters/{datacenter:id}/iscsibonds
- **GET** /datacenters/{datacenter:id}/iscsibonds
- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}
- **PUT** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}

- **DELETE** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}
- **POST** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network
- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network
- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}
- **PUT** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}
- **DELETE** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/网络/{network:id}
- **POST**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/networklabels
- **GET**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/networklabels
- **GET**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/networklabels/{label:id}
- **DELETE**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/networklabels/{label:id}
- **POST**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/permissions
- **GET**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/permissions

- **GET**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/permissions/{permission:id}`
- **DELETE**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/permissions/{permission:id}`
- **POST**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/vnicprofiles`
- **GET**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/vnicprofiles`
- **GET**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/vnicprofiles/{profile:id}`
- **DELETE**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/vnicprofiles/{profile:id}`
- **POST**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/vnicprofiles/{profile:id}/permissions`
- **GET**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/vnicprofiles/{profile:id}/permissions`
- **GET**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/vnicprofiles/{profile:id}/permissions/{permission:id}`
- **DELETE**
`/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/network/{network:id}/vnicprofiles/{profile:id}/permissions/{permission:id}`

- **POST**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections
- **GET** **/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections**
- **GET**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections/{storage connection:id}
- **PUT**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections/{storage connection:id}
- **DELETE**
/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections/{storage connection:id}
- **POST** **/datacenters/{datacenter:id}/network**
- **GET** **/datacenters/{datacenter:id}/network**
- **GET** **/datacenters/{datacenter:id}/network/{network:id}**
- **DELETE** **/datacenters/{datacenter:id}/network/{network:id}**
- **PUT** **/datacenters/{datacenter:id}/network/{network:id}**
- **POST** **/datacenters/{datacenter:id}/permissions**
- **GET** **/datacenters/{datacenter:id}/permissions**

- **GET** /datacenters/{datacenter:id}/permissions/{permission:id}
- **DELETE** /datacenters/{datacenter:id}/permissions/{permission:id}
- **POST** /datacenters/{datacenter:id}/qoss
- **GET** /datacenters/{datacenter:id}/qoss
- **GET** /datacenters/{datacenter:id}/qoss/{qos:id}
- **PUT** /datacenters/{datacenter:id}/qoss/{qos:id}
- **DELETE** /datacenters/{datacenter:id}/qoss/{qos:id}
- **POST** /datacenters/{datacenter:id}/quota
- **GET** /datacenters/{datacenter:id}/quota
- **GET** /datacenters/{datacenter:id}/quota/{quota:id}
- **PUT** /datacenters/{datacenter:id}/quota/{quota:id}
- **DELETE** /datacenters/{datacenter:id}/quota/{quota:id}
- **POST** /datacenters/{datacenter:id}/quota/{quota:id}/权限
- **GET** /datacenters/{datacenter:id}/quota/{quota:id}/权限

- **GET** /datacenters/{datacenter:id}/quota/{quota:id}/permissions/{permission:id}
- **DELETE** /datacenters/{datacenter:id}/quota/{quota:id}/permissions/{permission:id}
- **POST** /datacenters/{datacenter:id}/quota/{quota:id}/quotaclusterlimits
- **GET** /datacenters/{datacenter:id}/quota/{quota:id}/quotaclusterlimits
- **GET** /datacenters/{datacenter:id}/quota/{quota:id}/quotaclusterlimits/{limit:id}
- **DELETE** /datacenters/{datacenter:id}/quota/{quota:id}/quotaclusterlimits/{limit:id}
- **POST** /datacenters/{datacenter:id}/quota/{quota:id}/quotastorageimits
- **GET** /datacenters/{datacenter:id}/quota/{quota:id}/quotastorageimits
- **GET** /datacenters/{datacenter:id}/quota/{quota:id}/quotastorageimits/{limit:id}
- **DELETE** /datacenters/{datacenter:id}/quota/{quota:id}/quotastorageimits/{limit:id}
- **POST** /datacenters/{datacenter:id}/setmaster
- **POST** /datacenters/{datacenter:id}/storagedomains
- **GET** /datacenters/{datacenter:id}/storagedomains
- **GET** /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}

- **DELETE** /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}
- **POST** /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/activate
- **POST** /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/deactivate
- **POST** /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/磁盘
- **GET** /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/磁盘
- **PUT** /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}
- **GET** /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}
- **DELETE**
/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}
- **POST**
/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/copy
- **POST**
/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/export
- **POST**
/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/move
- **POST**
/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/permissions
- **GET**
/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/permissions

ns

- **GET**
`/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/permissions/{permission:id}`
- **DELETE**
`/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/permissions/{permission:id}`
- **POST**
`/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/register`
- **POST**
`/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/sparsify`
- **GET**
`/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/statistics`
- **GET**
`/datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disk/{disk:id}/statistics/{statistic:id}`
- **POST** /diskprofiles
- **GET** /diskprofiles
- **GET** /diskprofiles/{diskprofile:id}
- **PUT** /diskprofiles/{diskprofile:id}
- **DELETE** /diskprofiles/{diskprofile:id}

- **POST** /diskprofiles/{diskprofile:id}/permissions
- **GET** /diskprofiles/{diskprofile:id}/permissions
- **GET** /diskprofiles/{diskprofile:id}/permissions/{permission:id}
- **DELETE** /diskprofiles/{diskprofile:id}/permissions/{permission:id}
- **POST** /磁盘
- **GET** /磁盘
- **PUT** /磁盘/{disk:id}
- **GET** /disk/{disk:id}
- **DELETE** /磁盘/{disk:id}
- **POST** /磁盘/{disk:id}/convert
- **POST** /磁盘/{disk:id}/copy
- **GET** /disk/{disk:id}/diskshots
- **GET** /disk/{disk:id}/diskshots/{snapshot:id}
- **DELETE** /disk/{disk:id}/diskshots/{snapshot:id}

- **POST /磁盘/{disk:id}/export**
- **POST /磁盘/{disk:id}/move**
- **POST /磁盘/{disk:id}/权限**
- **GET /disk/{disk:id}/permissions**
- **GET /disk/{disk:id}/permissions/{permission:id}**
- **DELETE /disk/{disk:id}/permissions/{permission:id}**
- **POST /磁盘/{disk:id}/reduce**
- **POST /磁盘/{disk:id}/refreshlun**
- **POST /磁盘/{disk:id}/sparsify**
- **GET /disk/{disk:id}/statistics**
- **GET /disk/{disk:id}/statistics/{statistic:id}**
- **GET /domain**
- **GET /domain/{domain:id}**
- **GET /domain/{domain:id}/groups**

- **GET /domain/{domain:id}/groups/{group:id}**
- **GET /domain/{domain:id}/用户**
- **GET /domain/{domain:id}/用户/{user:id}**
- **POST/事件**
- **GET/事件**
- **POST /事件/取消删除**
- **GET /events/{event:id}**
- **DELETE /事件/{event:id}**
- **POST /externalhostproviders**
- **GET /externalhostproviders**
- **GET /externalhostproviders/{provider:id}**
- **PUT /externalhostproviders/{provider:id}**
- **DELETE /externalhostproviders/{provider:id}**
- **GET /externalhostproviders/{provider:id}/certificate**

- **GET** /externalhostproviders/{provider:id}/certificate/{certificate:id}
- **GET** /externalhostproviders/{provider:id}/computeresources
- **GET** /externalhostproviders/{provider:id}/computeresources/{resource:id}
- **GET** /externalhostproviders/{provider:id}/discoveredhosts
- **GET** /externalhostproviders/{provider:id}/discoveredhosts/{host:id}
- **GET** /externalhostproviders/{provider:id}/hostgroups
- **GET** /externalhostproviders/{provider:id}/hostgroups/{group:id}
- **GET** /externalhostproviders/{provider:id}/hosts
- **GET** /externalhostproviders/{provider:id}/hosts/{host:id}
- **POST** /externalhostproviders/{provider:id}/importcertificates
- **POST** /externalhostproviders/{provider:id}/testconnectivity
- **POST** /externaltemplateimports
- **POST** /externalvmimports
- **POST**/组

- **GET /groups**
- **GET /groups/{group:id}**
- **DELETE /groups/{group:id}**
- **POST /groups/{group:id}/权限**
- **GET /groups/{group:id}/权限**
- **GET /groups/{group:id}/permissions/{permission:id}**
- **DELETE /groups/{group:id}/permissions/{permission:id}**
- **GET /groups/{group:id}/roles**
- **GET /groups/{group:id}/roles/{role:id}**
- **DELETE /groups/{group:id}/roles/{role:id}**
- **PUT /groups/{group:id}/roles/{role:id}**
- **POST /groups/{group:id}/roles/{role:id}/allow**
- **GET /groups/{group:id}/roles/{role:id}/allow**
- **GET /groups/{group:id}/roles/{role:id}/allow/{permit:id}**

- **DELETE** /groups/{group:id}/roles/{role:id}/allow/{permit:id}
- **POST** /groups/{group:id}/tags
- **GET** /groups/{group:id}/tags
- **GET** /groups/{group:id}/tags/{tag:id}
- **DELETE** /groups/{group:id}/tags/{tag:id}
- **POST**/主机
- **GET** /hosts
- **GET** /hosts/{host:id}
- **PUT** /hosts/{host:id}
- **DELETE** /hosts/{host:id}
- **POST** /hosts/{host:id}/activate
- **POST** /hosts/{host:id}/affinitylabels
- **GET** /hosts/{host:id}/affinitylabels
- **GET** /hosts/{host:id}/affinitylabels/{label:id}

- **DELETE** /hosts/{host:id}/affinitylabels/{label:id}
- **POST** /hosts/{host:id}/批准
- **POST** /hosts/{host:id}/commitnetconfig
- **POST** /hosts/{host:id}/copyhostnetworks
- **GET** /hosts/{host:id}/cpuunits
- **POST** /hosts/{host:id}/deactivate
- **GET** /hosts/{host:id}/devices
- **GET** /hosts/{host:id}/devices/{device:id}
- **POST** /hosts/{host:id}/discoveriscsi
- **POST** /hosts/{host:id}/registrationcertificate
- **GET** /hosts/{host:id}/externalnetworkproviderconfigurations
- **GET** /hosts/{host:id}/externalnetworkproviderconfigurations/{configuration:id}
- **POST** /hosts/{host:id}/fence
- **POST** /hosts/{host:id}/fenceagents

- [GET /hosts/{host:id}/fenceagents](#)
- [GET /hosts/{host:id}/fenceagents/{agent:id}](#)
- [PUT /hosts/{host:id}/fenceagents/{agent:id}](#)
- [DELETE /hosts/{host:id}/fenceagents/{agent:id}](#)
- [POST /hosts/{host:id}/forceelectspm](#)
- [GET /hosts/{host:id}/hook](#)
- [GET /hosts/{host:id}/hooks/{hook:id}](#)
- [POST /hosts/{host:id}/install](#)
- [POST /hosts/{host:id}/iscsidiscover](#)
- [POST /hosts/{host:id}/iscsilogin](#)
- [GET /hosts/{host:id}/katelloerrata](#)
- [GET /hosts/{host:id}/katelloerrata/{katelloerratum:id}](#)
- [POST /hosts/{host:id}/网络连接](#)
- [GET /hosts/{host:id}/网络连接](#)

- **GET /hosts/{host:id}/networkattachments/{attachment:id}**
- **PUT /hosts/{host:id}/networkattachments/{attachment:id}**
- **DELETE /hosts/{host:id}/网络连接/{attachment:id}**
- **GET /hosts/{host:id}/nics**
- **GET /hosts/{host:id}/nics/{nic:id}**
- **GET /hosts/{host:id}/nics/{nic:id}/linklayerdiscoveryprotocolements**
- **POST /hosts/{host:id}/nics/{nic:id}/networkattachments**
- **GET /hosts/{host:id}/nics/{nic:id}/networkattachments**
- **GET /hosts/{host:id}/nics/{nic:id}/networkattachments/{attachment:id}**
- **PUT /hosts/{host:id}/nics/{nic:id}/networkattachments/{attachment:id}**
- **DELETE /hosts/{host:id}/nics/{nic:id}/networkattachments/{attachment:id}**
- **POST /hosts/{host:id}/nics/{nic:id}/networklabels**
- **GET /hosts/{host:id}/nics/{nic:id}/networklabels**
- **GET /hosts/{host:id}/nics/{nic:id}/networklabels/{label:id}**

- **DELETE** /hosts/{host:id}/nics/{nic:id}/networklabels/{label:id}
- **GET** /hosts/{host:id}/nics/{nic:id}/statistics
- **GET** /hosts/{host:id}/nics/{nic:id}/statistics/{statistic:id}
- **POST** /hosts/{host:id}/nics/{nic:id}/updatevirtualfunctionsconfiguration
- **POST** /hosts/{host:id}/nics/{nic:id}/virtualFeaturelowedlabels
- **GET** /hosts/{host:id}/nics/{nic:id}/virtualFeaturelowedlabels
- **GET** /hosts/{host:id}/nics/{nic:id}/virtualFeaturelowedlabels/{label:id}
- **DELETE** /hosts/{host:id}/nics/{nic:id}/virtualFeaturelowedlabels/{label:id}
- **POST** /hosts/{host:id}/nics/{nic:id}/virtualFeaturelowednetworks
- **GET** /hosts/{host:id}/nics/{nic:id}/virtualFeaturelowednetworks
- **GET** /hosts/{host:id}/nics/{nic:id}/virtualFeaturelowednetworks/{network:id}
- **DELETE** /hosts/{host:id}/nics/{nic:id}/virtualFeaturelowednetworks/{network:id}
- **GET** /hosts/{host:id}/numanodes
- **GET** /hosts/{host:id}/numanodes/{node:id}

- [GET /hosts/{host:id}/numanodes/{node:id}/statistics](#)
- [GET /hosts/{host:id}/numanodes/{node:id}/statistics/{statistic:id}](#)
- [POST /hosts/{host:id}/权限](#)
- [GET /hosts/{host:id}/permissions](#)
- [GET /hosts/{host:id}/permissions/{permission:id}](#)
- [DELETE /hosts/{host:id}/permissions/{permission:id}](#)
- [POST /hosts/{host:id}/refresh](#)
- [POST /hosts/{host:id}/setupnetworks](#)
- [GET /hosts/{host:id}/statistics](#)
- [GET /hosts/{host:id}/statistics/{statistic:id}](#)
- [GET /hosts/{host:id}/storage](#)
- [GET /hosts/{host:id}/storage/{storage:id}](#)
- [POST /hosts/{host:id}/storageconnectionextensions](#)
- [GET /hosts/{host:id}/storageconnectionextensions](#)

- **GET** /hosts/{host:id}/storageconnectionextensions/{storageconnectionextension:id}
- **PUT** /hosts/{host:id}/storageconnectionextensions/{storageconnectionextension:id}
- **DELETE**
/hosts/{host:id}/storageconnectionextensions/{storageconnectionextension:id}
- **POST** /hosts/{host:id}/syncallnetworks
- **POST** /hosts/{host:id}/tags
- **GET** /hosts/{host:id}/tags
- **GET** /hosts/{host:id}/tags/{tag:id}
- **DELETE** /hosts/{host:id}/tags/{tag:id}
- **GET** /hosts/{host:id}/非受管网络
- **GET** /hosts/{host:id}/非受管网络/{unmanagednetwork:id}
- **DELETE** /hosts/{host:id}/非受管网络/{unmanagednetwork:id}
- **POST** /hosts/{host:id}/unregisteredstoragedomainsdiscover
- **POST** /hosts/{host:id}/upgrade
- **POST** /hosts/{host:id}/upgradecheck

- [GET /图标](#)
- [GET /图标/{icon:id}](#)
- [POST/imagetransfers](#)
- [GET /imagetransfers](#)
- [GET /imagetransfers/{imagetransfer:id}](#)
- [POST /imagetransfers/{imagetransfer:id}/cancel](#)
- [POST /imagetransfers/{imagetransfer:id}/Extended](#)
- [POST /imagetransfers/{imagetransfer:id}/finalize](#)
- [POST /imagetransfers/{imagetransfer:id}/pause](#)
- [POST /imagetransfers/{imagetransfer:id}/restore](#)
- [POST/instancetypes](#)
- [GET /instancetypes](#)
- [GET /instancetypes/{instancetype:id}](#)
- [PUT /instancetypes/{instancetype:id}](#)

- **DELETE** /instancetypes/{instancetype:id}
- **POST** /instancetypes/{instancetype:id}/graphicsconsoles
- **GET** /instancetypes/{instancetype:id}/graphicsconsoles
- **GET** /instancetypes/{instancetype:id}/graphicsconsoles/{console:id}
- **DELETE** /instancetypes/{instancetype:id}/graphicsconsoles/{console:id}
- **POST** /instancetypes/{instancetype:id}/nics
- **GET** /instancetypes/{instancetype:id}/nics
- **GET** /instancetypes/{instancetype:id}/nics/{nic:id}
- **PUT** /instancetypes/{instancetype:id}/nics/{nic:id}
- **DELETE** /instancetypes/{instancetype:id}/nics/{nic:id}
- **POST** /instancetypes/{instancetype:id}/watchdogs
- **GET** /instancetypes/{instancetype:id}/watchdogs
- **GET** /instancetypes/{instancetype:id}/watchdogs/{watchdog:id}
- **PUT** /instancetypes/{instancetype:id}/watchdogs/{watchdog:id}

- **DELETE** /instancetypes/{instancetype:id}/watchdogs/{watchdog:id}
- **POST** /任务
- **GET** /Job
- **GET** /jobs/{job:id}
- **POST** /jobs/{job:id}/clear
- **POST** /job/{job:id}/end
- **POST** /job/{job:id}/steps
- **GET** /jobs/{job:id}/steps
- **GET** /jobs/{job:id}/steps/{step:id}
- **POST** /job/{job:id}/steps/{step:id}/end
- **GET** /jobs/{job:id}/steps/{step:id}/statistics
- **GET** /jobs/{job:id}/steps/{step:id}/statistics/{statistic:id}
- **GET** /katelloerrata
- **GET** /katelloerrata/{katelloerratum:id}

- **POST /macpools**
- **GET /macpools**
- **GET /macpools/{macpool:id}**
- **PUT /macpools/{macpool:id}**
- **DELETE /macpools/{macpool:id}**
- **POST /macpools/{macpool:id}/权限**
- **GET /macpools/{macpool:id}/permissions**
- **GET /macpools/{macpool:id}/permissions/{permission:id}**
- **DELETE /macpools/{macpool:id}/permissions/{permission:id}**
- **GET /networkfilters**
- **GET /networkfilters/{networkfilter:id}**
- **POST/网络**
- **GET /network**
- **GET /network/{network:id}**

- **PUT** /网络/{network:id}
- **DELETE** /网络/{network:id}
- **POST** /network/{network:id}/networklabels
- **GET** /network/{network:id}/networklabels
- **GET** /network/{network:id}/networklabels/{label:id}
- **DELETE** /network/{network:id}/networklabels/{label:id}
- **POST** /网络/{network:id}/权限
- **GET** /network/{network:id}/permissions
- **GET** /network/{network:id}/permissions/{permission:id}
- **DELETE** /网络/{network:id}/permissions/{permission:id}
- **POST** /network/{network:id}/vnicprofiles
- **GET** /network/{network:id}/vnicprofiles
- **GET** /network/{network:id}/vnicprofiles/{profile:id}
- **DELETE** /network/{network:id}/vnicprofiles/{profile:id}

- **POST** /network/{network:id}/vnicprofiles/{profile:id}/permissions
- **GET** /network/{network:id}/vnicprofiles/{profile:id}/permissions
- **GET** /network/{network:id}/vnicprofiles/{profile:id}/permissions/{permission:id}
- **DELETE** /network/{network:id}/vnicprofiles/{profile:id}/permissions/{permission:id}
- **POST**/openstackimageproviders
- **GET** /openstackimageproviders
- **GET** /openstackimageproviders/{provider:id}
- **PUT** /openstackimageproviders/{provider:id}
- **DELETE** /openstackimageproviders/{provider:id}
- **GET** /openstackimageproviders/{provider:id}/certificate
- **GET** /openstackimageproviders/{provider:id}/certificate/{certificate:id}
- **GET** /openstackimageproviders/{provider:id}/images
- **GET** /openstackimageproviders/{provider:id}/images/{image:id}
- **POST** /openstackimageproviders/{provider:id}/images/{image:id}/import

- **POST** /openstackimageproviders/{provider:id}/importcertificates
- **POST** /openstackimageproviders/{provider:id}/testconnectivity
- **POST** /openstacknetworkproviders
- **GET** /openstacknetworkproviders
- **GET** /openstacknetworkproviders/{provider:id}
- **PUT** /openstacknetworkproviders/{provider:id}
- **DELETE** /openstacknetworkproviders/{provider:id}
- **GET** /openstacknetworkproviders/{provider:id}/certificate
- **GET** /openstacknetworkproviders/{provider:id}/certificate/{certificate:id}
- **POST** /openstacknetworkproviders/{provider:id}/importcertificates
- **GET** /openstacknetworkproviders/{provider:id}/network
- **GET** /openstacknetworkproviders/{provider:id}/network/{network:id}
- **POST** /openstacknetworkproviders/{provider:id}/network/{network:id}/import
- **POST** /openstacknetworkproviders/{provider:id}/network/{network:id}/subnets

- **GET** /openstacknetworkproviders/{provider:id}/network/{network:id}/subnets
- **GET**
/openstacknetworkproviders/{provider:id}/network/{network:id}/subnets/{subnet:id}
- **DELETE**
/openstacknetworkproviders/{provider:id}/network/{network:id}/subnets/{subnet:id}
- **POST** /openstacknetworkproviders/{provider:id}/testconnectivity
- **POST** /openstackvolumeproviders
- **GET** /openstackvolumeproviders
- **GET** /openstackvolumeproviders/{provider:id}
- **PUT** /openstackvolumeproviders/{provider:id}
- **DELETE** /openstackvolumeproviders/{provider:id}
- **POST** /openstackvolumeproviders/{provider:id}/authenticationkeys
- **GET** /openstackvolumeproviders/{provider:id}/authenticationkeys
- **GET** /openstackvolumeproviders/{provider:id}/authenticationkeys/{key:id}
- **PUT** /openstackvolumeproviders/{provider:id}/authenticationkeys/{key:id}
- **DELETE** /openstackvolumeproviders/{provider:id}/authenticationkeys/{key:id}

- **GET** /openstackvolumeproviders/{provider:id}/certificate
- **GET** /openstackvolumeproviders/{provider:id}/certificate/{certificate:id}
- **POST** /openstackvolumeproviders/{provider:id}/importcertificates
- **POST** /openstackvolumeproviders/{provider:id}/testconnectivity
- **GET** /openstackvolumeproviders/{provider:id}/volumetypes
- **GET** /openstackvolumeproviders/{provider:id}/volumetypes/{type:id}
- **GET** /操作系统
- **GET** /操作系统/{operatingsystem:id}
- **GET** /options/{option:id}
- **POST** /权限
- **GET** /permissions
- **GET** /permissions/{permission:id}
- **DELETE** /permissions/{permission:id}
- **POST** /角色

- **GET/角色**
- **GET /roles/{role:id}**
- **DELETE /roles/{role:id}**
- **PUT /角色/{role:id}**
- **POST /角色/{role:id}/允许**
- **GET /roles/{role:id}/allow**
- **GET /roles/{role:id}/allow/{permit:id}**
- **DELETE /roles/{role:id}/allow/{permit:id}**
- **POST /schedulingpolicies**
- **GET /schedulingpolicies**
- **GET /schedulingpolicies/{policy:id}**
- **PUT /schedulingpolicies/{policy:id}**
- **DELETE /schedulingpolicies/{policy:id}**
- **POST /schedulingpolicies/{policy:id}/balance**

- [GET /schedulingpolicies/{policy:id}/balance](#)
- [GET /schedulingpolicies/{policy:id}/balance/{balance:id}](#)
- [DELETE /schedulingpolicies/{policy:id}/balance/{balance:id}](#)
- [POST /schedulingpolicies/{policy:id}/filter](#)
- [GET /schedulingpolicies/{policy:id}/filter](#)
- [GET /schedulingpolicies/{policy:id}/filter/{filter:id}](#)
- [DELETE /schedulingpolicies/{policy:id}/filter/{filter:id}](#)
- [POST /schedulingpolicies/{policy:id}/weights](#)
- [GET /schedulingpolicies/{policy:id}/weights](#)
- [GET /schedulingpolicies/{policy:id}/weights/{weight:id}](#)
- [DELETE /schedulingpolicies/{policy:id}/weights/{weight:id}](#)
- [GET /schedulingpolicyunits](#)
- [GET /schedulingpolicyunits/{unit:id}](#)
- [DELETE /schedulingpolicyunits/{unit:id}](#)

- [POST/存储连接](#)
- [GET /storageconnections](#)
- [GET /storageconnections/{storageconnection:id}](#)
- [PUT/存储连接/ {storageconnection:id}](#)
- [DELETE /storageconnections/{storageconnection:id}](#)
- [POST /storagedomains](#)
- [GET /storagedomains](#)
- [GET /storagedomains/{storagedomain:id}](#)
- [PUT /storagedomains/{storagedomain:id}](#)
- [DELETE /storagedomains/{storagedomain:id}](#)
- [POST /storagedomains/{storagedomain:id}/diskprofiles](#)
- [GET /storagedomains/{storagedomain:id}/diskprofiles](#)
- [GET /storagedomains/{storagedomain:id}/diskprofiles/{profile:id}](#)
- [DELETE /storagedomains/{storagedomain:id}/diskprofiles/{profile:id}](#)

- **POST** /storagedomains/{storagedomain:id}/磁盘
- **GET** /storagedomains/{storagedomain:id}/磁盘
- **PUT** /storagedomains/{storagedomain:id}/disk/{disk:id}
- **GET** /storagedomains/{storagedomain:id}/disk/{disk:id}
- **DELETE** /storagedomains/{storagedomain:id}/disk/{disk:id}
- **POST** /storagedomains/{storagedomain:id}/disk/{disk:id}/copy
- **POST** /storagedomains/{storagedomain:id}/disk/{disk:id}/export
- **POST** /storagedomains/{storagedomain:id}/disk/{disk:id}/move
- **POST** /storagedomains/{storagedomain:id}/disk/{disk:id}/permissions
- **GET** /storagedomains/{storagedomain:id}/disk/{disk:id}/permissions
- **GET** /storagedomains/{storagedomain:id}/disk/{disk:id}/permissions/{permission:id}
- **DELETE**
/storagedomains/{storagedomain:id}/disk/{disk:id}/permissions/{permission:id}
- **POST** /storagedomains/{storagedomain:id}/disk/{disk:id}/reduce
- **POST** /storagedomains/{storagedomain:id}/disk/{disk:id}/sparsify

- **GET** /storagedomains/{storagedomain:id}/disk/{disk:id}/statistics
- **GET** /storagedomains/{storagedomain:id}/disk/{disk:id}/statistics/{statistic:id}
- **GET** /storagedomains/{storagedomain:id}/disksnapshots
- **GET** /storagedomains/{storagedomain:id}/disksnapshots/{snapshot:id}
- **DELETE** /storagedomains/{storagedomain:id}/disksnapshots/{snapshot:id}
- **GET** /storagedomains/{storagedomain:id}/文件
- **GET** /storagedomains/{storagedomain:id}/files/{file:id}
- **GET** /storagedomains/{storagedomain:id}/images
- **GET** /storagedomains/{storagedomain:id}/images/{image:id}
- **POST** /storagedomains/{storagedomain:id}/images/{image:id}/import
- **POST** /storagedomains/{storagedomain:id}/isattached
- **POST** /storagedomains/{storagedomain:id}/权限
- **GET** /storagedomains/{storagedomain:id}/permissions
- **GET** /storagedomains/{storagedomain:id}/permissions/{permission:id}

- **DELETE** /storagedomains/{storagedomain:id}/permissions/{permission:id}
- **POST** /storagedomains/{storagedomain:id}/reduceluns
- **POST** /storagedomains/{storagedomain:id}/refreshluns
- **POST** /storagedomains/{storagedomain:id}/storageconnections
- **GET** /storagedomains/{storagedomain:id}/storageconnections
- **GET** /storagedomains/{storagedomain:id}/storageconnections/{connection:id}
- **DELETE** /storagedomains/{storagedomain:id}/storageconnections/{connection:id}
- **GET** /storagedomains/{storagedomain:id}/templates
- **GET** /storagedomains/{storagedomain:id}/templates/{template:id}
- **DELETE** /storagedomains/{storagedomain:id}/templates/{template:id}
- **GET** /storagedomains/{storagedomain:id}/templates/{template:id}/disk
- **GET** /storagedomains/{storagedomain:id}/templates/{template:id}/disk/{disk:id}
- **POST** /storagedomains/{storagedomain:id}/templates/{template:id}/import
- **POST** /storagedomains/{storagedomain:id}/templates/{template:id}/register

- **POST** /storagedomains/{storagedomain:id}/updateovfstore
- **GET** /storagedomains/{storagedomain:id}/vms
- **GET** /storagedomains/{storagedomain:id}/vms/{vm:id}
- **DELETE** /storagedomains/{storagedomain:id}/vms/{vm:id}
- **GET** /storagedomains/{storagedomain:id}/vms/{vm:id}/diskattachments
- **GET**
/storagedomains/{storagedomain:id}/vms/{vm:id}/diskattachments/{attachment:id}
- **GET** /storagedomains/{storagedomain:id}/vms/{vm:id}/磁盘
- **GET** /storagedomains/{storagedomain:id}/vms/{vm:id}/disk/{disk:id}
- **POST** /storagedomains/{storagedomain:id}/vms/{vm:id}/import
- **POST** /storagedomains/{storagedomain:id}/vms/{vm:id}/register
- **POST** /标签
- **GET** /tags
- **GET** /tags/{tag:id}
- **PUT** /tags/{tag:id}

- **DELETE** /tags/{tag:id}
- **POST**/模板
- **GET** /模板
- **GET** /templates/{template:id}
- **PUT** /templates/{template:id}
- **DELETE** /templates/{template:id}
- **GET** /templates/{template:id}/cdroms
- **GET** /templates/{template:id}/cdroms/{cdrom:id}
- **GET** /templates/{template:id}/diskattachments
- **GET** /templates/{template:id}/diskattachments/{attachment:id}
- **DELETE** /templates/{template:id}/diskattachments/{attachment:id}
- **POST** /templates/{template:id}/export
- **POST** /templates/{template:id}/graphicsconsoles
- **GET** /templates/{template:id}/graphicsconsoles

- **GET** /templates/{template:id}/graphicsconsoles/{console:id}
- **DELETE** /templates/{template:id}/graphicsconsoles/{console:id}
- **POST** /templates/{template:id}/mediateddevices
- **GET** /templates/{template:id}/mediateddevices
- **GET** /templates/{template:id}/mediateddevices/{device:id}
- **PUT** /templates/{template:id}/mediateddevices/{device:id}
- **DELETE** /templates/{template:id}/mediateddevices/{device:id}
- **POST** /templates/{template:id}/nics
- **GET** /templates/{template:id}/nics
- **GET** /templates/{template:id}/nics/{nic:id}
- **PUT** /templates/{template:id}/nics/{nic:id}
- **DELETE** /templates/{template:id}/nics/{nic:id}
- **POST** /templates/{template:id}/权限
- **GET** /templates/{template:id}/permissions

- **GET** /templates/{template:id}/permissions/{permission:id}
- **DELETE** /templates/{template:id}/permissions/{permission:id}
- **POST** /templates/{template:id}/tags
- **GET** /templates/{template:id}/tags
- **GET** /templates/{template:id}/tags/{tag:id}
- **DELETE** /templates/{template:id}/tags/{tag:id}
- **POST** /templates/{template:id}/watchdogs
- **GET** /templates/{template:id}/watchdogs
- **GET** /templates/{template:id}/watchdogs/{watchdog:id}
- **PUT** /templates/{template:id}/watchdogs/{watchdog:id}
- **DELETE** /templates/{template:id}/watchdogs/{watchdog:id}
- **POST**/用户
- **GET**/用户
- **GET** /users/{user:id}

- **PUT** /用户 /{user:id}
- **DELETE** /用户/{user:id}
- **POST** /users/{user:id}/eventsubscriptions
- **GET** /users/{user:id}/eventsubscriptions
- **GET** /users/{user:id}/eventsubscriptions/{eventsubscription:id}
- **DELETE** /用户/{user:id}/eventsubscriptions/{eventsubscription:id}
- **GET** /users/{user:id}/groups
- **POST** /users/{user:id}/选项
- **GET** /users/{user:id}/options
- **GET** /users/{user:id}/options/{option:id}
- **DELETE** /用户/{user:id}/options/{option:id}
- **POST** /用户/{user:id}/权限
- **GET** /users/{user:id}/permissions
- **GET** /users/{user:id}/permissions/{permission:id}

- **DELETE** /用户/{user:id}/permissions/{permission:id}
- **GET** /users/{user:id}/roles
- **GET** /users/{user:id}/roles/{role:id}
- **DELETE** /用户/{user:id}/roles/{role:id}
- **PUT** /用户/{user:id}/roles/{role:id}
- **POST** /users/{user:id}/roles/{role:id}/allow
- **GET** /users/{user:id}/roles/{role:id}/allow
- **GET** /users/{user:id}/roles/{role:id}/allow/{permit:id}
- **DELETE** /用户/{user:id}/roles/{role:id}/allow/{permit:id}
- **POST** /users/{user:id}/sshpublickeys
- **GET** /users/{user:id}/sshpublickeys
- **GET** /users/{user:id}/sshpublickeys/{key:id}
- **PUT** /用户/{user:id}/sshpublickeys/{key:id}
- **DELETE** /用户/{user:id}/sshpublickeys/{key:id}

- **POST** /users/{user:id}/tags
- **GET** /users/{user:id}/tags
- **GET** /users/{user:id}/tags/{tag:id}
- **DELETE** /用户/{user:id}/tags/{tag:id}
- **POST** /vmpools
- **GET** /vmpools
- **GET** /vmpools/{pool:id}
- **PUT** /vmpools/{pool:id}
- **DELETE** /vmpools/{pool:id}
- **POST** /vmpools/{pool:id}/allocatevm
- **POST** /vmpools/{pool:id}/权限
- **GET** /vmpools/{pool:id}/permissions
- **GET** /vmpools/{pool:id}/permissions/{permission:id}
- **DELETE** /vmpools/{pool:id}/permissions/{permission:id}

- **POST /vms**
- **GET /vms**
- **GET /vms/{vm:id}**
- **PUT /vms/{vm:id}**
- **DELETE /vms/{vm:id}**
- **POST /vms/{vm:id}/affinitylabels**
- **GET /vms/{vm:id}/affinitylabels**
- **GET /vms/{vm:id}/affinitylabels/{label:id}**
- **DELETE /vms/{vm:id}/affinitylabels/{label:id}**
- **GET /vms/{vm:id}/应用程序**
- **GET /vms/{vm:id}/applications/{application:id}**
- **POST /vms/{vm:id}/autopincpuandnumanodes**
- **POST /vms/{vm:id}/backups**
- **GET /vms/{vm:id}/backups**

- **GET /vms/{vm:id}/backups/{backup:id}**
- **GET /vms/{vm:id}/backups/{backup:id}/disks**
- **GET /vms/{vm:id}/backups/{backup:id}/disk/{disk:id}**
- **POST /vms/{vm:id}/backups/{backup:id}/finalize**
- **POST /vms/{vm:id}/cancelmigration**
- **POST /vms/{vm:id}/cdroms**
- **GET /vms/{vm:id}/cdroms**
- **GET /vms/{vm:id}/cdroms/{cdrom:id}**
- **PUT /vms/{vm:id}/cdroms/{cdrom:id}**
- **GET /vms/{vm:id}/checkpoints**
- **GET /vms/{vm:id}/checkpoints/{checkpoint:id}**
- **DELETE /vms/{vm:id}/checkpoints/{checkpoint:id}**
- **GET /vms/{vm:id}/checkpoints/{checkpoint:id}/磁盘**
- **GET /vms/{vm:id}/checkpoints/{checkpoint:id}/disk/{disk:id}**

- **POST** /vms/{vm:id}/clone
- **POST** /vms/{vm:id}/commitsnapshot
- **POST** /vms/{vm:id}/detach
- **POST** /vms/{vm:id}/diskattachments
- **GET** /vms/{vm:id}/diskattachments
- **GET** /vms/{vm:id}/diskattachments/{attachment:id}
- **DELETE** /vms/{vm:id}/磁盘附件/{attachment:id}
- **PUT** /vms/{vm:id}/diskattachments/{attachment:id}
- **POST** /vms/{vm:id}/export
- **POST** /vms/{vm:id}/freezefilesystems
- **POST** /vms/{vm:id}/graphicsconsoles
- **GET** /vms/{vm:id}/graphicsconsoles
- **GET** /vms/{vm:id}/graphicsconsoles/{console:id}
- **DELETE** /vms/{vm:id}/graphicsconsoles/{console:id}

- **POST** /vms/{vm:id}/graphicsconsoles/{console:id}/proxyticket
- **POST** /vms/{vm:id}/graphicsconsoles/{console:id}/remoteviewerconnectionfile
- **POST** /vms/{vm:id}/graphicsconsoles/{console:id}/ticket
- **POST** /vms/{vm:id}/hostdevices
- **GET** /vms/{vm:id}/hostdevices
- **GET** /vms/{vm:id}/hostdevices/{device:id}
- **DELETE** /vms/{vm:id}/hostdevices/{device:id}
- **GET** /vms/{vm:id}/katelloerrata
- **GET** /vms/{vm:id}/katelloerrata/{katelloerratum:id}
- **POST** /vms/{vm:id}/logon
- **POST** /vms/{vm:id}/maintenance
- **POST** /vms/{vm:id}/mediateddevices
- **GET** /vms/{vm:id}/mediateddevices
- **GET** /vms/{vm:id}/mediateddevices/{device:id}

- **PUT** /vms/{vm:id}/mediateddevices/{device:id}
- **DELETE** /vms/{vm:id}/mediateddevices/{device:id}
- **POST** /vms/{vm:id}/migrate
- **POST** /vms/{vm:id}/nics
- **GET** /vms/{vm:id}/nics
- **GET** /vms/{vm:id}/nics/{nic:id}
- **PUT** /vms/{vm:id}/nics/{nic:id}
- **DELETE** /vms/{vm:id}/nics/{nic:id}
- **POST** /vms/{vm:id}/nics/{nic:id}/activate
- **POST** /vms/{vm:id}/nics/{nic:id}/deactivate
- **GET** /vms/{vm:id}/nics/{nic:id}/networkfilterparameters
- **POST** /vms/{vm:id}/nics/{nic:id}/networkfilterparameters
- **GET** /vms/{vm:id}/nics/{nic:id}/networkfilterparameters/{parameter:id}
- **PUT** /vms/{vm:id}/nics/{nic:id}/networkfilterparameters/{parameter:id}

- **DELETE** /vms/{vm:id}/nics/{nic:id}/networkfilterparameters/{parameter:id}
- **GET** /vms/{vm:id}/nics/{nic:id}/reporteddevices
- **GET** /vms/{vm:id}/nics/{nic:id}/reportdevices/{reporteddevice:id}
- **GET** /vms/{vm:id}/nics/{nic:id}/statistics
- **GET** /vms/{vm:id}/nics/{nic:id}/statistics/{statistic:id}
- **POST** /vms/{vm:id}/numanodes
- **GET** /vms/{vm:id}/numanodes
- **GET** /vms/{vm:id}/numanodes/{node:id}
- **PUT** /vms/{vm:id}/numanodes/{node:id}
- **DELETE** /vms/{vm:id}/numanodes/{node:id}
- **POST** /vms/{vm:id}/权限
- **GET** /vms/{vm:id}/permissions
- **GET** /vms/{vm:id}/permissions/{permission:id}
- **DELETE** /vms/{vm:id}/permissions/{permission:id}

- **POST** /vms/{vm:id}/previewsnapshot
- **POST** /vms/{vm:id}/reboot
- **POST** /vms/{vm:id}/reordermacaddresses
- **GET** /vms/{vm:id}/reporteddevices
- **GET** /vms/{vm:id}/reporteddevices/{reporteddevice:id}
- **POST** /vms/{vm:id}/reset
- **POST** /vms/{vm:id}/screenshot
- **GET** /vms/{vm:id}/sessions
- **GET** /vms/{vm:id}/sessions/{session:id}
- **POST** /vms/{vm:id}/shutdown
- **POST** /vms/{vm:id}/快照
- **GET** /vms/{vm:id}/快照
- **GET** /vms/{vm:id}/快照/{snapshot:id}
- **DELETE** /vms/{vm:id}/快照/{snapshot:id}

- **GET /vms/{vm:id}/snapshot/{snapshot:id}/cdroms**
- **GET /vms/{vm:id}/快照/{snapshot:id}/cdroms/{cdrom:id}**
- **GET /vms/{vm:id}/snapshot/{snapshot:id}/disks**
- **GET /vms/{vm:id}/快照/{snapshot:id}/disk/{disk:id}**
- **GET /vms/{vm:id}/snapshot/{snapshot:id}/nics**
- **GET /vms/{vm:id}/snapshot/{snapshot:id}/nics/{nic:id}**
- **POST /vms/{vm:id}/snapshot/{snapshot:id}/restore**
- **POST /vms/{vm:id}/start**
- **GET /vms/{vm:id}/statistics**
- **GET /vms/{vm:id}/statistics/{statistic:id}**
- **POST /vms/{vm:id}/stop**
- **POST /vms/{vm:id}/suspend**
- **POST /vms/{vm:id}/tags**
- **GET /vms/{vm:id}/tags**

- **GET** /vms/{vm:id}/tags/{tag:id}
- **DELETE** /vms/{vm:id}/tags/{tag:id}
- **POST** /vms/{vm:id}/thawfilesystems
- **POST** /vms/{vm:id}/ticket
- **POST** /vms/{vm:id}/undosnapshot
- **POST** /vms/{vm:id}/watchdogs
- **GET** /vms/{vm:id}/watchdogs
- **GET** /vms/{vm:id}/watchdogs/{watchdog:id}
- **PUT** /vms/{vm:id}/watchdogs/{watchdog:id}
- **DELETE** /vms/{vm:id}/watchdogs/{watchdog:id}
- **POST** /vnicprofiles
- **GET** /vnicprofiles
- **GET** /vnicprofiles/{profile:id}
- **PUT** /vnicprofiles/{profile:id}

- **DELETE** /vnicprofiles/{profile:id}
- **POST** /vnicprofiles/{profile:id}/permissions
- **GET** /vnicprofiles/{profile:id}/permissions
- **GET** /vnicprofiles/{profile:id}/permissions/{permission:id}
- **DELETE** /vnicprofiles/{profile:id}/permissions/{permission:id}

第 6 章 服务

本节枚举 API 中提供的所有服务。

6.1. AFFINITYGROUP

此服务管理单个关联性组。

表 6.1. 方法概述

名称	概述
get	检索关联性组详细信息。
remove	移除 affinity 组。
update	更新 affinity 组。

6.1.1. 获取 GET

检索关联性组详细信息。

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

表 6.2. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
group	AffinityGroup	out	关联性组。

6.1.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.1.2. 删除 DELETE

移除 affinity 组。

DELETE /ovirt-engine/api/clusters/000-000/affinitygroups/123-456

表 6.3. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.1.3. 更新 PUT

更新 affinity 组。

表 6.4. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
group	AffinityGroup	in/Out	关联性组。

6.2. AFFINITYGROUPHOST

此服务管理单个主机与关联性组分配。

表 6.5. 方法概述

名称	概述
remove	从 affinity 组移除主机。

6.2.1. 删除 DELETE

从 affinity 组移除主机。

表 6.6. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.3. AFFINITYGROUPHOSTLABEL

此服务管理分配至关联性组的单一主机标签。

表 6.7. 方法概述

名称	概述
remove	从 affinity 组移除此标签。

6.3.1. 删除 DELETE

从 affinity 组移除此标签。

表 6.8. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.4. AFFINITYGROUPHOSTLABELS

此服务管理分配到关联性组的所有主机标签的集合。

表 6.9. 方法概述

名称	概述
add	将主机标签添加到 affinity 组。
list	列出分配到此关联性组的所有主机标签。

6.4.1. 添加 POST

将主机标签添加到 affinity 组。

例如，要将 label 789 添加到集群 123 的关联性组 456 中，请发送类似如下的请求：

POST /ovirt-engine/api/clusters/123/affinitygroups/456/hostlabels

使用以下机构：

```
<affinity_label id="789"/>
```

表 6.10. 参数摘要

名称	类型	方向	概述
label	AffinityLabel	in/Out	要添加到关联性组中的 AffinityLabel 对象。

6.4.2. 列出 GET

列出分配到此关联性组的所有主机标签。

返回标签的顺序不受保证。

表 6.11. 参数摘要

名称	类型	方向	概述
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
labels	AffinityLabel []	out	分配给 affinity 组的主机标签。
max	整数	in	设置要返回的最大主机标签数。

6.4.2.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.4.2.2. Max

设置要返回的最大主机标签数。如果没有指定，则返回所有标签。

6.5. AFFINITYGROUPHOSTS

此服务管理分配到关联性组的所有主机的集合。

表 6.12. 方法概述

名称	概述
add	将主机添加到关联性组。
list	列出分配到此关联性组的所有主机。

6.5.1. 添加 POST

将主机添加到关联性组。

例如，要将 host 789 添加到集群 123 的关联性组 456 中，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/123/affinitygroups/456/hosts
```

使用以下机构：

```
<host id="789"/>
```

表 6.13. 参数摘要

名称	类型	方向	概述
host	主机	in/Out	要添加到关联性组中的主机。

6.5.2. 列出 GET

列出分配到此关联性组的所有主机。

返回的主机的顺序不受保证。

表 6.14. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
主机	Host[]	out	分配给此关联性组的主机列表。
max	整数	in	设置要返回的最大主机数。

6.5.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.5.2.2. Max

设置要返回的最大主机数。如果未指定，则返回所有主机。

6.6. AFFINITYGROUPVPM

此服务管理单个虚拟机与组分配关联。

表 6.15. 方法概述

名称	概述
remove	从关联性组中移除此虚拟机。

6.6.1. 删除 DELETE

从关联性组中移除此虚拟机。

表 6.16. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.7. AFFINITYGROUPVMLABEL

此服务管理分配至关联性组的单一虚拟机标签。

表 6.17. 方法概述

名称	概述
remove	从 affinity 组移除此标签。

6.7.1. 删除 DELETE

从 affinity 组移除此标签。

表 6.18. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.8. AFFINITYGROUPVMLABELS

此服务管理分配到关联性组的所有虚拟机标签的集合。

表 6.19. 方法概述

名称	概述
add	向关联性组添加虚拟机标签。
list	列出分配到此关联性组的所有虚拟机标签。

6.8.1. 添加 POST

向关联性组添加虚拟机标签。

例如，要将 label 789 添加到集群 123 的关联性组 456 中，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/123/affinitygroups/456/vmlabels
```

使用以下机构：

```
<affinity_label id="789"/>
```

表 6.20. 参数摘要

名称	类型	方向	概述
label	AffinityLabel	in/Out	要添加到关联性组中的 AffinityLabel 对象。

6.8.2. 列出 GET

列出分配到此关联性组的所有虚拟机标签。

返回标签的顺序不受保证。

表 6.21. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
labels	AffinityLabel []	out	分配给关联性组的虚拟机标签。
max	整数	in	设置要返回的最大虚拟机标签数。

6.8.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.8.2.2. Max

设置要返回的最大虚拟机标签数。如果没有指定，则返回所有标签。

6.9. AFFINITYGROUPVMS

此服务管理分配到关联性组的所有虚拟机的集合。

表 6.22. 方法概述

名称	概述
add	将虚拟机添加到关联性组。

名称	概述
list	列出分配到此关联性组的所有虚拟机。

6.9.1. 添加 POST

将虚拟机添加到关联性组。

例如，要将虚拟机 789 添加到集群 123 的关联性组 456 中，请发送如下请求：

```
POST /ovirt-engine/api/clusters/123/affinitygroups/456/vms
```

使用以下机构：

```
<vm id="789"/>
```

表 6.23. 参数摘要

名称	类型	方向	概述
vm	VM	in/Out	

6.9.2. 列出 GET

列出分配到此关联性组的所有虚拟机。

返回的虚拟机的顺序无法保证。

表 6.24. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大虚拟机数量。
vms	Vm[]	out	

6.9.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.9.2.2. Max

设置要返回的最大虚拟机数量。如果未指定，则返回所有虚拟机。

6.10. AFFINITYGROUPS

关联性组服务管理虚拟机关系和依赖项。

表 6.25. 方法概述

名称	概述
add	创建新的关联性组。
list	列出现有的关联性组。

6.10.1. 添加 POST

创建新的关联性组。

在下例中发布一个请求以创建新关联性组：

```
POST /ovirt-engine/api/clusters/000-000/affinitygroups
```

并在正文中使用以下示例：

```
<affinity_group>
  <name>AF_GROUP_001 </name>
  <hosts_rule>
    <enforcing>true</enforcing>
    <positive>true</positive>
  </hosts_rule>
  <vms_rule>
    <enabled>false</enabled>
  </vms_rule>
</affinity_group>
```

表 6.26. 参数摘要

名称	类型	方向	概述
group	AffinityGroup	in/Out	要创建的关联性组对象。

6.10.2. 列出 GET

列出现有的关联性组。

关联组结果的顺序不受保证。

表 6.27. 参数摘要

名称	类型	方向	概述
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
groups	AffinityGroup[]	out	现有关联性组的列表。
max	整数	in	设置要返回的关联性组的最大数量。

6.10.2.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.10.2.2. Max

设置要返回的关联性组的最大数量。如果未指定，则返回所有关联性组。

6.11. AFFINITYLABEL

单个关联性标签的详细信息。

表 6.28. 方法概述

名称	概述
get	检索标签的详细信息。

名称	概述
remove	从系统中删除标签并清除已删除标签的所有分配。
update	更新标签。

6.11.1. 获取 GET

检索标签的详细信息。

表 6.29. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
label	AffinityLabel	out	

6.11.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.11.2. 删除 DELETE

从系统中删除标签并清除已删除标签的所有分配。

6.11.3. 更新 PUT

更新标签。此调用将更新所有元数据，如名称或描述。

表 6.30. 参数摘要

名称	类型	方向	概述
label	AffinityLabel	in/Out	

6.12. AFFINITYLABELHOST

此服务表示在通过 `affinitylabels/hosts` 子集合访问时具有特定标签的主机。

表 6.31. 方法概述

名称	概述
<code>get</code>	检索分配了此标签的主机的详细信息。
<code>remove</code>	从主机中删除标签。

6.12.1. 获取 GET

检索分配了此标签的主机的详细信息。

表 6.32. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>host</code>	主机	out	

6.12.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.12.2. 删除 DELETE

从主机中删除标签。

6.13. AFFINITYLABELHOSTS

此服务代表通过 `affinitylabels/hosts` 子集合访问时具有特定标签的主机列表。

表 6.33. 方法概述

名称	概述
<code>add</code>	向主机添加标签。

名称	概述
list	列出具有 标签的所有主机。

6.13.1. 添加 POST

向主机添加标签。

表 6.34. 参数摘要

名称	类型	方向	概述
host	主机	in/Out	

6.13.2. 列出 GET

列出具有 标签的所有主机。

返回的主机的顺序不受保证。

表 6.35. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
主机	Host[]	out	

6.13.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.14. AFFINITYLABELVM

此服务代表 vm，它在通过 `affinitylabels/vms` 子集合访问时具有特定标签。

表 6.36. 方法概述

名称	概述
get	检索分配了此标签的 vm 的详细信息。
remove	从 vm 中删除标签。

6.14.1. 获取 GET

检索分配了此标签的 vm 的详细信息。

表 6.37. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
vm	VM	out	

6.14.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.14.2. 删除 DELETE

从 vm 中删除标签。

6.15. AFFINITYLABELVMS

此服务代表通过 `affinitylabels/vms` 子集合访问时具有特定标签的 vms 列表。

表 6.38. 方法概述

名称	概述
add	向 vm 添加标签。
list	列出所有具有 标签的虚拟机。

6.15.1. 添加 POST

向 **vm** 添加标签。

表 6.39. 参数摘要

名称	类型	方向	概述
vm	VM	in/Out	

6.15.2. 列出 GET

列出所有具有 标签的虚拟机。

返回的虚拟机的顺序无法保证。

表 6.40. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
vms	Vm[]	out	

6.15.2.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.16. AFFINITYLABELS

管理系统中可用的关联性标签。

表 6.41. 方法概述

名称	概述
add	创建新标签。
list	列出系统中存在的所有标签。

6.16.1. 添加 POST

创建新标签。标签自动附加到 `vms` 或 `hosts` 列表中提及的所有实体。

表 6.42. 参数摘要

名称	类型	方向	概述
<code>label</code>	<code>AffinityLabel</code>	in/Out	

6.16.2. 列出 GET

列出系统中存在的所有标签。

返回标签的顺序不受保证。

表 6.43. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>labels</code>	<code>AffinityLabel</code> []	out	
<code>max</code>	整数	in	设置要返回的最大标签数。

6.16.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.16.2.2. Max

设置要返回的最大标签数。如果没有指定，则返回所有标签。

6.17. 区域

此注释旨在指定与以下内容相关的带注释概念的 oVirt 区域：目前，以下领域正在使用中，它们与 oVirt 团队紧密相关，但不一定是相同的：

- 基础架构
- Network
- SLA
- 存储
- 虚拟化

概念可以与多个区域相关联，或者与任何区域相关联。

此注解的值仅用于报告，它不会影响生成的所有代码或模型的有效性

6.18. ASSIGNEDAFFINITYLABEL

此服务表示在使用实体/关联性标签子集合访问时，实体分配的一个标签。

表 6.44. 方法概述

名称	概述
get	检索所附加标签的详细信息。
remove	从实体中删除该标签。

6.18.1. 获取 GET

检索所附加标签的详细信息。

表 6.45. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。

名称	类型	方向	概述
label	AffinityLabel	out	

6.18.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.18.2. 删除 DELETE

从实体中删除该标签。请勿接触标签本身。

6.19. ASSIGNEDAFFINITYLABELS

此服务用于列出和操作在使用实体/关联性标签访问时分配给支持的实体的关联性标签。

表 6.46. 方法概述

名称	概述
add	将标签附加到实体。
list	列出附加到实体的所有标签。

6.19.1. 添加 POST

将标签附加到实体。

表 6.47. 参数摘要

名称	类型	方向	概述
label	AffinityLabel	in/Out	

6.19.2. 列出 GET

列出附加到实体的所有标签。

返回实体的顺序不受保证。

表 6.48. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
label	AffinityLabel []	out	

6.19.2.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.20. ASSIGNEDCPUPROFILE

表 6.49. 方法概述

名称	概述
get	
remove	

6.20.1. 获取 GET

表 6.50. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
profile	CpuProfile	out	

6.20.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.20.2. 删除 DELETE

表 6.51. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.21. ASSIGNEDCPUFILES

表 6.52. 方法概述

名称	概述
add	为集群添加新 cpu 配置集。
list	列出分配给集群的 CPU 配置集。

6.21.1. 添加 POST

为集群添加新 cpu 配置集。

表 6.53. 参数摘要

名称	类型	方向	概述
profile	CpuProfile	in/Out	

6.21.2. 列出 GET

列出分配给集群的 CPU 配置集。

返回的 CPU 配置文件的顺序不受保证。

表 6.54. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大配置集数。
profiles	CpuProfile[]	out	

6.21.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.21.2.2. Max

设置要返回的最大配置集数。如果没有指定，则返回所有配置集。

6.22. ASSIGNEDDISKPROFILE

表 6.55. 方法概述

名称	概述
get	
remove	

6.22.1. 获取 GET

表 6.56. 参数摘要

名称	类型	方向	概述
disk_profile	DiskProfile	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.22.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.22.2. 删除 DELETE

表 6.57. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.23. ASSIGNEDDISKPROFILES

表 6.58. 方法概述

名称	概述
add	为存储域添加新磁盘配置文件。
list	返回分配给存储域的磁盘配置文件列表。

6.23.1. 添加 POST

为存储域添加新磁盘配置文件。

表 6.59. 参数摘要

名称	类型	方向	概述
profile	DiskProfile	in/Out	

6.23.2. 列出 GET

返回分配给存储域的磁盘配置文件列表。

返回的磁盘配置文件的顺序无法保证。

表 6.60. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大配置集数。
profiles	DiskProfile[]	out	

6.23.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.23.2.2. Max

设置要返回的最大配置集数。如果没有指定，则返回所有配置集。

6.24. ASSIGNEDPERMISSIONS

表示权限子集合，按用户、组或某些实体类型的范围。

表 6.61. 方法概述

名称	概述
add	为特定实体为用户或组分配新权限。
list	列出特定实体的所有权限。

6.24.1. 添加 POST

为特定实体为用户或组分配新权限。

例如，要为 ID 为 123 的虚拟机分配 **UserVmManager** 角色给 ID 为 456 的用户发送类似如下的请求：

POST /ovirt-engine/api/vms/123/permissions

这样的请求正文：

```
<permission>
  <role>
    <name>UserVmManager</name>
  </role>
  <user id="456"/>
</permission>
```

要将 **SuperUser** 角色分配给 ID 为 456 的用户，请发送如下请求：

POST /ovirt-engine/api/permissions

这样的请求正文：

```
<permission>
  <role>
    <name>SuperUser</name>
```



```

</role>
<user id="456"/>
</permission>

```

如果要为组而不是用户分配权限，请将 **user** 元素替换为 **组** 元素的适当 ID。例如，将 **UserRole** 角色分配给 ID 为 123 的集群，将 id 789 的组分配如下：

POST /ovirt-engine/api/clusters/123/permissions

这样的请求正文：

```

<permission>
  <role>
    <name>UserRole</name>
  </role>
  <group id="789"/>
</permission>

```

表 6.62. 参数摘要

名称	类型	方向	概述
权限	权限	in/Out	权限。

6.24.2. 列出 GET

列出特定实体的所有权限。

例如，列出 ID 为 123 的集群的所有权限 发送 如下请求：

GET /ovirt-engine/api/clusters/123/permissions

```

<permissions>
  <permission id="456">
    <cluster id="123"/>
    <role id="789"/>
    <user id="451"/>
  </permission>
  <permission id="654">
    <cluster id="123"/>
    <role id="789"/>
    <group id="127"/>
  </permission>
</permissions>

```

返回的权限顺序不受保证。

表 6.63. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
权限	permission[]	out	权限列表。

6.24.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.25. ASSIGNEDROLES

代表子集合的角色，如用户作用域。

表 6.64. 方法概述

名称	概述
list	返回分配给权限的角色。

6.25.1. 列出 GET

返回分配给权限的角色。

返回角色的顺序不受保证。

表 6.65. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的角色的最大数量。
角色	role[]	out	

6.25.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.25.1.2. Max

设置要返回的角色的最大数量。如果未指定，则返回所有角色。

6.26. ASSIGNEDTAG

种服务，用于管理为系统中特定实体分配特定标签的服务。

表 6.66. 方法概述

名称	概述
get	获取关于分配标签的信息。
remove	从系统中的特定实体取消分配标签。

6.26.1. 获取 GET

获取关于分配标签的信息。

例如，要检索 ID 为 456 的标签信息，该标签分配给 id 123 的虚拟机，发送如下请求：

```
GET /ovirt-engine/api/vms/123/tags/456
```

```
<tag href="/ovirt-engine/api/tags/456" id="456">
  <name>root</name>
  <description>root</description>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</tag>
```

表 6.67. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
tag	Tag	out	分配的标签。

6.26.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.26.2. 删除 DELETE

从系统中的特定实体取消分配标签。

例如，要从 ID 123 的虚拟机中取消分配 ID 为 456 的标签，发送类似如下的请求：

```
DELETE /ovirt-engine/api/vms/123/tags/456
```

表 6.68. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.27. ASSIGNEDTAGS

此服务用于管理向系统中特定实体分配标签的服务。

表 6.69. 方法概述

名称	概述
add	为系统中的特定实体分配标签。
list	列出分配给特定实体的所有标签。

6.27.1. 添加 POST

为系统中的特定实体分配标签。

例如，将标签 mytag 分配给 ID 为 123 的虚拟机发送如下请求：

```
POST /ovirt-engine/api/vms/123/tags
```

这样的请求正文：

```
<tag>
  <name>mytag</name>
</tag>
```

表 6.70. 参数摘要

名称	类型	方向	概述
tag	Tag	in/Out	分配的标签。

6.27.2. 列出 GET

列出分配给特定实体的所有标签。

例如，要列出 ID 为 123 的虚拟机的所有标签，发送如下请求：

GET /ovirt-engine/api/vms/123/tags

```
<tags>
  <tag href="/ovirt-engine/api/tags/222" id="222">
    <name>mytag</name>
    <description>mytag</description>
    <vm href="/ovirt-engine/api/vms/123" id="123"/>
  </tag>
</tags>
```

返回标签的顺序不受保证。

表 6.71. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大标签数。
tags	Tag[]	out	所分配标签的列表。

6.27.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.27.2.2. Max

设置要返回的最大标签数。如果未指定，则返回所有标签。

6.28. ASSIGNEDVNICPROFILE

表 6.72. 方法概述

名称	概述
get	
remove	

6.28.1. 获取 GET

表 6.73. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
profile	VnicProfile	out	

6.28.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.28.2. 删除 DELETE

表 6.74. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.29. ASSIGNEDVNICPROFILES

表 6.75. 方法概述

名称	概述
add	为网络添加新的虚拟网络接口卡配置文件。
list	将 VNIC 配置集列表返回给网络。

6.29.1. 添加 POST

为网络添加新的虚拟网络接口卡配置文件。

表 6.76. 参数摘要

名称	类型	方向	概述
profile	VnicProfile	in/Out	

6.29.2. 列出 GET

将 VNIC 配置集列表返回给网络。

返回的 VNIC 配置集的顺序无法保证。

表 6.77. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大配置集数。
profiles	VnicProfile[]	out	

6.29.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.29.2.2. Max

设置要返回的最大配置集数。如果没有指定，则返回所有配置集。

6.30. ATTACHEDSTORAGEDOMAIN

表 6.78. 方法概述

名称	概述
激活	此操作将激活已连接的存储域。
取消激活	此操作将停用已连接的存储域。
get	
remove	

6.30.1. 激活 POST

此操作将激活已连接的存储域。激活存储域后，该存储域就已准备好用于数据中心。

```
POST /ovirt-engine/api/datacenters/123/storagedomains/456/activate
```

激活操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

表 6.79. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行激活。

6.30.2. 取消激活 POST

此操作将停用已连接的存储域。存储域被停用后，该存储域将不会与数据中心一起使用。例如，要停用存储域 456，请发送以下请求：

```
POST /ovirt-engine/api/datacenters/123/storagedomains/456/deactivate
```

这样的请求正文：

```
<action/>
```


如果 `force` 参数为 `true`，则操作将成功，即使取消激活存储域失败前进行 OVF 更新也是如此。如果 `force` 参数为 `false`，OVF 更新失败，则存储域的停用也将失败。

表 6.80. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行取消激活。
<code>force</code>	布尔值	in	指明操作是否应该成功，并且存储域应移到停用的状态，即使存储域的 OVF 更新失败。

6.30.2.1. force

指明操作是否应该成功，并且存储域应移到停用的状态，即使存储域的 OVF 更新失败。例如，要使用 `force` 标志停用存储域 456，请发送以下请求：

```
POST /ovirt-engine/api/datacenters/123/storagedomains/456/deactivate
```

这样的请求正文：

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

此参数是可选的，默认值为 `false`。

6.30.3. 获取 GET

表 6.81. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>storage_domain</code>	StorageDomain	out	

6.30.3.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

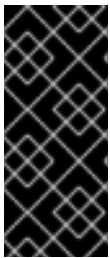
6.30.4. 删除 DELETE

表 6.82. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.31. ATTACHEDSTORAGEDOMAINDISK

管理附加到数据中心的存储域中的单个磁盘。



重要

由于引擎版本 4.2 旨在仅列出存储域中可用的磁盘，并注册未注册的磁盘。所有其他操作（如复制磁盘、移动磁盘等）已被弃用，并将在以后的版本中删除。[要执行这些操作，请使用管理系统所有磁盘的服务，或者使用管理特定磁盘的服务。](#)

表 6.83. 方法概述

名称	概述
copy	将磁盘复制到指定的存储域。
export	将磁盘导出到导出存储域。
get	检索磁盘的描述。
Move	将磁盘移动到其他存储域。
Register	注册未注册的磁盘。
remove	删除磁盘。
sparsify	稀疏磁盘。
update	更新磁盘。

6.31.1. 复制 POST

将磁盘复制到指定的存储域。

**重要**

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。[要复制磁盘，请使用管理该磁盘的服务的副本操作。](#)

表 6.84. 参数摘要

名称	类型	方向	概述
disk	Disk	in	生成磁盘的描述。
storage_domain	StorageDomain	in	创建新磁盘的存储域。

6.31.2. 导出 POST

将磁盘导出到导出存储域。

**重要**

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。[若要导出磁盘，可使用管理该磁盘的服务的导出操作。](#)

表 6.85. 参数摘要

名称	类型	方向	概述
storage_domain	StorageDomain	in	磁盘应导出到的导出存储域。

6.31.3. 获取 GET

检索磁盘的描述。

表 6.86. 参数摘要

名称	类型	方向	概述
disk	Disk	out	磁盘的描述。
follow	字符串	in	指明应遵循哪些内联链接。

6.31.3.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.31.4. 移动 POST

将磁盘移动到其他存储域。



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。[要移动磁盘，请使用管理该磁盘的服务的移动操作。](#)

表 6.87. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移动。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
storage_domain	StorageDomain	in	磁盘将移动到的存储域。

6.31.5. 注册 POST

注册未注册的磁盘。

6.31.6. 删除 DELETE

删除磁盘。



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。[要删除磁盘，请使用管理该磁盘的服务的 `remove` 操作。](#)

6.31.7. 稀疏 POST

稀疏磁盘。



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。要删除磁盘，请使用管理该磁盘的服务的 [remove](#) 操作。

6.31.8. 更新 PUT

更新磁盘。



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。若要更新磁盘，可使用管理该磁盘的服务的更新操作。

表 6.88. 参数摘要

名称	类型	方向	概述
disk	Disk	in/Out	应用到磁盘的更新。

6.32. ATTACHEDSTORAGEDOMAINDISKS

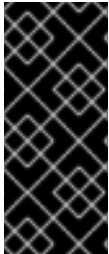
管理附加到数据中心的存储域中可用磁盘的集合。

表 6.89. 方法概述

名称	概述
add	添加或注册磁盘。
list	检索存储域中可用的磁盘的列表。

6.32.1. 添加 POST

添加或注册磁盘。



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。要添加新磁盘，请使用管理系统磁盘的服务的 **add** 操作。要注册一个未注册的磁盘，请使用管理该磁盘的服务注册操作。

表 6.90. 参数摘要

名称	类型	方向	概述
disk	Disk	in/Out	要添加或注册的磁盘。
unregistered	布尔值	in	指明是否应添加新磁盘，或者是否应注册现有的未注册磁盘。

6.32.1.1. unregistered

指明是否应添加新磁盘，或者是否应注册现有的未注册磁盘。如果值为 **true**，则需要提供要注册的磁盘标识符。例如，使用 **id 456** 注册磁盘会发送如下请求：

```
POST /ovirt-engine/api/storagedomains/123/disks?unregistered=true
```

这样的请求正文：

```
<disk id="456"/>
```

如果值为 **false**，则会在存储域中创建新磁盘。在这种情况下，**provisioning_size**、**格式**和 **name** 属性是必需的。例如：要在 1 GiB 的写入磁盘上创建新副本，请发送如下请求：

```
POST /ovirt-engine/api/storagedomains/123/disks
```

这样的请求正文：

```
<disk>
  <name>mydisk</name>
  <format>cow</format>
  <provisioned_size>1073741824</provisioned_size>
</disk>
```

默认值为 **false**。

6.32.2. 列出 GET

检索存储域中可用的磁盘的列表。

表 6.91. 参数摘要

名称	类型	方向	概述
磁盘	<code>disk[]</code>	out	检索的磁盘列表。
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>max</code>	整数	in	设置要返回的最大磁盘数。

6.32.2.1. 磁盘

检索的磁盘列表。

返回的磁盘的顺序无法保证。

6.32.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.32.2.3. Max

设置要返回的最大磁盘数。如果没有指定，则返回所有磁盘。

6.33. ATTACHEDSTORAGEDOMAINS

管理附加到数据中心的存储域。

表 6.92. 方法概述

名称	概述
<code>add</code>	将现有存储域附加到数据中心。
<code>list</code>	返回附加到数据中心的存储域的列表。

6.33.1. 添加 POST

将现有存储域附加到数据中心。

表 6.93. 参数摘要

名称	类型	方向	概述
storage_domain	StorageDomain	in/Out	要连接到数据中心的存储域。

6.33.2. 列出 GET

返回附加到数据中心的存储域的列表。

返回的存储域的顺序不受保证。

表 6.94. 参数摘要

名称	类型	方向	概述
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的存储域的最大数量。
storage_domains	StorageDomain[]	out	附加到数据中心的存储域的列表。

6.33.2.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.33.2.2. Max

设置要返回的存储域的最大数量。如果没有指定，则返回所有存储域。

6.34. BALANCE

表 6.95. 方法概述

名称	概述
get	
remove	

6.34.1. 获取 GET

表 6.96. 参数摘要

名称	类型	方向	概述
balance	balance	out	
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。

6.34.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.34.2. 删除 DELETE

表 6.97. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.35. 平衡

表 6.98. 方法概述

名称	概述
add	向指定的用户定义的调度策略中添加 balance 模块。
list	返回调度策略使用的平衡模块列表。

6.35.1. 添加 POST

向指定的用户定义的调度策略中添加 **balance** 模块。

表 6.99. 参数摘要

名称	类型	方向	概述
balance	balance	in/Out	

6.35.2. 列出 GET

返回调度策略使用的平衡模块列表。

返回的平衡模块的顺序无法保证。

表 6.100. 参数摘要

名称	类型	方向	概述
平衡	balance[]	out	
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置返回的最大平衡数。

6.35.2.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.35.2.2. Max

设置返回的最大平衡数。如果未指定，将返回所有平衡。

6.36. 书签

用于管理书签的服务。

表 6.101. 方法概述

名称	概述
get	获取书签.
remove	删除书签.
update	更新书签.

6.36.1. 获取 GET

获取书签.

获取书签的示例：

```
GET /ovirt-engine/api/bookmarks/123
```

```
<bookmark href="/ovirt-engine/api/bookmarks/123" id="123">
  <name>example_vm</name>
  <value>vm: name=example*</value>
</bookmark>
```

表 6.102. 参数摘要

名称	类型	方向	概述
bookmark	书签	out	请求的书签。
follow	字符串	in	指明应遵循哪些内联链接。

6.36.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.36.2. 删除 DELETE

删除书签.

删除书签的示例：

```
DELETE /ovirt-engine/api/bookmarks/123
```

表 6.103. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.36.3. 更新 PUT

更新书签。

更新书签的示例：

```
PUT /ovirt-engine/api/bookmarks/123
```

使用请求正文：

```
<bookmark>
  <name>new_example_vm</name>
  <value>vm: name=new_example*</value>
</bookmark>
```

表 6.104. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
bookmark	书签	in/Out	更新的书签。

6.37. 书签

用于管理书签的服务。

表 6.105. 方法概述

名称	概述
add	添加新书签。
list	列出所有可用的书签。

6.37.1. 添加 POST

添加新书签。

添加书签示例：

```
POST /ovirt-engine/api/bookmarks
```

```
<bookmark>
  <name>new_example_vm</name>
  <value>vm: name=new_example*</value>
</bookmark>
```

表 6.106. 参数摘要

名称	类型	方向	概述
bookmark	书签	in/Out	添加的书签.

6.37.2. 列出 GET

列出所有可用的书签。

列出书签示例：

```
GET /ovirt-engine/api/bookmarks
```

```
<bookmarks>
  <bookmark href="/ovirt-engine/api/bookmarks/123" id="123">
    <name>database</name>
    <value>vm: name=database*</value>
  </bookmark>
  <bookmark href="/ovirt-engine/api/bookmarks/456" id="456">
    <name>example</name>
    <value>vm: name=example*</value>
  </bookmark>
</bookmarks>
```

返回的书签的顺序无法保证。

表 6.107. 参数摘要

名称	类型	方向	概述
bookmarks	书签[]	out	可用书签列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大书签数。

6.37.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.37.2.2. Max

设置要返回的最大书签数。如果没有指定，则返回所有书签。

6.38. CLUSTER

管理特定集群的服务。

表 6.108. 方法概述

名称	概述
get	获取有关集群的信息。
refreshglusterhealstatus	刷新集群中所有卷的 Gluster 修复信息。
remove	从系统中删除集群。
resetemulatedmachine	
syncallnetworks	同步集群中的所有网络。
update	更新有关集群的信息。
upgrade	根据操作值启动、更新或完成集群的升级过程。

6.38.1. 获取 GET

获取有关集群的信息。

获取集群的示例：

GET /ovirt-engine/api/clusters/123

```
<cluster href="/ovirt-engine/api/clusters/123" id="123">
  <actions>
    <link href="/ovirt-engine/api/clusters/123/resetemulatedmachine" rel="resetemulatedmachine"/>
  </actions>
  <name>Default</name>
  <description>The default server cluster</description>
  <link href="/ovirt-engine/api/clusters/123/networks" rel="networks"/>
  <link href="/ovirt-engine/api/clusters/123/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/clusters/123/glustervolumes" rel="glustervolumes"/>
  <link href="/ovirt-engine/api/clusters/123/glusterhooks" rel="glusterhooks"/>
  <link href="/ovirt-engine/api/clusters/123/affinitygroups" rel="affinitygroups"/>
  <link href="/ovirt-engine/api/clusters/123/cpuprofiles" rel="cpuprofiles"/>
  <ballooning_enabled>false</ballooning_enabled>
  <cpu>
    <architecture>x86_64</architecture>
    <type>Intel Nehalem Family</type>
  </cpu>
  <error_handling>
    <on_error>migrate</on_error>
  </error_handling>
  <fencing_policy>
    <enabled>true</enabled>
    <skip_if_connectivity_broken>
      <enabled>false</enabled>
      <threshold>50</threshold>
    </skip_if_connectivity_broken>
    <skip_if_sd_active>
      <enabled>false</enabled>
    </skip_if_sd_active>
  </fencing_policy>
  <gluster_service>false</gluster_service>
  <ha_reservation>false</ha_reservation>
  <ksm>
    <enabled>true</enabled>
    <merge_across_nodes>true</merge_across_nodes>
  </ksm>
  <memory_policy>
    <over_commit>
      <percent>100</percent>
    </over_commit>
    <transparent_hugepages>
      <enabled>true</enabled>
    </transparent_hugepages>
  </memory_policy>
  <migration>
    <auto_converge>inherit</auto_converge>
  </migration>
</cluster>
```

```

<bandwidth>
  <assignment_method>auto</assignment_method>
</bandwidth>
<compressed>inherit</compressed>
</migration>
<required_rng_sources>
  <required_rng_source>random</required_rng_source>
</required_rng_sources>
<scheduling_policy href="/ovirt-engine/api/schedulingpolicies/456" id="456"/>
<threads_as_cores>false</threads_as_cores>
<trusted_service>false</trusted_service>
<tunnel_migration>false</tunnel_migration>
<version>
  <major>4</major>
  <minor>0</minor>
</version>
<virt_service>true</virt_service>
<data_center href="/ovirt-engine/api/datacenters/111" id="111"/>
</cluster>

```

表 6.109. 参数摘要

名称	类型	方向	概述
cluster	Cluster	out	
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。

6.38.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.38.2. refreshglusterhealstatus POST

刷新集群中所有卷的 Gluster 修复信息。

例如，Cluster 123，发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/123/refreshglusterhealstatus
```

6.38.3. 删除 DELETE

从系统中删除集群。

```
DELETE /ovirt-engine/api/clusters/00000000-0000-0000-0000-000000000000
```

表 6.110. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.38.4. resetemulatedmachine POST

表 6.111. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行重置。

6.38.5. syncallnetworks POST

同步集群中的所有网络。

```
POST /ovirt-engine/api/clusters/123/syncallnetworks
```

这样的请求正文：

```
<action/>
```

表 6.112. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.38.6. 更新 PUT

更新有关集群的信息。

只有指定的字段会被更新，其他字段保持不变。

例如，更新集群的 CPU：

```
PUT /ovirt-engine/api/clusters/123
```

这样的请求正文：

```
<cluster>
  <cpu>
    <type>Intel Haswell-noTSX Family</type>
  </cpu>
</cluster>
```

表 6.113. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
cluster	Cluster	in/Out	

6.38.7. 升级 POST

根据操作值启动、更新或完成集群的升级过程。此操作将集群标记为升级，更新进度，或者根据启动值、停止或 `update_progress` 的值来清除在集群中运行的升级标志。

```
POST /ovirt-engine/api/clusters/123/upgrade
```

使用这个请求正文来标记集群进行升级：

```
<action>
  <upgrade_action>
    start
  </upgrade_action>
</action>
```

在开始升级后，请使用请求正文将进度更新为 15%：

```
<action>
  <upgrade_action>
    update_progress
  </upgrade_action>
  <upgrade_percent_complete>
```

15

```
</upgrade_percent_complete>
</action>
```

表 6.114. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
correlation_id	字符串	in	明确设置升级相关性标识符。
upgrade_action	ClusterUpgradeAction	in	要执行的操作。
upgrade_percent_complete	整数	in	将升级的过程作为总过程的百分比更新。

6.38.7.1. correlation_id

明确设置升级相关性标识符。使用 将详细说明集群升级的事件与升级本身相关联。如果不指定，则使用来自 **Correlation-Id** http 标头的关联 ID。

6.39. CLUSTERENABLEDFEATURE

代表为集群启用的功能。

表 6.115. 方法概述

名称	概述
get	提供有关启用集群功能的信息。
remove	禁用集群功能。

6.39.1. 获取 GET

提供有关启用集群功能的信息。

例如，要查找集群 123 启用的功能 456 的详细信息，请发送如下请求：

-

```
GET /ovirt-engine/api/clusters/123/enabledfeatures/456
```

这将返回包含名称的 **ClusterFeature** 对象：

```
<cluster_feature id="456">
  <name>libgfapi_supported</name>
</cluster_feature>
```

表 6.116. 参数摘要

名称	类型	方向	概述
功能	ClusterFeature	out	检索启用的集群功能。
follow	字符串	in	指明应遵循哪些内联链接。

6.39.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.39.2. 删除 DELETE

禁用集群功能。

例如，要禁用集群 123 的功能 456，发送如下请求：

```
DELETE /ovirt-engine/api/clusters/123/enabledfeatures/456
```

6.40. CLUSTERENABLEDFEATURES

提供有关为此集群启用的其他功能的信息。启用的功能是集群级别的可用功能

表 6.117. 方法概述

名称	概述
add	为集群启用额外功能。
list	列出为集群启用的其他功能。

6.40.1. 添加 POST

为集群启用额外功能。

例如，要在集群 123 上启用功能 456，请发送如下请求：

```
POST /ovirt-engine/api/clusters/123/enabledfeatures
```

请求正文应类似如下：

```
<cluster_feature id="456"/>
```

表 6.118. 参数摘要

名称	类型	方向	概述
功能	ClusterFeature	in/Out	

6.40.2. 列出 GET

列出为集群启用的其他功能。

例如，要为集群 123 启用功能，请发送类似如下的请求：

```
GET /ovirt-engine/api/clusters/123/enabledfeatures
```

这将返回一个功能列表：

```
<enabled_features>
  <cluster_feature id="123">
    <name>test_feature</name>
  </cluster_feature>
  ...
</enabled_features>
```

表 6.119. 参数摘要

名称	类型	方向	概述
功能	ClusterFeature[]	out	检索的功能。
follow	字符串	in	指明应遵循哪些内联链接。

6.40.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.41. CLUSTEREXTERNALPROVIDERS

此服务列出了外部提供程序。

表 6.120. 方法概述

名称	概述
list	返回外部提供程序列表。

6.41.1. 列出 GET

返回外部提供程序列表。

无法保证返回的供应商列表的顺序。

表 6.121. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
供应商	ExternalProvider[]	out	

6.41.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.42. CLUSTERFEATURE

代表为集群级别启用的功能

表 6.122. 方法概述

名称	概述
get	提供有关集群级别支持的集群功能的信息。

6.42.1. 获取 GET

提供有关集群级别支持的集群功能的信息。

例如，要查找集群级别 4.1 的集群功能 456 的详情，请发送类似如下的请求：

```
GET /ovirt-engine/api/clusterlevels/4.1/clusterfeatures/456
```

这将返回包含名称的 **ClusterFeature** 对象：

```
<cluster_feature id="456">
  <name>libgfapi_supported</name>
</cluster_feature>
```

表 6.123. 参数摘要

名称	类型	方向	概述
功能	ClusterFeature	out	检索的集群功能。
follow	字符串	in	指明应遵循哪些内联链接。

6.42.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.43. CLUSTERFEATURES

提供有关集群级别支持的集群功能的信息。

表 6.124. 方法概述

名称	概述
list	列出集群级别支持的集群功能。

6.43.1. 列出 GET

列出集群级别支持的集群功能。

GET /ovirt-engine/api/clusterlevels/4.1/clusterfeatures

这将返回集群级别支持的集群功能列表：

```
<cluster_features>
  <cluster_feature id="123">
    <name>test_feature</name>
  </cluster_feature>
  ...
</cluster_features>
```

表 6.125. 参数摘要

名称	类型	方向	概述
功能	ClusterFeature[]	out	检索的功能。
follow	字符串	in	指明应遵循哪些内联链接。

6.43.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.44. CLUSTERLEVEL

提供有关特定集群级别的信息。如需更多信息，请参阅 [ClusterLevels](#) 服务。

表 6.126. 方法概述

名称	概述
get	提供有关此服务管理的特定集群级别功能的信息。

6.44.1. 获取 GET

提供有关此服务管理的特定集群级别功能的信息。

例如，要查找级别 3.6 支持哪些 CPU 类型，您可以发送类似如下的请求：

```
GET /ovirt-engine/api/clusterlevels/3.6
```

这将返回一个包含支持的 CPU 类型的 **ClusterLevel** 对象，以及描述集群级别的其他信息：

```
<cluster_level id="3.6">
  <cpu_types>
    <cpu_type>
      <name>Intel Nehalem Family</name>
      <level>3</level>
      <architecture>x86_64</architecture>
    </cpu_type>
    ...
  </cpu_types>
  <permits>
    <permit id="1">
      <name>create_vm</name>
      <administrative>false</administrative>
    </permit>
    ...
  </permits>
</cluster_level>
```

表 6.127. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
level	ClusterLevel	out	Retrieved 集群级别。

6.44.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.45. CLUSTERLEVELS

提供有关引擎支持的不同集群级别功能的信息。引擎的版本 4.0 支持级别 4.0 和 3.6。每个级别都支持不同的 CPU 类型集，例如：此服务提供该信息。

表 6.128. 方法概述

名称	概述
<code>list</code>	列出系统支持的集群级别。

6.45.1. 列出 GET

列出系统支持的集群级别。

GET /ovirt-engine/api/clusterlevels

这将返回可用集群级别的列表。

```
<cluster_levels>
  <cluster_level id="4.0">
    ...
  </cluster_level>
  ...
</cluster_levels>
```

返回的集群级别的顺序不受保证。

表 6.129. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>level</code>	ClusterLevel []	out	检索的集群级别。

6.45.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.46. CLUSTERNETWORK

管理特定集群网络的服务。

表 6.130. 方法概述

名称	概述
get	检索集群网络详情。
remove	从集群中取消分配网络。
update	更新集群中的网络。

6.46.1. 获取 GET

检索集群网络详情。

表 6.131. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
network	Network	out	集群网络。

6.46.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.46.2. 删除 DELETE

从集群中取消分配网络。

6.46.3. 更新 PUT

更新集群中的网络。

表 6.132. 参数摘要

名称	类型	方向	概述
network	Network	in/Out	集群网络。

6.47. CLUSTER NETWORKS

用于管理集群网络的服务。

表 6.133. 方法概述

名称	概述
add	将网络分配给集群。
list	列出分配到集群的网络。

6.47.1. 添加 POST

将网络分配给集群。

在以下示例中发布一个将网络分配给集群的请求：

```
POST /ovirt-engine/api/clusters/123/networks
```

在正文中使用以下示例：

```
<network id="123" />
```

表 6.134. 参数摘要

名称	类型	方向	概述
network	Network	in/Out	要分配给集群的网络对象。

6.47.2. 列出 GET

列出分配到集群的网络。

返回的集群的顺序不受保证。

表 6.135. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大网络数。
网络	network[]	out	分配给集群的网络列表。

6.47.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.47.2.2. Max

设置要返回的最大网络数。如果没有指定，则会返回所有网络。

6.48. 集群

用于管理集群的服务。

表 6.136. 方法概述

名称	概述
add	创建新集群。
list	返回系统集群列表。

6.48.1. 添加 POST

创建新集群。

这需要 `name`、`cpu.type` 和 `data_center` 属性。使用 `id` 或 `name` 属性识别数据中心。

POST /ovirt-engine/api/clusters

这样的请求正文：

```
<cluster>
  <name>mycluster</name>
  <cpu>
    <type>Intel Nehalem Family</type>
  </cpu>
  <data_center id="123"/>
</cluster>
```

要创建具有外部网络供应商的集群，请在添加到集群中的每个主机上部署，请发送如下请求：

```
POST /ovirt-engine/api/clusters
```

使用一个请求正文，其中包含对所需供应商的引用：

```
<cluster>
  <name>mycluster</name>
  <cpu>
    <type>Intel Nehalem Family</type>
  </cpu>
  <data_center id="123"/>
  <external_network_providers>
    <external_provider name="ovirt-provider-ovn"/>
  </external_network_providers>
</cluster>
```

表 6.137. 参数摘要

名称	类型	方向	概述
cluster	Cluster	in/Out	

6.48.2. 列出 GET

返回系统群集列表。

只有 **search** 参数中包含 **orderby** 子句时，才会保证返回的集群的顺序。

表 6.138. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下执行搜索。
clusters	cluster[]	out	
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大集群数量。
search	字符串	in	用于限制返回的集群的查询字符串。

6.48.2.1. case_sensitive

指明是否应该在考虑大小写的情况下执行搜索。默认值为 **true**，这意味着会考虑这个情况。要搜索忽略大小写，将其设置为 **false**。

6.48.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.48.2.3. Max

设置要返回的最大集群数量。如果没有指定，则会返回所有集群。

6.49. COPYABLE

表 6.139. 方法概述

名称	概述
copy	

6.49.1. 复制 POST

表 6.140. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明复制应异步执行。

6.50. CPUPROFILE

表 6.141. 方法概述

名称	概述
get	
remove	
update	更新系统中指定的 cpu 配置集。

6.50.1. 获取 GET

表 6.142. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
profile	CpuProfile	out	

6.50.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.50.2. 删除 DELETE

表 6.143. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.50.3. 更新 PUT

更新系统中指定的 cpu 配置集。

表 6.144. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
profile	CpuProfile	in/Out	

6.51. CPUPROFILES

表 6.145. 方法概述

名称	概述
add	向系统添加新 cpu 配置集。
list	返回系统的 CPU 配置文件列表。

6.51.1. 添加 POST

向系统添加新 cpu 配置集。

表 6.146. 参数摘要

名称	类型	方向	概述
profile	CpuProfile	in/Out	

6.51.2. 列出 GET

返回系统的 CPU 配置文件列表。

CPU 配置文件返回列表的顺序是随机的。

表 6.147. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大配置集数。
profile	CpuProfile[]	out	

6.51.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.51.2.2. Max

设置要返回的最大配置集数。如果没有指定，则会返回所有配置集。

6.52. DATACENTER

用于管理数据中心的的服务。

表 6.148. 方法概述

名称	概述
cleanfinishedtasks	目前，如果 SPM 不参与任务，存储池管理器(SPM)无法切换到其他主机。
get	获取数据中心。
remove	移除数据中心。
setmaster	用于将数据中心中的存储域手动设置为主域。
update	更新数据中心。

6.52.1. 清理finishedtasks POST

目前，如果 SPM 不参与任务，存储池管理器(SPM)无法切换到其他主机。清除所有已完成的任务可启用 SPM 切换。

例如，要清理 ID 为 123 的数据中心中的所有已完成任务，请发送类似如下的请求：

```
POST /ovirt-engine/api/datacenters/123/cleanfinishedtasks
```

这样的请求正文：

```
<action/>
```

表 6.149. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.52.2. 获取 GET

获取数据中心。

获取数据中心的示例：

GET /ovirt-engine/api/datacenters/123

```
<data_center href="/ovirt-engine/api/datacenters/123" id="123">
  <name>Default</name>
  <description>The default Data Center</description>
  <link href="/ovirt-engine/api/datacenters/123/clusters" rel="clusters"/>
  <link href="/ovirt-engine/api/datacenters/123/storagedomains" rel="storagedomains"/>
  <link href="/ovirt-engine/api/datacenters/123/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/datacenters/123/networks" rel="networks"/>
  <link href="/ovirt-engine/api/datacenters/123/quotas" rel="quotas"/>
  <link href="/ovirt-engine/api/datacenters/123/qoss" rel="qoss"/>
  <link href="/ovirt-engine/api/datacenters/123/iscsibonds" rel="iscsibonds"/>
  <local>false</local>
  <quota_mode>disabled</quota_mode>
  <status>up</status>
  <storage_format>v3</storage_format>
  <supported_versions>
    <version>
      <major>4</major>
      <minor>0</minor>
    </version>
  </supported_versions>
  <version>
    <major>4</major>
    <minor>0</minor>
  </version>
  <mac_pool href="/ovirt-engine/api/macpools/456" id="456"/>
</data_center>
```

表 6.150. 参数摘要

名称	类型	方向	概述
data_center	DataCenter	out	

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。

6.52.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.52.3. 删除 DELETE

移除数据中心。

```
DELETE /ovirt-engine/api/datacenters/123
```

如果不使用任何特殊参数，附加到数据中心的存储域将被分离，然后从存储中删除。如果执行此操作时出现问题，例如没有主机可用于从存储中删除存储域，则完整的操作将失败。

如果 **force** 参数为 **true**，则操作将始终成功，即使某些操作在删除一个存储域时也会失败，例如：故障只是被忽略，数据中心仍然从数据库中删除。

表 6.151. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。
force	布尔值	in	指明操作是否应该成功，并且存储域从数据库中删除，即使操作期间出现问题。

6.52.3.1. force

指明操作是否应该成功，并且存储域从数据库中删除，即使操作期间出现问题。

此参数是可选的，默认值为 **false**。

6.52.4. setmaster POST

用于将数据中心中的存储域手动设置为主域。例如，若要在 ID 为 '123' 的数据中心上将 ID 为 "456" 的存储域设置为 **master**，请发送类似如下的请求：

```
POST /ovirt-engine/api/datacenters/123/setmaster
```

这样的请求正文：

```
<action>
  <storage_domain id="456"/>
</action>
```

也可以通过其名称来指定新的主存储域。

表 6.152. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
storage_domain	StorageDomain	in	数据中心的新主存储域。

6.52.5. 更新 PUT

更新数据中心。

name、**description**、**storage_type**、**version**、**storage_format** 和 **mac_pool** 元素是 **updatable** 后创建。例如，要更改数据中心 123 的名称和描述，请发送如下请求：

```
PUT /ovirt-engine/api/datacenters/123
```

这样的请求正文：

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

表 6.153. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
data_center	DataCenter	in/Out	正在更新的数据中心。

6.53. DATACENTERNETWORK

用于管理特定数据中心网络的服务。

表 6.154. 方法概述

名称	概述
get	检索数据中心网络详细信息。
remove	移除网络。
update	更新数据中心中的网络。

6.53.1. 获取 GET

检索数据中心网络详细信息。

表 6.155. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
network	Network	out	数据中心网络。

6.53.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.53.2. 删除 DELETE

移除网络。

6.53.3. 更新 PUT

更新数据中心中的网络。

表 6.156. 参数摘要

名称	类型	方向	概述
network	Network	in/Out	数据中心网络.

6.54. DATACENTERNETWORKS

用于管理数据中心网络的服务。

表 6.157. 方法概述

名称	概述
add	在数据中心中创建新网络。
list	列出数据中心中的网络.

6.54.1. 添加 POST

在数据中心中创建新网络。

在下例中发出一个请求，以在 ID 为 123 的数据中心中创建一个新网络。

```
POST /ovirt-engine/api/datacenters/123/networks
```

在正文中使用以下示例：

```
<network>
  <name>mynetwork</name>
</network>
```

表 6.158. 参数摘要

名称	类型	方向	概述
network	Network	in/Out	要在数据中心的网络对象。

6.54.2. 列出 GET

列出数据中心的网络。

网络返回列表的顺序无法保证。

表 6.159. 参数摘要

名称	类型	方向	概述
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的最大网络数。
网络	network[]	out	数据中心中的网络列表。

6.54.2.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.54.2.2. Max

设置要返回的最大网络数。 如果没有指定，则会返回所有网络。

6.55. DATACENTERS

用于管理数据中心的的服务。

表 6.160. 方法概述

名称	概述
add	创建新的数据中心。
list	列出数据中心。

6.55.1. 添加 POST

创建新的数据中心。

创建新数据中心需要名称和本地元素。例如，要创建名为 **mydc** 的数据中心，它使用共享存储（NFS、iSCSI 或光纤通道）发送如下的请求：

```
POST /ovirt-engine/api/datacenters
```

这样的请求正文：

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

表 6.161. 参数摘要

名称	类型	方向	概述
data_center	DataCenter	in/Out	正在添加的数据中心。

6.55.2. 列出 GET

列出数据中心。

以下请求检索数据中心的表示法：

```
GET /ovirt-engine/api/datacenters
```

以上请求通过 **curl** 执行：

```
curl \
--request GET \
--cacert /etc/pki/ovirt-engine/ca.pem \
--header "Version: 4" \
--header "Accept: application/xml" \
--user "admin@internal:mypassword" \
https://myengine.example.com/ovirt-engine/api/datacenters
```

以下是响应示例：

```
<data_center href="/ovirt-engine/api/datacenters/123" id="123">
  <name>Default</name>
  <description>The default Data Center</description>
  <link href="/ovirt-engine/api/datacenters/123/networks" rel="networks"/>
  <link href="/ovirt-engine/api/datacenters/123/storagedomains" rel="storagedomains"/>
  <link href="/ovirt-engine/api/datacenters/123/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/datacenters/123/clusters" rel="clusters"/>
  <link href="/ovirt-engine/api/datacenters/123/qoss" rel="qoss"/>
  <link href="/ovirt-engine/api/datacenters/123/iscsibonds" rel="iscsibonds"/>
  <link href="/ovirt-engine/api/datacenters/123/quotas" rel="quotas"/>
  <local>false</local>
  <quota_mode>disabled</quota_mode>
  <status>up</status>
  <supported_versions>
    <version>
      <major>4</major>
      <minor>0</minor>
    </version>
  </supported_versions>
  <version>
    <major>4</major>
    <minor>0</minor>
  </version>
</data_center>
```

注意您的 Default 数据中心的 id 代码。此代码相对于虚拟环境的其他资源标识此数据中心。

数据中心还包含指向存储域集合的链接。数据中心使用此集合从存储域主集合中附加存储域。

只有 search 参数中包含 sortby 子句时，才会保证返回的数据中心列表的顺序。

表 6.162. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
data_centers	DataCenter[]	out	
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。

名称	类型	方向	概述
max	整数	in	设置要返回的数据中心的最大数量。
search	字符串	in	用于限制返回的数据中心的查询字符串。

6.55.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 **search** 参数执行搜索。默认值为 **true**，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 **false**。

6.55.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.55.2.3. Max

设置要返回的数据中心的最大数量。如果没有指定，则会返回所有数据中心。

6.56. DISK

管理单个磁盘。

表 6.163. 方法概述

名称	概述
convert	转换磁盘格式和/或预分配模式。
copy	此操作将磁盘复制到指定的存储域。
export	将磁盘导出到导出存储域。
get	检索磁盘的描述。
Move	将磁盘移动到其他存储域。
减少	减少磁盘镜像的大小。
refreshlun	使用存储中的最新信息刷新直接 LUN 磁盘。

名称	概述
remove	删除磁盘。
sparsify	稀疏磁盘。
update	更新指定磁盘的参数。

6.56.1. 转换 POST

转换磁盘格式和/或预分配模式。

例如，要将磁盘格式从预分配转换为稀疏镜像，请发送如下请求：

```
POST /ovirt-engine/api/disks/123/convert
```

使用以下请求正文：

```
<action>
  <disk>
    <sparse>true</sparse>
    <format>raw</format>
  </disk>
</action>
```

表 6.164. 参数摘要

名称	类型	方向	概述
disk	Disk	in	磁盘的描述。
follow	字符串	in	指明应遵循哪些内联链接。

6.56.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.56.2. 复制 POST

此操作将磁盘复制到指定的存储域。

例如，可使用以下请求复制磁盘：

```
POST /ovirt-engine/api/disks/123/copy
```

这样的请求正文：

```
<action>
  <storage_domain id="456"/>
  <disk>
    <name>mydisk</name>
  </disk>
</action>
```

如果没有为新存储域定义磁盘配置文件或磁盘当前使用的配额，则可以明确指定它们。如果没有指定它们，则使用第一个可用磁盘配置文件和默认配额。

例如，要指定磁盘配置集 987 和配额 753，请发送如下请求正文：

```
<action>
  <storage_domain id="456"/>
  <disk_profile id="987"/>
  <quota id="753"/>
</action>
```

表 6.165. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明复制应异步执行。
disk	Disk	in	
disk_profile	DiskProfile	in	新存储域中磁盘的磁盘配置文件。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
quota	Quota	in	新存储域中的磁盘配额。
storage_domain	StorageDomain	in	创建新磁盘的存储域。

6.56.2.1. disk_profile

新存储域中磁盘的磁盘配置文件。

为存储域定义磁盘配置文件，因此旧磁盘配置文件不会存在于新存储域中。如果未使用此参数，则将向磁盘分配用户具有权限的新存储域中的第一个磁盘配置文件。

6.56.2.2. quota

新存储域中的磁盘配额。

此可选参数可用于指定磁盘的新配额，因为可能不会为新存储域定义当前的配额。如果未为新存储域定义此参数，并且未为新存储域定义旧配额，则将向磁盘分配默认（无限）配额。

6.56.2.3. storage_domain

创建新磁盘的存储域。这可以使用 `id` 或 `name` 属性指定。例如，要将磁盘复制到名为 `mydata` 的存储域，请发送类似如下的请求：

POST /ovirt-engine/api/storagedomains/123/disks/789

这样的请求正文：

```
<action>
  <storage_domain>
    <name>mydata</name>
  </storage_domain>
</action>
```

6.56.3. 导出 POST

将磁盘导出到导出存储域。

表 6.166. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行导出。
<code>filter</code>	布尔值	in	指明是否应根据用户的权限过滤结果。

名称	类型	方向	概述
storage_domain	StorageDomain	in	磁盘要导出到的导出存储域。

6.56.4. 获取 GET

检索磁盘的描述。

表 6.167. 参数摘要

名称	类型	方向	概述
all_content	布尔值	in	指明是否应将磁盘的所有属性包含在响应中。
disk	Disk	out	磁盘的描述。
follow	字符串	in	指明应遵循哪些内联链接。

6.56.4.1. all_content

指明是否应将磁盘的所有属性包含在响应中。

默认情况下，排除以下磁盘属性：

- **vms**

例如，要检索磁盘 '123' 的完整表示法：

```
GET /ovirt-engine/api/disks/123?all_content=true
```

6.56.4.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.56.5. 移动 POST

将磁盘移动到其他存储域。

例如，要将标识符为 **123** 的磁盘移动到标识符 **456** 的存储域中，发送以下请求：

```
POST /ovirt-engine/api/disks/123/move
```

使用以下请求正文：

```
<action>
  <storage_domain id="456"/>
</action>
```

如果没有为新存储域定义磁盘配置文件或当前磁盘使用的配额，则可以明确指定它们。如果没有，则使用第一个可用磁盘配置文件和默认配额。

例如，要显式使用磁盘 **profile 987**，配额 **753** 会发送如下请求正文：

```
<action>
  <storage_domain id="456"/>
  <disk_profile id="987"/>
  <quota id="753"/>
</action>
```

表 6.168. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移动。
disk_profile	DiskProfile	in	新存储域中磁盘的磁盘配置文件。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
quota	Quota	in	新存储域中的磁盘配额。
storage_domain	StorageDomain	in	磁盘将移动到的存储域。

6.56.5.1. disk_profile

新存储域中磁盘的磁盘配置文件。

为存储域定义磁盘配置文件，因此旧磁盘配置文件不会存在于新存储域中。如果未使用此参数，则将向磁盘分配用户具有权限的新存储域中的第一个磁盘配置文件。

6.56.5.2. quota

新存储域中的磁盘配额。

此可选参数可用于指定磁盘的新配额，因为可能不会为新存储域定义当前的配额。如果未为新存储域定义此参数，并且未为新存储域定义旧配额，则将向磁盘分配默认（无限）配额。

6.56.6. 减少 POST

减少磁盘镜像的大小。

在逻辑卷中调用 `less`（即，这仅适用于块存储域）。这适用于附加到未运行虚拟机的浮动磁盘和磁盘。无需指定大小，因为会自动计算最佳大小。

表 6.169. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行移除。

6.56.7. refreshlun POST

使用存储中的最新信息刷新直接 LUN 磁盘。

在以下情况下刷新直接 LUN 磁盘非常有用：

- LUN 在没有 `host` 参数的情况下使用 API 添加，因此不包含存储中的任何信息（请参阅 [DisksService::add](#)）。

- 有关 LUN 的新信息可在存储中获取，您希望使用它更新 LUN。

要使用主机 456 刷新直接 LUN 磁盘 123，请发送以下请求：

```
POST /ovirt-engine/api/disks/123/refreshlun
```

使用以下请求正文：

```
<action>
  <host id='456' />
</action>
```

表 6.170. 参数摘要

名称	类型	方向	概述
host	主机	in	用于刷新直接 LUN 磁盘的主机。

6.56.8. 删除 DELETE

删除磁盘。

表 6.171. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.56.9. 稀疏 POST

稀疏磁盘。

解析可在其文件系统不使用的磁盘映像中释放空间。因此，镜像会在存储中占用较少的空间。

目前稀疏仅适用于没有快照的磁盘。也不允许有派生磁盘的磁盘。

6.56.10. 更新 PUT

更新指定磁盘的参数。

此操作允许更新以下浮动磁盘属性：

- 对于映像磁盘：`provisioned_size`、`alias`、`description`、`wipe_after_delete`、可共享、备份和 `disk_profile`。
- LUN 磁盘：别名、说明 和可共享。
- Cinder 集成已被 Managed Block Storage 替代。
- 对于 Managed Block 磁盘：`provisioned_size`、别名 和描述。
- 对于虚拟机附加的磁盘，也可更新 `qcow_version`。

例如，可以使用以下请求完成磁盘的更新：

```
PUT /ovirt-engine/api/disks/123
```

这样的请求正文：

```
<disk>
  <qcow_version>qcow2_v3</qcow_version>
  <alias>new-alias</alias>
  <description>new-desc</description>
</disk>
```

由于后端操作是异步的，返回给用户的磁盘元素可能无法与更改的属性同步。

表 6.172. 参数摘要

名称	类型	方向	概述
<code>disk</code>	Disk	in/Out	应用到磁盘的更新。

6.57. DISKATTACHMENT

此服务管理磁盘附加到虚拟机。

表 6.173. 方法概述

名称	概述
get	返回附加的详细信息，包括可引导标志和磁盘链接。
remove	删除磁盘附加。
update	更新磁盘附加及其中的磁盘属性。

6.57.1. 获取 GET

返回附加的详细信息，包括可引导标志和磁盘链接。

获取磁盘附加的示例：

```
GET /ovirt-engine/api/vms/123/diskattachments/456
```

```
<disk_attachment href="/ovirt-engine/api/vms/123/diskattachments/456" id="456">
  <active>true</active>
  <bootable>true</bootable>
  <interface>virtio</interface>
  <disk href="/ovirt-engine/api/disks/456" id="456"/>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</disk_attachment>
```

表 6.174. 参数摘要

名称	类型	方向	概述
Attach	DiskAttachm ent	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.57.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.57.2. 删除 DELETE

删除磁盘附加。

这只会将磁盘从虚拟机分离，但不会将其从系统中删除，除非 `detach_only` 参数为 `false`。

删除磁盘附加的示例：

```
DELETE /ovirt-engine/api/vms/123/diskattachments/456?detach_only=true
```

表 6.175. 参数摘要

名称	类型	方向	概述
<code>detach_only</code>	布尔值	in	指明磁盘是否应该从虚拟机分离，但不应从系统中删除。

6.57.2.1. detach_only

指明磁盘是否应该从虚拟机分离，但不应从系统中删除。默认值为 `true`，这不会从系统中删除磁盘。

6.57.3. 更新 PUT

更新磁盘附加及其中的磁盘属性。

```
PUT /vms/{vm:id}/disksattachments/{attachment:id}
<disk_attachment>
  <bootable>true</bootable>
  <interface>ide</interface>
  <active>true</active>
  <disk>
    <name>mydisk</name>
    <provisioned_size>1024</provisioned_size>
    ...
  </disk>
</disk_attachment>
```

表 6.176. 参数摘要

名称	类型	方向	概述
disk_attachment	DiskAttachment	in/Out	

6.58. DISKATTACHMENTS

此服务管理附加到虚拟机的磁盘集合。每个附加的磁盘都由 [DiskAttachment](#) 表示，其中包含可引导标志、磁盘接口和磁盘引用。

表 6.177. 方法概述

名称	概述
add	将新磁盘附加添加到虚拟机。
list	列出附加到虚拟机的磁盘。

6.58.1. 添加 POST

将新磁盘附加添加到虚拟机。如果磁盘已存在，附加 参数可以只包含一个引用：

```
<disk_attachment>
  <bootable>true</bootable>
  <pass_discard>true</pass_discard>
  <interface>ide</interface>
  <active>true</active>
  <disk id="123"/>
</disk_attachment>
```

或者，如果磁盘还不存在，它可以包含磁盘的完整表示法：

```
<disk_attachment>
  <bootable>true</bootable>
  <pass_discard>true</pass_discard>
  <interface>ide</interface>
  <active>true</active>
  <disk>
    <name>mydisk</name>
    <provisioned_size>1024</provisioned_size>
    ...
  </disk>
</disk_attachment>
```

在这种情况下，将创建磁盘并附加到虚拟机。

在这两种情况下，为 ID 为 345 的虚拟机使用以下 URL：

POST /ovirt-engine/api/vms/345/diskattachments



重要

服务器接受不包含 **active** 属性但效果未定义的请求。在某些情况下，磁盘将被自动激活，在其他情况下则不会激活。为避免出现问题，强烈建议始终使用所需值包含 **active** 属性。

表 6.178. 参数摘要

名称	类型	方向	概述
Attach	DiskAttachment	in/Out	要添加到虚拟机的磁盘附件。

6.58.2. 列出 GET

列出附加到虚拟机的磁盘。

无法保证返回的磁盘附件列表的顺序。

表 6.179. 参数摘要

名称	类型	方向	概述
附件	DiskAttachment[]	out	附加到虚拟机的磁盘附件列表。
follow	字符串	in	指明应遵循哪些内联链接。

6.58.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.59. DISKPROFILE

表 6.180. 方法概述

名称	概述
get	
remove	
update	更新系统中指定的磁盘配置文件。

6.59.1. 获取 GET

表 6.181. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
profile	DiskProfile	out	

6.59.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.59.2. 删除 DELETE

表 6.182. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.59.3. 更新 PUT

更新系统中指定的磁盘配置文件。

表 6.183. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
profile	DiskProfile	in/Out	

6.60. DISKPROFILES

表 6.184. 方法概述

名称	概述
add	向系统添加新磁盘配置文件。
list	返回系统的磁盘配置文件列表。

6.60.1. 添加 POST

向系统添加新磁盘配置文件。

表 6.185. 参数摘要

名称	类型	方向	概述
profile	DiskProfile	in/Out	

6.60.2. 列出 GET

返回系统的磁盘配置文件列表。

无法保证返回的磁盘配置文件列表的顺序。

表 6.186. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大配置集数。
profile	DiskProfile[]	out	

6.60.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.60.2.2. Max

设置要返回的最大配置集数。如果没有指定，则返回所有配置集。

6.61. DISKSNAPSHOT

表 6.187. 方法概述

名称	概述
get	
remove	

6.61.1. 获取 GET

表 6.188. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
snapshot	DiskSnapshot	out	

6.61.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.61.2. 删除 DELETE

表 6.189. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.62. DISKSNAPSHOTS

管理存储域中可用的磁盘快照的集合。

表 6.190. 方法概述

名称	概述
list	返回存储域的磁盘快照列表。

6.62.1. 列出 GET

返回存储域的磁盘快照列表。

无法保证磁盘快照返回列表的顺序。

表 6.191. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
include_active	布尔值	in	如果 true 也返回活动快照。
include_template	布尔值	in	如果 true 也返回模板快照。
max	整数	in	设置要返回的快照的最大数量。
snapshots	DiskSnapshot[]	out	

6.62.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.62.1.2. include_active

如果 true 也返回活动快照。如果未指定活动快照，则不返回。

6.62.1.3. include_template

如果 true 也返回模板快照。如果没有返回指定的模板快照。

6.62.1.4. Max

设置要返回的快照的最大数量。如果没有指定，则返回所有快照。

6.63. 磁盘

管理系统中可用的磁盘集合。

表 6.192. 方法概述

名称	概述
add	添加新浮动磁盘。
list	获取磁盘列表。

6.63.1. 添加 POST

添加新浮动磁盘。

可以添加三种类型的磁盘 - 磁盘镜像、直接 LUN 和 Managed Block 磁盘。[Cinder](#) 集成已被 Managed Block Storage 替代。

添加新镜像磁盘：

在创建新的浮动映像 [磁盘](#) 时，API 需要 `storage_domain`、`provision_size` 和 `format` 属性。

请注意，块存储域（例如 iSCSI 或 FCP [存储类型的](#) 存储域）不支持将原始格式与 `sparse=true` 的组合，因此必须明确说明 `sparse=false`。

要使用指定的 `provisioned_size` 创建一个新的浮动镜像磁盘，在存储域上格式化和名称（id 123 并启用了增量备份），请发送如下请求：

```
POST /ovirt-engine/api/disks
```

请求正文如下：

```
<disk>
  <storage_domains>
    <storage_domain id="123"/>
  </storage_domains>
  <name>mydisk</name>
  <provisioned_size>1048576</provisioned_size>
  <format>cow</format>
  <backup>incremental</backup>
</disk>
```

添加新直接 LUN 磁盘：

在通过 API 添加新浮动直接 LUN 时，可以使用两种类别：

1. 对于 主机 元素 - 在这种情况下，主机用于完整性检查（例如，LUN 是可见的）并检索有关 LUN 的基本信息（如大小和串行）。
2. 如果没有 主机 元素 - 在这种情况下，操作仅为数据库，并且不会访问存储。

要使用 ID 为 123、指定 别名、类型和 logical_unit 的主机 元素创建一个新的浮动直接 LUN 磁盘（具有属性 地址、端口 和目标），请发送请求，如下所示：

```
POST /ovirt-engine/api/disks
```

请求正文如下：

```
<disk>
  <alias>mylun</alias>
  <lun_storage>
    <host id="123"/>
    <type>iscsi</type>
    <logical_units>
      <logical_unit id="456">
        <address>10.35.10.20</address>
        <port>3260</port>
        <target>iqn.2017-01.com.myhost:444</target>
      </logical_unit>
    </logical_units>
  </lun_storage>
</disk>
```

若要创建新的浮动直接 LUN 磁盘，而不使用主机，请删除 主机 元素。

添加新 Cinder 磁盘：

Cinder 集成已被 Managed Block Storage 替代。

添加浮动磁盘以上传磁盘快照：

由于引擎的版本 4.2，可以使用快照上传磁盘。此请求应当用于创建映像链的基础镜像（镜像），在创建快照时使用 `disk-attachments` 元素创建。

磁盘必须使用与上传镜像相同的磁盘标识符和映像标识符创建。i. 标识符应保存为备份过程的一部分。也可以使用 `qemu-img info` 命令来获取映像标识符。例如，如果磁盘映像存储在名为 `b7a4c6c5-443b-47c5-967f-6abc79675e8b/myimage.img` 的文件中：

```
$ qemu-img info b7a4c6c5-443b-47c5-967f-6abc79675e8b/myimage.img
image: b548366b-fb51-4b41-97be-733c887fe305
file format: qcow2
virtual size: 1.0G (1073741824 bytes)
disk size: 196K
cluster_size: 65536
backing file: ad58716a-1fe9-481f-815e-664de1df04eb
backing file format: raw
```

要使用上面显示的 `qemu-img info` 命令获取的磁盘标识符和映像标识符创建磁盘，请发送类似如下的请求：

```
POST /ovirt-engine/api/disks
```

请求正文如下：

```
<disk id="b7a4c6c5-443b-47c5-967f-6abc79675e8b">
  <image_id>b548366b-fb51-4b41-97be-733c887fe305</image_id>
  <storage_domains>
    <storage_domain id="123"/>
  </storage_domains>
  <name>mydisk</name>
```

```
<provisioned_size>1048576</provisioned_size>
<format>cow</format>
</disk>
```

表 6.193. 参数摘要

名称	类型	方向	概述
disk	Disk	in/Out	磁盘.

6.63.2. 列出 GET

获取磁盘列表.

GET /ovirt-engine/api/disks

您将获得类似如下的 XML 响应：

```
<disks>
  <disk id="123">
    <actions>...</actions>
    <name>MyDisk</name>
    <description>MyDisk description</description>
    <link href="/ovirt-engine/api/disks/123/permissions" rel="permissions"/>
    <link href="/ovirt-engine/api/disks/123/statistics" rel="statistics"/>
    <actual_size>5345845248</actual_size>
    <alias>MyDisk alias</alias>
    ...
    <status>ok</status>
    <storage_type>image</storage_type>
    <wipe_after_delete>false</wipe_after_delete>
    <disk_profile id="123"/>
    <quota id="123"/>
    <storage_domains>...</storage_domains>
  </disk>
  ...
</disks>
```

只有 `search` 参数中包含 `sortby` 子句时，才会保证返回的磁盘列表的顺序。

表 6.194. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
磁盘	disk[]	out	检索的磁盘列表。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的最大磁盘数。
search	字符串	in	用于限制返回的磁盘的查询字符串。

6.63.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 **search** 参数执行搜索。默认值为 **true**，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 **false**。

6.63.2.2. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.63.2.3. Max

设置要返回的最大磁盘数。如果没有指定，则返回所有磁盘。

6.64. 域

此服务可用于查看系统中身份验证域的详细信息。

表 6.195. 方法概述

名称	概述
get	获取身份验证域信息。

6.64.1. 获取 GET

获取身份验证域信息。

使用方法：

GET /ovirt-engine/api/domains/5678

将返回域信息：

```
<domain href="/ovirt-engine/api/domains/5678" id="5678">
  <name>internal-authz</name>
  <link href="/ovirt-engine/api/domains/5678/users" rel="users"/>
  <link href="/ovirt-engine/api/domains/5678/groups" rel="groups"/>
  <link href="/ovirt-engine/api/domains/5678/users?search={query}" rel="users/search"/>
  <link href="/ovirt-engine/api/domains/5678/groups?search={query}" rel="groups/search"/>
</domain>
```

表 6.196. 参数摘要

名称	类型	方向	概述
domain	域	out	身份验证域。
follow	字符串	in	指明应遵循哪些内联链接。

6.64.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.65. DOMAINGROUP

表 6.197. 方法概述

名称	概述
get	

6.65.1. 获取 GET

表 6.198. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
get	组	out	

6.65.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.66. DOMAINGROUPS

表 6.199. 方法概述

名称	概述
list	返回组列表。

6.66.1. 列出 GET

返回组列表。

无法保证返回的组列表的顺序。

表 6.200. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
follow	字符串	in	指明应遵循哪些内联链接。
groups	group[]	out	
max	整数	in	设置要返回的最大组数。
search	字符串	in	用于限制返回的组的查询字符串。

6.66.1.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 `search` 参数执行搜索。默认值为 `true`，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 `false`。

6.66.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.66.1.3. Max

设置要返回的最大组数。如果没有指定，则返回所有组。

6.67. DOMAINUSER

用于查看系统中域用户的服务。

表 6.201. 方法概述

名称	概述
<code>get</code>	获取域用户信息.

6.67.1. 获取 GET

获取域用户信息.

使用方法：

GET /ovirt-engine/api/domains/5678/users/1234

将返回域用户信息：

```
<user href="/ovirt-engine/api/users/1234" id="1234">
  <name>admin</name>
  <namespace>*</namespace>
  <principal>admin</principal>
  <user_name>admin@internal-authz</user_name>
  <domain href="/ovirt-engine/api/domains/5678" id="5678">
    <name>internal-authz</name>
```

```

</domain>
<groups/>
</user>

```

表 6.202. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
user	用户	out	域用户。

6.67.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.68. DOMAINUSERGROUPS

在 AAA 扩展中显示用户的组成员资格的服务。

表 6.203. 方法概述

名称	概述
list	返回用户所属组的列表：

6.68.1. 列出 GET

返回用户所属组的列表：

表 6.204. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
groups	group[]	out	用户所属组的列表。

6.68.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.69. DOMAINUSERS

用于列出系统中的所有域用户的服务。

表 6.205. 方法概述

名称	概述
list	列出 域中的所有用户。

6.69.1. 列出 GET

列出 域中的所有用户。

使用方法：

GET /ovirt-engine/api/domains/5678/users

将返回域中的用户列表：

```
<users>
  <user href="/ovirt-engine/api/domains/5678/users/1234" id="1234">
    <name>admin</name>
    <namespace>*</namespace>
    <principal>admin</principal>
    <user_name>admin@internal-authz</user_name>
    <domain href="/ovirt-engine/api/domains/5678" id="5678">
      <name>internal-authz</name>
    </domain>
    <groups/>
  </user>
</users>
```

无法保证返回的用户列表的顺序。

表 6.206. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大用户数。
search	字符串	in	用于限制返回的用户的查询字符串。
用户	user[]	out	域中的用户列表。

6.69.1.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 **search** 参数执行搜索。默认值为 **true**，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 **false**。

6.69.1.2. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.69.1.3. Max

设置要返回的最大用户数。如果未指定，则返回所有用户。

6.70. 域

用于列出系统中的所有身份验证域的服务。

表 6.207. 方法概述

名称	概述
list	列出系统中的所有身份验证域。

6.70.1. 列出 GET

列出系统中的所有身份验证域。

使用方法：

GET /ovirt-engine/api/domains

将返回域列表：

```
<domains>
  <domain href="/ovirt-engine/api/domains/5678" id="5678">
    <name>internal-authz</name>
    <link href="/ovirt-engine/api/domains/5678/users" rel="users"/>
    <link href="/ovirt-engine/api/domains/5678/groups" rel="groups"/>
    <link href="/ovirt-engine/api/domains/5678/users?search={query}" rel="users/search"/>
    <link href="/ovirt-engine/api/domains/5678/groups?search={query}" rel="groups/search"/>
  </domain>
</domains>
```

无法保证返回的域列表的顺序。

表 6.208. 参数摘要

名称	类型	方向	概述
domains	Domain[]	out	域列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大域数。

6.70.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.70.1.2. Max

设置要返回的最大域数。如果未指定，则返回所有域。

6.71. ENGINEKATELLOERRATA

此服务用于管理分配给引擎的 Katello 勘误表。该信息是从 Katello 检索而来。

表 6.209. 方法概述

名称	概述
list	检索 Katello 勘误表的表示形式。

6.71.1. 列出 GET

检索 Katello 勘误表的表示形式。

GET /ovirt-engine/api/katelloerrata

您将在 XML 中收到类似以下的响应：

```
<katello_errata>
  <katello_erratum href="/ovirt-engine/api/katelloerrata/123" id="123">
    <name>RHBA-2013:XYZ</name>
    <description>The description of the erratum</description>
    <title>some bug fix update</title>
    <type>bugfix</type>
    <issued>2013-11-20T02:00:00.000+02:00</issued>
    <solution>Few guidelines regarding the solution</solution>
    <summary>Updated packages that fix one bug are now available for XYZ</summary>
    <packages>
      <package>
        <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
      </package>
      ...
    </packages>
  </katello_erratum>
  ...
</katello_errata>
```

返回的勘误列表的顺序无法保证。

表 6.210. 参数摘要

名称	类型	方向	概述
勘误	KatelloErratum[]	out	Katello 勘误表的表示。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的勘误表的最大数量。

6.71.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.71.1.2. Max

设置要返回的勘误表的最大数量。如果没有指定，则会返回所有勘误。

6.72. EVENT

用于管理系统中事件的服务。

表 6.211. 方法概述

名称	概述
get	获取事件。
remove	从内部审计日志中移除事件。

6.72.1. 获取 GET

获取事件。

获取事件的示例：

```
GET /ovirt-engine/api/events/123
```

```
<event href="/ovirt-engine/api/events/123" id="123">
  <description>Host example.com was added by admin@internal-authz.</description>
  <code>42</code>
  <correlation_id>135</correlation_id>
  <custom_id>-1</custom_id>
  <flood_rate>30</flood_rate>
  <origin>oVirt</origin>
  <severity>normal</severity>
  <time>2016-12-11T11:13:44.654+02:00</time>
  <cluster href="/ovirt-engine/api/clusters/456" id="456"/>
  <host href="/ovirt-engine/api/hosts/789" id="789"/>
  <user href="/ovirt-engine/api/users/987" id="987"/>
</event>
```

请注意，字段数量根据事件中驻留的信息而变化。例如，对于与存储域相关的事件，您将获得存储域参考以及该存储域所在数据中心的引用。

表 6.212. 参数摘要

名称	类型	方向	概述
event	event	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.72.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.72.2. 删除 DELETE

从内部审计日志中移除事件。

通过发送以下请求可以删除事件

```
DELETE /ovirt-engine/api/events/123
```

表 6.213. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.73. EVENTSUBSCRIPTION

在系统中管理特定事件订阅的服务。

表 6.214. 方法概述

名称	概述
get	获取与 event-subscription 相关的信息。
remove	从系统中删除 event-subscription。

6.73.1. 获取 GET

获取与 `event-subscription` 相关的信息。

例如，要将用户 '123' 订阅到事件 'vm_console_detected' 的信息检索到事件 'vm_console_detected'：

```
GET /ovirt-engine/api/users/123/vm_console_detected
```

```
<event-subscription href="/ovirt-engine/api/users/123/event-subscriptions/vm_console_detected">
  <event>vm_console_detected</event>
  <notification_method>smtp</notification_method>
  <user href="/ovirt-engine/api/users/123" id="123"/>
  <address>a@b.com</address>
</event-subscription>
```

表 6.215. 参数摘要

名称	类型	方向	概述
<code>event_subscription</code>	<code>EventSubscription</code>	out	event-subscription。

6.73.2. 删除 DELETE

从系统中删除 `event-subscription`。

例如，要删除用户 123 的订阅到 `vm_console_detected` 事件：

```
DELETE /ovirt-engine/api/users/123/vm_console_detected
```

表 6.216. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行移除。

6.74. EVENTSUBSCRIPTIONS

代表用于管理用户事件订阅集合的服务。

表 6.217. 方法概述

名称	概述
add	向系统添加新的 event-subscription。
list	列出提供的用户的 event-subscriptions。

6.74.1. 添加 POST

向系统添加新的 event-subscription。

event-subscription 始终在用户的上下文中添加。例如，要为用户 123 为 host_high_cpu_use 添加新的 event-subscription，并将通知发送到电子邮件地址：a@b.com，发送如下请求：

POST /ovirt-engine/api/users/123/eventsubscriptions

这样的请求正文：

```
<event_subscription>
  <event>host_high_cpu_use</event>
  <address>a@b.com</address>
</event_subscription>
```

该事件名称将成为新 event-subscription 实体的 ID：**GET ... /api/users/123/eventsubscriptions/host_high_cpu_use**

请注意，请求正文中不提供用户 ID。这是因为，API 已经从上下文中了解 user-id（本例中为 123）。另请注意，event-subscription 实体包含 notification-method 字段，但它也不会请求正文中提供。这是因为当前总是设置为 SMTP，因为 API 层仍不支持 SNMP 通知。

表 6.218. 参数摘要

名称	类型	方向	概述
event_subscription	EventSubscription	in/Out	添加的 event-subscription。

6.74.2. 列出 GET

列出提供的用户的 event-subscriptions。

例如，列出用户 123 的 event-subscriptions：

GET /ovirt-engine/api/users/123/event-subscriptions

```
<event-subscriptions>
  <event-subscription href="/ovirt-engine/api/users/123/event-subscriptions/host_install_failed">
    <event>host_install_failed</event>
    <notification_method>smtp</notification_method>
    <user href="/ovirt-engine/api/users/123" id="123"/>
    <address>a@b.com</address>
  </event-subscription>
  <event-subscription href="/ovirt-engine/api/users/123/event-subscriptions/vm_paused">
    <event>vm_paused</event>
    <notification_method>smtp</notification_method>
    <user href="/ovirt-engine/api/users/123" id="123"/>
    <address>a@b.com</address>
  </event-subscription>
</event-subscriptions>
```

表 6.219. 参数摘要

名称	类型	方向	概述
event_subscriptions	EventSubscription[]	out	指定用户的 event-subscriptions 列表
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大 event-subscriptions 数。

6.74.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.74.2.2. Max

设置要返回的最大 event-subscriptions 数。如果没有指定所有 event-subscriptions，则会返回。

6.75. 事件

用于管理系统中事件的服务。

表 6.220. 方法概述

名称	概述
add	将外部事件添加到内部审计日志中。
list	获取事件列表。
undelele	

6.75.1. 添加 POST

将外部事件添加到内部审计日志中。

这可用于与检测或生成与系统管理员相关的事件的外部系统集成。例如，外部监控工具可以检测文件系统是否完全在虚拟机的客户机操作系统内。此事件可添加到内部审计日志中，如下所示发送请求：

```
POST /ovirt-engine/api/events
<event>
  <description>File system /home is full</description>
  <severity>alert</severity>
  <origin>mymonitor</origin>
  <custom_id>1467879754</custom_id>
</event>
```

事件也可以链接到特定的对象。例如，以上事件可使用 **vm** 链接到发生该事件的特定虚拟机：

```
POST /ovirt-engine/api/events
<event>
  <description>File system /home is full</description>
  <severity>alert</severity>
  <origin>mymonitor</origin>
  <custom_id>1467879754</custom_id>
  <vm id="aae98225-5b73-490d-a252-899209af17e9"/>
</event>
```



注意

使用链接时，如上例中的 **vm**，仅接受 **id** 属性。**name** 属性（如果提供）被简单地忽略。

表 6.221. 参数摘要

名称	类型	方向	概述
event	event	in/Out	

6.75.2. 列出 GET

获取事件列表.

GET /ovirt-engine/api/events

对上述请求，我们得到以下响应：

```
<events>
  <event href="/ovirt-engine/api/events/2" id="2">
    <description>User admin@internal-authz logged out.</description>
    <code>31</code>
    <correlation_id>1e892ea9</correlation_id>
    <custom_id>-1</custom_id>
    <flood_rate>30</flood_rate>
    <origin>oVirt</origin>
    <severity>normal</severity>
    <time>2016-09-14T12:14:34.541+02:00</time>
    <user href="/ovirt-engine/api/users/57d91d48-00da-0137-0138-000000000244" id="57d91d48-00da-0137-0138-000000000244"/>
  </event>
  <event href="/ovirt-engine/api/events/1" id="1">
    <description>User admin logged in.</description>
    <code>30</code>
    <correlation_id>1fbd81f4</correlation_id>
    <custom_id>-1</custom_id>
    <flood_rate>30</flood_rate>
    <origin>oVirt</origin>
    <severity>normal</severity>
    <time>2016-09-14T11:54:35.229+02:00</time>
    <user href="/ovirt-engine/api/users/57d91d48-00da-0137-0138-000000000244" id="57d91d48-00da-0137-0138-000000000244"/>
  </event>
</events>
```

发生以下事件：

- ID="1" - admin 用户帐户中的 API 日志。

- **ID="2" - API 注销 admin 用户帐户。**

返回的事件列表的顺序始终是巨大的。如果 `search` 参数中包含 `sortby` 子句，则事件将根据该子句排序。如果没有包括 `sortby` 子句，则事件将按 `id` 属性的数字值排序，从最高值开始。这与 `max` 参数相结合，简化了获取最新事件的过程：

GET /ovirt-engine/api/events?max=1

表 6.222. 参数摘要

名称	类型	方向	概述
<code>case_sensitive</code>	布尔值	in	指明是否应该在考虑大小写的情况下使用 <code>search</code> 参数执行搜索。
事件	<code>Event[]</code>	out	
<code>follow</code>	字符串	in	<i>指明应遵循哪些内联链接。</i>
来自	整数	in	指定应返回的事件索引。
<code>max</code>	整数	in	设置要返回的最大事件数。
<code>search</code>	字符串	in	<code>events</code> 服务提供类似于其他资源服务的搜索查询。

6.75.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 `search` 参数执行搜索。默认值为 `true`，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 `false`。

6.75.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.75.2.3. 来自

指定应返回的事件索引。事件索引严格增加，因此当仅返回具有更大索引的事件时，才会返回此参数。例如，以下请求将仅返回索引大于 123 的事件：

GET /ovirt-engine/api/events?from=123

这个参数是可选的，如果没有指定，则返回的第一个事件将最近生成。

6.75.2.4. Max

设置要返回的最大事件数。如果没有指定，则返回所有事件。

6.75.2.5. search

events 服务提供类似于其他资源服务的搜索查询。

我们可以通过提供特定的严重性来搜索。

```
GET /ovirt-engine/api/events?search=severity%3Dnormal
```

在上述请求中，我们获得一个严重性等于正常的事件列表：

```
<events>
  <event href="/ovirt-engine/api/events/2" id="2">
    <description>User admin@internal-authz logged out.</description>
    <code>31</code>
    <correlation_id>1fbd81f4</correlation_id>
    <custom_id>-1</custom_id>
    <flood_rate>30</flood_rate>
    <origin>oVirt</origin>
    <severity>normal</severity>
    <time>2016-09-14T11:54:35.229+02:00</time>
    <user href="/ovirt-engine/api/users/57d91d48-00da-0137-0138-000000000244" id="57d91d48-00da-0137-0138-000000000244"/>
  </event>
  <event href="/ovirt-engine/api/events/1" id="1">
    <description>Affinity Rules Enforcement Manager started.</description>
    <code>10780</code>
    <custom_id>-1</custom_id>
    <flood_rate>30</flood_rate>
    <origin>oVirt</origin>
    <severity>normal</severity>
    <time>2016-09-14T11:52:18.861+02:00</time>
  </event>
</events>
```

一段时间后，虚拟化环境会产生大量事件。但是，API 仅显示一个搜索查询的默认事件数。若要显示超过默认值，API 在搜索查询中使用 **page** 命令将结果分隔到页面中。以下搜索查询告知 API 使用页面值和 **sortby** 子句来分页结果：

```
sortby time asc page 1
```

以下示例对事件资源进行分页。URL 编码的请求是：

```
GET /ovirt-engine/api/events?search=sortby%20time%20asc%20page%201
```

增加页面值以查看下一页结果。

```
GET /ovirt-engine/api/events?search=sortby%20time%20asc%20page%202
```

6.75.3. 取消删除 POST

表 6.223. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行取消删除。

6.76. EXTERNALCOMPUTERESOURCE

管理单个外部计算资源。

计算资源是一个主机外部提供程序术语。外部提供程序还需要知道调配的主机需要注册的位置。引擎的登录详情保存为外部提供程序端的计算资源。

表 6.224. 方法概述

名称	概述
<code>get</code>	检索外部计算资源详细信息。

6.76.1. 获取 GET

检索外部计算资源详细信息。

例如，要获取 供应商 123 的计算资源详情，请发送类似如下的请求：

```
GET /ovirt-engine/api/externalhostproviders/123/computeresources/234
```

它将返回类似如下的响应：

```
<external_compute_resource href="/ovirt-engine/api/externalhostproviders/123/computeresources/234" id="234">
  <name>hostname</name>
  <provider>oVirt</provider>
  <url>https://hostname/api</url>
  <user>admin@internal</user>
  <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
</external_compute_resource>
```

表 6.225. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
resource	ExternalComputeResource	out	外部计算资源信息

6.76.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.77. EXTERNALCOMPUTERESOURCES

管理一系列外部计算资源。

计算资源是一个主机外部提供程序术语。外部提供程序还需要知道调配的主机需要注册的位置。引擎的登录详情保存为外部提供程序端的计算资源。

表 6.226. 方法概述

名称	概述
list	检索外部计算资源的列表。

6.77.1. 列出 GET

检索外部计算资源的列表。

例如，要检索外部主机供应商 123 的计算资源，请发送如下请求：

```
GET /ovirt-engine/api/externalhostproviders/123/computeresources
```

它将返回类似如下的响应：

```
<external_compute_resources>
  <external_compute_resource href="/ovirt-
engine/api/externalhostproviders/123/computeresources/234" id="234">
    <name>hostname</name>
    <provider>oVirt</provider>
    <url>https://address/api</url>
    <user>admin@internal</user>
    <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
  </external_compute_resource>
  ...
</external_compute_resources>
```

无法保证返回的计算资源列表的顺序。

表 6.227. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的资源的最大数量。
资源	ExternalComputeResource[]	out	外部计算机资源列表。

6.77.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.77.1.2. Max

设置要返回的资源的最大数量。如果没有指定，则返回所有资源。

6.78. EXTERNALDISCOVEREDHOST

此服务管理单个发现的主机。

表 6.228. 方法概述

名称	概述
get	获取发现的主机信息.

6.78.1. 获取 GET

获取发现的主机信息.

检索有关在外部提供程序管理系统中管理的主机的信息，如 Foreman。该信息包括主机名、地址、子网、基础镜像等。

例如，要从供应商 123 获取主机 234 的详细信息，请发送类似如下的请求：

```
GET /ovirt-engine/api/externalhostproviders/123/discoveredhosts/234
```

结果如下：

```
<external_discovered_host href="/ovirt-engine/api/externalhostproviders/123/discoveredhosts/234"
id="234">
  <name>mac001a4ad04040</name>
  <ip>10.34.67.43</ip>
  <last_report>2017-04-24 11:05:41 UTC</last_report>
  <mac>00:1a:4a:d0:40:40</mac>
  <subnet_name>sat0</subnet_name>
  <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
</external_discovered_host>
```

表 6.229. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
host	ExternalDisc overedHost	out	主机的硬件和配置信息。

6.78.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.79. EXTERNALDISCOVEREDHOSTS

此服务管理外部发现的主机。

表 6.230. 方法概述

名称	概述
<code>list</code>	获取发现的主机信息列表。

6.79.1. 列出 GET

获取发现的主机信息列表。

从 Foreman 等第三方提供程序获取发现的主机。

要列出供应商 123 的所有发现的主机，请执行以下操作：

```
GET /ovirt-engine/api/externalhostproviders/123/discoveredhost
```

```
<external_discovered_hosts>
  <external_discovered_host href="/ovirt-engine/api/externalhostproviders/123/discoveredhosts/456"
  id="456">
    <name>mac001a4ad04031</name>
    <ip>10.34.67.42</ip>
    <last_report>2017-04-24 11:05:41 UTC</last_report>
    <mac>00:1a:4a:d0:40:31</mac>
    <subnet_name>sat0</subnet_name>
    <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
  </external_discovered_host>
  <external_discovered_host href="/ovirt-engine/api/externalhostproviders/123/discoveredhosts/789"
  id="789">
    <name>mac001a4ad04040</name>
    <ip>10.34.67.43</ip>
    <last_report>2017-04-24 11:05:41 UTC</last_report>
    <mac>00:1a:4a:d0:40:40</mac>
    <subnet_name>sat0</subnet_name>
    <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
  </external_discovered_host>
  ...
</external_discovered_hosts>
```

无法保证返回的主机列表的顺序。

表 6.231. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
主机	ExternalDisc overedHost[]	out	发现的主机列表
max	整数	in	设置要返回的最大主机数。

6.79.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.79.1.2. Max

设置要返回的最大主机数。如果未指定，则返回所有主机。

6.80. EXTERNALHOST

表 6.232. 方法概述

名称	概述
get	

6.80.1. 获取 GET

表 6.233. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
host	ExternalHost	out	

6.80.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。详情请查看[此处](#)。

6.81. EXTERNALHOSTGROUP

此服务管理单个主机组信息。

主机组是主机提供程序的术语，主机组中包含应用于新发现主机的调配详细信息。子网、操作系统、域等信息。

表 6.234. 方法概述

名称	概述
get	获取主机组信息。

6.81.1. 获取 GET

获取主机组信息。

例如，要获取供应商 123 的 hostgroup 234 的详细信息，请发送类似如下的请求：

```
GET /ovirt-engine/api/externalhostproviders/123/hostgroups/234
```

它将返回类似如下的响应：

```
<external_host_group href="/ovirt-engine/api/externalhostproviders/123/hostgroups/234" id="234">
  <name>rhel7</name>
  <architecture_name>x86_64</architecture_name>
  <domain_name>s.com</domain_name>
  <operating_system_name>RedHat 7.3</operating_system_name>
  <subnet_name>sat0</subnet_name>
  <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
</external_host_group>
```

表 6.235. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。

名称	类型	方向	概述
group	ExternalHost Group	out	主机组信息.

6.81.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.82. EXTERNALHOSTGROUPS

此服务管理主机组。

表 6.236. 方法概述

名称	概述
list	从外部主机提供程序获取主机组列表。

6.82.1. 列出 GET

从外部主机提供程序获取主机组列表。

主机组是主机提供程序的一个术语 - 主机组中包含调配详细信息。此 API 返回外部提供程序公开的所有可能的主机组。

例如，要获取供应商 123 的所有主机组的详情，请发送如下请求：

```
GET /ovirt-engine/api/externalhostproviders/123/hostgroups
```

回答如下：

```
<external_host_groups>
  <external_host_group href="/ovirt-engine/api/externalhostproviders/123/hostgroups/234" id="234">
    <name>rhel7</name>
    <architecture_name>x86_64</architecture_name>
    <domain_name>example.com</domain_name>
```

```

<operating_system_name>RedHat 7.3</operating_system_name>
<subnet_name>sat0</subnet_name>
<external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
</external_host_group>
...
</external_host_groups>

```

主机组返回列表的顺序不保证。

表 6.237. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
groups	ExternalHost Group[]	out	可用于外部主机提供程序的所有主机组的列表
max	整数	in	设置要返回的最大组数。

6.82.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.82.1.2. Max

设置要返回的最大组数。如果没有指定，则返回所有组。

6.83. EXTERNALHOSTPROVIDER

代表外部主机提供程序，如 Foreman 或卫星。

有关 Foreman 的更多详细信息，请参见 <https://www.theforeman.org/>。有关 Red Hat Satellite 的详情，请查看 <https://access.redhat.com/products/red-hat-satellite>。

表 6.238. 方法概述

名称	概述
get	<p>获取外部主机提供程序信息</p> <p>主机提供程序 Foreman 或卫星可以在 ovirt 中设置为外部提供程序。</p>

名称	概述
importcertificates	导入外部主机提供程序的 SSL 证书。
remove	
testconnectivity	为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。
update	更新系统中指定的外部主机供应商。

6.83.1. 获取 GET

获取外部主机提供程序信息

主机提供程序 Foreman 或卫星可以在 ovirt 中设置为外部提供程序。要查看附加到 ovirt 的特定主机提供程序的详细信息，请使用这个 API。

例如，要获取主机供应商 123 的详细信息，请发送类似如下的请求：

```
GET /ovirt-engine/api/externalhostproviders/123
```

回答如下：

```
<external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123">
  <name>mysatellite</name>
  <requires_authentication>true</requires_authentication>
  <url>https://mysatellite.example.com</url>
  <username>admin</username>
</external_host_provider>
```

表 6.239. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
provider	ExternalHost Provider	out	

6.83.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.83.2. importcertificates POST

导入外部主机提供程序的 SSL 证书。

表 6.240. 参数摘要

名称	类型	方向	概述
证书	certificate[]	in	

6.83.3. 删除 DELETE

表 6.241. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.83.4. 测试连接 POST

为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。

```
POST /ovirt-engine/api/externalhostproviders/123/testconnectivity
```

表 6.242. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行测试。

6.83.5. 更新 PUT

更新系统中指定的外部主机供应商。

表 6.243. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。

名称	类型	方向	概述
provider	ExternalHost Provider	in/Out	

6.84. EXTERNALHOSTPROVIDERS

表 6.244. 方法概述

名称	概述
add	向系统添加新的外部主机提供程序。
list	返回外部主机提供程序列表。

6.84.1. 添加 POST

向系统添加新的外部主机提供程序。

表 6.245. 参数摘要

名称	类型	方向	概述
provider	ExternalHost Provider	in/Out	

6.84.2. 列出 GET

返回外部主机提供程序列表。

主机提供程序返回列表的顺序不受保证。

表 6.246. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大供应商数量。
供应商	ExternalHost Provider[]	out	

名称	类型	方向	概述
search	字符串	in	用于限制返回的外部主机供应商的查询字符串。

6.84.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.84.2.2. Max

设置要返回的最大供应商数量。如果没有指定，则返回所有供应商。

6.85. EXTERNALHOSTS

表 6.247. 方法概述

名称	概述
list	返回外部主机列表。

6.85.1. 列出 GET

返回外部主机列表。

无法保证返回的主机列表的顺序。

表 6.248. 参数摘要

名称	类型	方向	概述
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
主机	ExternalHost []	out	
max	整数	in	设置要返回的最大主机数。

6.85.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.85.1.2. Max

设置要返回的最大主机数。如果未指定，则返回所有主机。

6.86. EXTERNALNETWORKPROVIDERCONFIGURATION

描述主机上系统如何调配外部网络提供程序。

表 6.249. 方法概述

名称	概述
get	返回主机上外部网络提供程序的信息。

6.86.1. 获取 GET

返回主机上外部网络提供程序的信息。

表 6.250. 参数摘要

名称	类型	方向	概述
配置	ExternalNetworkProviderConfiguration	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.86.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.87. EXTERNALNETWORKPROVIDERCONFIGURATIONS

此服务用于列出系统在主机上调配的所有外部网络提供程序。

表 6.251. 方法概述

名称	概述
list	返回主机上所有外部网络提供程序的列表。

6.87.1. 列出 GET

返回主机上所有外部网络提供程序的列表。

无法保证返回的网络列表的顺序。

表 6.252. 参数摘要

名称	类型	方向	概述
配置	ExternalNetworkProviderConfiguration[]	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.87.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.88. EXTERNALPROVIDER

提供管理外部提供程序的功能。

表 6.253. 方法概述

名称	概述
importcertificates	导入外部主机提供程序的 SSL 证书。
testconnectivity	为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。

6.88.1. importcertificates POST

导入外部主机提供程序的 SSL 证书。

表 6.254. 参数摘要

名称	类型	方向	概述
证书	certificate[]	in	

6.88.2. 测试连接 POST

为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。

```
POST /ovirt-engine/api/externalhostproviders/123/testconnectivity
```

表 6.255. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行测试。

6.89. EXTERNALPROVIDERCERTIFICATE

用于查看外部提供程序特定证书的服务。

表 6.256. 方法概述

名称	概述
get	获取特定证书。

6.89.1. 获取 GET

获取特定证书。

```
GET /ovirt-engine/api/externalhostproviders/123/certificate/0
```

以下是响应示例：

```
<certificate id="0">
```

```
<organization>provider.example.com</organization>
<subject>CN=provider.example.com</subject>
<content>...</content>
</certificate>
```

表 6.257. 参数摘要

名称	类型	方向	概述
certificate	证书	out	证书的详细信息。
follow	字符串	in	指明应遵循哪些内联链接。

6.89.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.90. EXTERNALPROVIDERCERTIFICATES

用于查看外部供应商证书的服务。

表 6.258. 方法概述

名称	概述
list	返回外部供应商提供的证书链。

6.90.1. 列出 GET

返回外部供应商提供的证书链。

```
GET /ovirt-engine/api/externalhostproviders/123/certificates
```

以下是响应示例：

```
<certificates>
  <certificate id="789">...</certificate>
  ...
</certificates>
```

返回的证书的顺序始终被保证为符号顺序：第一个是服务器本身的证书，第二个是签署第一个证书的 CA 的证书，以此类推。

表 6.259. 参数摘要

名称	类型	方向	概述
证书	certificate[]	out	包含证书详细信息的列表。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的最大证书数。

6.90.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.90.1.2. Max

设置要返回的最大证书数。如果没有指定，则返回所有证书。

6.91. EXTERNALTEMPLATEIMPORTS

提供导入外部模板的功能。目前仅支持 OVA。

表 6.260. 方法概述

名称	概述
add	此操作用于从外部虚拟机监控程序导入模板。

6.91.1. 添加 POST

此操作用于从外部虚拟机监控程序导入模板。

例如，可以使用以下请求来简化模板 OVA 的导入：

```
POST /externaltemplateimports
```

使用 `ExternalTemplateImport` 类型的请求正文，例如：

```
<external_template_import>
  <template>
    <name>my_template</name>
  </template>
  <cluster id="2b18aca2-4469-11eb-9449-482ae35a5f83" />
  <storage_domain id="8bb5ade5-e988-4000-8b93-dbf6717fe50" />
  <url>ova:///mnt/ova/ova_template.ova</url>
  <host id="8bb5ade5-e988-4000-8b93-dbf6717fe50" />
</external_template_import>
```

表 6.261. 参数摘要

名称	类型	方向	概述
import	ExternalTemplateImport	in/Out	

6.92. EXTERNALVMIMPORTS

提供导入外部虚拟机的功能。

表 6.262. 方法概述

名称	概述
add	此操作用于从外部管理程序（如 KVM、XEN 或 VMware）导入虚拟机。

6.92.1. 添加 POST

此操作用于从外部管理程序（如 KVM、XEN 或 VMware）导入虚拟机。

例如，可以使用以下请求来简化从 VMware 导入虚拟机：

```
POST /externalvmimports
```

使用 **ExternalVmImport** 类型的请求正文，例如：

```
<external_vm_import>
  <vm>
    <name>my_vm</name>
  </vm>
  <cluster id="360014051136c20574f743bdbd28177fd" />
  <storage_domain id="8bb5ade5-e988-4000-8b93-dbf6717fe50" />
```

```

<name>vm_name_as_is_in_vmware</name>
<sparse>true</sparse>
<username>vmware_user</username>
<password>123456</password>
<provider>VMWARE</provider>
<url>vpx://wmware_user@vcenter-host/DataCenter/Cluster/esxi-host?no_verify=1</url>
<drivers_iso id="virtio-win-1.6.7.iso" />
</external_vm_import>

```

表 6.263. 参数摘要

名称	类型	方向	概述
import	ExternalVml mport	in/Out	

6.93. FENCEAGENT

用于管理特定主机的隔离代理的服务。

表 6.264. 方法概述

名称	概述
get	获取该隔离代理的详情。
remove	删除特定主机的隔离代理。
update	更新 fence-agent。

6.93.1. 获取 GET

获取该隔离代理的详情。

```
GET /ovirt-engine/api/hosts/123/fenceagents/0
```

以下是响应示例：

```

<agent id="0">
  <type>apc</type>
  <order>1</order>
  <ip>192.168.1.101</ip>
  <user>user</user>
  <password>xxx</password>

```

```
<port>9</port>
<options>name1=value1, name2=value2</options>
</agent>
```

表 6.265. 参数摘要

名称	类型	方向	概述
agent	agent	out	隔离代理详细信息.
follow	字符串	in	指明应遵循哪些内联链接。

6.93.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.93.2. 删除 DELETE

删除特定主机的隔离代理。

```
DELETE /ovirt-engine/api/hosts/123/fenceagents/0
```

表 6.266. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.93.3. 更新 PUT

更新 fence-agent。

表 6.267. 参数摘要

名称	类型	方向	概述
agent	agent	in/Out	隔离代理详细信息.
async	布尔值	in	指明是否应异步执行更新。

6.94. FENCEAGENTS

用于管理特定主机的隔离代理的服务。

表 6.268. 方法概述

名称	概述
add	向主机添加新的 fence-agent。
list	返回为主机配置的隔离代理列表。

6.94.1. 添加 POST

向主机添加新的 **fence-agent**。

POST /ovirt-engine/api/hosts/123/fenceagents

You should consult the /usr/sbin/fence_<agent_name> manual page for the legal parameters to [name1=value1, name2=value2,...] in the options field. If any parameter in options appears by name that means that it is mandatory. For example in <options>slot=7[,name1=value1, name2=value2,...]</options> slot is mandatory.

APC, bladecenter, wti 隔离代理/s 示例请求 :

```
<agent>
  <type>apc</type>
  <order>1</order>
  <ip>192.168.1.101</ip>
  <user>user</user>
  <password>xxx</password>
  <port>9</port>
  <options>slot=7[,name1=value1, name2=value2,...]</options>
</agent>
```

apc_snmp, hblade, ilo, ilo2, ilo_ssh, redfish, rsa 隔离代理/s 示例请求 :

```
<agent>
  <type>apc_snmp</type>
  <order>1</order>
  <ip>192.168.1.101</ip>
  <user>user</user>
  <password>xxx</password>
  <port>9</port>
  <options>[name1=value1, name2=value2,...]</options>
</agent>
```

cisco_ucs, drac5, eps 隔离代理/s 示例请求：

```
<agent>
  <type>cisco_ucs</type>
  <order>1</order>
  <ip>192.168.1.101</ip>
  <user>user</user>
  <password>xxx</password>
  <options>slot=7[,name1=value1, name2=value2,...]</options>
</agent>
```

drac7、ilo3、ilo4、ipmilan、rsb 隔离代理/s 示例请求：

```
<agent>
  <type>drac7</type>
  <order>1</order>
  <ip>192.168.1.101</ip>
  <user>user</user>
  <password>xxx</password>
  <options>[name1=value1, name2=value2,...]</options>
</agent>
```

表 6.269. 参数摘要

名称	类型	方向	概述
agent	agent	in/Out	

6.94.2. 列出 GET

返回为主机配置的隔离代理列表。

GET /ovirt-engine/api/hosts/123/fenceagents

以下是响应示例：

```
<agents>
  <agent id="0">
    <type>apc</type>
    <order>1</order>
    <ip>192.168.1.101</ip>
    <user>user</user>
    <password>xxx</password>
    <port>9</port>
```



```
<options>name1=value1, name2=value2</options>
</agent>
</agents>
```

无法保证返回的隔离代理列表的顺序。

表 6.270. 参数摘要

名称	类型	方向	概述
代理	agent[]	out	隔离代理详情列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的代理的最大数量。

6.94.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.94.2.2. Max

设置要返回的代理的最大数量。如果没有指定，则返回所有代理。

6.95. FILE

表 6.271. 方法概述

名称	概述
get	

6.95.1. 获取 GET

表 6.272. 参数摘要

名称	类型	方向	概述
file	File	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.95.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.96. 文件

提供了一种方式，供客户端列出可用的文件。

此服务专用于 ISO 存储域，后者包含管理员上传的 ISO 映像和虚拟软盘(VFD)。

向虚拟机中添加 CD-ROM 设备需要来自 ISO 存储域文件的 ISO 映像。

表 6.273. 方法概述

名称	概述
list	返回存储域中可用的 ISO 映像和虚拟软盘的列表。

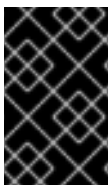
6.96.1. 列出 GET

返回存储域中可用的 ISO 映像和虚拟软盘的列表。无法保证返回列表的顺序。

如果刷新参数为 `false`，则返回的列表可能无法反映对存储域的最新更改；例如，它可能不包含最近添加的新 ISO 文件。这是因为服务器缓存文件列表以提高性能。要获得最新结果，请将 `refresh` 参数设置为 `true`。

`refresh` 参数的默认值为 `true`，但可使用配置值 `force RefreshDomainFilesByDefault` 更改：

```
# engine-config -s ForceRefreshDomainFilesByDefault=false
```



重要

将 `refresh` 参数的值设置为 `true` 会影响服务器的性能。必要时才使用。

表 6.274. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否使用 search 参数进行的搜索应考虑大小写。
file	file[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大文件数。
refresh	布尔值	in	指明是否应从存储域刷新文件列表，而不显示以特定间隔更新的缓存结果。
search	字符串	in	用于限制返回文件的查询字符串。

6.96.1.1. case_sensitive

指明是否使用 **search** 参数进行的搜索应考虑大小写。默认值为 **true**。

6.96.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.96.1.3. Max

设置要返回的最大文件数。如果未指定，则返回所有文件。

6.97. FILTER

表 6.275. 方法概述

名称	概述
get	
remove	

6.97.1. 获取 GET

表 6.276. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
结果	filter	out	

6.97.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.97.2. 删除 DELETE

表 6.277. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.98. 过滤器

管理调度策略使用的过滤器。

表 6.278. 方法概述

名称	概述
add	向指定用户定义的调度策略添加过滤器。
list	返回调度策略使用的过滤器列表。

6.98.1. 添加 POST

向指定用户定义的调度策略添加过滤器。

表 6.279. 参数摘要

名称	类型	方向	概述
filter	filter	in/Out	

6.98.2. 列出 GET

返回调度策略使用的过滤器列表。

返回的过滤器列表的顺序无法保证。

表 6.280. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
过滤器	filter[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大过滤器数。

6.98.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.98.2.2. Max

设置要返回的最大过滤器数。如果没有指定，则会返回所有过滤器。

6.99. 关注

6.100. GLUSTERBRICK

此服务管理单个 gluster brick。

表 6.281. 方法概述

名称	概述
get	获取 brick 的详细信息。
remove	删除 brick。

名称	概述
replace	将此 brick 替换为新的 brick。

6.100.1. 获取 GET

获取 brick 的详细信息。

从底层 gluster 卷检索 brick 的状态详细信息，并标头 All-Content 设为 true。这等同于运行 gluster 卷状态 <volumename> <brickname> 详情。

例如，若要获取 gluster 卷 123 的 brick 234 的详细信息，请发送类似如下的请求：

```
GET /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/234
```

这会返回类似如下的响应正文：

```
<brick id="234">
  <name>host1:/rhgs/data/brick1</name>
  <brick_dir>/rhgs/data/brick1</brick_dir>
  <server_id>111</server_id>
  <status>up</status>
  <device>/dev/mapper/RHGS_vg1-lv_vmaddl disks</device>
  <fs_name>xfs</fs_name>
  <gluster_clients>
    <gluster_client>
      <bytes_read>2818417648</bytes_read>
      <bytes_written>1384694844</bytes_written>
      <client_port>1011</client_port>
      <host_name>client2</host_name>
    </gluster_client>
  </gluster_clients>
  <memory_pools>
    <memory_pool>
      <name>data-server:fd_t</name>
      <alloc_count>1626348</alloc_count>
      <cold_count>1020</cold_count>
      <hot_count>4</hot_count>
      <max_alloc>23</max_alloc>
      <max_stdalloc>0</max_stdalloc>
      <padded_size>140</padded_size>
      <pool_misses>0</pool_misses>
    </memory_pool>
  </memory_pools>
</brick>
```

```
<mnt_options>rw,seclabel,noatime,nodiratime,attr2,inode64,sunit=512,swidth=2048,noquota</mnt_options>
<pid>25589</pid>
<port>49155</port>
</brick>
```

表 6.282. 参数摘要

名称	类型	方向	概述
brick	GlusterBrick	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.100.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.100.2. 删除 DELETE

删除 brick。

从底层 gluster 卷移除 brick，并从数据库中删除条目。这只能在没有数据迁移的情况下删除单个 brick 时使用。若要删除多个 brick 和 数据迁移，请改为使用 migrate。

例如，要从 gluster 卷 123 中删除 brick 234，请发送如下所示的请求：

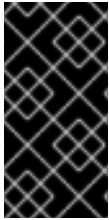
```
DELETE /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/234
```

表 6.283. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.100.3. 替换 POST

将此 brick 替换为新的 brick。

**重要**

自引擎版本 3.5 起，此操作已弃用，并将在以后的版本中删除。改为使用 **add brick**，再迁移 **brick**。

表 6.284. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行替换。
force	布尔值	in	

6.101. GLUSTERBRICKS

此服务管理 **gluster** 卷中的 **gluster brick**

表 6.285. 方法概述

名称	概述
激活	在删除 brick 操作的数据迁移后激活 brick。
add	添加 brick 列表到 gluster 卷。
list	列出 gluster 卷的 brick。
migrate	在删除 brick 之前开始迁移数据。
remove	从 gluster 卷移除 brick。
stopmigrate	停止从 brick 操作迁移 brick 数据。

6.101.1. 激活 POST

在删除 **brick** 操作的数据迁移后激活 **brick**。

用于在 **brick** 的数据迁移完成后激活 **brick**，但用户不想再删除 **brick**。之前标记为移除的 **brick** 现在用作普通 **brick**。

例如，若要保留迁移数据的 `glustervolume 123` 上的 `brick`，请发送如下请求：

```
POST /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/activate
```

这样的请求正文：

```
<action>
  <bricks>
    <brick>
      <name>host1:/rhgs/brick1</name>
    </brick>
  </bricks>
</action>
```

表 6.286. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行激活。
brick	GlusterBrick[]	in	需要重新激活的 brick 列表。

6.101.2. 添加 POST

添加 `brick` 列表到 `gluster` 卷。

用于通过添加 `brick` 来扩展 `gluster` 卷。对于复制卷类型，需要传递参数 `replica_count`。如果副本数增加，则 `brick` 的数量需要等于副本集的数量。

例如，要将 `brick` 添加到 `gluster` 卷 `123`，请发送如下所示的请求：

```
POST /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks
```

这样的请求正文：

```
<bricks>
  <brick>
    <server_id>111</server_id>
```

```
<brick_dir>/export/data/brick3</brick_dir>
</brick>
</bricks>
```

表 6.287. 参数摘要

名称	类型	方向	概述
brick	GlusterBrick[]	in/Out	要添加到卷中的 brick 列表
replica_count	整数	in	添加后操作的副本数。
stripe_count	整数	in	添加后操作的条带计数。

6.101.3. 列出 GET

列出 gluster 卷的 brick。

例如，要列出 gluster 卷 123 的 brick，请发送如下所示的请求：

```
GET /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks
```

提供如下输出：

```
<bricks>
  <brick id="234">
    <name>host1:/rhgs/data/brick1</name>
    <brick_dir>/rhgs/data/brick1</brick_dir>
    <server_id>111</server_id>
    <status>up</status>
  </brick>
  <brick id="233">
    <name>host2:/rhgs/data/brick1</name>
    <brick_dir>/rhgs/data/brick1</brick_dir>
    <server_id>222</server_id>
    <status>up</status>
  </brick>
</bricks>
```

返回列表的顺序基于在创建 gluster 卷时提供的 brick 顺序。

表 6.288. 参数摘要

名称	类型	方向	概述
brick	GlusterBrick[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大 brick 数。

6.101.3.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.101.3.2. Max

设置要返回的最大 brick 数。如果未指定，则返回所有 brick。

6.101.4. 迁移 POST

在删除 brick 之前开始迁移数据。

删除 brick 的过程包含两步，其中要移除 brick 上的数据首先迁移到剩余的 brick。迁移完成后，通过 API 删除确认 brick 删除。如果随时需要取消该操作，则必须调用该操作。

例如，要从 ID 为 123 的 gluster 卷中删除 brick，请发送请求：

```
POST /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/migrate
```

这样的请求正文：

```
<action>
  <bricks>
    <brick>
      <name>host1:/rhgs/brick1</name>
    </brick>
  </bricks>
</action>
```

可以使用作业和作业中的步骤从 [API 返回的作业 id](#) 中跟踪迁移过程。

表 6.289. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行迁移。
brick	GlusterBrick []	in	需要为其启动数据迁移的 brick 的列表。

6.101.5. 删除 DELETE

从 gluster 卷移除 brick。

在不丢失数据的情况下删除 brick 的建议方法是首先使用 [stopmigrate](#) 来迁移数据，然后删除它们。如果删除之前未在 brick 上调用迁移，则 brick 将被删除，而无需数据迁移，这可能会导致数据丢失。

例如，要从 gluster 卷 123 中删除 brick，请发送如下所示的请求：

```
DELETE /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks
```

这样的请求正文：

```
<bricks>
  <brick>
    <name>host:brick_directory</name>
  </brick>
</bricks>
```

表 6.290. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。
brick	GlusterBrick []	in	要删除的 brick 的列表
replica_count	整数	in	添加后操作的副本数。

6.101.6. stopmigrate POST

停止从 brick 操作迁移 brick 数据。

若要取消在 2 步中启动的数据迁移，如果用户希望继续使用 brick，则删除 brick 进程。标记为移除的 brick 在执行此操作后将作为普通 brick 发挥作用。

例如，若要停止从 gluster 卷 123 的 brick 迁移数据，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/stopmigrate
```

这样的请求正文：

```
<bricks>
  <brick>
    <name>host:brick_directory</name>
  </brick>
</bricks>
```

表 6.291. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
brick	GlusterBrick[]	in	需要停止数据迁移的 brick 列表。

6.101.6.1. brick

需要停止数据迁移的 brick 列表。此列表应与传递给迁移的参数匹配。

6.102. GLUSTERHOOK

表 6.292. 方法概述

名称	概述
disable	通过禁用集群中的所有服务器中的 Gluster hook，解决集群中服务器之间的 hook 状态冲突。

名称	概述
启用	通过禁用集群中的所有服务器中的 Gluster hook，解决集群中服务器之间的 hook 状态冲突。
get	
remove	从集群中的所有服务器移除此 Gluster hook，并将它从数据库中删除。
resolve	根据解析类型解决缺少的 hook 冲突。

6.102.1. 禁用 POST

通过禁用集群中的所有服务器中的 Gluster hook，解决集群中服务器之间的 hook 状态冲突。这会将 hook 状态更新为数据库中的 **DISABLED**。

表 6.293. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.102.2. 启用 POST

通过禁用集群中的所有服务器中的 Gluster hook，解决集群中服务器之间的 hook 状态冲突。这会将 hook 状态更新为数据库中的 **DISABLED**。

表 6.294. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.102.3. 获取 GET

表 6.295. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
hook	GlusterHook	out	

6.102.3.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.102.4. 删除 DELETE

从集群中的所有服务器移除此 Gluster hook，并将它从数据库中删除。

表 6.296. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.102.5. 解析 POST

根据解析类型解决缺少的 hook 冲突。

对于 ADD，通过将存储在引擎数据库中的 hook 复制到缺少 hook 的所有服务器上来解决：引擎维护缺少 hook 的所有服务器的列表。

对于 COPY，通过将存储在引擎数据库中的 hook 复制到缺少 hook 的所有服务器来解决 hook 内容冲突。引擎维护内容有冲突的所有服务器的列表。如果主机 ID 作为参数传递，则服务器的 hook 内容将用作主服务器以复制到集群中的其他服务器。

表 6.297. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
host	主机	in	
resolution_type	字符串	in	

6.103. GLUSTERHOOKS

表 6.298. 方法概述

名称	概述
list	返回 hook 列表。

6.103.1. 列出 GET

返回 hook 列表。

返回的 hook 列表的顺序不受保证。

表 6.299. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
Hook	GlusterHook []	out	
max	整数	in	设置要返回的最大 hook 数。

6.103.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.103.1.2. Max

设置要返回的最大 hook 数。如果没有指定，则返回所有 hook。

6.104. GLUSTERVOLUME

此服务管理单个 gluster 卷。

表 6.300. 方法概述

名称	概述
get	获取 gluster 卷详细信息。

名称	概述
getprofilestatistics	获取 gluster 卷配置文件统计信息。
rebalance	重新平衡 gluster 卷。
remove	删除 gluster 卷。
resetalloptions	重置 gluster 卷中设置的所有选项。
resetoption	重置 gluster 卷中的特定选项。
setoption	在 gluster 卷中设置特定选项。
start	启动 gluster 卷。
startprofile	启动 gluster 卷性能分析。
stop	停止 gluster 卷。
stopprofile	停止分析 gluster 卷。
stoprebalance	停止重新平衡 gluster 卷。

6.104.1. 获取 GET

获取 gluster 卷详细信息。

例如，若要获取群集 456 中标识符为 123 的 gluster 卷详情，请发送类似如下的请求：

```
GET /ovirt-engine/api/clusters/456/glustervolumes/123
```

此 GET 请求将返回以下输出：

```
<gluster_volume id="123">
  <name>data</name>
  <link href="/ovirt-engine/api/clusters/456/glustervolumes/123/glusterbricks" rel="glusterbricks"/>
  <disperse_count>0</disperse_count>
  <options>
    <option>
      <name>storage.owner-gid</name>
```

```

    <value>36</value>
  </option>
  <option>
    <name>performance.io-cache</name>
    <value>off</value>
  </option>
  <option>
    <name>cluster.data-self-heal-algorithm</name>
    <value>full</value>
  </option>
</options>
<redundancy_count>0</redundancy_count>
<replica_count>3</replica_count>
<status>up</status>
<stripe_count>0</stripe_count>
<transport_types>
  <transport_type>tcp</transport_type>
</transport_types>
<volume_type>replicate</volume_type>
</gluster_volume>

```

表 6.301. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
卷	GlusterVolume	out	gluster 卷的表示形式。

6.104.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.104.2. getprofilestatistics POST

获取 gluster 卷配置文件统计信息。

例如，若要获取群集 456 中标识符为 123 的 gluster 卷的配置集统计信息，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/getprofilestatistics
```

表 6.302. 参数摘要

名称	类型	方向	概述
详情	GlusterVolumeProfileDetails	out	Gluster 卷性能分析从操作返回的信息。

6.104.3. 重新平衡 POST

重新平衡 gluster 卷。

重新平衡 gluster 卷有助于在所有 brick 之间均匀分布数据。在扩展或缩小 gluster 卷（无需迁移数据）后，我们需要在 brick 之间重新平衡数据。在非复制卷中，所有 brick 都应在线，以执行重新平衡操作。在复制卷中，副本中应至少有一个 brick 在线。

例如，若要在集群 456 中使用标识符 123 重新平衡 gluster 卷，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/rebalance
```

表 6.303. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行重新平衡。
fix_layout	布尔值	in	如果设为 true，则重新平衡将仅修复布局，以便添加到卷的新数据在所有主机上分布。
force	布尔值	in	指明是否应强制启动重新平衡。

6.104.3.1. fix_layout

如果设为 true，则重新平衡将仅修复布局，以便添加到卷的新数据在所有主机上分布。但它不会迁移/重新平衡现有数据。默认为 false。

6.104.3.2. force

指明是否应强制启动重新平衡。重新平衡命令可以使用 force 选项执行，即使旧的客户端连接到集群时也是如此。但是，这会导致数据丢失的情况。默认为 false。

6.104.4. 删除 DELETE

删除 gluster 卷。

例如，要删除在集群 456 中带有 123 的卷，请发送类似如下的请求：

```
DELETE /ovirt-engine/api/clusters/456/glustervolumes/123
```

表 6.304. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.104.5. resetalloptions POST

重置 gluster 卷中设置的所有选项。

例如，若要重置群集 456 中标识符为 123 的 gluster 卷中的所有选项，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/resetalloptions
```

表 6.305. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行重置。

6.104.6. resetoption POST

重置 gluster 卷中的特定选项。

例如，若要在群集 456 中使用标识符 123 重置 gluster 卷中的特定选项 option1，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/resetoption
```

使用以下请求正文：

```
<action>
<option name="option1"/>
</action>
```

表 6.306. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行重置。
force	布尔值	in	
选项	选项	in	重置选项.

6.104.7. setoption POST

在 **gluster** 卷中设置特定选项。

例如，要在群集 456 中为 **gluster** 卷设置 值为 **value1** 的 **option1**，请在群集 456 中使用标识符 123 设置 **option1**，发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/setoption
```

使用以下请求正文：

```
<action>
<option name="option1" value="value1"/>
</action>
```

表 6.307. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
选项	选项	in	要设置的选项.

6.104.8. 启动 POST

启动 **gluster** 卷。

应当启动 **Gluster** 卷以读/写数据。例如，要在集群 **456** 中启动一个带有标识符 **123** 的 **gluster** 卷，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/start
```

表 6.308. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
force	布尔值	in	指明是否应强制启动该卷。

6.104.8.1. force

指明是否应强制启动该卷。如果 **gluster** 卷已启动，但很少/所有 **brick** 已停机，则可以使用 **force start** 来启动所有 **brick**。默认为 **false**。

6.104.9. startprofile POST

启动 **gluster** 卷性能分析。

例如，要在群集 **456** 中使用标识符 **123** 开始分析 **gluster** 卷，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/startprofile
```

表 6.309. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.104.10. 停止 POST

停止 **gluster** 卷。

停止卷将使其数据无法访问。

例如，若要在集群 456 中停止带有标识符 123 的 gluster 卷，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/stop
```

表 6.310. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
force	布尔值	in	

6.104.11. stopprofile POST

停止分析 gluster 卷。

例如，要在集群 456 中停止使用标识符 123 分析 gluster 卷，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/stopprofile
```

表 6.311. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.104.12. 停止平衡 POST

停止重新平衡 gluster 卷。

例如，若要停止在集群 456 中使用标识符 123 重新平衡 gluster 卷，请发送类似如下的请求：

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/stoprebalance
```

表 6.312. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.105. GLUSTERVOLUMES

此服务管理集群中可用的 **gluster** 卷的集合。

表 6.313. 方法概述

名称	概述
add	创建新的 gluster 卷。
list	列出集群中的所有 gluster 卷。

6.105.1. 添加 POST

创建新的 **gluster** 卷。

卷基于 **volume** 参数的属性创建。需要属性 **名称**、**volume_type** 和 **brick**。

例如，要将名为 **myvolume** 的卷添加到集群 **123** 中，请发送以下请求：

```
POST /ovirt-engine/api/clusters/123/glustervolumes
```

使用以下请求正文：

```
<gluster_volume>
  <name>myvolume</name>
  <volume_type>replicate</volume_type>
  <replica_count>3</replica_count>
  <bricks>
    <brick>
      <server_id>server1</server_id>
      <brick_dir>exp1</brick_dir>
    </brick>
    <brick>
      <server_id>server2</server_id>
      <brick_dir>exp1</brick_dir>
    </brick>
  </bricks>
</gluster_volume>
```



```

<brick>
  <server_id>server3</server_id>
  <brick_dir>exp1</brick_dir>
</brick>
<bricks>
</gluster_volume>

```

表 6.314. 参数摘要

名称	类型	方向	概述
卷	GlusterVolume	in/Out	要创建卷的 gluster 卷定义作为输入传递，并且新创建的卷返回。

6.105.2. 列出 GET

列出集群中的所有 gluster 卷。

例如，若要列出群集 456 中的所有 Gluster 卷，请发送如下所示的请求：

```
GET /ovirt-engine/api/clusters/456/glustervolumes
```

无法保证返回的卷列表的顺序。

表 6.315. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大卷数。
search	字符串	in	用于限制返回的卷的查询字符串。
卷	GlusterVolume[]	out	

6.105.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 **search** 参数执行搜索。默认值为 **true**，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 **false**。

6.105.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.105.2.3. Max

设置要返回的最大卷数。如果没有指定，则返回所有卷。

6.106. 组

管理一组用户。使用此服务获取组详细信息或删除组。[要添加新组，请使用管理组集合的服务。](#)

表 6.316. 方法概述

名称	概述
get	获取系统组信息.
remove	删除系统组。

6.106.1. 获取 GET

获取系统组信息.

使用方法：

GET /ovirt-engine/api/groups/123

将返回组信息：

```
<group href="/ovirt-engine/api/groups/123" id="123">
  <name>mygroup</name>
  <link href="/ovirt-engine/api/groups/123/roles" rel="roles"/>
  <link href="/ovirt-engine/api/groups/123/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/groups/123/tags" rel="tags"/>

  <domain_entry_id>476652557A382F67696B6D2B32762B37796E46476D513D3D</domain_entry_id
>
```

```
<namespace>DC=example,DC=com</namespace>
<domain href="/ovirt-engine/api/domains/ABCDEF" id="ABCDEF">
  <name>myextension-authz</name>
</domain>
</group>
```

表 6.317. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
get	组	out	系统组.

6.106.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.106.2. 删除 DELETE

删除系统组。

使用方法：

```
DELETE /ovirt-engine/api/groups/123
```

表 6.318. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.107. 组

管理用户组的集合。

表 6.319. 方法概述

名称	概述
add	从目录服务添加组.

名称	概述
list	列出系统中的所有组。

6.107.1. 添加 POST

从目录服务添加组.请注意，域名是授权供应商的名称。

例如，从 `internal-Authz` 授权供应商添加 `Developers` 组，发送类似如下的请求：

POST /ovirt-engine/api/groups

这样的请求正文：

```
<group>
  <name>Developers</name>
  <domain>
    <name>internal-Authz</name>
  </domain>
</group>
```

表 6.320. 参数摘要

名称	类型	方向	概述
group	组	in/Out	要添加的组。

6.107.2. 列出 GET

列出系统中的所有组。

使用方法：

GET /ovirt-engine/api/groups

将返回组列表：

```
<groups>
```

```

<group href="/ovirt-engine/api/groups/123" id="123">
  <name>mygroup</name>
  <link href="/ovirt-engine/api/groups/123/roles" rel="roles"/>
  <link href="/ovirt-engine/api/groups/123/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/groups/123/tags" rel="tags"/>

<domain_entry_id>476652557A382F67696B6D2B32762B37796E46476D513D3D</domain_entry_id
>
  <namespace>DC=example,DC=com</namespace>
  <domain href="/ovirt-engine/api/domains/ABCDEF" id="ABCDEF">
    <name>myextension-authz</name>
  </domain>
</group>
...
</groups>

```

无法保证返回的组列表的顺序。

表 6.321. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
groups	group[]	out	组列表。
max	整数	in	设置要返回的最大组数。
search	字符串	in	用于限制返回的组的查询字符串。

6.107.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 **search** 参数执行搜索。默认值为 **true**，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 **false**。

6.107.2.2. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.107.2.3. Max

设置要返回的最大组数。如果没有指定，则返回所有组。

6.108. 主机

用于管理主机的服务。

表 6.322. 方法概述

名称	概述
激活	激活要使用的主机，例如用于运行虚拟机的主机。
批准	批准预安装的系统管理程序主机，以便在虚拟化环境中使用。
commitnetconfig	将网络配置标记为良好的，并在主机中持久保留。
copyhostnetworks	将指定主机的网络配置复制到当前主机。
取消激活	停用主机以执行维护任务。
discoveriscsi	使用启动器详细信息，在主机上发现 iSCSI 目标。
注册证书	注册主机的证书。
fence	控制主机的电源管理设备。
forceselectspm	将主机手动设置为存储池管理器(SPM)。
get	获取主机详细信息。
install	在主机上安装最新版本的 VDSM 和相关软件。
iscsidiscover	这个方法自 Engine 版本 4 起已弃用。
iscsilogin	使用目标详细信息登录到主机上的 iSCSI 目标。
refresh	刷新主机设备和功能。
remove	从系统中删除主机。
setupnetworks	此方法用于更改主机网络接口的配置。

名称	概述
syncallnetworks	要同步主机上的所有网络，请发送如下请求： [source] ---- POST /ovirt-engine/api/hosts/123/syncallnetworks ---- 这样的请求正文： [source,xml] ---- <action/> ----
unregisteredstoragedomainsdiscover	发现要导入到设置的块存储域。
update	更新主机属性。
upgrade	升级主机上的 VDSM 和选定的软件。
upgradecheck	检查是否有适用于主机的升级。

6.108.1. 激活 POST

激活要使用的主机，例如用于运行虚拟机的主机。

表 6.323. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行激活。

6.108.2. 批准 POST

批准预安装的系统管理程序主机，以便在虚拟化环境中使用。

此操作还接受用于定义此主机的目标集群的可选集群元素。

表 6.324. 参数摘要

名称	类型	方向	概述
激活	布尔值	in	当设置为"true"时，此主机将在其批准完成后激活。
async	布尔值	in	指明是否应当异步执行批准。

名称	类型	方向	概述
cluster	Cluster	in	在批准主机后将添加主机的群集。
host	主机	in	要批准的主机。
reboot	布尔值	in	指明是否应当在安装成功后重新启动该主机。

6.108.2.1. 激活

当设置为"true"时，此主机将在其批准完成后激活。当设置为"false"时，主机在批准后将保持"维护"状态。缺少此参数将解释为 'true'，因为所需的默认行为是在批准后激活主机。

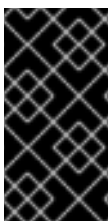
6.108.2.2. reboot

指明是否应当在安装成功后重新启动该主机。默认值为 true。

6.108.3. commitnetconfig POST

将网络配置标记为良好的，并在主机中持久保留。

API 用户提交网络配置以持久保留主机网络接口附加或分离，或持久创建和删除绑定接口。



重要

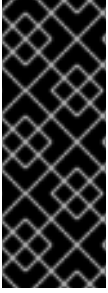
只有在引擎确定了主机连接不会因为配置更改而丢失后，才会提交网络配置。如果主机连接丢失，则主机需要重新引导，并且自动恢复到先前的网络配置。

例如，要提交 ID 为 123 的主机的网络配置，发送如下请求：

```
POST /ovirt-engine/api/hosts/123/commitnetconfig
```

这样的请求正文：

```
<action/>
```


**重要**

由于 Red Hat Virtualization Manager 4.3, 也可以在 `setupnetworks` 请求中指定 `commit_on_success`, 在这种情况下, 新配置会在完成设置并重新建立 {hypervisor-name} 和 Red Hat Virtualization Manager 之间的连接后自动保存在 {hypervisor-name} 和 Red Hat Virtualization Manager, 且无需等待单独的 `commitnetconfig` 请求。

表 6.325. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行该操作。

6.108.4. copyhostnetworks POST

将指定主机的网络配置复制到当前主机。

**重要**

源主机上不存在的任何网络附加都会通过复制操作从目标主机上清除。

要从另一个主机复制网络, 请发送如下请求 :

```
POST /ovirt-engine/api/hosts/123/copyhostnetworks
```

这样的请求正文 :

```
<action>
  <source_host id="456"/>
</action>
```

表 6.326. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行该操作。
<code>source_host</code>	主机	in	要从中复制网络的主机。

6.108.5. 取消激活 POST

停用主机以执行维护任务。

表 6.327. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行取消激活。
reason	字符串	in	
stop_gluster_service	布尔值	in	指明是否应当在停用主机时停止 gluster 服务。

6.108.5.1. stop_gluster_service

指明是否应当在停用主机时停止 **gluster** 服务。它可用于在 **gluster** 主机上执行维护操作。此变量的默认值为 **false**。

6.108.6. 发现iscsi POST

使用启动器详细信息，在主机上发现 iSCSI 目标。返回包含已发现数据的 **IscsiDetails** 对象的列表。

例如，要从主机 123 发现 **myiscsi.example.com** 中可用的 iSCSI 目标，请发送如下请求：

```
POST /ovirt-engine/api/hosts/123/discoveriscsi
```

这样的请求正文：

```
<action>
  <iscsi>
    <address>myiscsi.example.com</address>
  </iscsi>
</action>
```

结果如下：

```
<discovered_targets>
  <iscsi_details>
```

```

<address>10.35.1.72</address>
<port>3260</port>
<portal>10.35.1.72:3260,1</portal>
<target>iqn.2015-08.com.tgt:444</target>
</iscsi_details>
</discovered_targets>

```



重要

当使用此方法发现 **iscsi** 目标时，您可以使用 **FQDN** 或 **IP 地址**，但必须使用发现的目标中的 **iscsi** 详情才能使用 **iscsilogin** 方法登录。

表 6.328. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行发现。
discovered_targets	<code>iscsiDetails[]</code>	out	发现的目标包括所有连接信息。
iscsi	<code>iscsiDetails</code>	in	目标 iSCSI 设备。

6.108.7. 注册certificate POST

注册主机的证书。如果您收到即将过期或已过期的警告，则很有用。

表 6.329. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行注册。

6.108.8. 隔离 POST

控制主机的电源管理设备。

例如，要启动主机：这可以通过以下方法完成：

```

#!/bin/sh -ex

url="https://engine.example.com/ovirt-engine/api"
user="admin@internal"

```

```
password="..."

curl \
--verbose \
--cacert /etc/pki/ovirt-engine/ca.pem \
--user "${user}:${password}" \
--request POST \
--header "Version: 4" \
--header "Content-Type: application/xml" \
--header "Accept: application/xml" \
--data '
<action>
  <fence_type>start</fence_type>
</action>
'\
"${url}/hosts/123/fence"
```

表 6.330. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行隔离。
fence_type	字符串	in	
maintenanc e_after_re start	布尔值	in	指明是否应在重新启动后将主机置于维护状态。
power_ma nagement	PowerManag ement	out	

6.108.9. forceelectspm POST

将主机手动设置为存储池管理器(SPM)。

```
POST /ovirt-engine/api/hosts/123/forceelectspm
```

这样的请求正文：

```
<action/>
```

表 6.331. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.108.10. 获取 GET

获取主机详细信息。

```
GET /ovirt-engine/api/hosts/123
```

表 6.332. 参数摘要

名称	类型	方向	概述
all_content	布尔值	in	指明是否应将主机的所有属性包含在响应中。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
host	主机	out	查询的主机。

6.108.10.1. all_content

指明是否应将主机的所有属性包含在响应中。

默认情况下排除以下属性：

- **hosted_engine**

例如，要检索主机 '123' 的完整表示法：

```
GET /ovirt-engine/api/hosts/123?all_content=true
```



注意

这些属性默认不包含在内，因为检索它们会影响性能。它们很少使用，需要对数据库进行其他查询。谨慎使用此参数，且仅在有特殊需要时才使用。

6.108.10.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.108.11. 安装 POST

在主机上安装最新版本的 VDSM 和相关软件。

该操作还会在主机上执行在向引擎中添加主机时执行的每个配置步骤：kdump 配置、hosted-engine 部署、内核选项更改等。

主机类型定义操作的其他参数。

使用 curl 和 JSON 安装主机示例：

```
curl \
--verbose \
--cacert /etc/pki/ovirt-engine/ca.pem \
--request PUT \
--header "Content-Type: application/json" \
--header "Accept: application/json" \
--header "Version: 4" \
--user "admin@internal:..." \
--data '
{
  "root_password": "myrootpassword"
}
' \
"https://engine.example.com/ovirt-engine/api/hosts/123"
```

使用带有托管引擎组件的 curl 和 JSON 安装主机示例：

```
curl \
curl \
--verbose \
--cacert /etc/pki/ovirt-engine/ca.pem \
--request PUT \
--header "Content-Type: application/json" \
--header "Accept: application/json" \
--header "Version: 4" \
--user "admin@internal:..." \
--data '
{
  "root_password": "myrootpassword"
  "deploy_hosted_engine" : "true"
}
' \
"https://engine.example.com/ovirt-engine/api/hosts/123"
```



重要

自引擎版本 4.1.2 起，在重新安装主机时，我们默认覆盖主机防火墙定义。

表 6.333. 参数摘要

名称	类型	方向	概述
激活	布尔值	in	当设置为"true"时，此主机将在安装完成后激活。
async	布尔值	in	指明是否应异步执行安装。
deploy_hosted_engine	布尔值	in	如果设置为 true ，此主机也会部署自托管引擎组件。
host	主机	in	override_iptables 属性用于指示防火墙配置是否应该替换为默认配置。
image	字符串	in	安装 {hypervisor-name} 时，需要一个 ISO 镜像文件。
reboot	布尔值	in	指明是否应当在安装成功后重新启动该主机。
root_password	字符串	in	用于通过 SSH 连接到主机的 root 用户的密码。
ssh	ssh	in	用于连接主机的 SSH 详细信息。
undeploy_hosted_engine	布尔值	in	如果设置为 true ，此主机将取消部署自托管引擎组件，此主机将不充当高可用性群集的一部分。

6.108.11.1. 激活

当设置为"true"时，此主机将在安装完成后激活。当设置为"false"时，主机在安装后将保持"维护"状态。缺少此参数将解释为"true"，因为所需的默认行为是在安装后激活主机。

6.108.11.2. deploy_hosted_engine

如果设置为 **true**，此主机也会部署自托管引擎组件。缺少的值被视为 **true**，即 **deploy**。省略此参数表示假，且不会在自托管引擎区域执行任何操作。

6.108.11.3. reboot

指明是否应当在安装成功后重新启动该主机。默认值为 `true`。

6.108.11.4. undeploy_hosted_engine

如果设置为 `true`，此主机将取消部署自托管引擎组件，此主机将不充当高可用性群集的一部分。缺少的值被视为 `true`，即 `un-deploy`。省略此参数表示假，且不会在自托管引擎区域执行任何操作。

6.108.12. iscsidiscover POST

这个方法自 Engine 版本 4.4.6 起已弃用。应改为使用 `DiscoverIscsi`。

使用启动器详细信息，在主机上发现 iSCSI 目标。返回包含发现数据的字符串数组。

例如，要从主机 123 发现 `myiscsi.example.com` 中可用的 iSCSI 目标，请发送如下请求：

```
POST /ovirt-engine/api/hosts/123/iscsidiscover
```

这样的请求正文：

```
<action>
  <iscsi>
    <address>myiscsi.example.com</address>
  </iscsi>
</action>
```

表 6.334. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行发现。
<code>iscsi</code>	<code>IscsiDetails</code>	in	目标 iSCSI 设备。
<code>iscsi_targets</code>	<code>string[]</code>	out	iSCSI 目标。

6.108.12.1. iscsi_targets

iSCSI 目标。*

6.108.13. iscsilogin POST

使用目标详细信息登录到主机上的 iSCSI 目标。



重要

使用此方法登录时，您必须使用发现的目标中的 `iscsi` 详情会导致 `discoveriscsi` 方法。

表 6.335. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行登录。
<code>iscsi</code>	<code>iscsiDetails</code>	in	目标 iSCSI 设备。

6.108.14. 刷新 POST

刷新主机设备和功能。

表 6.336. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行刷新。

6.108.15. 删除 DELETE

从系统中删除主机。

```
#!/bin/sh -ex

url="https://engine.example.com/ovirt-engine/api"
user="admin@internal"
password="..."

curl \
--verbose \
```

```
--cacert /etc/pki/ovirt-engine/ca.pem \
--user "${user}:${password}" \
--request DELETE \
--header "Version: 4" \
"${url}/hosts/1ff7a191-2f3b-4eff-812b-9f91a30c3acc"
```

表 6.337. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。
force	布尔值	in	表示该主机应被删除，即使它不响应，或者即使它属于 Gluster 存储群集并且有卷 brick。

6.108.16. setupnetworks POST

此方法用于更改主机网络接口的配置。

例如，如果您有一个具有三个网络接口的主机 `eth0`、`eth1` 和 `eth2`，并且您想要使用 `eth0` 和 `eth1` 配置一个新绑定，并在上面放置 VLAN。使用一个简单的 `shell` 脚本和 `curl` 命令行 HTTP 客户端，可以按如下方式完成：

```
#!/bin/sh -ex

url="https://engine.example.com/ovirt-engine/api"
user="admin@internal"
password="..."

curl \
--verbose \
--cacert /etc/pki/ovirt-engine/ca.pem \
--user "${user}:${password}" \
--request POST \
--header "Version: 4" \
--header "Content-Type: application/xml" \
--header "Accept: application/xml" \
--data '
<action>
  <modified_bonds>
    <host_nic>
      <name>bond0</name>
      <bonding>
        <options>
          <option>
            <name>mode</name>
            <value>4</value>
          </option>
          <option>
            <name>miimon</name>
```

```

    <value>100</value>
  </option>
</options>
<slaves>
  <host_nic>
    <name>eth1</name>
  </host_nic>
  <host_nic>
    <name>eth2</name>
  </host_nic>
</slaves>
</bonding>
</host_nic>
</modified_bonds>
<modified_network_attachments>
  <network_attachment>
    <network>
      <name>myvlan</name>
    </network>
    <host_nic>
      <name>bond0</name>
    </host_nic>
    <ip_address_assignments>
      <ip_address_assignment>
        <assignment_method>static</assignment_method>
        <ip>
          <address>192.168.122.10</address>
          <netmask>255.255.255.0</netmask>
        </ip>
      </ip_address_assignment>
    </ip_address_assignments>
    <dns_resolver_configuration>
      <name_servers>
        <name_server>1.1.1.1</name_server>
        <name_server>2.2.2.2</name_server>
      </name_servers>
    </dns_resolver_configuration>
  </network_attachment>
</modified_network_attachments>
</action>
'\
"${url}/hosts/1ff7a191-2f3b-4eff-812b-9f91a30c3acc/setupnetworks"

```



注意

这对 **API 版本 4** 有效。在以前的版本中，部分元素表示为 XML 属性而不是 XML 元素。特别是，选项和 ip 元素的表示如下：

```

<options name="mode" value="4"/>
<options name="miimon" value="100"/>
<ip address="192.168.122.10" netmask="255.255.255.0"/>

```

同样的操作也可使用 Python SDK 及以下代码完成：

```
# Find the service that manages the collection of hosts:
hosts_service = connection.system_service().hosts_service()

# Find the host:
host = hosts_service.list(search='name=myhost')[0]

# Find the service that manages the host:
host_service = hosts_service.host_service(host.id)

# Configure the network adding a bond with two slaves and attaching it to a
# network with an static IP address:
host_service.setup_networks(
    modified_bonds=[
        types.HostNic(
            name='bond0',
            bonding=types.Bonding(
                options=[
                    types.Option(
                        name='mode',
                        value='4',
                    ),
                    types.Option(
                        name='miimon',
                        value='100',
                    ),
                ],
            ),
            slaves=[
                types.HostNic(
                    name='eth1',
                ),
                types.HostNic(
                    name='eth2',
                ),
            ],
        ),
    ],
    modified_network_attachments=[
        types.NetworkAttachment(
            network=types.Network(
                name='myvlan',
            ),
            host_nic=types.HostNic(
                name='bond0',
            ),
            ip_address_assignments=[
                types.IpAddressAssignment(
                    assignment_method=types.BootProtocol.STATIC,
                    ip=types.Ip(
                        address='192.168.122.10',
                        netmask='255.255.255.0',
                    ),
                ),
            ],
        ),
    ],
)
```

```

    ),
  ],
  dns_resolver_configuration=types.DnsResolverConfiguration(
    name_servers=[
      '1.1.1.1',
      '2.2.2.2',
    ],
  ),
),
],
)

# After modifying the network configuration it is very important to make it
# persistent:
host_service.commit_net_config()

```

重要

为确保网络配置已保存到主机上，并且主机重启时会应用它，请记得调用 `commitnetconfig`。

重要

由于 Red Hat Virtualization Manager 4.3，也可以在 `setupnetworks` 请求中指定 `commit_on_success`，在这种情况下，新配置会在完成设置并重新建立 {hypervisor-name} 和 Red Hat Virtualization Manager 之间的连接后自动保存在 {hypervisor-name} 和 Red Hat Virtualization Manager，且无需等待单独的 `commitnetconfig` 请求。

表 6.338. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行该操作。
<code>check_connectivity</code>	布尔值	in	
<code>commit_on_success</code>	布尔值	in	指定在完成设置并重新建立 {hypervisor-name} 和 Red Hat Virtualization Manager 之间的连接后，是否会在 {hypervisor-name} 和 Red Hat Virtualization Manager 中自动保存配置，且不会等待单独的 <code>commitnetconfig</code> 请求。
<code>connectivity_timeout</code>	整数	in	
<code>modified_bonds</code>	<code>HostNic[]</code>	in	

名称	类型	方向	概述
modified_labels	NetworkLabel[]	in	
modified_network_attachments	NetworkAttachment[]	in	
removed_bonds	HostNic[]	in	
removed_labels	NetworkLabel[]	in	
removed_network_attachments	NetworkAttachment[]	in	
synchronized_network_attachments	NetworkAttachment[]	in	将同步的网络附加列表。

6.108.16.1. commit_on_success

指定在完成设置并重新建立 {hypervisor-name} 和 Red Hat Virtualization Manager 之间的连接后，是否会在 {hypervisor-name} 和 Red Hat Virtualization Manager 中自动保存配置，且不会等待单独的 [commitnetconfig](#) 请求。默认值为 **false**，这意味着配置不会自动保存。

6.108.17. syncallnetworks POST

要同步主机上的所有网络，请发送如下请求：

```
POST /ovirt-engine/api/hosts/123/syncallnetworks
```

这样的请求正文：

```
<action/>
```

表 6.339. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.108.18. unregisteredstoragedomainsdiscover POST

发现要导入到设置的块存储域。对于 FCP，不需要参数。

表 6.340. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行发现。
iscsi	iscsiDetails	in	
storage_domains	StorageDomainain[]	out	

6.108.19. 更新 PUT

更新主机属性。

例如，更新主机的内核命令行可发送如下请求：

```
PUT /ovirt-engine/api/hosts/123
```

借助此类请求正文：

```
<host>
  <os>
    <custom_kernel_cmdline>vfio_iommu_type1.allow_unsafe_interrupts=1</custom_kernel_cmdline>
  >
  </os>
</host>
```

表 6.341. 参数摘要

名称	类型	方向	概述
----	----	----	----

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
host	主机	in/Out	

6.108.20. 升级 POST

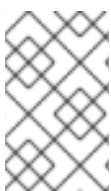
升级主机上的 VDSM 和选定的软件。

表 6.342. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行升级。
image	字符串	in	此属性不再相关，因为 Vintage 节点不再被支持，并且已弃用。
reboot	布尔值	in	指明是否应在升级后重新引导该主机。
timeout	整数	in	升级超时。

6.108.20.1. reboot

指明是否应在升级后重新引导该主机。默认情况下，将重新引导主机。



注意

{hypervisor-name} 会忽略这个参数，它总是在升级后重启。

6.108.20.2. timeout

升级超时。

等待升级完成的最长时间（以分钟为单位）。默认值由 `ANSIBLE_PLAYBOOK_EXEC_DEFAULT_TIMEOUT` 配置选项指定。

6.108.21. upgradecheck POST

检查是否有适用于主机的升级。如果有可用的升级，管理门户中主机状态图标旁边将显示一个图标。也添加了审计日志消息，以指示升级的可用性。升级可以从 [webadmin](#) 或 [升级主机操作启动](#)。

6.109. HOSTCPUUNITS

表 6.343. 方法概述

名称	概述
list	返回所有主机的 CPU 列表，其中包含关于拓扑（套接字、核心）的详细信息，以及有关当前 CPU 固定的信息。

6.109.1. 列出 GET

返回所有主机的 CPU 列表，其中包含关于拓扑（套接字、核心）的详细信息，以及有关当前 CPU 固定的信息。

表 6.344. 参数摘要

名称	类型	方向	概述
cpu_units	HostCpuUnit []	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.109.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处](#)。

6.110. HOSTDEVICE

用于访问主机特定设备的服务。

表 6.345. 方法概述

名称	概述
get	检索特定主机设备的信息。

6.110.1. 获取 GET

检索特定主机设备的信息。

获取主机设备的示例：

```
GET /ovirt-engine/api/hosts/123/devices/456
```

```
<host_device href="/ovirt-engine/api/hosts/123/devices/456" id="456">
  <name>usb_1_9_1_1_0</name>
  <capability>usb</capability>
  <host href="/ovirt-engine/api/hosts/123" id="123"/>
  <parent_device href="/ovirt-engine/api/hosts/123/devices/789" id="789">
    <name>usb_1_9_1</name>
  </parent_device>
</host_device>
```

表 6.346. 参数摘要

名称	类型	方向	概述
device	HostDevice	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.110.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.111. HOSTDEVICES

用于访问主机设备的服务。

表 6.347. 方法概述

名称	概述
list	列出主机的设备。

6.111.1. 列出 GET

列出主机的设备。

无法保证返回的设备列表的顺序。

表 6.348. 参数摘要

名称	类型	方向	概述
devices	HostDevice[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大设备数。

6.111.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.111.1.2. Max

设置要返回的最大设备数。如果没有指定，则返回所有设备。

6.112. HOSTHOOK

表 6.349. 方法概述

名称	概述
get	

6.112.1. 获取 GET

表 6.350. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
hook	Hook	out	

6.112.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.113. HOSTHOOKS

表 6.351. 方法概述

名称	概述
list	返回为主机配置的 hook 列表。

6.113.1. 列出 GET

返回为主机配置的 hook 列表。

返回的 hook 列表的顺序是随机的。

表 6.352. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
Hook	Hook[]	out	
max	整数	in	设置要返回的最大 hook 数。

6.113.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.113.1.2. Max

设置要返回的最大 hook 数。如果没有指定，则会返回所有 hook。

6.114. HOSTNIC

用于管理主机的网络接口的服务。

表 6.353. 方法概述

名称	概述
<code>get</code>	
<code>updatevirtualfunctionsconfiguration</code>	如果当前资源代表启用了 SR-IOV 的 NIC，则该操作会更新虚拟功能配置。

6.114.1. 获取 GET

表 6.354. 参数摘要

名称	类型	方向	概述
<code>all_content</code>	布尔值	in	指明是否应将主机网络接口的所有属性包含在响应中。
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>nic</code>	HostNic	out	

6.114.1.1. all_content

指明是否应将主机网络接口的所有属性包含在响应中。

默认情况下排除以下属性：

- `virtual_functions_configuration`

例如，要检索主机 '123' 的完整表示网络接口 '456'：

```
GET /ovirt-engine/api/hosts/123/nics/456?all_content=true
```



注意

这些属性默认不包含在内，因为检索它们会影响性能。它们很少使用，需要对数据库进行其他查询。谨慎使用此参数，且仅在有特殊需要时才使用。

6.114.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.114.2. 更新虚拟功能配置 POST

如果当前资源代表启用了 SR-IOV 的 NIC，则该操作会更新虚拟功能配置。输入应至少包含以下属性之一：

- **allNetworksAllowed**
- **numberOfVirtualFunctions**

请参阅 `HostNicVirtualFunctionsConfiguration` 类型以了解属性的含义。

表 6.355. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
virtual_functions_configuration	<code>HostNicVirtualFunctionsConfiguration</code>	in	

6.115. HOSTNICS

用于管理主机的网络接口的服务。

表 6.356. 方法概述

名称	概述
list	返回主机的网络接口列表。

6.115.1. 列出 GET

返回主机的网络接口列表。

无法保证返回的网络接口列表的顺序。

表 6.357. 参数摘要

名称	类型	方向	概述
all_content	布尔值	in	指明是否应将主机网络接口的所有属性包含在响应中。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大 NIC 数。
nics	HostNic[]	out	

6.115.1.1. all_content

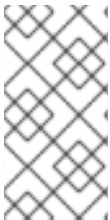
指明是否应将主机网络接口的所有属性包含在响应中。

默认情况下排除以下属性：

- **virtual_functions_configuration**

例如，要检索主机 '123' 的网络接口 '456' 的完整表示法：

```
GET /ovirt-engine/api/hosts/123/nics?all_content=true
```



注意

这些属性默认不包含在内，因为检索它们会影响性能。它们很少使用，需要对数据库进行其他查询。谨慎使用此参数，且仅在有特殊需要时才使用。

6.115.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.115.1.3. Max

设置要返回的最大 NIC 数。如果没有指定，则会返回所有 NIC。

6.116. HOSTNUMANODE

表 6.358. 方法概述

名称	概述
get	

6.116.1. 获取 GET

表 6.359. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
node	NumaNode	out	

6.116.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.117. HOSTNUMANODES

表 6.360. 方法概述

名称	概述
list	返回主机的 NUMA 节点列表。

6.117.1. 列出 GET

返回主机的 NUMA 节点列表。

NUMA 节点返回列表的顺序不受保证。

表 6.361. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大节点数。
节点	NumaNode[]	out	

6.117.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.117.1.2. Max

设置要返回的最大节点数。如果没有指定，则返回所有节点。

6.118. HOSTSTORAGE

用于管理主机存储的服务。

表 6.362. 方法概述

名称	概述
list	获取存储列表。

6.118.1. 列出 GET

获取存储列表。

GET /ovirt-engine/api/hosts/123/storage

您得到的 XML 响应如下：

```
<host_storages>
  <host_storage id="123">
    ...
  </host_storage>
  ...
</host_storages>
```

无法保证返回的存储列表的顺序。

表 6.363. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
report_status	布尔值	in	指明是否应检查存储中的 LUN 状态。
Storages	HostStorage []	out	检索的存储列表。

6.118.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.118.1.2. report_status

指明是否应检查存储中的 LUN 状态。检查 LUN 状态是一种重量型操作，用户并不总是需要这个数据。此参数将提供选项，使其不执行 LUN 的状态检查。

默认为 向后 兼容。

这里有一个 LUN 状态为 的示例：

```
<host_storage id="123">
  <logical_units>
    <logical_unit id="123">
      <lun_mapping>0</lun_mapping>
      <paths>1</paths>
      <product_id>lun0</product_id>
      <serial>123</serial>
      <size>10737418240</size>
      <status>used</status>
      <vendor_id>LIO-ORG</vendor_id>
      <volume_group_id>123</volume_group_id>
    </logical_unit>
  </logical_units>
  <type>iscsi</type>
</host id="123"/>
</host_storage>
```

这里有一个没有 LUN 状态的示例：

```
<host_storage id="123">
  <logical_units>
    <logical_unit id="123">
      <lun_mapping>0</lun_mapping>
      <paths>1</paths>
      <product_id>lun0</product_id>
      <serial>123</serial>
      <size>10737418240</size>
      <vendor_id>LIO-ORG</vendor_id>
      <volume_group_id>123</volume_group_id>
    </logical_unit>
  </logical_units>
  <type>iscsi</type>
</host id="123"/>
</host_storage>
```

6.119. 主机

管理主机的服务。

表 6.364. 方法概述

名称	概述
add	创建新主机.
list	获取所有可用主机的列表。

6.119.1. 添加 POST

创建新主机.

主机基于 **host** 参数的属性创建。**name**、**address** 和 **root_password** 属性是必需的。

例如，要添加主机，请发送以下请求：

```
POST /ovirt-engine/api/hosts
```

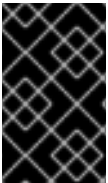
使用以下请求正文：

```
<host>
  <name>myhost</name>
  <address>myhost.example.com</address>
  <root_password>myrootpassword</root_password>
</host>
```



注意

root_password 元素仅包含在客户端提供的初始表示法中，不会在从后续请求返回的表示中公开。



重要

自引擎版本 4.1.2 起，当新添加主机时，主机防火墙定义将默认被覆盖。

要添加托管引擎主机，请使用可选的 **deploy_hosted_engine** 参数：

```
POST /ovirt-engine/api/hosts?deploy_hosted_engine=true
```

如果集群有一个支持自动部署的默认外部网络供应商，则在添加主机时会部署外部网络供应商。只有 OVN 的外部网络提供程序支持自动部署。要部署集群中定义的外部网络供应商，请在添加主机时覆盖外部网络供应商，请发送以下请求：

```
POST /ovirt-engine/api/hosts
```

使用一个请求正文，其中包含对 **external_network_provider_configuration** 中所需提供程序的引用：

```
<host>
  <name>myhost</name>
  <address>myhost.example.com</address>
  <root_password>123456</root_password>
  <external_network_provider_configurations>
    <external_network_provider_configuration>
      <external_network_provider name="ovirt-provider-ovn"/>
    </external_network_provider_configuration>
  </external_network_provider_configurations>
</host>
```

表 6.365. 参数摘要

名称	类型	方向	概述
激活	布尔值	in	当设置为 true 时，此主机将在安装完成后激活。
deploy_hosted_engine	布尔值	in	当设置为 true 时，此主机将部署托管引擎组件。
host	主机	in/Out	创建新主机的主机定义作为参数传递，而新创建的主机将返回。
reboot	布尔值	in	指明是否应当在安装成功后重新启动该主机。
undeploy_hosted_engine	布尔值	in	当设置为 true 时，此主机将取消部署托管引擎组件，且不会作为高可用性集群的一部分运行。

6.119.1.1. 激活

当设置为 **true** 时，此主机将在安装完成后激活。当设置为 **false** 时，主机安装后将保持 **维护** 状态。如果没有此参数，将解释为 **true**，因为所需的默认行为是在安装后激活主机。

6.119.1.2. deploy_hosted_engine

当设置为 **true** 时，此主机将部署托管引擎组件。缺少的值被视为 **true**，即部署托管引擎组件。省略此参数等于 **false**，主机不会在托管引擎区域执行任何操作。

6.119.1.3. reboot

指明是否应当在安装成功后重新启动该主机。默认值为 **true**。

6.119.1.4. undeploy_hosted_engine

当设置为 **true** 时，此主机将取消部署托管引擎组件，且不会作为高可用性集群的一部分运行。缺少的值被视为 **true**，即 **un-deploy**。省略此参数等于 **false**，主机不会在托管引擎区域执行任何操作。

6.119.2. 列出 GET

获取所有可用主机的列表。

例如，列出发送以下请求的主机：

GET /ovirt-engine/api/hosts

响应正文类似如下：

```
<hosts>
  <host href="/ovirt-engine/api/hosts/123" id="123">
    ...
  </host>
  <host href="/ovirt-engine/api/hosts/456" id="456">
    ...
  </host>
  ...
</host>
```

只有 `search` 参数中包含 `sortby` 子句时，才会保证返回的主机列表的顺序。

表 6.366. 参数摘要

名称	类型	方向	概述
<code>all_content</code>	布尔值	in	指明是否应将主机的所有属性包含在响应中。
<code>case_sensitive</code>	布尔值	in	指明是否应该在考虑大小写的情况下使用 <code>search</code> 参数执行搜索。
<code>check_vms_in_affinity_closure</code>	布尔值	in	此参数可与 <code>migration_target_of</code> 一起使用，为列出的虚拟机和所有其他与列出的虚拟机执行密切关系的虚拟机获取有效的迁移目标。
<code>filter</code>	布尔值	in	指明是否应根据用户的权限过滤结果。
<code>follow</code>	字符串	in	<i>指明应遵循哪些内联链接。</i>
主机	Host[]	out	
<code>max</code>	整数	in	设置要返回的最大主机数。
<code>migration_target_of</code>	字符串	in	接受以逗号分隔的虚拟机 ID 列表，并返回可迁移到这些虚拟机的主机。
<code>search</code>	字符串	in	用于限制返回的主机的查询字符串。

6.119.2.1. all_content

指明是否应将主机的所有属性包含在响应中。

默认情况下，排除以下主机属性：

- `hosted_engine`

例如，要检索主机的完整表示法：

```
GET /ovirt-engine/api/hosts?all_content=true
```



注意

这些属性默认不包含在内，因为检索它们会影响性能。它们很少使用，需要对数据库进行其他查询。谨慎使用此参数，且仅在有特殊需要时才使用。

6.119.2.2. case_sensitive

指明是否应该在考虑大小写的情况下使用 `search` 参数执行搜索。默认值为 `true`，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 `false`。

6.119.2.3. check_vms_in_affinity_closure

此参数可与 `migration_target_of` 一起使用，为列出的虚拟机和所有其他与列出的虚拟机执行密切关系的虚拟机获取有效的迁移目标。

这在虚拟机与正关联组中的其他虚拟机一起迁移时非常有用。

默认值为 `false`。

```
GET /ovirt-engine/api/hosts?  
migration_target_of=123,456&check_vms_in_affinity_closure=true
```

6.119.2.4. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.119.2.5. Max

设置要返回的最大主机数。如果未指定，则返回所有主机。

6.119.2.6. migration_target_of

接受以逗号分隔的虚拟机 ID 列表，并返回可迁移到这些虚拟机的主机。

例如，要检索 ID 为 123 的虚拟机和 ID 为 456 的虚拟机可以迁移到的主机列表，请发送以下请求：

```
GET /ovirt-engine/api/hosts?migration_target_of=123,456
```

6.120. ICON

用于管理图标的服务（只读）。

表 6.367. 方法概述

名称	概述
get	获取图标.

6.120.1. 获取 GET

获取图标.

```
GET /ovirt-engine/api/icons/123
```

您将获得类似以下的 XML 响应：

```
<icon id="123">
  <data>Some binary data here</data>
  <media_type>image/png</media_type>
</icon>
```

表 6.368. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
icon	icon	out	检索图标。

6.120.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.121. ICONS

用于管理图标的服务。

表 6.369. 方法概述

名称	概述
list	获取图标列表。

6.121.1. 列出 GET

获取图标列表。

GET /ovirt-engine/api/icons

您将获得类似以下的 XML 响应：

```
<icons>
  <icon id="123">
    <data>...</data>
    <media_type>image/png</media_type>
  </icon>
  ...
</icons>
```

无法保证返回的图标列表的顺序。

表 6.370. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
icons	Icon[]	out	检索到的图标列表。
max	整数	in	设置要返回的最大图标数。

6.121.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.121.1.2. Max

设置要返回的最大图标数。如果没有指定，则返回所有图标。

6.122. IMAGE

表 6.371. 方法概述

名称	概述
get	
import	导入映像。

6.122.1. 获取 GET

表 6.372. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
image	Image	out	

6.122.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.122.2. 导入 POST

导入映像。

如果 `import_as_template` 参数为 `true`，则映像将作为模板导入，否则它将作为磁盘导入。

将模板导入为模板时，可以通过可选的 `template.name` 参数指定模板的名称。如果未指定该参数，则引擎将自动将该模板的名称指定为 `GlanceTemplate-x`（其中 `x` 是七个随机十六进制字符）。

作为磁盘导入时，可以通过可选的 `disk.name` 参数指定磁盘名称。如果未指定该参数，则引擎将自动将该磁盘名称指定为 `GlanceDisk-x`（其中 `x` 将是映像标识符的七个十六进制字符）。

建议始终明确指定模板或磁盘名称，以避免引擎生成的这些自动名称。

表 6.373. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行导入。
<code>cluster</code>	Cluster	in	如果 <code>import_as_template</code> 参数设为 <code>true</code> ，则镜像应导入到的集群。
<code>disk</code>	Disk	in	要导入的磁盘。
<code>import_as_template</code>	布尔值	in	指定是否应从导入磁盘创建模板。
<code>storage_domain</code>	StorageDomain	in	磁盘应导入到的存储域。
模板	Template	in	如果 <code>import_as_template</code> 参数设为 <code>true</code> ，则所创建的模板的名称。

6.123. IMAGETRANSFER

此服务提供控制镜像传输的机制。客户端必须使用 [第 6.124 节 “ImageTransfers” 服务添加来创建传输](#)，指示要在其中传输数据的镜像。

执行此操作后，转让由此服务管理。

使用 oVirt 的 Python SDK:

上传 ID 123 的磁盘（在数据中心的随机主机上）：

```
transfers_service = system_service.image_transfers_service()
transfer = transfers_service.add(
    types.ImageTransfer(
        disk=types.Disk(
            id='123'
        )
    )
)
```

在主机 ID 456 上上传 ID 123 的磁盘：

```
transfers_service = system_service.image_transfers_service()
transfer = transfers_service.add(
    types.ImageTransfer(
        disk=types.Disk(
            id='123'
        ),
        host=types.Host(
            id='456'
        )
    )
)
```

如果用户想要下载磁盘而不是上传，则应将 `下载` 指定为传输的 `方向` 属性。这将授予镜像的读取权限，而不是写入权限。

例如：

```
transfers_service = system_service.image_transfers_service()
transfer = transfers_service.add(
    types.ImageTransfer(
        disk=types.Disk(
            id='123'
        ),
        direction=types.ImageTransferDirection.DOWNLOAD
    )
)
```

传输具有控制上传/下载流程的阶段。实施此类流程的客户端应轮询/检查转让的阶段，并相应地操作。所有可能的阶段都可以在 `ImageTransferPhase` 中找到。

添加新转移后，其阶段将初始化。在转让更改之前，客户端必须轮询传输阶段。当阶段开始转移后，会话就可以开始转移。

例如：

```
transfer_service = transfers_service.image_transfer_service(transfer.id)
while transfer.phase == types.ImageTransferPhase.INITIALIZING:
    time.sleep(3)
    transfer = transfer_service.get()
```

在该阶段，如果转让的阶段是 `paused_system`，则无法成功建立会话。这可能是因为在 `ovirt-imageio-daemon` 没有在选择进行传输的主机中运行。可以通过调用管理它的服务恢复传输。

如果会话成功建立 - 返回的传输实体将包含 `transfer_url` 和 `proxy_url` 属性，客户端需要使用这些属性来传输所需的数据。客户端可以选择任何技术和工具来发送带有镜像数据的 HTTPS 请求。

- `transfer_url` 是其中一个虚拟机监控程序上运行的 `imageio` 服务器的地址。
- `proxy_url` 是 `imageio` 代理服务器的地址，如果您无法访问 `transfer_url`，则可以使用该地址。

若要传输映像，建议使用 `imageio` 客户端 `python` 库。

```
from ovirt_imageio import client

# Upload qcow2 image to virtual disk:
client.upload("disk.qcow2", transfer.transfer_url)

# Download virtual disk to qcow2 image:
client.download(transfer.transfer_url, "disk.qcow2")
```

您还可以使用 `imageio` REST API 上传和下载。如需了解更多相关信息，请参阅 `imageio` API 文档：

<http://ovirt.github.io/ovirt-imageio/images.html>

完成转让后，用户应致电完成。这将为完成转移过程做出最终调整和验证。

例如：

```
transfer_service.finalize()
```

如果出现错误，传输阶段将更改为 **finished_failure**，磁盘的状态将更改为 **Illegal**。否则，它将更改为 **completed_success**，磁盘将可供使用。在这两种情况下，转让实体将在后面很快删除。

使用 HTTP 和 cURL 调用：

- 对于上传，首先创建一个新磁盘：
 - 以字节为单位指定 'initial_size' 和 'provisioned_size'。
 - 'initial_size' 必须大于或与上传数据的大小相同。

```
POST /ovirt-engine/api/disks
```

请求正文如下：

```
<disk>
  <storage_domains>
    <storage_domain id="123"/>
  </storage_domains>
  <alias>mydisk</alias>
  <initial_size>1073741824</initial_size>
  <provisioned_size>1073741824</provisioned_size>
  <format>raw</format>
</disk>
```

- 创建用于下载/卸载 ID 为 456 的磁盘 的新镜像传输：

```
POST /ovirt-engine/api/imagetransfers
```

请求正文如下：

```
<image_transfer>
  <disk id="456"/>
  <direction>upload|download</direction>
</image_transfer>
```

将响应：

```
<image_transfer id="123">
  <direction>download|upload</direction>
  <phase>initializing|transferring</phase>
  <proxy_url>https://proxy_fqdn:54323/images/41c732d4-2210-4e7b-9e5c-
4e2805baadbb</proxy_url>
  <transfer_url>https://daemon_fqdn:54322/images/41c732d4-2210-4e7b-9e5c-
4e2805baadbb</transfer_url>
  ...
</image_transfer>
```

注意：如果阶段为"初始化"，请轮询 `image_transfer`，直到其阶段变为"传输"。

- 使用 'transfer_url' 或 'proxy_url' 来调用 curl 命令：
- 使用 'transfer_url' 直接从/到 `ovirt-imageio-daemon` 传输，或者使用 'proxy_url' 从/ `ovirt-imageio-proxy` 传输。注意：使用代理可以缓解没有直接连接到守护进程计算机的情况，例如 `vdsm` 机器位于与引擎不同的网络中。

-usersdownload：

```
$ curl --cacert /etc/pki/ovirt-engine/ca.pem https://daemon_fqdn:54322/images/41c732d4-2210-
4e7b-9e5c-4e2805baadbb -o <output_file>
```

- SASupload：

```
$ curl --cacert /etc/pki/ovirt-engine/ca.pem --upload-file <file_to_upload> -X PUT
https://daemon_fqdn:54322/images/41c732d4-2210-4e7b-9e5c-4e2805baadbb
```

- 通过调用操作来结束镜像传输：

POST /ovirt-engine/api/imagetransfers/123/finalize

请求正文如下：

```
<action />
```

表 6.374. 方法概述

名称	概述
取消	取消镜像传输会话。
extend	扩展映像传输会话。
Finish	在完成传输数据后，完成传输。
get	获取映像传输实体。
pause	暂停映像传输会话。
resume	恢复镜像传输会话。

6.123.1. 取消 POST

取消镜像传输会话。这会终止传输操作并移除部分镜像。

6.123.2. 扩展 POST

扩展映像传输会话。

6.123.3. 最终 POST

在完成传输数据后，完成传输。

这样将确保传输的数据有效，并且适合传输中的目标映像实体。具体来说，将验证映像实体是否为 QCOW 磁盘，上传的数据实际上是 QCOW 文件，并且映像没有后备文件。

6.123.4. 获取 GET

获取映像传输实体。

表 6.375. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>image_transfer</code>	<code>ImageTransfer</code>	out	

6.123.4.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.123.5. 暂停 POST

暂停映像传输会话。

6.123.6. 恢复 POST

恢复镜像传输会话。客户需要轮询传输阶段，直到这与恢复不同。例如：

```
transfer_service = transfers_service.image_transfer_service(transfer.id)
transfer_service.resume()
transfer = transfer_service.get()
```

```
while transfer.phase == types.ImageTransferPhase.RESUMING:
    time.sleep(1)
    transfer = transfer_service.get()
```

6.124. IMAGETRANSFERS

此服务管理镜像传输，以用于在红帽虚拟化中执行镜像 I/O API。[有关更多文档，请参阅图像传送](#)。

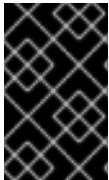
表 6.376. 方法概述

名称	概述
<code>add</code>	添加新映像传输。

名称	概述
list	检索当前正在执行的镜像传输的列表。

6.124.1. 添加 POST

添加新映像传输。需要指定映像、磁盘或磁盘快照才能进行新的传输。



重要

image 属性已弃用，因为引擎的版本 4.2。改为使用 **disk** 或 **snapshot** 属性。

创建用于下载或上传 **磁盘** 的新镜像传输：

要创建镜像传输来下载或上传 ID 为 **123** 的磁盘，请发送以下请求：

```
POST /ovirt-engine/api/imagetransfers
```

这样的请求正文：

```
<image_transfer>
  <disk id="123"/>
  <direction>upload|download</direction>
</image_transfer>
```

创建用于下载或上传 **disk_snapshot** 的新映像传输：

要创建映像传输来下载或上传 ID 为 **456** 的 **disk_snapshot**，请发送以下请求：

```
POST /ovirt-engine/api/imagetransfers
```

这样的请求正文：

```
<image_transfer>
  <snapshot id="456"/>
  <direction>download|upload</direction>
```

`</image_transfer>`

表 6.377. 参数摘要

名称	类型	方向	概述
<code>image_transfer</code>	<code>ImageTransfer</code>	in/Out	要添加的镜像传输。

6.124.2. 列出 GET

检索当前正在执行的镜像传输的列表。

返回的镜像传输列表的顺序不受保证。

表 6.378. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>image_transfer</code>	<code>ImageTransfer[]</code>	out	当前正在执行的镜像传输列表。

6.124.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.125. 镜像

管理存储域或 OpenStack 映像提供程序中可用的镜像集合。

表 6.379. 方法概述

名称	概述
<code>list</code>	返回存储域或提供程序中可用映像的列表。

6.125.1. 列出 GET

返回存储域或提供程序中可用映像的列表。

无法保证返回的镜像列表的顺序。

表 6.380. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
images	image[]	out	
max	整数	in	设置要返回的最大镜像数。

6.125.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.125.1.2. Max

设置要返回的最大镜像数。如果没有指定，则返回所有镜像。

6.126. INSTANCETYPE

表 6.381. 方法概述

名称	概述
get	获取特定实例类型及其属性。
remove	从系统移除特定的实例类型。
update	更新特定的实例类型及其属性。

6.126.1. 获取 GET

获取特定实例类型及其属性。

GET /ovirt-engine/api/instancetypees/123

表 6.382. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
instance_type	InstanceType	out	

6.126.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.126.2. 删除 DELETE

从系统移除特定的实例类型。

如果在删除实例类型后使用实例类型 X 创建虚拟机，则虚拟机的实例类型将设置为 **custom**。

DELETE /ovirt-engine/api/instancetypees/123

表 6.383. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.126.3. 更新 PUT

更新特定的实例类型及其属性。

所有属性均可在创建后编辑。如果使用实例类型 X 创建虚拟机，并且更新了实例类型 X 中的一些配置，引擎将自动更新虚拟机的配置。

PUT /ovirt-engine/api/instancetypees/123

例如，要将实例类型 123 的内存更新为 1 GiB，并将 cpu 拓扑设置为 2 个插槽和 1 个内核，请发送类似如下的请求：

```

<instance_type>
  <memory>1073741824</memory>
  <cpu>
    <topology>
      <cores>1</cores>
      <sockets>2</sockets>
      <threads>1</threads>
    </topology>
  </cpu>
</instance_type>

```

表 6.384. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
instance_type	InstanceType	in/Out	

6.127. INSTANCETYPEGRAPHICSCONSOLE

表 6.385. 方法概述

名称	概述
get	获取实例类型的图形控制台配置。
remove	从实例类型中删除图形控制台。

6.127.1. 获取 GET

获取实例类型的图形控制台配置。

表 6.386. 参数摘要

名称	类型	方向	概述
console	GraphicsConsole	out	关于实例类型的图形控制台的信息。
follow	字符串	in	指明应遵循哪些内联链接。

6.127.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.127.2. 删除 DELETE

从实例类型中删除图形控制台。

表 6.387. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.128. INSTANCETYPEGRAPHICSCONSOLES

表 6.388. 方法概述

名称	概述
add	向实例类型添加新的图形控制台。
list	列出实例类型的所有配置的图形控制台。

6.128.1. 添加 POST

向实例类型添加新的图形控制台。

表 6.389. 参数摘要

名称	类型	方向	概述
console	GraphicsConsole	in/Out	

6.128.2. 列出 GET

列出实例类型的所有配置的图形控制台。

无法保证返回的图形控制台列表的顺序。

表 6.390. 参数摘要

名称	类型	方向	概述
consoles	GraphicsConsole[]	out	实例类型的图形控制台列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大控制台数。

6.128.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.128.2.2. Max

设置要返回的最大控制台数。如果没有指定，则返回所有控制台。

6.129. INSTANCETYPENIC

表 6.391. 方法概述

名称	概述
get	获取实例类型的网络接口配置。
remove	从实例类型中删除网络接口。
update	更新实例类型的网络接口配置。

6.129.1. 获取 GET

获取实例类型的网络接口配置。

表 6.392. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
nic	NIC	out	

6.129.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.129.2. 删除 DELETE

从实例类型中删除网络接口。

表 6.393. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.129.3. 更新 PUT

更新实例类型的网络接口配置。

表 6.394. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
nic	NIC	in/Out	

6.130. INSTANCETYPEENICS

表 6.395. 方法概述

名称	概述
add	向实例类型添加新网络接口。
list	列出实例类型的所有已配置网络接口。

6.130.1. 添加 POST

向实例类型添加新网络接口。

表 6.396. 参数摘要

名称	类型	方向	概述
nic	NIC	in/Out	

6.130.2. 列出 GET

列出实例类型的所有已配置网络接口。

无法保证返回的网络接口列表的顺序。

表 6.397. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大 NIC 数。
nics	Nic[]	out	
search	字符串	in	用于限制返回的模板的查询字符串。

6.130.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.130.2.2. Max

设置要返回的最大 NIC 数。如果没有指定，则会返回所有 NIC。

6.131. INSTANCETYPEWATCHDOG

表 6.398. 方法概述

名称	概述
get	获取实例类型的 watchdog 配置。
remove	从实例类型中删除 watchdog。

名称	概述
update	更新实例类型的 watchdog 配置。

6.131.1. 获取 GET

获取实例类型的 watchdog 配置。

表 6.399. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
watchdog	Watchdog	out	

6.131.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.131.2. 删除 DELETE

从实例类型中删除 watchdog。

表 6.400. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.131.3. 更新 PUT

更新实例类型的 watchdog 配置。

表 6.401. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。

名称	类型	方向	概述
watchdog	Watchdog	in/Out	

6.132. INSTANCETYPEWATCHDOGS

表 6.402. 方法概述

名称	概述
add	向实例类型添加新的 watchdog。
list	列出实例类型的所有配置的 watchdog。

6.132.1. 添加 POST

向实例类型添加新的 watchdog。

表 6.403. 参数摘要

名称	类型	方向	概述
watchdog	Watchdog	in/Out	

6.132.2. 列出 GET

列出实例类型的所有配置的 watchdog。

返回的 watchdogs 列表的顺序不受保证。

表 6.404. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大 watchdogs 数。
search	字符串	in	用于限制返回的模板的查询字符串。
watchdogs	Watchdog[]	out	

6.132.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.132.2.2. Max

设置要返回的最大 `watchdogs` 数。如果未指定，则返回所有 `watchdogs`。

6.133. INSTANCETYPES

表 6.405. 方法概述

名称	概述
<code>add</code>	创建新的实例类型。
<code>list</code>	列出系统中的所有现有实例类型。

6.133.1. 添加 POST

创建新的实例类型。

这只需要 `name` 属性，并且可以包含虚拟机的所有硬件配置。

```
POST /ovirt-engine/api/instancetypes
```

这样的请求正文：

```
<instance_type>
  <name>myinstancetype</name>
</template>
```

使用请求正文创建包含所有硬件配置的实例类型，如下方所示：

```
<instance_type>
  <name>myinstancetype</name>
  <console>
    <enabled>true</enabled>
  </console>
  <cpu>
```

```

<topology>
  <cores>2</cores>
  <sockets>2</sockets>
  <threads>1</threads>
</topology>
</cpu>
<custom_cpu_model>AMD Opteron_G2</custom_cpu_model>
<custom_emulated_machine>q35</custom_emulated_machine>
<display>
  <monitors>1</monitors>
  <single_qxl_pci>true</single_qxl_pci>
  <smartcard_enabled>true</smartcard_enabled>
  <type>spice</type>
</display>
<high_availability>
  <enabled>true</enabled>
  <priority>1</priority>
</high_availability>
<io>
  <threads>2</threads>
</io>
<memory>4294967296</memory>
<memory_policy>
  <ballooning>true</ballooning>
  <guaranteed>268435456</guaranteed>
</memory_policy>
<migration>
  <auto_converge>inherit</auto_converge>
  <compressed>inherit</compressed>
  <policy id="00000000-0000-0000-0000-000000000000"/>
</migration>
<migration_downtime>2</migration_downtime>
<os>
  <boot>
    <devices>
      <device>hd</device>
    </devices>
  </boot>
</os>
<rng_device>
  <rate>
    <bytes>200</bytes>
    <period>2</period>
  </rate>
  <source>urandom</source>
</rng_device>
<soundcard_enabled>true</soundcard_enabled>
<usb>
  <enabled>true</enabled>
  <type>native</type>
</usb>
<virtio_scsi>
  <enabled>true</enabled>
</virtio_scsi>
</instance_type>

```

表 6.406. 参数摘要

名称	类型	方向	概述
<code>instance_type</code>	<code>InstanceType</code>	in/Out	

6.133.2. 列出 GET

列出系统中的所有现有实例类型。

无法保证返回的实例类型列表的顺序。

表 6.407. 参数摘要

名称	类型	方向	概述
<code>case_sensitive</code>	布尔值	in	指明是否应该在考虑大小写的情况下使用 <code>search</code> 参数执行搜索。
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>instance_type</code>	<code>InstanceType[]</code>	out	
<code>max</code>	整数	in	设置要返回的实例类型的最大数量。
<code>search</code>	字符串	in	用于限制返回的模板的查询字符串。

6.133.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 `search` 参数执行搜索。默认值为 `true`，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 `false`。

6.133.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.133.2.3. Max

设置要返回的实例类型的最大数量。如果没有指定，则返回所有实例类型。

6.134. ISCSIBOND

表 6.408. 方法概述

名称	概述
get	
remove	删除现有 iSCSI 绑定。
update	更新 iSCSI 绑定。

6.134.1. 获取 GET

表 6.409. 参数摘要

名称	类型	方向	概述
bond	iscsiBond	out	iSCSI 绑定。
follow	字符串	in	指明应遵循哪些内联链接。

6.134.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.134.2. 删除 DELETE

删除现有 iSCSI 绑定。

例如，要删除 iSCSI 绑定 456，发送类似如下的请求：

```
DELETE /ovirt-engine/api/datacenters/123/iscsibonds/456
```

表 6.410. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.134.3. 更新 PUT

更新 iSCSI 绑定。

iSCSI 绑定的更新只能在 名称和 描述 属性上完成。例如，要更新数据中心 123 的 iSCSI bond 456，请发送如下请求：

```
PUT /ovirt-engine/api/datacenters/123/iscsibonds/1234
```

请求正文应类似如下：

```
<iscsi_bond>
  <name>mybond</name>
  <description>My iSCSI bond</description>
</iscsi_bond>
```

表 6.411. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
bond	iscsiBond	in/Out	要更新的 iSCSI 绑定。

6.135. ISCSIBONDS

表 6.412. 方法概述

名称	概述
add	在数据中心上创建新的 iSCSI 绑定。
list	返回数据中心的 iSCSI 绑定列表。

6.135.1. 添加 POST

在数据中心上创建新的 iSCSI 绑定。

例如，要使用存储连接 456 和 789 在数据中心 123 上创建新的 iSCSI 绑定，请发送如下请求：

```
POST /ovirt-engine/api/datacenters/123/iscsibonds
```

请求正文应类似如下：

```
<iscsi_bond>
  <name>mybond</name>
  <storage_connections>
    <storage_connection id="456"/>
    <storage_connection id="789"/>
  </storage_connections>
  <networks>
    <network id="abc"/>
  </networks>
</iscsi_bond>
```

表 6.413. 参数摘要

名称	类型	方向	概述
bond	lscsiBond	in/Out	

6.135.2. 列出 GET

返回数据中心的 iSCSI 绑定列表。

无法保证返回的 iSCSI 绑定列表的顺序。

表 6.414. 参数摘要

名称	类型	方向	概述
bonds	lscsiBond[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大绑定数。

6.135.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.135.2.2. Max

设置要返回的最大绑定数。如果没有指定所有绑定，则会返回所有绑定。

6.136. JOB

用于管理作业的服务。

表 6.415. 方法概述

名称	概述
clear	设置系统要清除的外部作业执行。
end	将外部作业执行标记为结束。
get	检索作业。

6.136.1. 清除 POST

设置系统要清除的外部作业执行。

例如，要设置标识符为 123 的作业，发送以下请求：

```
POST /ovirt-engine/api/jobs/clear
```

使用以下请求正文：

```
<action/>
```

表 6.416. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.136.2. 最终 POST

将外部作业执行标记为结束。

例如，要终止标识符为 **123** 的作业，发送以下请求：

```
POST /ovirt-engine/api/jobs/end
```

使用以下请求正文：

```
<action>
  <force>true</force>
  <status>finished</status>
</action>
```

表 6.417. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
force	布尔值	in	指明该作业是否应该被强制终止。
succeeded	布尔值	in	表示作业应标记为成功完成或失败。

6.136.2.1. succeeded

表示作业应标记为成功完成或失败。

此参数是可选的，默认值为 **true**。

6.136.3. 获取 GET

检索作业。

```
GET /ovirt-engine/api/jobs/123
```

您将在 **XML** 中收到类似以下的响应：

```
<job href="/ovirt-engine/api/jobs/123" id="123">
  <actions>
    <link href="/ovirt-engine/api/jobs/123/clear" rel="clear"/>
    <link href="/ovirt-engine/api/jobs/123/end" rel="end"/>
  </actions>
```

```

<description>Adding Disk</description>
<link href="/ovirt-engine/api/jobs/123/steps" rel="steps"/>
<auto_cleared>true</auto_cleared>
<end_time>2016-12-12T23:07:29.758+02:00</end_time>
<external>false</external>
<last_updated>2016-12-12T23:07:29.758+02:00</last_updated>
<start_time>2016-12-12T23:07:26.593+02:00</start_time>
<status>failed</status>
<owner href="/ovirt-engine/api/users/456" id="456"/>
</job>

```

表 6.418. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
job	Job	out	检索作业的表达法。

6.136.3.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.137. JOBS

用于管理作业的服务。

表 6.419. 方法概述

名称	概述
add	添加外部作业。
list	检索作业的表达法。

6.137.1. 添加 POST

添加外部作业。

例如，使用以下请求添加作业：

```
POST /ovirt-engine/api/jobs
```

使用以下请求正文：

```
<job>
  <description>Doing some work</description>
  <auto_cleared>true</auto_cleared>
</job>
```

响应应类似于：

```
<job href="/ovirt-engine/api/jobs/123" id="123">
  <actions>
    <link href="/ovirt-engine/api/jobs/123/clear" rel="clear"/>
    <link href="/ovirt-engine/api/jobs/123/end" rel="end"/>
  </actions>
  <description>Doing some work</description>
  <link href="/ovirt-engine/api/jobs/123/steps" rel="steps"/>
  <auto_cleared>true</auto_cleared>
  <external>true</external>
  <last_updated>2016-12-13T02:15:42.130+02:00</last_updated>
  <start_time>2016-12-13T02:15:42.130+02:00</start_time>
  <status>started</status>
  <owner href="/ovirt-engine/api/users/456" id="456"/>
</job>
```

表 6.420. 参数摘要

名称	类型	方向	概述
job	Job	in/Out	将添加的作业。

6.137.2. 列出 GET

检索作业的表达法。

GET /ovirt-engine/api/jobs

您将在 XML 中收到类似以下的响应：

```
<jobs>
  <job href="/ovirt-engine/api/jobs/123" id="123">
    <actions>
      <link href="/ovirt-engine/api/jobs/123/clear" rel="clear"/>
      <link href="/ovirt-engine/api/jobs/123/end" rel="end"/>
    </actions>
    <description>Adding Disk</description>
```

```

<link href="/ovirt-engine/api/jobs/123/steps" rel="steps"/>
<auto_cleared>true</auto_cleared>
<end_time>2016-12-12T23:07:29.758+02:00</end_time>
<external>false</external>
<last_updated>2016-12-12T23:07:29.758+02:00</last_updated>
<start_time>2016-12-12T23:07:26.593+02:00</start_time>
<status>failed</status>
<owner href="/ovirt-engine/api/users/456" id="456"/>
</job>
...
</jobs>

```

无法保证返回的作业列表的顺序。

表 6.421. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
follow	字符串	in	指明应遵循哪些内联链接。
任务	Job[]	out	作业表示。
max	整数	in	设置要返回的最大作业数。
search	字符串	in	用于限制返回的作业的查询字符串。

6.137.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 **search** 参数执行搜索。默认值为 **true**，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 **false**。

6.137.2.2. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.137.2.3. Max

设置要返回的最大作业数。如果未指定，则返回所有作业。

6.138. KATELLOERRATA

用于管理 Katello 勘误表的服务。该信息是从 Katello 检索而来。

表 6.422. 方法概述

名称	概述
list	检索 Katello 勘误表的表示形式。

6.138.1. 列出 GET

检索 Katello 勘误表的表示形式。

GET /ovirt-engine/api/katelloerrata

您将在 XML 中收到类似以下的响应：

```
<katello_errata>
  <katello_erratum href="/ovirt-engine/api/katelloerrata/123" id="123">
    <name>RHBA-2013:XYZ</name>
    <description>The description of the erratum</description>
    <title>some bug fix update</title>
    <type>bugfix</type>
    <issued>2013-11-20T02:00:00.000+02:00</issued>
    <solution>Few guidelines regarding the solution</solution>
    <summary>Updated packages that fix one bug are now available for XYZ</summary>
    <packages>
      <package>
        <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
      </package>
      ...
    </packages>
  </katello_erratum>
  ...
</katello_errata>
```

返回的勘误列表的顺序无法保证。

表 6.423. 参数摘要

名称	类型	方向	概述
勘误	KatelloErratum[]	out	Katello 勘误表的表示。

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的勘误表的最大数量。

6.138.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.138.1.2. Max

设置要返回的勘误表的最大数量。如果没有指定，则会返回所有勘误。

6.139. KATELLOERRATUM

用于管理 Katello 勘误的服务。

表 6.424. 方法概述

名称	概述
get	检索 Katello 勘误。

6.139.1. 获取 GET

检索 Katello 勘误。

GET /ovirt-engine/api/katelloerrata/123

您将在 XML 中收到类似以下的响应：

```
<katello_erratum href="/ovirt-engine/api/katelloerrata/123" id="123">
  <name>RHBA-2013:XYZ</name>
  <description>The description of the erratum</description>
  <title>some bug fix update</title>
  <type>bugfix</type>
  <issued>2013-11-20T02:00:00.000+02:00</issued>
  <solution>Few guidelines regarding the solution</solution>
  <summary>Updated packages that fix one bug are now available for XYZ</summary>
```

```

<packages>
  <package>
    <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
  </package>
  ...
</packages>
</katello_erratum>

```

表 6.425. 参数摘要

名称	类型	方向	概述
erratum	KatelloErratum	out	检索 Katello 勘误的表示形式。
follow	字符串	in	指明应遵循哪些内联链接。

6.139.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.140. LINKLAYERDISCOVERYPROTOCOL

用于获取由链路层发现协议(LLDP)接收的信息元素的服务。

表 6.426. 方法概述

名称	概述
list	获取 LLDP 接收的信息元素。

6.140.1. 列出 GET

获取 LLDP 接收的信息元素。

表 6.427. 参数摘要

名称	类型	方向	概述
元素	LinkLayerDiscoveryProtocolElement[]	out	检索 LLDP 接收的信息元素列表。
follow	字符串	in	指明应遵循哪些内联链接。

6.140.1.1. 元素

检索 LLDP 接收的信息元素列表。

例如，要检索主机 123 上的 NIC 321 上接收的信息元素，请发送类似如下的请求：

```
GET ovirt-engine/api/hosts/123/nics/321/linklayerdiscoveryprotocolelements
```

它将返回类似如下的响应：

```
<link_layer_discovery_protocol_elements>
...
<link_layer_discovery_protocol_element>
  <name>Port Description</name>
  <properties>
    <property>
      <name>port description</name>
      <value>Summit300-48-Port 1001</value>
    </property>
  </properties>
  <type>4</type>
</link_layer_discovery_protocol_element>
...
</link_layer_discovery_protocol_elements>
```

6.140.1.2. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.141. MACPOOL

表 6.428. 方法概述

名称	概述
get	
remove	删除 MAC 地址池。
update	更新 MAC 地址池。

6.141.1. 获取 GET

表 6.429. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
池	MacPool	out	

6.141.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.141.2. 删除 DELETE

删除 MAC 地址池。

例如，要删除 ID 123 的 MAC 地址池，发送类似如下的请求：

```
DELETE /ovirt-engine/api/macpools/123
```

表 6.430. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.141.3. 更新 PUT

更新 MAC 地址池。

可以更新 `name`、`description`、`allow_duplicates` 和 `ranges` 属性。

例如，更新 id 123 的 MAC 地址池可发送如下请求：

```
PUT /ovirt-engine/api/macpools/123
```

这样的请求正文：

```

<mac_pool>
  <name>UpdatedMACPool</name>
  <description>An updated MAC address pool</description>
  <allow_duplicates>false</allow_duplicates>
  <ranges>
    <range>
      <from>00:1A:4A:16:01:51</from>
      <to>00:1A:4A:16:01:e6</to>
    </range>
    <range>
      <from>02:1A:4A:01:00:00</from>
      <to>02:1A:4A:FF:FF:FF</to>
    </range>
  </ranges>
</mac_pool>

```

表 6.431. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
池	MacPool	in/Out	

6.142. MACPOOLS

表 6.432. 方法概述

名称	概述
add	创建新的 MAC 地址池。
list	返回系统的 MAC 地址池列表。

6.142.1. 添加 POST

创建新的 MAC 地址池。

创建 MAC 地址池需要名称和范围属性值。

例如，要创建 MAC 地址池发送如下请求：

```
POST /ovirt-engine/api/macpools
```

这样的请求正文：

```
<mac_pool>
  <name>MACPool</name>
  <description>A MAC address pool</description>
  <allow_duplicates>true</allow_duplicates>
  <default_pool>false</default_pool>
  <ranges>
    <range>
      <from>00:1A:4A:16:01:51</from>
      <to>00:1A:4A:16:01:e6</to>
    </range>
  </ranges>
</mac_pool>
```

表 6.433. 参数摘要

名称	类型	方向	概述
池	MacPool	in/Out	

6.142.2. 列出 GET

返回系统的 MAC 地址池列表。

MAC 地址池返回的列表不受保证。

表 6.434. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大池数。
pools	MacPool[]	out	

6.142.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.142.2.2. Max

设置要返回的最大池数。如果没有指定，则返回所有池。

6.143. 可测量

6.144. 可移动

表 6.435. 方法概述

名称	概述
Move	

6.144.1. 移动 POST

表 6.436. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移动。

6.145. NETWORK

管理网络的服务

表 6.437. 方法概述

名称	概述
get	获取逻辑网络.
remove	删除逻辑网络或逻辑网络与数据中心的关联。
update	更新逻辑网络.

6.145.1. 获取 GET

获取逻辑网络.

例如：

GET /ovirt-engine/api/networks/123

将响应：

```
<network href="/ovirt-engine/api/networks/123" id="123">
  <name>ovirtmgmt</name>
  <description>Default Management Network</description>
  <link href="/ovirt-engine/api/networks/123/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/networks/123/vnicprofiles" rel="vnicprofiles"/>
  <link href="/ovirt-engine/api/networks/123/networklabels" rel="networklabels"/>
  <mtu>0</mtu>
  <stp>false</stp>
  <usages>
    <usage>vm</usage>
  </usages>
  <data_center href="/ovirt-engine/api/datacenters/456" id="456"/>
</network>
```

表 6.438. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
network	Network	out	

6.145.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.145.2. 删除 DELETE

删除逻辑网络或逻辑网络与数据中心的关联。

例如，要删除逻辑网络 123，发送类似如下的请求：

DELETE /ovirt-engine/api/networks/123

每个网络都完全绑定到一个数据中心。因此，如果我们将网络与数据中心解除关联，其结果就与我们只是删除该网络一样。但是，可能更具体地指出我们删除了数据中心 123 的网络 456。

例如，要删除网络 456 到数据中心 123 的关联，请发送类似如下的请求：

```
DELETE /ovirt-engine/api/datacenters/123/networks/456
```



注意

要移除外部逻辑网络，必须通过 [OpenStack 网络 API](#) 将网络直接从其提供程序中删除。如果为提供商启用了 `auto_sync`，则代表红帽虚拟化内部外部网络的实体将自动删除，否则必须使用此方法移除该实体。

表 6.439. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行移除。

6.145.3. 更新 PUT

更新逻辑网络。

可以更新 `name`、`description`、`ip`、`vlan`、`stp` 和 `display` 属性。

例如，要更新逻辑网络 123 的描述，可发送如下请求：

```
PUT /ovirt-engine/api/networks/123
```

这样的请求正文：

```
<network>
  <description>My updated description</description>
</network>
```

网络的最大传输单元使用 `PUT` 请求来设置，以指定 `mtu` 属性的整数值。

例如，要设置最大传输单元会发送如下请求：

PUT /ovirt-engine/api/datacenters/123/networks/456

这样的请求正文：

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



注意

更新外部网络不会传播到提供程序。

表 6.440. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
network	Network	in/Out	

6.146. NETWORKATTACHMENT

表 6.441. 方法概述

名称	概述
get	
remove	
update	更新主机上指定的网络附加。

6.146.1. 获取 GET

表 6.442. 参数摘要

名称	类型	方向	概述
Attach	NetworkAttachment	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.146.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.146.2. 删除 DELETE

表 6.443. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.146.3. 更新 PUT

更新主机上指定的网络附加。

表 6.444. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
Attach	NetworkAttachment	in/Out	

6.147. NETWORKATTACHMENTS

管理主机或主机 NIC 的网络附加集合。

表 6.445. 方法概述

名称	概述
add	向网络接口添加新网络附加。
list	返回主机或主机 NIC 的网络附加列表。

6.147.1. 添加 POST

向网络接口添加新网络附加。

表 6.446. 参数摘要

名称	类型	方向	概述
Attach	NetworkAttachment	in/Out	

6.147.2. 列出 GET

返回主机或主机 NIC 的网络附加列表。

网络附加列表返回的顺序不受保证。

表 6.447. 参数摘要

名称	类型	方向	概述
附件	NetworkAttachment[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置返回的最大附件数。

6.147.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.147.2.2. Max

设置返回的最大附件数。如果没有指定，则返回所有附件。

6.148. NETWORKFILTER

管理网络过滤器。

```
<network_filter id="00000019-0019-0019-0019-00000000026b">
  <name>example-network-filter-b</name>
  <version>
    <major>4</major>
    <minor>0</minor>
    <build>-1</build>
```

```

<revision>-1</revision>
</version>
</network_filter>

```

请注意，版本指的是特定过滤器的最小支持版本。

表 6.448. 方法概述

名称	概述
get	检索网络过滤器的表示法。

6.148.1. 获取 GET

检索网络过滤器的表示法。

表 6.449. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
network_filter	NetworkFilter	out	

6.148.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.149. NETWORKFILTERS

表示只读网络过滤器子集合。

网络过滤器允许根据定义的规则过滤发送到/从虚拟机的 nic 发送的数据包。如需更多信息，请参阅 [NetworkFilter 服务文档](#)

从版本 3.0 开始，支持网络过滤器的不同版本。

为每个 **vnic** 配置集定义一个网络过滤器。

为特定网络定义 **vnic** 配置集。

网络可以分配给多个不同的集群。未来，将在集群级别上定义每个网络。

目前，每个网络都在数据中心级别上定义。每个网络的潜在网络过滤器由网络的数据中心兼容性版本 **V** 决定。**V** 必须是 **>=** 网络过滤器版本，以便为特定网络配置这个网络过滤器。请注意，如果将网络分配给带有支持网络过滤器的版本的集群，则该过滤器可能会因为数据中心版本较小而不可用，然后网络过滤器的版本便不可用。

列出特定集群的所有支持的网络过滤器示例：

```
GET http://localhost:8080/ovirt-engine/api/clusters/{cluster:id}/networkfilters
```

输出：

```
<network_filters>
  <network_filter id="00000019-0019-0019-0019-00000000026c">
    <name>example-network-filter-a</name>
    <version>
      <major>4</major>
      <minor>0</minor>
      <build>-1</build>
      <revision>-1</revision>
    </version>
  </network_filter>
  <network_filter id="00000019-0019-0019-0019-00000000026b">
    <name>example-network-filter-b</name>
    <version>
      <major>4</major>
      <minor>0</minor>
      <build>-1</build>
      <revision>-1</revision>
    </version>
  </network_filter>
  <network_filter id="00000019-0019-0019-0019-00000000026a">
    <name>example-network-filter-a</name>
    <version>
      <major>3</major>
      <minor>0</minor>
      <build>-1</build>
      <revision>-1</revision>
```

```

</version>
</network_filter>
</network_filters>

```

表 6.450. 方法概述

名称	概述
list	检索网络过滤器的表示法。

6.149.1. 列出 GET

检索网络过滤器的表示法。

网络过滤器返回列表的顺序不受保证。

表 6.451. 参数摘要

名称	类型	方向	概述
过滤器	NetworkFilter[]	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.149.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.150. NETWORKLABEL

表 6.452. 方法概述

名称	概述
get	
remove	从逻辑网络中删除标签。

6.150.1. 获取 GET

表 6.453. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
label	NetworkLabel	out	

6.150.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.150.2. 删除 DELETE

从逻辑网络中删除标签。

例如，要从 ID 为 123 的逻辑网络中删除标签 示例，发送以下请求：

```
DELETE /ovirt-engine/api/networks/123/networklabels/exemplary
```

表 6.454. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.151. NETWORKLABELS

管理附加到网络或主机 NIC 的标签 **ser**。

表 6.455. 方法概述

名称	概述
add	将标签附加到逻辑网络。
list	返回附加到网络或主机 NIC 的标签列表。

6.151.1. 添加 POST

将标签附加到逻辑网络。

您可以将标签附加到逻辑网络，从而将该逻辑网络与已附加相同标签的物理主机网络接口自动关联。

例如，要将标签 `mylabel` 附加到 ID 123 的逻辑网络上，可发送如下所示的请求：

```
POST /ovirt-engine/api/networks/123/networklabels
```

这样的请求正文：

```
<network_label id="mylabel"/>
```

表 6.456. 参数摘要

名称	类型	方向	概述
<code>label</code>	<code>NetworkLabel</code>	in/Out	

6.151.2. 列出 GET

返回附加到网络或主机 NIC 的标签列表。

返回的标签列表的顺序不受保证。

表 6.457. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>labels</code>	<code>NetworkLabel[]</code>	out	
<code>max</code>	整数	in	设置要返回的最大标签数。

6.151.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.151.2.2. Max

设置要返回的最大标签数。如果没有指定，则返回所有标签。

6.152. 网络

管理逻辑网络。

引擎会在安装时创建一个默认的 `ovirtmgmt` 网络。此网络充当访问虚拟机监控程序主机的管理网络。此网络与 `Default` 集群关联，是 `Default` 数据中心的成员。

表 6.458. 方法概述

名称	概述
<code>add</code>	创建新的逻辑网络，或者将现有网络与数据中心相关联。
<code>list</code>	列出逻辑网络。

6.152.1. 添加 POST

创建新的逻辑网络，或者将现有网络与数据中心相关联。

创建新网络需要 `name` 和 `data_center` 元素。

例如，要为数据中心创建名为 `mynetwork` 的网络 `123` 可发送如下请求：

```
POST /ovirt-engine/api/networks
```

这样的请求正文：

```
<network>
  <name>mynetwork</name>
  <data_center id="123"/>
</network>
```

要将现有网络 456 与数据中心 123 关联，可发送如下所示的请求：

```
POST /ovirt-engine/api/datacenters/123/networks
```

这样的请求正文：

```
<network>
  <name>ovirtmgmt</name>
</network>
```

要在外部 *OpenStack* 网络提供商 456 上创建名为 *exnetwork* 的网络，请发送如下请求：

```
POST /ovirt-engine/api/networks
```

```
<network>
  <name>exnetwork</name>
  <external_provider id="456"/>
  <data_center id="123"/>
</network>
```

表 6.459. 参数摘要

名称	类型	方向	概述
network	Network	in/Out	

6.152.2. 列出 GET

列出逻辑网络.

例如：

```
GET /ovirt-engine/api/networks
```

将响应：

```
<networks>
  <network href="/ovirt-engine/api/networks/123" id="123">
    <name>ovirtmgmt</name>
    <description>Default Management Network</description>
```

```

<link href="/ovirt-engine/api/networks/123/permissions" rel="permissions"/>
<link href="/ovirt-engine/api/networks/123/vnicprofiles" rel="vnicprofiles"/>
<link href="/ovirt-engine/api/networks/123/networklabels" rel="networklabels"/>
<mtu>0</mtu>
<stp>false</stp>
<usages>
  <usage>vm</usage>
</usages>
<data_center href="/ovirt-engine/api/datacenters/456" id="456"/>
</network>
...
</networks>

```

只有 `search` 参数中包含 `sortby` 子句时，才能保证返回的网络列表的顺序。

表 6.460. 参数摘要

名称	类型	方向	概述
<code>case_sensitive</code>	布尔值	in	指明是否应该在考虑大小写的情况下使用 <code>search</code> 参数执行搜索。
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>max</code>	整数	in	设置要返回的最大网络数。
网络	<code>network[]</code>	out	
<code>search</code>	字符串	in	用于限制返回的网络的查询字符串。

6.152.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 `search` 参数执行搜索。默认值为 `true`，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 `false`。

6.152.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.152.2.3. Max

设置要返回的最大网络数。如果没有指定，则会返回所有网络。

6.153. NICNETWORKFILTERPARAMETER

此服务管理网络过滤器的参数。

表 6.461. 方法概述

名称	概述
get	检索网络过滤器参数表示法。
remove	删除过滤器参数。
update	更新网络过滤器参数。

6.153.1. 获取 GET

检索网络过滤器参数表示法。

表 6.462. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
参数	NetworkFilterParameter	out	网络过滤器参数的表示。

6.153.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.153.2. 删除 DELETE

删除过滤器参数。

例如，要删除在虚拟机 789 的 NIC 456 上 ID 123 的 filter 参数，发送类似如下的请求：

```
DELETE /ovirt-engine/api/vms/789/nics/456/networkfilterparameters/123
```

6.153.3. 更新 PUT

更新网络过滤器参数。

例如，要更新在虚拟机 789 的 NIC 456 上带有 id 123 的网络过滤器参数发送如下请求：

```
PUT /ovirt-engine/api/vms/789/nics/456/networkfilterparameters/123
```

这样的请求正文：

```
<network_filter_parameter>
  <name>updatedName</name>
  <value>updatedValue</value>
</network_filter_parameter>
```

表 6.463. 参数摘要

名称	类型	方向	概述
参数	NetworkFilterParameter	in/Out	正在更新的网络过滤器参数。

6.154. NICNETWORKFILTERPARAMETERS

此服务管理网络过滤器的一组参数。

表 6.464. 方法概述

名称	概述
add	添加网络过滤器参数。
list	检索网络过滤器参数的表示法。

6.154.1. 添加 POST

添加网络过滤器参数。

例如，要在虚拟机 789 的 NIC 456 上为网络过滤器添加参数，发送类似如下的请求：

```
POST /ovirt-engine/api/vms/789/nics/456/networkfilterparameters
```

这样的请求正文：

```
<network_filter_parameter>
  <name>IP</name>
  <value>10.0.1.2</value>
</network_filter_parameter>
```

表 6.465. 参数摘要

名称	类型	方向	概述
参数	NetworkFilterParameter	in/Out	正在添加的网络过滤器参数。

6.154.2. 列出 GET

检索网络过滤器参数的表示法。

网络过滤器返回列表的顺序不受保证。

表 6.466. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
parameters	NetworkFilterParameter[]	out	网络过滤器参数列表。

6.154.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.155. OPENSTACKIMAGE

表 6.467. 方法概述

名称	概述
get	
import	从 Glance 映像存储域导入虚拟机。

6.155.1. 获取 GET

表 6.468. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
image	OpenStack image	out	

6.155.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.155.2. 导入 POST

从 Glance 映像存储域导入虚拟机。

例如，要从存储域导入标识符为 456 的镜像，标识符为 123，发送类似如下的请求：

```
POST /ovirt-engine/api/openstackimageproviders/123/images/456/import
```

这样的请求正文：

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

表 6.469. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行导入。
cluster	Cluster	in	如果使用 import_as_template ，则此参数是必需的，并指明哪个集群应用于导入 glance 镜像作为模板。

名称	类型	方向	概述
disk	Disk	in	
import_as_template	布尔值	in	指明是否应当将映像导入为模板。
storage_domain	StorageDomain	in	
模板	Template	in	

6.156. OPENSTACKIMAGEPROVIDER

表 6.470. 方法概述

名称	概述
get	
importcertificates	导入外部主机提供程序的 SSL 证书。
remove	
testconnectivity	为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。
update	更新系统中指定的 OpenStack 镜像提供程序。

6.156.1. 获取 GET

表 6.471. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
provider	OpenStackImageProvider	out	

6.156.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.156.2. importcertificates POST

导入外部主机提供程序的 SSL 证书。

表 6.472. 参数摘要

名称	类型	方向	概述
证书	certificate[]	in	

6.156.3. 删除 DELETE

表 6.473. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.156.4. 测试连接 POST

为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。

```
POST /ovirt-engine/api/externalhostproviders/123/testconnectivity
```

表 6.474. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行测试。

6.156.5. 更新 PUT

更新系统中指定的 OpenStack 镜像提供程序。

表 6.475. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
provider	OpenStack imageProvider	in/Out	

6.157. OPENSTACKIMAGEPROVIDERS

表 6.476. 方法概述

名称	概述
add	向系统添加新的 OpenStack 映像提供程序。
list	返回提供程序列表。

6.157.1. 添加 POST

向系统添加新的 OpenStack 映像提供程序。

表 6.477. 参数摘要

名称	类型	方向	概述
provider	OpenStackImageProvider	in/Out	

6.157.2. 列出 GET

返回提供程序列表。

返回的供应商列表的顺序不受保证。

表 6.478. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大供应商数量。
供应商	OpenStackImageProvider[]	out	
search	字符串	in	用于限制返回的 OpenStack 镜像提供程序的查询字符串。

6.157.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.157.2.2. Max

设置要返回的最大供应商数量。如果没有指定，则返回所有供应商。

6.158. OPENSTACKIMAGES

表 6.479. 方法概述

名称	概述
list	列出 Glance 映像存储域的映像。

6.158.1. 列出 GET

列出 Glance 映像存储域的映像。

无法保证返回的镜像列表的顺序。

表 6.480. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
images	OpenStackImage[]	out	
max	整数	in	设置要返回的最大镜像数。

6.158.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.158.1.2. Max

设置要返回的最大镜像数。如果没有指定，则返回所有镜像。

6.159. OPENSTACKNETWORK

表 6.481. 方法概述

名称	概述
get	
import	此操作将外部网络导入到红帽虚拟化中。

6.159.1. 获取 GET

表 6.482. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
network	OpenStackNetwork	out	

6.159.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.159.2. 导入 POST

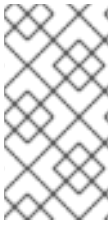
此操作将外部网络导入到红帽虚拟化中。网络将添加到指定的数据中心。

表 6.483. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行导入。
data_center	DataCenter	in	网络要导入到的数据中心。

6.159.2.1. data_center

网络要导入到的数据中心。数据中心是必需的，可使用 `id` 或 `name` 属性指定。将忽略其余属性。



注意

如果为供应商启用了 `auto_sync`，则网络可能会自动导入。要防止这种情况，可以通过将 `auto_sync` 设置为 `false` 来禁用自动导入，并在导入网络后再次启用它。

6.160. OPENSTACKNETWORKPROVIDER

此服务管理 OpenStack 网络提供程序。

表 6.484. 方法概述

名称	概述
<code>get</code>	返回此服务管理的对象表示法。
<code>importcertificates</code>	导入外部主机提供程序的 SSL 证书。
<code>remove</code>	删除供应商。
<code>testconnectivity</code>	为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。
<code>update</code>	更新提供程序。

6.160.1. 获取 GET

返回此服务管理的对象表示法。

例如，要获取标识符为 1234 的 OpenStack 网络提供程序，请发送如下请求：

```
GET /ovirt-engine/api/openstacknetworkproviders/1234
```

表 6.485. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。

名称	类型	方向	概述
provider	OpenStackNetworkProvider	out	

6.160.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.160.2. importcertificates POST

导入外部主机提供程序的 SSL 证书。

表 6.486. 参数摘要

名称	类型	方向	概述
证书	certificate[]	in	

6.160.3. 删除 DELETE

删除供应商。

例如，要删除带有标识符 1234 的 OpenStack 网络提供程序，请发送类似如下的请求：

```
DELETE /ovirt-engine/api/openstacknetworkproviders/1234
```

表 6.487. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.160.4. 测试连接 POST

为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。

POST /ovirt-engine/api/externalhostproviders/123/testconnectivity

表 6.488. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行测试。

6.160.5. 更新 PUT

更新提供程序。

例如，要更新 `provider_name`、`Requires_authentication`、`url`、`tenant_name` 和 `type` 属性，请向带有标识符 1234 的 OpenStack 网络供应商发送如下请求：

PUT /ovirt-engine/api/openstacknetworkproviders/1234

这样的请求正文：

```
<openstack_network_provider>
  <name>ovn-network-provider</name>
  <requires_authentication>false</requires_authentication>
  <url>http://some_server_url.domain.com:9696</url>
  <tenant_name>oVirt</tenant_name>
  <type>external</type>
</openstack_network_provider>
```

表 6.489. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
provider	OpenStackNetworkProvider	in/Out	要更新的供应商。

6.161. OPENSTACKNETWORKPROVIDERS

此服务管理 OpenStack 网络提供程序。

表 6.490. 方法概述

名称	概述
add	该操作向系统添加新的网络提供程序。
list	返回提供程序列表。

6.161.1. 添加 POST

该操作向系统添加新的网络提供程序。如果 `类型` 属性不存在，将使用默认值 `NEUTRON`。

表 6.491. 参数摘要

名称	类型	方向	概述
provider	OpenStackNetworkProvider	in/Out	

6.161.2. 列出 GET

返回提供程序列表。

返回的供应商列表的顺序不受保证。

表 6.492. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大供应商数量。
供应商	OpenStackNetworkProvider[]	out	
search	字符串	in	用于限制返回的 OpenStack 网络提供程序的查询字符串。

6.161.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.161.2.2. Max

设置要返回的最大供应商数量。如果没有指定，则返回所有供应商。

6.162. OPENSTACKNETWORKS

表 6.493. 方法概述

名称	概述
list	返回网络列表。

6.162.1. 列出 GET

返回网络列表。

网络返回列表的顺序无法保证。

表 6.494. 参数摘要

名称	类型	方向	概述
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的最大网络数。
网络	OpenStackNetwork[]	out	

6.162.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.162.1.2. Max

设置要返回的最大网络数。如果没有指定，则会返回所有网络。

6.163. OPENSTACKSUBNET

表 6.495. 方法概述

名称	概述
get	
remove	

6.163.1. 获取 GET

表 6.496. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
subnet	OpenStackSubnet	out	

6.163.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.163.2. 删除 DELETE

表 6.497. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.164. OPENSTACKSUBNETS

表 6.498. 方法概述

名称	概述
add	
list	返回子网列表。

6.164.1. 添加 POST

表 6.499. 参数摘要

名称	类型	方向	概述
subnet	OpenStackSubnet	in/Out	

6.164.2. 列出 GET

返回子网列表。

返回的子网列表的顺序不受保证。

表 6.500. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大子网数。
子网	OpenStackSubnet[]	out	

6.164.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.164.2.2. Max

设置要返回的最大子网数。如果没有指定，则返回所有子网。

6.165. OPENSTACKVOLUMEAUTHENTICATIONKEY

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 6.501. 方法概述

名称	概述
get	
remove	
update	更新指定的身份验证密钥。

6.165.1. 获取 GET

表 6.502. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
key	OpenstackVolumeAuthenticationKey	out	

6.165.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.165.2. 删除 DELETE

表 6.503. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.165.3. 更新 PUT

更新指定的身份验证密钥。

表 6.504. 参数摘要

名称	类型	方向	概述
key	OpenstackVolumeAuthenticationKey	in/Out	

6.166. OPENSTACKVOLUMEAUTHENTICATIONKEYS

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 6.505. 方法概述

名称	概述
add	添加新的身份验证密钥到 OpenStack 卷提供程序。
list	返回身份验证密钥列表。

6.166.1. 添加 POST

添加新的身份验证密钥到 OpenStack 卷提供程序。

表 6.506. 参数摘要

名称	类型	方向	概述
key	OpenstackVolumeAuthenticationKey	in/Out	

6.166.2. 列出 GET

返回身份验证密钥列表。

身份验证密钥返回列表的顺序不保证。

表 6.507. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
Keys	OpenstackVolumeAuthenticationKey[]	out	
max	整数	in	设置要返回的最大密钥数。

6.166.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.166.2.2. Max

设置要返回的最大密钥数。如果没有指定，则返回所有键。

6.167. OPENSTACKVOLUMEPROVIDER

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 6.508. 方法概述

名称	概述
get	
importcertificates	导入外部主机提供程序的 SSL 证书。
remove	
testconnectivity	为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。
update	更新系统中指定的 OpenStack 卷提供程序。

6.167.1. 获取 GET

表 6.509. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
provider	OpenStackVolumeProvider	out	

6.167.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.167.2. importcertificates POST

导入外部主机提供程序的 SSL 证书。

表 6.510. 参数摘要

名称	类型	方向	概述
证书	certificate[]	in	

6.167.3. 删除 DELETE

表 6.511. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。
force	布尔值	in	指明操作是否应该成功，并且供应商从数据库中删除，即使操作期间出现问题。

6.167.3.1. force

指明操作是否应该成功，并且供应商从数据库中删除，即使操作期间出现问题。

此参数是可选的，默认值为 **false**。

6.167.4. 测试连接 POST

为了测试外部提供程序的连接性，我们需要运行以下请求，其中 123 是提供程序的 ID。

```
POST /ovirt-engine/api/externalhostproviders/123/testconnectivity
```

表 6.512. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行测试。

6.167.5. 更新 PUT

更新系统中指定的 OpenStack 卷提供程序。

表 6.513. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
provider	OpenStackVolumeProvider	in/Out	

6.168. OPENSTACKVOLUMEPROVIDERS

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 6.514. 方法概述

名称	概述
add	添加新卷提供程序。
list	检索卷提供程序的列表。

6.168.1. 添加 POST

添加新卷提供程序。

例如：

```
POST /ovirt-engine/api/openstackvolumeproviders
```

这样的请求正文：

```
<openstack_volume_provider>
  <name>mycinder</name>
  <url>https://mycinder.example.com:8776</url>
  <data_center>
    <name>mydc</name>
  </data_center>
  <requires_authentication>true</requires_authentication>
```

```
<username>admin</username>
<password>mypassword</password>
<tenant_name>mytenant</tenant_name>
</openstack_volume_provider>
```

表 6.515. 参数摘要

名称	类型	方向	概述
provider	OpenStackVolumeProvider	in/Out	

6.168.2. 列出 GET

检索卷提供程序的列表。

无法保证返回的卷提供程序列表的顺序。

表 6.516. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大供应商数量。
供应商	OpenStackVolumeProvider[]	out	
search	字符串	in	用于限制返回的卷提供程序的查询字符串。

6.168.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.168.2.2. Max

设置要返回的最大供应商数量。如果没有指定，则返回所有供应商。

6.169. OPENSTACKVOLUMETYPE

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 6.517. 方法概述

名称	概述
get	

6.169.1. 获取 GET

表 6.518. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
type	OpenStackVolumeType	out	

6.169.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.170. OPENSTACKVOLUMETYPES

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 6.519. 方法概述

名称	概述
list	返回卷类型列表。

6.170.1. 列出 GET

返回卷类型列表。

无法保证返回的卷类型列表的顺序。

表 6.520. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大卷类型数。
类型	OpenStackVolumeType[]	out	

6.170.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.170.1.2. Max

设置要返回的最大卷类型数。如果没有指定，则返回所有卷类型。

6.171. OPERATINGSYSTEM

表 6.521. 方法概述

名称	概述
get	

6.171.1. 获取 GET

表 6.522. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
operating_system	OperatingSystemInfo	out	

6.171.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.172. OPERATINGSYSTEMS

管理系统中可用的操作系统集合。

表 6.523. 方法概述

名称	概述
list	返回系统中可用操作系统类型的列表。

6.172.1. 列出 GET

返回系统中可用操作系统类型的列表。

无法保证返回的操作系统列表的顺序。

表 6.524. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大网络数。
operating_system	OperatingSystemInfo[]	out	

6.172.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.172.1.2. Max

设置要返回的最大网络数。如果没有指定，则会返回所有网络。

6.173. 权限

表 6.525. 方法概述

名称	概述
get	

名称	概述
remove	

6.173.1. 获取 GET

表 6.526. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
权限	权限	out	

6.173.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.173.2. 删除 DELETE

表 6.527. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.174. ALLOW

用于管理该角色特定权限的服务。

表 6.528. 方法概述

名称	概述
get	获取有关角色允许的信息。
remove	从角色中删除允许。

6.174.1. 获取 GET

获取有关角色允许的信息。

例如，要检索有关角色 ID 为 456 的允许信息，id 123 会发送如下请求：

```
GET /ovirt-engine/api/roles/123/permits/456
```

```
<permit href="/ovirt-engine/api/roles/123/permits/456" id="456">
  <name>change_vm_cd</name>
  <administrative>false</administrative>
  <role href="/ovirt-engine/api/roles/123" id="123"/>
</permit>
```

表 6.529. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
permit	Allow	out	角色的允许。

6.174.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.174.2. 删除 DELETE

从角色中删除允许。

例如，要从 id 123 角色中删除 id 456 的允许，可发送类似如下的请求：

```
DELETE /ovirt-engine/api/roles/123/permits/456
```

表 6.530. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.175. ALLOW

表示允许特定角色的子集合。

表 6.531. 方法概述

名称	概述
add	向角色添加允许。
list	列出角色允许。

6.175.1. 添加 POST

向角色添加允许。允许名称可以从 [第 6.45 节 “ClusterLevels” 服务检索](#)。

例如，要将允许 `create_vm` 分配给具有 `id 123` 的角色，可发送如下所示的请求：

POST /ovirt-engine/api/roles/123/permits

这样的请求正文：

```
<permit>
  <name>create_vm</name>
</permit>
```

表 6.532. 参数摘要

名称	类型	方向	概述
permit	<code>Allow</code>	in/Out	要添加的允许。

6.175.2. 列出 GET

列出角色允许。

例如，使用 `id 123` 列出角色允许，发送类似如下的请求：

GET /ovirt-engine/api/roles/123/permits

```
<permits>
  <permit href="/ovirt-engine/api/roles/123/permits/5" id="5">
    <name>change_vm_cd</name>
    <administrative>false</administrative>
  </role href="/ovirt-engine/api/roles/123" id="123"/>
```



```

</permit>
<permit href="/ovirt-engine/api/roles/123/permits/7" id="7">
  <name>connect_to_vm</name>
  <administrative>false</administrative>
  <role href="/ovirt-engine/api/roles/123" id="123"/>
</permit>
</permits>

```

无法保证返回的许可列表的顺序。

表 6.533. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置返回的最大允许数。
Allow	Permit[]	out	允许列表。

6.175.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.175.2.2. Max

设置返回的最大允许数。如果没有指定，则返回所有允许。

6.176. QOS

表 6.534. 方法概述

名称	概述
get	在数据中心的获取指定的 QoS。
remove	从数据中心移除指定的 QoS。
update	更新 dataCenter 中指定的 QoS。

6.176.1. 获取 GET

在数据中心的获取指定的 QoS。

```
GET /ovirt-engine/api/datacenters/123/qoss/123
```

您将获得如下响应：

```
<qos href="/ovirt-engine/api/datacenters/123/qoss/123" id="123">
  <name>123</name>
  <description>123</description>
  <max_iops>1</max_iops>
  <max_throughput>1</max_throughput>
  <type>storage</type>
  <data_center href="/ovirt-engine/api/datacenters/123" id="123"/>
</qos>
```

表 6.535. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
qos	QoS	out	查询的 QoS 对象。

6.176.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.176.2. 删除 DELETE

从数据中心移除指定的 QoS。

```
DELETE /ovirt-engine/api/datacenters/123/qoss/123
```

表 6.536. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.176.3. 更新 PUT

更新 dataCenter 中指定的 QoS。

```
PUT /ovirt-engine/api/datacenters/123/qoss/123
```

例如，使用 curl：

```
curl -u admin@internal:123456 -X PUT -H "content-type: application/xml" -d \
"<qos><name>321</name><description>321</description><max_iops>10</max_iops></qos>" \
https://engine/ovirt-engine/api/datacenters/123/qoss/123
```

您将收到如下响应：

```
<qos href="/ovirt-engine/api/datacenters/123/qoss/123" id="123">
  <name>321</name>
  <description>321</description>
  <max_iops>10</max_iops>
  <max_throughput>1</max_throughput>
  <type>storage</type>
  <data_center href="/ovirt-engine/api/datacenters/123" id="123"/>
</qos>
```

表 6.537. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
qos	QoS	in/Out	更新了 QoS 对象。

6.177. QOSS

管理数据中心的可用的一组服务质量配置。

表 6.538. 方法概述

名称	概述
add	向 dataCenter 添加新 QoS。
list	返回数据中心的可用服务质量配置列表。

6.177.1. 添加 POST

向 dataCenter 添加新 QoS。

POST /ovirt-engine/api/datacenters/123/qoss

回复方式如下：

```
<qos href="/ovirt-engine/api/datacenters/123/qoss/123" id="123">
  <name>123</name>
  <description>123</description>
  <max_iops>10</max_iops>
  <type>storage</type>
  <data_center href="/ovirt-engine/api/datacenters/123" id="123"/>
</qos>
```

表 6.539. 参数摘要

名称	类型	方向	概述
qos	QoS	in/Out	添加了 QoS 对象。

6.177.2. 列出 GET

返回数据中心中可用的服务质量配置列表。

GET /ovirt-engine/api/datacenter/123/qoss

您将获得类似如下的响应：

```
<qoss>
  <qos href="/ovirt-engine/api/datacenters/123/qoss/1" id="1">...</qos>
  <qos href="/ovirt-engine/api/datacenters/123/qoss/2" id="2">...</qos>
  <qos href="/ovirt-engine/api/datacenters/123/qoss/3" id="3">...</qos>
</qoss>
```

返回的服务质量配置列表无法保证。

表 6.540. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。

名称	类型	方向	概述
max	整数	in	设置要返回的最大 QoS 描述符数。
qoss	Qos[]	out	查询的 QoS 对象列表。

6.177.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.177.2.2. Max

设置要返回的最大 QoS 描述符数。如果没有指定，则会返回所有描述符。

6.178. QUOTA

表 6.541. 方法概述

名称	概述
get	检索配额.
remove	删除配额.
update	更新配额.

6.178.1. 获取 GET

检索配额.

获取配额的示例：

```
GET /ovirt-engine/api/datacenters/123/quotas/456
```

```
<quota id="456">
  <name>myquota</name>
  <description>My new quota for virtual machines</description>
  <cluster_hard_limit_pct>20</cluster_hard_limit_pct>
  <cluster_soft_limit_pct>80</cluster_soft_limit_pct>
```

```
<storage_hard_limit_pct>20</storage_hard_limit_pct>
<storage_soft_limit_pct>80</storage_soft_limit_pct>
</quota>
```

表 6.542. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
quota	Quota	out	

6.178.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.178.2. 删除 DELETE

删除配额。

删除配额的示例：

```
DELETE /ovirt-engine/api/datacenters/123-456/quotas/654-321
-0472718ab224 HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

表 6.543. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.178.3. 更新 PUT

更新配额。

更新配额的示例：

```
PUT /ovirt-engine/api/datacenters/123/quotas/456
```

```

<quota>
  <cluster_hard_limit_pct>30</cluster_hard_limit_pct>
  <cluster_soft_limit_pct>70</cluster_soft_limit_pct>
  <storage_hard_limit_pct>20</storage_hard_limit_pct>
  <storage_soft_limit_pct>80</storage_soft_limit_pct>
</quota>

```

表 6.544. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
quota	Quota	in/Out	

6.179. QUOTACLUSTERLIMIT

表 6.545. 方法概述

名称	概述
get	
remove	

6.179.1. 获取 GET

表 6.546. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
limit	QuotaClusterLimit	out	

6.179.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.179.2. 删除 DELETE

表 6.547. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.180. QUOTA_CLUSTER_LIMITS

管理为集群配置的配额限值集合。

表 6.548. 方法概述

名称	概述
add	将集群限制添加到指定的配额。
list	返回为集群配置的配额限值集合。

6.180.1. 添加 POST

将集群限制添加到指定的配额。

表 6.549. 参数摘要

名称	类型	方向	概述
limit	QuotaClusterLimit	in/Out	

6.180.2. 列出 GET

返回为集群配置的配额限值集合。

返回的配额限制列表不受保证。

表 6.550. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
limits	QuotaClusterLimit[]	out	

名称	类型	方向	概述
max	整数	in	设置返回的最大限值数。

6.180.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.180.2.2. Max

设置返回的最大限值数。如果没有指定，则会返回所有限制。

6.181. QUOTASTORAGELIMIT

表 6.551. 方法概述

名称	概述
get	
remove	

6.181.1. 获取 GET

表 6.552. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
limit	QuotaStorageLimit	out	

6.181.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.181.2. 删除 DELETE

表 6.553. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。

6.182. QUOTASTORAGELIMITS

管理为配额配置的存储限值集合。

表 6.554. 方法概述

名称	概述
add	为指定的配额添加存储限制。
list	返回为配额配置的存储限值列表。

6.182.1. 添加 POST

为指定的配额添加存储限制。

要为数据中心中的所有存储域创建 100GiB 存储限值，请发送如下请求：

```
POST /ovirt-engine/api/datacenters/123/quotas/456/quotastoragelimits
```

这样的请求正文：

```
<quota_storage_limit>
  <limit>100</limit>
</quota_storage_limit>
```

要为 ID 为 000 的存储域创建 50GiB 存储限值，请发送如下请求：

```
POST /ovirt-engine/api/datacenters/123/quotas/456/quotastoragelimits
```

这样的请求正文：

```
<quota_storage_limit>
```

```
<limit>50</limit>
<storage_domain id="000"/>
</quota_storage_limit>
```

表 6.555. 参数摘要

名称	类型	方向	概述
limit	QuotaStorageLimit	in/Out	

6.182.2. 列出 GET

返回为配额配置的存储限值列表。

无法保证返回的存储限值列表的顺序。

表 6.556. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
limits	QuotaStorageLimit[]	out	
max	整数	in	设置返回的最大限值数。

6.182.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.182.2.2. Max

设置返回的最大限值数。如果没有指定，则会返回所有限制。

6.183. 配额

管理为数据中心配置的一组配额。

表 6.557. 方法概述

名称	概述
add	创建新配额.
list	列出数据中心的配额.

6.183.1. 添加 POST

创建新配额.

创建新配额的示例：

```
POST /ovirt-engine/api/datacenters/123/quotas
```

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

表 6.558. 参数摘要

名称	类型	方向	概述
quota	Quota	in/Out	

6.183.2. 列出 GET

列出数据中心的配额.

无法保证返回的配额列表的顺序。

表 6.559. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大配额描述符数。
quotas	Quota[]	out	

6.183.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.183.2.2. Max

设置要返回的最大配额描述符数。如果没有指定，则会返回所有描述符。

6.184. 角色

表 6.560. 方法概述

名称	概述
get	获取角色.
remove	删除角色。
update	更新角色.

6.184.1. 获取 GET

获取角色.

GET /ovirt-engine/api/roles/123

您将收到类似以下的 XML 响应：

```
<role id="123">
  <name>MyRole</name>
  <description>MyRole description</description>
  <link href="/ovirt-engine/api/roles/123/permits" rel="permits"/>
  <administrative>true</administrative>
  <mutable>false</mutable>
</role>
```

表 6.561. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。

名称	类型	方向	概述
role	角色	out	检索的角色。

6.184.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.184.2. 删除 DELETE

删除角色。

要删除需要知道其 ID 的角色，然后发送类似如下的请求：

```
DELETE /ovirt-engine/api/roles/{role_id}
```

表 6.562. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.184.3. 更新 PUT

更新角色.您可以在角色创建后更新 名称、描述 和管理属性。在此端点中，您不能添加或删除角色，因此您需要使用管理角色允许的服务。

例如，要更新角色的名称，描述 和管理 属性 会发送类似如下的请求：

```
PUT /ovirt-engine/api/roles/123
```

这样的请求正文：

```
<role>
  <name>MyNewRoleName</name>
  <description>My new description of the role</description>
  <administrative>true</administrative>
</group>
```

表 6.563. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
role	角色	in/Out	更新的角色。

6.185. 角色

提供对全局角色集的只读访问权限

表 6.564. 方法概述

名称	概述
add	创建新角色。
list	列出角色。

6.185.1. 添加 POST

创建新角色。该角色可以是管理或非管理员角色，可以具有不同的许可。

例如，要将 **MyRole** 非管理员角色添加到 允许登录并创建虚拟机发送类似此请求（请注意，您必须传递允许 ID）：

```
POST /ovirt-engine/api/roles
```

这样的请求正文：

```
<role>
  <name>MyRole</name>
  <description>My custom role to create virtual machines</description>
  <administrative>false</administrative>
  <permits>
    <permit id="1"/>
    <permit id="1300"/>
  </permits>
</group>
```

表 6.565. 参数摘要

名称	类型	方向	概述
role	角色	in/Out	将添加的角色.

6.185.2. 列出 GET

列出角色.

GET /ovirt-engine/api/roles

您将在 XML 中收到类似以下的响应：

```
<roles>
  <role id="123">
    <name>SuperUser</name>
    <description>Roles management administrator</description>
    <link href="/ovirt-engine/api/roles/123/permits" rel="permits"/>
    <administrative>true</administrative>
    <mutable>false</mutable>
  </role>
  ...
</roles>
```

无法保证返回的角色列表的顺序。

表 6.566. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的角色的最大数量。
角色	role[]	out	检索的角色列表。

6.185.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.185.2.2. Max

设置要返回的角色的最大数量。如果未指定，则返回所有角色。

6.186. SCHEDULINGPOLICIES

管理系统中可用的一组调度策略。

表 6.567. 方法概述

名称	概述
add	向系统添加新调度策略。
list	返回系统中可用的调度策略列表。

6.186.1. 添加 POST

向系统添加新调度策略。

表 6.568. 参数摘要

名称	类型	方向	概述
policy	SchedulingPolicy	in/Out	

6.186.2. 列出 GET

返回系统中可用的调度策略列表。

无法保证返回的调度策略列表的顺序。

表 6.569. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的策略的最大数量。

名称	类型	方向	概述
policies	SchedulingPolicy[]	out	

6.186.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.186.2.2. Max

设置要返回的策略的最大数量。如果没有指定，则返回所有策略。

6.187. SCHEDULINGPOLICY

表 6.570. 方法概述

名称	概述
get	
remove	
update	更新系统中指定的用户定义的调度策略。

6.187.1. 获取 GET

表 6.571. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
policy	SchedulingPolicy	out	

6.187.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.187.2. 删除 DELETE

表 6.572. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.187.3. 更新 PUT

更新系统中指定的用户定义的调度策略。

表 6.573. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
policy	SchedulingPolicy	in/Out	

6.188. SCHEDULINGPOLICYUNIT

表 6.574. 方法概述

名称	概述
get	
remove	

6.188.1. 获取 GET

表 6.575. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
unit	SchedulingPolicyUnit	out	

6.188.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.188.2. 删除 DELETE

表 6.576. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.189. SCHEDULINGPOLICYUNITS

管理系统中可用的一组调度策略单元。

表 6.577. 方法概述

名称	概述
list	返回系统中可用的调度策略单元列表。

6.189.1. 列出 GET

返回系统中可用的调度策略单元列表。

无法保证返回的调度策略单元列表的顺序。

表 6.578. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大策略单元数。
unit	SchedulingPolicyUnit[]	out	

6.189.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.189.1.2. Max

设置要返回的最大策略单元数。如果没有指定，则返回所有策略单元。

6.190. SNAPSHOT

表 6.579. 方法概述

名称	概述
get	
remove	
restore	恢复虚拟机快照.

6.190.1. 获取 GET

表 6.580. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
snapshot	snapshot	out	

6.190.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.190.2. 删除 DELETE

表 6.581. 参数摘要

名称	类型	方向	概述
all_content	布尔值	in	指明是否应将虚拟机快照的所有属性包含在响应中。
async	布尔值	in	指明是否应异步执行移除。

6.190.2.1. all_content

指明是否应将虚拟机快照的所有属性包含在响应中。

默认情况下，排除属性 `initialization.configuration.data`。

例如，要检索 ID 为 456 且 ID 为 123 的虚拟机的快照的完整表示形式，发送如下请求：

```
GET /ovirt-engine/api/vms/123/snapshots/456?all_content=true
```

6.190.3. 恢复 POST

恢复虚拟机快照。

例如，要恢复带有标识符 456 的虚拟机的快照，并使用标识符 123 发送类似如下的请求：

```
POST /ovirt-engine/api/vms/123/snapshots/456/restore
```

在正文中有一个空操作：

```
<action/>
```



注意

确认提交操作已完成，并且虚拟机已停机，然后运行虚拟机。

表 6.582. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行恢复。
磁盘	<code>disk[]</code>	in	指定快照恢复中包含的磁盘。
<code>restore_memory</code>	布尔值	in	

6.190.3.1. 磁盘

指定快照恢复中包含的磁盘。

对于每个磁盘参数，还需要指定其 `image_id`。

例如，要恢复具有标识符 123（包括标识符为 111）和 `image_id` 为 222 的虚拟机的标识符 456 的快照，请发送如下请求：

```
POST /ovirt-engine/api/vms/123/snapshots/456/restore
```

请求正文：

```
<action>
<disks>
  <disk id="111">
    <image_id>222</image_id>
  </disk>
</disks>
</action>
```

6.191. SNAPSHOTCDROM

表 6.583. 方法概述

名称	概述
<code>get</code>	

6.191.1. 获取 GET

表 6.584. 参数摘要

名称	类型	方向	概述
<code>cdrom</code>	<code>cdrom</code>	out	
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。

6.191.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.192. SNAPSHOTCDROMS

管理虚拟机快照的 CD-ROM 设备集合。

表 6.585. 方法概述

名称	概述
<code>list</code>	返回快照的 CD-ROM 设备列表。

6.192.1. 列出 GET

返回快照的 CD-ROM 设备列表。

CD-ROM 设备返回列表的顺序不受保证。

表 6.586. 参数摘要

名称	类型	方向	概述
<code>cdroms</code>	<code>Cdrom[]</code>	out	
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>max</code>	整数	in	设置要返回的最大 CDROMS 数。

6.192.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.192.1.2. Max

设置要返回的最大 CDROMS 数。如果没有指定所有 CDROMS，则会返回。

6.193. SNAPSHOTDISK

表 6.587. 方法概述

名称	概述
get	

6.193.1. 获取 GET

表 6.588. 参数摘要

名称	类型	方向	概述
disk	Disk	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.193.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.194. SNAPSHOTDISKS

管理快照的磁盘集合。

表 6.589. 方法概述

名称	概述
list	返回快照的磁盘列表。

6.194.1. 列出 GET

返回快照的磁盘列表。

无法保证返回的磁盘列表的顺序。

表 6.590. 参数摘要

名称	类型	方向	概述
磁盘	disk[]	out	

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大磁盘数。

6.194.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.194.1.2. Max

设置要返回的最大磁盘数。如果没有指定，则返回所有磁盘。

6.195. SNAPSHOTNIC

表 6.591. 方法概述

名称	概述
get	

6.195.1. 获取 GET

表 6.592. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
nic	NIC	out	

6.195.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.196. SNAPSHOTNICS

管理快照的一组 NIC。

表 6.593. 方法概述

名称	概述
<code>list</code>	返回快照的 NIC 列表。

6.196.1. 列出 GET

返回快照的 NIC 列表。

NIC 返回列表的顺序不受保证。

表 6.594. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>max</code>	整数	in	设置要返回的最大 NIC 数。
<code>nics</code>	Nic[]	out	

6.196.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.196.1.2. Max

设置要返回的最大 NIC 数。如果没有指定，则会返回所有 NIC。

6.197. 快照

管理存储域或虚拟机的一组快照。

表 6.595. 方法概述

名称	概述
<code>add</code>	创建虚拟机快照。

名称	概述
list	返回存储域或虚拟机的快照列表。

6.197.1. 添加 POST

创建虚拟机快照。

例如，要为虚拟机创建新快照 123 发送如下请求：

```
POST /ovirt-engine/api/vms/123/snapshots
```

这样的请求正文：

```
<snapshot>
  <description>My snapshot</description>
</snapshot>
```

对于仅将一组磁盘包含在快照中，请将 `disk_attachments` 元素添加到请求正文。请注意，在 `disk_attachments` 元素中未指定的磁盘将不属于快照的一部分。如果传递了空的 `disk_attachments` 元素，快照将仅包含虚拟机配置。如果没有传递 `disk_attachments` 元素，则所有磁盘都将包含在快照中。

对于每个磁盘，可以指定 `image_id` 元素来设置新的活动镜像 ID。这是为了从备份中恢复镜像链。i. 在恢复具有快照的磁盘时，应当为每个快照指定相关的 `image_id`（因此磁盘快照的标识符与备份相同）。

```
<snapshot>
  <description>My snapshot</description>
  <disk_attachments>
    <disk_attachment>
      <disk id="123">
        <image_id>456</image_id>
      </disk>
    </disk_attachment>
  </disk_attachments>
</snapshot>
```

重要

创建快照时，`persist_memorystate` 属性的默认值为 `true`。这意味着虚拟机的内存内容将包含在快照中，这也意味着虚拟机将暂停更长时间。这可能会对对计时非常敏感的应用程序（例如，NTP 服务器）造成负面影响。在这些情况下，请确保将属性设置为 `false`：

```
<snapshot>
  <description>My snapshot</description>
  <persist_memorystate>false</persist_memorystate>
</snapshot>
```

表 6.596. 参数摘要

名称	类型	方向	概述
<code>snapshot</code>	<code>snapshot</code>	in/Out	

6.197.2. 列出 GET

返回存储域或虚拟机的快照列表。

快照返回列表的顺序无法保证。

表 6.597. 参数摘要

名称	类型	方向	概述
<code>all_content</code>	布尔值	in	指明是否应将虚拟机快照的所有属性包含在响应中。
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>max</code>	整数	in	设置要返回的快照的最大数量。
<code>snapshots</code>	<code>Snapshot[]</code>	out	

6.197.2.1. all_content

指明是否应将虚拟机快照的所有属性包含在响应中。

默认情况下，排除属性 `initialization.configuration.data`。

例如，要检索 ID 123 快照的虚拟机的完整表示法，发送如下请求：

```
GET /ovirt-engine/api/vms/123/snapshots?all_content=true
```

6.197.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.197.2.3. Max

设置要返回的快照的最大数量。如果没有指定，则返回所有快照。

6.198. SSHPUBLICKEY

表 6.598. 方法概述

名称	概述
get	
remove	
update	使用新资源替换密钥。

6.198.1. 获取 GET

表 6.599. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
key	SshPublicKey	out	

6.198.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.198.2. 删除 DELETE

表 6.600. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.198.3. 更新 PUT

使用新资源替换密钥。

**重要**

因为引擎的 4.4.8 版本已弃用，并只保留向后兼容。以后会将其移除。请使用 **DELETE** 后跟 [添加操作](#)。

表 6.601. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
key	SshPublicKey	in/Out	

6.199. SSHPUBLICKEYS

表 6.602. 方法概述

名称	概述
add	
list	返回用户的 SSH 公钥列表。

6.199.1. 添加 POST

表 6.603. 参数摘要

名称	类型	方向	概述
key	SshPublicKey	in/Out	

6.199.2. 列出 GET

返回用户的 SSH 公钥列表。

例如，要检索标识符为 123 的 SSH 密钥列表，发送类似如下的请求：

```
GET /ovirt-engine/api/users/123/sshpublickeys
```

其结果将是以下 XML 文档：

```
<ssh_public_keys>
  <ssh_public_key href="/ovirt-engine/api/users/123/sshpublickeys/456" id="456">
    <content>ssh-rsa ...</content>
    <user href="/ovirt-engine/api/users/123" id="123"/>
  </ssh_public_key>
</ssh_public_keys>
```

或以下 JSON 对象

```
{
  "ssh_public_key": [
    {
      "content": "ssh-rsa ...",
      "user": {
        "href": "/ovirt-engine/api/users/123",
        "id": "123"
      },
      "href": "/ovirt-engine/api/users/123/sshpublickeys/456",
      "id": "456"
    }
  ]
}
```

无法保证返回的密钥列表的顺序。

表 6.604. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
Keys	SshPublicKey[]	out	

名称	类型	方向	概述
max	整数	in	设置要返回的最大密钥数。

6.199.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.199.2.2. Max

设置要返回的最大密钥数。如果没有指定，则返回所有键。

6.200. 统计

表 6.605. 方法概述

名称	概述
get	

6.200.1. 获取 GET

表 6.606. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
statistic	统计	out	

6.200.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.201. STATISTICS

表 6.607. 方法概述

名称	概述
list	检索统计数据列表。

6.201.1. 列出 GET

检索统计数据列表。

例如，要检索虚拟机 123 的统计信息，发送如下请求：

```
GET /ovirt-engine/api/vms/123/statistics
```

结果如下：

```
<statistics>
  <statistic href="/ovirt-engine/api/vms/123/statistics/456" id="456">
    <name>memory.installed</name>
    <description>Total memory configured</description>
    <kind>gauge</kind>
    <type>integer</type>
    <unit>bytes</unit>
    <values>
      <value>
        <datum>1073741824</datum>
      </value>
    </values>
    <vm href="/ovirt-engine/api/vms/123" id="123"/>
  </statistic>
  ...
</statistics>
```

可以通过在 URI 末尾指定其 ID 来检索统计的单个部分。这意味着：

```
GET /ovirt-engine/api/vms/123/statistics/456
```

outputs:

```
<statistic href="/ovirt-engine/api/vms/123/statistics/456" id="456">
  <name>memory.installed</name>
  <description>Total memory configured</description>
  <kind>gauge</kind>
  <type>integer</type>
  <unit>bytes</unit>
  <values>
    <value>
      <datum>1073741824</datum>
    </value>
```

```

</values>
<vm href="/ovirt-engine/api/vms/123" id="123"/>
</statistic>

```

返回的统计数据列表的顺序无法保证。

表 6.608. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大统计数据数。
statistics	Statistic[]	out	

6.201.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.201.1.2. Max

设置要返回的最大统计数据数。如果没有指定，则返回所有统计信息。

6.202. 步骤

用于管理步骤的服务。

表 6.609. 方法概述

名称	概述
end	将外部步骤执行标记为结束。
get	检索步骤。

6.202.1. 最终 POST

将外部步骤执行标记为结束。

例如，要终止标识符 456 的步骤，该步骤属于标识符为 123 的作业，发送以下请求：

```
POST /ovirt-engine/api/jobs/123/steps/456/end
```

使用以下请求正文：

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

表 6.610. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
force	布尔值	in	指明是否应该强制终止该步骤。
succeeded	布尔值	in	表示该步骤应标记为成功完成还是失败。

6.202.1.1. succeeded

表示该步骤应标记为成功完成还是失败。

此参数是可选的，默认值为 `true`。

6.202.2. 获取 GET

检索步骤。

```
GET /ovirt-engine/api/jobs/123/steps/456
```

您将在 XML 中收到类似以下的响应：

```
<step href="/ovirt-engine/api/jobs/123/steps/456" id="456">
  <actions>
    <link href="/ovirt-engine/api/jobs/123/steps/456/end" rel="end"/>
  </actions>
  <description>Validating</description>
```

```

<end_time>2016-12-12T23:07:26.627+02:00</end_time>
<external>false</external>
<number>0</number>
<start_time>2016-12-12T23:07:26.605+02:00</start_time>
<status>finished</status>
<type>validating</type>
<job href="/ovirt-engine/api/jobs/123" id="123"/>
</step>

```

表 6.611. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
步骤	步骤	out	检索步骤表示法。

6.202.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.203. 步骤

用于管理步骤的服务。

表 6.612. 方法概述

名称	概述
add	向现有作业或现有步骤中添加外部步骤。
list	检索步骤的表示法。

6.203.1. 添加 POST

向现有作业或现有步骤中添加外部步骤。

例如，要为带有标识符 123 的作业 添加步骤，发送以下请求：

```
POST /ovirt-engine/api/jobs/123/steps
```

使用以下请求正文：

```
<step>
  <description>Validating</description>
  <start_time>2016-12-12T23:07:26.605+02:00</start_time>
  <status>started</status>
  <type>validating</type>
</step>
```

响应应类似于：

```
<step href="/ovirt-engine/api/jobs/123/steps/456" id="456">
  <actions>
    <link href="/ovirt-engine/api/jobs/123/steps/456/end" rel="end"/>
  </actions>
  <description>Validating</description>
  <link href="/ovirt-engine/api/jobs/123/steps/456/statistics" rel="statistics"/>
  <external>true</external>
  <number>2</number>
  <start_time>2016-12-13T01:06:15.380+02:00</start_time>
  <status>started</status>
  <type>validating</type>
  <job href="/ovirt-engine/api/jobs/123" id="123"/>
</step>
```

表 6.613. 参数摘要

名称	类型	方向	概述
步骤	步骤	in/Out	将要添加的步骤。

6.203.2. 列出 GET

检索步骤的表示法。

```
GET /ovirt-engine/api/job/123/steps
```

您将在 XML 中收到类似以下的响应：

```
<steps>
  <step href="/ovirt-engine/api/jobs/123/steps/456" id="456">
    <actions>
      <link href="/ovirt-engine/api/jobs/123/steps/456/end" rel="end"/>
    </actions>
    <description>Validating</description>
```

```

<link href="/ovirt-engine/api/jobs/123/steps/456/statistics" rel="statistics"/>
<external>true</external>
<number>2</number>
<start_time>2016-12-13T01:06:15.380+02:00</start_time>
<status>started</status>
<type>validating</type>
<job href="/ovirt-engine/api/jobs/123" id="123"/>
</step>
...
</steps>

```

返回的步骤列表的顺序无法保证。

表 6.614. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置返回的最大步骤数。
步骤	第[]步	out	步骤的表示。

6.203.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.203.2.2. Max

设置返回的最大步骤数。如果没有指定，则返回所有步骤。

6.204. 存储

表 6.615. 方法概述

名称	概述
get	

6.204.1. 获取 GET

表 6.616. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
report_status	布尔值	in	指明是否应检查存储中的 LUN 状态。
storage	HostStorage	out	

6.204.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.204.1.2. report_status

指明是否应检查存储中的 LUN 状态。检查 LUN 状态是一种重量型操作，用户并不总是需要这个数据。此参数将提供选项，使其不执行 LUN 的状态检查。

默认为 向后 兼容。

这里有一个 LUN 状态为 的示例：

```
<host_storage id="360014051136c20574f743bdbd28177fd">
  <logical_units>
    <logical_unit id="360014051136c20574f743bdbd28177fd">
      <lun_mapping>0</lun_mapping>
      <paths>1</paths>
      <product_id>lun0</product_id>
      <serial>SLIO-ORG_lun0_1136c205-74f7-43bd-bd28-177fd5ce6993</serial>
      <size>10737418240</size>
      <status>used</status>
      <vendor_id>LIO-ORG</vendor_id>
      <volume_group_id>O9Du7I-RahN-ECe1-dZ1w-nh0b-64io-MNzIBZ</volume_group_id>
    </logical_unit>
  </logical_units>
  <type>iscsi</type>
  <host id="8bb5ade5-e988-4000-8b93-dbfc6717fe50"/>
</host_storage>
```

这里有一个没有 LUN 状态的示例：

```
<host_storage id="360014051136c20574f743bdbd28177fd">
```



```

<logical_units>
  <logical_unit id="360014051136c20574f743bdbd28177fd">
    <lun_mapping>0</lun_mapping>
    <paths>1</paths>
    <product_id>lun0</product_id>
    <serial>SLIO-ORG_lun0_1136c205-74f7-43bd-bd28-177fd5ce6993</serial>
    <size>10737418240</size>
    <vendor_id>LIO-ORG</vendor_id>
    <volume_group_id>O9Du7I-RahN-ECe1-dZ1w-nh0b-64io-MNzIBZ</volume_group_id>
  </logical_unit>
</logical_units>
<type>iscsi</type>
<host id="8bb5ade5-e988-4000-8b93-dbfc6717fe50"/>
</host_storage>

```

6.205. STORAGEDOMAIN

表 6.617. 方法概述

名称	概述
get	检索存储域的描述。
isattached	用于查询存储域是否已通过 is_attached 布尔值字段连接到数据中心，该字段是存储服务器的一部分。
reduceluns	此操作可减少存储域中的逻辑单元。
refreshluns	此操作会刷新 LUN 大小。
remove	删除存储域。
update	更新存储域。
updateovfstore	此操作会强制更新此存储域的 OVF_STORE 。

6.205.1. 获取 GET

检索存储域的描述。

表 6.618. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。

名称	类型	方向	概述
storage_domain	StorageDomain	out	存储域的描述。

6.205.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.205.2. 附加 POST

用于查询存储域是否已通过 `is_attached` 布尔值字段连接到数据中心，该字段是存储服务器的一部分。重要信息：执行此 API 将会导致主机断开与存储域的连接。

表 6.619. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。
host	主机	in	表示数据中心的主机。
is_attached	布尔值	out	指明存储域是否已附加到数据中心。

6.205.3. 减少luns POST

此操作可减少存储域中的逻辑单元。

为此，存储在提供的逻辑单元上的数据将移到存储域的其他逻辑单元中，然后才会从存储域中减少。

例如，要从存储域减少两个逻辑单元，发送如下请求：

```
POST /ovirt-engine/api/storageDomains/123/reduceluns
```

这样的请求正文：

```
<action>
```

```

<logical_units>
  <logical_unit id="1IET_00010001"/>
  <logical_unit id="1IET_00010002"/>
</logical_units>
</action>

```

Note that this operation is only applicable to block storage domains (i.e., storage domains with the <<types/storage_type, storage type> of iSCSI or FCP).

表 6.620. 参数摘要

名称	类型	方向	概述
logical_units	LogicalUnit[]	in	需要从存储域中减少的逻辑单元。

6.205.4. refreshluns POST

此操作会刷新 LUN 大小。

在增大存储服务器上的底层 LUN 的大小后，用户可以刷新 LUN 大小。如果需要，此操作会强制重新扫描提供的 LUN，并使用新大小更新数据库。

例如，要刷新两个 LUN 的大小，可发送如下请求：

```
POST /ovirt-engine/api/storageDomains/262b056b-aede-40f1-9666-b883eff59d40/refreshluns
```

这样的请求正文：

```

<action>
  <logical_units>
    <logical_unit id="1IET_00010001"/>
    <logical_unit id="1IET_00010002"/>
  </logical_units>
</action>

```

表 6.621. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行刷新。

名称	类型	方向	概述
logical_units	LogicalUnit[]	in	需要刷新的 LUN。

6.205.5. 删除 DELETE

删除存储域。

若无任何特殊参数，存储域将从系统分离并从数据库中删除。然后，存储域可以导入到同一或不同的设置中，其中含有所有数据。如果无法访问存储，则操作会失败。

如果 **destroy** 参数为 **true**，那么操作将始终成功，即使存储无法访问，其故障只是被忽略，存储域也会从数据库中删除。

如果 **format** 参数为 **true**，则将格式化实际存储，并将元数据从 LUN 或目录中移除，因此无法再导入到同一设置或另一设置中。

表 6.622. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。
destroy	布尔值	in	指明操作是否应该成功，并且存储域已从数据库中删除，即使该存储无法访问。
format	布尔值	in	指明是否应格式化实际存储，从底层 LUN 或目录中删除所有元数据： [source] ---- DELETE /ovirt-engine/api/storageDomains/123?format=true ---- 此参数是可选的，默认值为 false 。
host	字符串	in	指明应使用哪个主机来删除存储域。

6.205.5.1. destroy

指明操作是否应该成功，并且存储域已从数据库中删除，即使该存储无法访问。

```
DELETE /ovirt-engine/api/storageDomains/123?destroy=true
```

此参数是可选的，默认值为 `false`。当 `destroy` 值为 `true` 时，`host` 参数将被忽略。

6.205.5.2. 主机

指明应使用哪个主机来删除存储域。

此参数是必需的，除非包含 `destroy` 参数且其值为 `true`，这时将忽略 `主机` 参数。

该值应包含主机的名称或标识符。例如，使用名为 `myhost` 的主机删除带有标识符 `123` 的存储域，发送类似如下的请求：

```
DELETE /ovirt-engine/api/storageDomains/123?host=myhost
```

6.205.6. 更新 PUT

更新存储域。

并非所有 `StorageDomain` 的属性在创建后都是 `updatable`。可以更新的有名称、描述、注释、`warning_low_space_indicator`、`critical_space_action_blocker` 和 `wipe_after_delete`。（请注意，更改 `wipe_after_delete` 属性不会在删除已存在的磁盘属性后更改擦除。）

要更新带有标识符 `123` 的存储域的名称和 `wipe_after_delete` 属性，请发送请求，如下所示：

```
PUT /ovirt-engine/api/storageDomains/123
```

请求正文如下：

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

表 6.623. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
storage_domain	StorageDomain	in/Out	更新的存储域。

6.205.7. updateovfstore POST

此操作会强制更新此存储域的 **OVF_STORE**。

OVF_STORE 是一个磁盘映像，包含驻留在存储域中的虚拟机和磁盘的元数据。如果域是从其他数据中心或其他数据中心导入或导出的，则使用此元数据。

默认情况下，会定期更新 **OVF_STORE**（默认设置为 60 分钟），但用户可能需要在重要更改后或认为 **OVF_STORE** 损坏时强制更新更新。

当用户启动时，无论是否需要更新，都将执行 **OVF_STORE** 更新。

表 6.624. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行 OVF_STORE 更新。

6.206. STORAGEDOMAINCONTENTDISK

表 6.625. 方法概述

名称	概述
get	

6.206.1. 获取 GET

表 6.626. 参数摘要

名称	类型	方向	概述
disk	Disk	out	

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。

6.206.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.207. STORAGEDOMAINCONTENTDISKS

管理存储域中可用的磁盘集合。

表 6.627. 方法概述

名称	概述
list	返回存储域中可用磁盘的列表。

6.207.1. 列出 GET

返回存储域中可用磁盘的列表。

只有 **search** 参数中包含 **sortby** 子句时，才会保证返回的磁盘列表的顺序。

表 6.628. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
磁盘	disk[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大磁盘数。
search	字符串	in	用于限制返回的磁盘的查询字符串。

6.207.1.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 `search` 参数执行搜索。默认值为 `true`，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 `false`。

6.207.1.2. follow

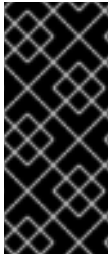
指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.207.1.3. Max

设置要返回的最大磁盘数。如果没有指定，则返回所有磁盘。

6.208. STORAGEDOMAINDISK

管理存储域中可用的单个磁盘。



重要

由于引擎版本 4.2 旨在仅列出存储域中可用的磁盘，并注册未注册的磁盘。所有其他操作（如复制磁盘、移动磁盘等）已被弃用，并将在以后的版本中删除。[要执行这些操作，请使用管理系统所有磁盘的服务，或者使用管理特定磁盘的服务。](#)

表 6.629. 方法概述

名称	概述
<code>copy</code>	将磁盘复制到指定的存储域。
<code>export</code>	将磁盘导出到导出存储域。
<code>get</code>	检索磁盘的描述。
<code>Move</code>	将磁盘移动到其他存储域。
<code>减少</code>	减少磁盘镜像的大小。
<code>remove</code>	删除磁盘。
<code>sparsify</code>	稀疏磁盘。

名称	概述
update	更新磁盘.

6.208.1. 复制 POST

将磁盘复制到指定的存储域.



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。[要复制磁盘，请使用管理该磁盘的服务的副本操作。](#)

表 6.630. 参数摘要

名称	类型	方向	概述
disk	Disk	in	生成磁盘的描述。
storage_domain	StorageDomain	in	创建新磁盘的存储域。

6.208.2. 导出 POST

将磁盘导出到导出存储域.



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。[若要导出磁盘，可使用管理该磁盘的服务的导出操作。](#)

表 6.631. 参数摘要

名称	类型	方向	概述
storage_domain	StorageDomain	in	磁盘应导出到的导出存储域。

6.208.3. 获取 GET

检索磁盘的描述。

表 6.632. 参数摘要

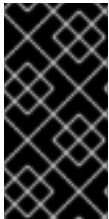
名称	类型	方向	概述
disk	Disk	out	磁盘的描述。
follow	字符串	in	指明应遵循哪些内联链接。

6.208.3.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.208.4. 移动 POST

将磁盘移动到其他存储域。



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。[要移动磁盘，请使用管理该磁盘的服务的移动操作](#)。

表 6.633. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移动。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
storage_domain	StorageDomain	in	磁盘将移动到的存储域。

6.208.5. 减少 POST

减少磁盘镜像的大小。

在逻辑卷中调用 `less`（即，这仅适用于块存储域）。这适用于附加到未运行虚拟机的浮动磁盘和磁

盘。无需指定大小，因为会自动计算最佳大小。

表 6.634. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行移除。

6.208.6. 删除 DELETE

删除磁盘。



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。要删除磁盘，请使用管理该磁盘的服务的 `remove` 操作。

6.208.7. 稀疏 POST

稀疏磁盘。



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。要删除磁盘，请使用管理该磁盘的服务的 `remove` 操作。

6.208.8. 更新 PUT

更新磁盘。



重要

由于引擎版本 4.2 已被弃用，并只保留此版本以向后兼容性。以后会将其移除。若要更新磁盘，可使用管理该磁盘的服务的更新操作。

表 6.635. 参数摘要

名称	类型	方向	概述
disk	Disk	in/Out	应用到磁盘的更新。

6.209. STORAGEDOMAINDISKS

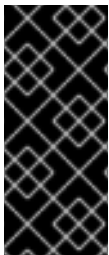
管理特定存储域中可用的磁盘集合。

表 6.636. 方法概述

名称	概述
add	添加或注册磁盘。
list	检索存储域中可用的磁盘的列表。

6.209.1. 添加 POST

添加或注册磁盘。



重要

自 Red Hat Virtualization Manager 版本 4.2 起，此操作已弃用，仅用于向后兼容。以后会将其移除。要添加新磁盘，请使用管理系统磁盘的服务的 **add** 操作。要注册一个未注册的磁盘，请使用管理该磁盘的服务注册操作。

表 6.637. 参数摘要

名称	类型	方向	概述
disk	Disk	in/Out	要添加或注册的磁盘。
unregistered	布尔值	in	指明是否应添加新磁盘，或者是否应注册现有的未注册磁盘。

6.209.1.1. unregistered

指明是否应添加新磁盘，或者是否应注册现有的未注册磁盘。如果值为 **true**，则需要提供要注册的磁盘标识符。例如，要使用 ID 456 注册磁盘，请发送类似如下的请求：

POST /ovirt-engine/api/storagedomains/123/disks?unregistered=true

这样的请求正文：

```
<disk id="456"/>
```

如果值为 `false`，则会在存储域中创建新磁盘。在这种情况下，`provisioning_size`、格式和 `name` 属性是必需的。例如：要在 1 GiB 的写入磁盘上创建新副本，请发送如下请求：

POST /ovirt-engine/api/storagedomains/123/disks

这样的请求正文：

```
<disk>
  <name>mydisk</name>
  <format>cow</format>
  <provisioned_size>1073741824</provisioned_size>
</disk>
```

默认值为 `false`。

从 Red Hat Virtualization Manager 的版本 4.2 开始，这个参数已弃用。

6.209.2. 列出 GET

检索存储域中可用的磁盘的列表。

无法保证返回的磁盘列表的顺序。

表 6.638. 参数摘要

名称	类型	方向	概述
磁盘	<code>disk[]</code>	out	检索的磁盘列表。
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>max</code>	整数	in	设置要返回的最大磁盘数。

名称	类型	方向	概述
unregistered	布尔值	in	指明是否检索存储域中已注册或未注册磁盘的列表。

6.209.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.209.2.2. Max

设置要返回的最大磁盘数。如果没有指定，则返回所有磁盘。

6.209.2.3. unregistered

指明是否检索存储域中已注册或未注册磁盘的列表。要获得存储域中未注册磁盘的列表，调用应指示未注册的标志。例如，要获得未注册磁盘的列表，REST API 调用应如下所示：

```
GET /ovirt-engine/api/storagedomains/123/disks?unregistered=true
```

unregistered 标志的默认值为 **false**。请求仅适用于附加的存储域。

6.210. STORAGEDOMAINSERVERCONNECTION

表 6.639. 方法概述

名称	概述
get	
remove	将存储连接从存储分离。

6.210.1. 获取 GET

表 6.640. 参数摘要

名称	类型	方向	概述
连接	StorageConnection	out	

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。

6.210.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.210.2. 删除 DELETE

将存储连接从存储分离。

表 6.641. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.211. STORAGEDOMAINSERVERCONNECTIONS

管理与存储域中存在的存储服务器的连接集合。

表 6.642. 方法概述

名称	概述
add	
list	将连接列表返回到存储域中存在的存储服务器。

6.211.1. 添加 POST

表 6.643. 参数摘要

名称	类型	方向	概述
连接	StorageConnection	in/Out	

6.211.2. 列出 GET

将连接列表返回到存储域中存在的存储服务器。

无法保证返回的连接列表的顺序。

表 6.644. 参数摘要

名称	类型	方向	概述
连接	StorageConnection[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大连接数。

6.211.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.211.2.2. Max

设置要返回的最大连接数。如果未指定，则返回所有连接。

6.212. STORAGEDOMAINTEMPLATE

表 6.645. 方法概述

名称	概述
get	
import	从导出存储域导入模板的操作。
Register	注册模板意味着要通过将模板和磁盘的配置插入到数据库中而无需复制过程来从数据域导入模板。
remove	

6.212.1. 获取 GET

表 6.646. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
模板	Template	out	

6.212.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.212.2. 导入 POST

从导出存储域导入模板的操作。

例如，要从存储域 123 导入模板 456，发送以下请求：

```
POST /ovirt-engine/api/storagedomains/123/templates/456/import
```

使用以下请求正文：

```
<action>
  <storage_domain>
    <name>myexport</name>
  </storage_domain>
  <cluster>
    <name>mycluster</name>
  </cluster>
</action>
```

如果您在没有指定集群 ID 或名称的情况下注册实体，将使用来自实体 OVF 的集群名称（除非注册请求也包含集群映射）。

表 6.647. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行导入。
clone	布尔值	in	使用可选的 clone 参数为导入模板及其实体生成新 UUID。

名称	类型	方向	概述
cluster	Cluster	in	
专用	布尔值	in	
storage_domain	StorageDomain	in	
模板	Template	in	
vm	VM	in	

6.212.2.1. clone

使用可选的 **clone** 参数为导入模板及其实体生成新 UUID。

从导出域导入模板时，您可以使用不同 Red Hat Virtualization 环境导出的模板导入克隆参数设置为 **false**。

6.212.3. 注册 POST

注册模板意味着要通过将模板和磁盘的配置插入到数据库中而无需复制过程来从数据域导入模板。

表 6.648. 参数摘要

名称	类型	方向	概述
allow_partial_import	布尔值	in	指明是否只允许使用其中一些磁盘注册模板。
async	布尔值	in	指明是否应异步执行注册。
clone	布尔值	in	
cluster	Cluster	in	
专用	布尔值	in	
registration_configuration	RegistrationConfiguration	in	此参数描述了模板的注册方式。

名称	类型	方向	概述
模板	Template	in	
vnic_profile_mappings	VnicProfileMapping[]	in	弃用了将在导入\register 过程中应用的虚拟 NIC 配置集的映射规则的属性。

6.212.3.1. allow_partial_import

指明是否只允许使用其中一些磁盘注册模板。

如果此标志为 **true**，则系统在未找到映像时不会在验证过程中失败，而是允许在不缺少磁盘的情况下注册模板。这主要用于注册模板时，当某些存储域不可用时。默认值为 **false**。

6.212.3.2. registration_configuration

此参数描述了模板的注册方式。

这个参数是可选的。如果未指定 参数，则模板将使用在创建它的原始环境中具有的不同配置进行注册。

6.212.3.3. vnic_profile_mappings

弃用了将在导入\register 过程中应用的虚拟 NIC 配置集的映射规则的属性。



警告

请注意，此属性自引擎版本 4.2.1 起已弃用，仅用于向后兼容。以后会将其移除。要指定 `vnic_profile_mappings`，请使用 `RegistrationConfiguration` 类型内的 `vnic_profile_mappings` 属性。

6.212.4. 删除 DELETE

表 6.649. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.213. STORAGEDOMAINTEMPLATES

管理存储域中可用的一组模板。

表 6.650. 方法概述

名称	概述
list	返回存储域中可用模板的列表。

6.213.1. 列出 GET

返回存储域中可用模板的列表。

返回的模板列表的顺序无法保证。

表 6.651. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的模板的最大数量。
templates	template[]	out	
unregistered	布尔值	in	指明要检索已注册或未注册的模板列表，这些模板是否包含存储域中的磁盘。

6.213.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.213.1.2. Max

设置要返回的模板的最大数量。如果没有指定，则会返回所有模板。

6.213.1.3. unregistered

指明要检索已注册或未注册的模板列表，这些模板是否包含存储域中的磁盘。要获得未注册模板的列表，调用应指示未注册的标志。例如，要获得未注册模板的列表，REST API 调用应如下所示：

```
GET /ovirt-engine/api/storagedomains/123/templates?unregistered=true
```

`unregisterd` 标志的默认值为 `false`。请求仅适用于附加的存储域。

6.214. STORAGEDOMAINVM

表 6.652. 方法概述

名称	概述
<code>get</code>	
<code>import</code>	从导出存储域导入虚拟机。
<code>Register</code>	
<code>remove</code>	从导出存储域中删除虚拟机。

6.214.1. 获取 GET

表 6.653. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>vm</code>	VM	out	

6.214.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.214.2. 导入 POST

从导出存储域导入虚拟机。

例如，发送如下请求：

```
POST /ovirt-engine/api/storagedomains/123/vms/456/import
```

这样的请求正文：

```
<action>
  <storage_domain>
    <name>mydata</name>
  </storage_domain>
  <cluster>
    <name>mycluster</name>
  </cluster>
</action>
```

要将虚拟机导入为新实体，请添加 **clone** 参数：

```
<action>
  <storage_domain>
    <name>mydata</name>
  </storage_domain>
  <cluster>
    <name>mycluster</name>
  </cluster>
  <clone>true</clone>
  <vm>
    <name>myvm</name>
  </vm>
</action>
```

包含可选的 **disk** 参数，以选择要导入的磁盘。例如，要导入具有标识符 123 和 456 的模板磁盘，发送以下请求正文：

```
<action>
  <cluster>
    <name>mycluster</name>
  </cluster>
  <vm>
    <name>myvm</name>
  </vm>
  <disks>
    <disk id="123"/>
  </disks>
</action>
```

```

<disk id="456"/>
</disks>
</action>

```

如果您在没有指定集群 ID 或名称的情况下注册实体，将使用来自实体 OVF 的集群名称（除非注册请求也包含集群映射）。

表 6.654. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行导入。
clone	布尔值	in	指明是否应重新生成导入虚拟机的标识符。
cluster	Cluster	in	
collapse_snapshots	布尔值	in	表示导入的虚拟机的快照应折叠，以便结果为没有快照的虚拟机。
专用	布尔值	in	
storage_domain	StorageDomain	in	
vm	VM	in	

6.214.2.1. clone

指明是否应重新生成导入虚拟机的标识符。

默认情况下，导入虚拟机时会保留标识符。这意味着同一虚拟机不能多次导入，因为该标识符需要是唯一的。要允许导入同一台机器多次将此参数设置为 **true**，因为默认值为 **false**。

6.214.2.2. collapse_snapshots

表示导入的虚拟机的快照应折叠，以便结果为没有快照的虚拟机。

这个参数是可选的，如果没有显式指定，则默认值为 **false**。

6.214.3. 注册 POST

表 6.655. 参数摘要

名称	类型	方向	概述
allow_partial_import	布尔值	in	指明是否只允许虚拟机注册到其中一些磁盘。
async	布尔值	in	指明是否应异步执行注册。
clone	布尔值	in	
cluster	Cluster	in	
reassign_bad_macs	布尔值	in	指明引擎是否应当在导入过程中重新分配有问题的 MAC 地址。
registration_configuration	Registration Configuration	in	此参数描述了虚拟机应如何注册。
vm	VM	in	
vnic_profile_mappings	VnicProfileMapping[]	in	弃用了将在导入 <code>register</code> 过程中应用的虚拟 NIC 配置集的映射规则的属性。

6.214.3.1. allow_partial_import

指明是否只允许虚拟机注册到其中一些磁盘。

如果此标志为 **true**，则如果未找到映像，引擎不会在验证过程中失败，而是允许在不缺少磁盘的情况下注册虚拟机。这主要用于注册虚拟机，当某些存储域不可用时。默认值为 **false**。

6.214.3.2. reassign_bad_macs

指明引擎是否应当在导入过程中重新分配有问题的 MAC 地址。

如果以下地址为 **true**，则 MAC 地址将被视为有问题的地址：

- 它与已在目标环境中分配给虚拟机的 MAC 地址冲突。

- 它不在目标 MAC 地址池的范围。

6.214.3.3. registration_configuration

此参数描述了虚拟机应如何注册。

这个参数是可选的。如果未指定 参数，虚拟机将使用创建它的原始环境中相同的配置进行注册。

6.214.3.4. vnic_profile_mappings

弃用了将在导入\register 过程中应用的虚拟 NIC 配置集的映射规则的属性。



警告

请注意，此属性自引擎版本 4.2.1 起已弃用，仅用于向后兼容。以后会将其移除。要指定 vnic_profile_mappings，请使用 [RegistrationConfiguration](#) 类型内的 vnic_profile_mappings 属性。

6.214.4. 删除 DELETE

从导出存储域中删除虚拟机。

例如，要从存储域 123 中删除虚拟机 456，请发送类似如下的请求：

```
DELETE /ovirt-engine/api/storagedomains/123/vms/456
```

表 6.656. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.215. STORAGEDOMAINVMDISKATTACHMENT

返回附加到导出域中虚拟机的虚拟机的磁盘详情。

表 6.657. 方法概述

名称	概述
get	返回附件的详细信息及其所有属性以及磁盘的链接。

6.215.1. 获取 GET

返回附件的详细信息及其所有属性以及磁盘的链接。

表 6.658. 参数摘要

名称	类型	方向	概述
Attach	DiskAttachm ent	out	磁盘附加。
follow	字符串	in	指明应遵循哪些内联链接。

6.215.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.216. STORAGEDOMAINVMDISKATTACHMENTS

返回附加到导出域中虚拟机的磁盘的详细信息。

表 6.659. 方法概述

名称	概述
list	列出附加到虚拟机的磁盘。

6.216.1. 列出 GET

列出附加到虚拟机的磁盘。

无法保证返回的磁盘附件列表的顺序。

表 6.660. 参数摘要

名称	类型	方向	概述
附件	DiskAttachm ent[]	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.216.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.217. STORAGEDOMAINVMS

列出导出存储域的虚拟机。

例如，要检索存储域中可用且标识符为 123 的虚拟机，发送以下请求：

```
GET /ovirt-engine/api/storagedomains/123/vms
```

这将返回以下响应正文：

```
<vms>
  <vm id="456" href="/api/storagedomains/123/vms/456">
    <name>vm1</name>
    ...
    <storage_domain id="123" href="/api/storagedomains/123"/>
    <actions>
      <link rel="import" href="/api/storagedomains/123/vms/456/import"/>
    </actions>
  </vm>
</vms>
```

这些集合中的虚拟机和模板与顶级 **Vm** 和模板集合中的对应方表示相似，但它们还包含 **StorageDomain** 参考和 导入操作。

表 6.661. 方法概述

名称	概述
list	返回导出存储域的虚拟机列表。

6.217.1. 列出 GET

返回导出存储域的虚拟机列表。

虚拟机返回列表的顺序无法保证。

表 6.662. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大虚拟机数量。
unregistered	布尔值	in	指明要检索包含存储域中磁盘的已注册或未注册虚拟机的列表。
vm	Vm[]	out	

6.217.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.217.1.2. Max

设置要返回的最大虚拟机数量。如果未指定，则返回所有虚拟机。

6.217.1.3. unregistered

指明要检索包含存储域中磁盘的已注册或未注册虚拟机的列表。要获得未注册的虚拟机列表，调用应指示未注册的标志。例如，要获得未注册虚拟机的列表，REST API 调用应如下所示：

```
GET /ovirt-engine/api/storagedomains/123/vms?unregistered=true
```

`unregisterd` 标志的默认值为 `false`。请求仅适用于附加的存储域。

6.218. STORAGEDOMAINS

管理系统中的一组存储域。

表 6.663. 方法概述

名称	概述
<code>add</code>	添加新存储域。
<code>list</code>	返回系统中存储域的列表。

6.218.1. 添加 POST

添加新存储域。

创建新 `StorageDomain` 需要名称、类型、主机和存储属性。使用 `id` 或 `name` 属性识别主机属性。在 Red Hat Virtualization 3.6 及更高版本中，您可以在存储域中默认启用 `wipe` 后删除选项。要配置此功能，请在 `POST` 请求中指定 `wipe_after_delete`。可以在创建域后编辑此选项，但是这样做不会在删除已存在的磁盘属性后更改擦除。

要添加具有指定名称、类型、`storage.type`、`storage.address` 和 `storage.path` 的新存储域，并使用 ID 123 的主机发送类似如下的请求：

```
POST /ovirt-engine/api/storageDomains
```

这样的请求正文：

```
<storage_domain>
  <name>mydata</name>
  <type>data</type>
  <storage>
    <type>nfs</type>
    <address>my nfs.example.com</address>
    <path>/exports/mydata</path>
  </storage>
</host>
```

```

<name>myhost</name>
</host>
</storage_domain>

```

要创建新的 NFS ISO 存储域，请发送如下请求：

```

<storage_domain>
  <name>myisos</name>
  <type>iso</type>
  <storage>
    <type>nfs</type>
    <address>my nfs.example.com</address>
    <path>/export/myisos</path>
  </storage>
</host>
  <name>myhost</name>
</host>
</storage_domain>

```

要创建新 iSCSI 存储域，请发送如下请求：

```

<storage_domain>
  <name>myiscsi</name>
  <type>data</type>
  <storage>
    <type>iscsi</type>
    <logical_units>
      <logical_unit id="3600144f09dbd050000004eedbd340001"/>
      <logical_unit id="3600144f09dbd050000004eedbd340002"/>
    </logical_units>
  </storage>
</host>
  <name>myhost</name>
</host>
</storage_domain>

```

表 6.664. 参数摘要

名称	类型	方向	概述
storage_domain	StorageDomain	in/Out	要添加的存储域。

6.218.2. 列出 GET

返回系统中存储域的列表。

只有 `search` 参数中包含 `sortby` 子句时，才会保证返回的存储域列表的顺序。

表 6.665. 参数摘要

名称	类型	方向	概述
<code>case_sensitive</code>	布尔值	in	指明是否应该在考虑大小写的情况下执行搜索。
<code>filter</code>	布尔值	in	指明是否应根据用户的权限过滤结果。
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>max</code>	整数	in	设置要返回的存储域的最大数量。
<code>search</code>	字符串	in	用于限制返回的存储域的查询字符串。
<code>storage_domains</code>	StorageDomain[]	out	系统中存储域的列表。

6.218.2.1. case_sensitive

指明是否应该在考虑大小写的情况下执行搜索。默认值为 `true`，这意味着会考虑这个情况。如果要搜索忽略大小写，将其设置为 `false`。

6.218.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.218.2.3. Max

设置要返回的存储域的最大数量。如果未指定，则返回所有存储域。

6.219. STORAGESEVERCONNECTION

表 6.666. 方法概述

名称	概述
<code>get</code>	
<code>remove</code>	删除存储连接。

名称	概述
update	更新存储连接。

6.219.1. 获取 GET

表 6.667. 参数摘要

名称	类型	方向	概述
connection	StorageConnection	out	
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>

6.219.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.219.2. 删除 DELETE

删除存储连接。

只有在存储域和 LUN 磁盘均未引用存储连接时，才能删除存储连接。主机名或 id 是可选的；提供它断开与该主机的连接（解除挂载）。

表 6.668. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。
host	字符串	in	从中卸载连接的主机的名称或标识符（断开连接）。

6.219.2.1. 主机

从中卸载连接的主机的名称或标识符（断开连接）。如果未提供，则不会断开任何主机。

例如，要使用标识符 456 的主机删除带标识符 123 的存储连接，发送类似如下的请求：


```
DELETE /ovirt-engine/api/storageconnections/123?host=456
```

6.219.3. 更新 PUT

更新存储连接。

例如：要更改 NFS 存储服务器的地址，请发送类似如下的请求：

```
PUT /ovirt-engine/api/storageconnections/123
```

这样的请求正文：

```
<storage_connection>
  <address>mynewnfs.example.com</address>
</storage_connection>
```

要更改 iSCSI 存储服务器的连接，请发送类似如下的请求：

```
PUT /ovirt-engine/api/storageconnections/123
```

这样的请求正文：

```
<storage_connection>
  <port>3260</port>
  <target>iqn.2017-01.com.myhost:444</target>
</storage_connection>
```

表 6.669. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
连接	StorageConnection	in/Out	
force	布尔值	in	指明无论相关存储域的状态如何（i.

6.219.3.1. force

指明无论相关存储域的状态如何，操作都应成功（例如，当存储域的状态不是维护时，更新也适用）。

此参数是可选的，默认值为 `false`。

6.220. STORAGESEVERCONNECTIONEXTENSION

表 6.670. 方法概述

名称	概述
<code>get</code>	
<code>remove</code>	
<code>update</code>	更新给定主机的存储服务器连接扩展。

6.220.1. 获取 GET

表 6.671. 参数摘要

名称	类型	方向	概述
扩展	StorageConnectionExtension	out	
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。

6.220.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.220.2. 删除 DELETE

表 6.672. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行移除。

6.220.3. 更新 PUT

更新给定主机的存储服务器连接扩展。

要更新主机 123 的存储连接 456，发送如下请求：

```
PUT /ovirt-engine/api/hosts/123/storageconnectionextensions/456
```

这样的请求正文：

```
<storage_connection_extension>
  <target>iqn.2016-01.com.example:mytarget</target>
  <username>myuser</username>
  <password>mypassword</password>
</storage_connection_extension>
```

表 6.673. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
扩展	StorageConnectionExtension	in/Out	

6.221. STORAGESERVERCONNECTIONEXTENSIONS

表 6.674. 方法概述

名称	概述
add	为给定主机创建新的存储服务器连接扩展名。
list	返回列表 os 存储连接扩展。

6.221.1. 添加 POST

为给定主机创建新的存储服务器连接扩展名。

扩展允许用户为特定主机的 iSCSI 目标定义凭据。例如，在从主机 123 连接到 iSCSI 目标时，使用 `myuser` 和 `mypassword` 作为凭证发送如下请求：

POST /ovirt-engine/api/hosts/123/storageconnectionextensions

这样的请求正文：

```
<storage_connection_extension>
  <target>iqn.2016-01.com.example:mytarget</target>
  <username>myuser</username>
  <password>mypassword</password>
</storage_connection_extension>
```

表 6.675. 参数摘要

名称	类型	方向	概述
扩展	StorageConnectionExtension	in/Out	

6.221.2. 列出 GET

返回列表 os 存储连接扩展。

存储连接返回列表的顺序不受保证。

表 6.676. 参数摘要

名称	类型	方向	概述
扩展	StorageConnectionExtension[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大扩展数。

6.221.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.221.2.2. Max

设置要返回的最大扩展数。如果没有指定，则会返回所有扩展。

6.222. STORAGESERVERCONNECTIONS

表 6.677. 方法概述

名称	概述
add	创建新的存储连接。
list	返回存储连接列表。

6.222.1. 添加 POST

创建新的存储连接。

例如，要为 NFS 服务器 `my nfs.example.com` 和 NFS 共享 `/export/mydata` 创建新存储连接，请发送如下请求：

```
POST /ovirt-engine/api/storageconnections
```

这样的请求正文：

```
<storage_connection>
  <type>nfs</type>
  <address>my nfs.example.com</address>
  <path>/export/mydata</path>
  <host>
    <name>myhost</name>
  </host>
</storage_connection>
```

表 6.678. 参数摘要

名称	类型	方向	概述
连接	StorageConnection	in/Out	

6.222.2. 列出 GET

返回存储连接列表。

无法保证返回的连接列表的顺序。

表 6.679. 参数摘要

名称	类型	方向	概述
连接	StorageConnection[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大连接数。

6.222.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.222.2.2. Max

设置要返回的最大连接数。如果未指定，则返回所有连接。

6.223. 系统

表 6.680. 方法概述

名称	概述
get	返回描述 API 的基本信息，如产品名称、版本号和对象数量的摘要。
重新载入配置	

6.223.1. 获取 GET

返回描述 API 的基本信息，如产品名称、版本号和对象数量的摘要。

GET /ovirt-engine/api

我们得到以下响应：

```

<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}"/>
</product_info>
  <name>oVirt Engine</name>
  <vendor>ovirt.org</vendor>
  <version>
    <build>4</build>
    <full_version>4.0.4</full_version>
    <major>4</major>
    <minor>0</minor>
    <revision>0</revision>
  </version>
</product_info>
<special_objects>
  <blank_template href="/ovirt-engine/api/templates/00000000-0000-0000-0000-000000000000"
id="00000000-0000-0000-0000-000000000000"/>
  <root_tag href="/ovirt-engine/api/tags/00000000-0000-0000-0000-000000000000" id="00000000-
0000-0000-0000-000000000000"/>
</special_objects>
<summary>
  <hosts>
    <active>0</active>
    <total>0</total>
  </hosts>
  <storage_domains>
    <active>0</active>
    <total>1</total>
  </storage_domains>
  <users>
    <active>1</active>

```

```

    <total>1</total>
  </users>
  <vms>
    <active>0</active>
    <total>0</total>
  </vms>
</summary>
<time>2016-09-14T12:00:48.132+02:00</time>
</api>

```

入口点为用户提供虚拟环境中集合的链接。每个集合链接的 `Therel` 属性 都会为每个链接提供一个参考点。

入口点还包含其他数据，如 `product_info`、`special_objects` 和 `summary`。

表 6.681. 参数摘要

名称	类型	方向	概述
<code>api</code>	<code>api</code>	out	
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。

6.223.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.223.2. 重新载入配置 POST

表 6.682. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行重新加载。

6.224. SYSTEMOPTION

此服务提供系统特定配置选项的值。

表 6.683. 方法概述

名称	概述
get	获取特定配置选项的值。

6.224.1. 获取 GET

获取特定配置选项的值。

例如，要检索配置选项 **MigrationPoliciesSupported** 的值，发送类似如下的请求：

```
GET /ovirt-engine/api/options/MigrationPoliciesSupported
```

对这个请求的响应如下：

```
<system_option href="/ovirt-engine/api/options/MigrationPoliciesSupported"
id="MigrationPoliciesSupported">
  <name>MigrationPoliciesSupported</name>
  <values>
    <system_option_value>
      <value>true</value>
      <version>4.0</version>
    </system_option_value>
    <system_option_value>
      <value>true</value>
      <version>4.1</version>
    </system_option_value>
    <system_option_value>
      <value>true</value>
      <version>4.2</version>
    </system_option_value>
    <system_option_value>
      <value>>false</value>
      <version>3.6</version>
    </system_option_value>
  </values>
</system_option>
```



注意

查询配置选项需要适当的权限。某些选项仅可由具备管理员权限的用户查询。



重要

没有向后兼容，无法保证选项的名称或值。可以删除选项，也可以随时更改其含义。

我们强烈建议将此服务用于与引擎同时发布的应用程序之外的应用程序。不支持其他应用程序的使用。因此，不会有文档列出可访问的配置选项。

表 6.684. 参数摘要

名称	类型	方向	概述
选项	SystemOption	out	系统返回的配置选项。
version	字符串	in	指定仅应返回特定版本的配置选项的可选 version 参数。

6.224.1.1. version

指定仅应返回特定版本的配置选项的可选 **version** 参数。如果未使用这个参数，则会返回所有版本。

例如，要获取 **MigrationPoliciesSupported** 选项的值，但只有版本 4.2 发送如下请求：

```
GET /ovirt-engine/api/options/MigrationPoliciesSupported?version=4.2
```

对这个请求的响应如下：

```
<system_option href="/ovirt-engine/api/options/MigrationPoliciesSupported"
id="MigrationPoliciesSupported">
  <name>MigrationPoliciesSupported</name>
  <values>
    <system_option_value>
      <value>true</value>
      <version>4.2</version>
    </system_option_value>
  </values>
</system_option>
```

6.225. SYSTEMOPTIONS

提供系统配置选项值的服务。

6.226. SYSTEMPERMISSIONS

此服务不添加新方法，只是注解的占位符，用于指定管理分配给系统对象的权限的资源的路径。

表 6.685. 方法概述

名称	概述
add	为特定实体为用户或组分配新权限。
list	列出特定实体的所有权限。

6.226.1. 添加 POST

为特定实体为用户或组分配新权限。

例如，要为 ID 为 123 的虚拟机分配 UserVmManager 角色给 ID 为 456 的用户发送类似如下的请求：

POST /ovirt-engine/api/vms/123/permissions

这样的请求正文：

```
<permission>
  <role>
    <name>UserVmManager</name>
  </role>
  <user id="456"/>
</permission>
```

要将 SuperUser 角色分配给 ID 为 456 的用户，请发送如下请求：

POST /ovirt-engine/api/permissions

这样的请求正文：

```
<permission>
```

```

<role>
  <name>SuperUser</name>
</role>
<user id="456"/>
</permission>

```

如果要为组而不是用户分配权限，请将 `user` 元素替换为 `组` 元素的适当 ID。例如，将 `UserRole` 角色分配给 ID 为 123 的集群，将 id 789 的组分配如下：

POST /ovirt-engine/api/clusters/123/permissions

这样的请求正文：

```

<permission>
  <role>
    <name>UserRole</name>
  </role>
  <group id="789"/>
</permission>

```

表 6.686. 参数摘要

名称	类型	方向	概述
权限	权限	in/Out	权限。

6.226.2. 列出 GET

列出特定实体的所有权限。

例如，列出 ID 为 123 的集群的所有权限 发送 如下请求：

GET /ovirt-engine/api/clusters/123/permissions

```

<permissions>
  <permission id="456">
    <cluster id="123"/>
    <role id="789"/>
    <user id="451"/>
  </permission>
  <permission id="654">
    <cluster id="123"/>
    <role id="789"/>

```

```
<group id="127"/>
</permission>
</permissions>
```

返回的权限顺序不受保证。

表 6.687. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
权限	permission[]	out	权限列表。

6.226.2.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.227. TAG

在系统中管理特定标签的服务。

表 6.688. 方法概述

名称	概述
get	获取有关标签的信息。
remove	从系统中删除该标签。
update	更新标签实体。

6.227.1. 获取 GET

获取有关标签的信息。

例如，要检索 ID 123 的标签信息，发送如下请求：

```
GET /ovirt-engine/api/tags/123
```

```
<tag href="/ovirt-engine/api/tags/123" id="123">
  <name>root</name>
  <description>root</description>
</tag>
```

表 6.689. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
tag	Tag	out	标签。

6.227.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.227.2. 删除 DELETE

从系统中删除该标签。

例如，要删除 ID 123 的标签，发送类似如下的请求：

```
DELETE /ovirt-engine/api/tags/123
```

表 6.690. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.227.3. 更新 PUT

更新标签实体。

例如，将父标签更新为带有 id 123 的标签 ID 为 456 时，发送类似如下的请求：

```
PUT /ovirt-engine/api/tags/123
```

借助请求正文，例如：

```
<tag>
  <parent id="456"/>
</tag>
```

您还可以指定标签名称而不是 id。例如，将父标签更新为带有 id 123 的标签名称为 **mytag** 时，发送类似如下的请求：

```
<tag>
  <parent>
    <name>mytag</name>
  </parent>
</tag>
```

表 6.691. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
tag	Tag	in/Out	更新的标签。

6.228. TAGS

代表用于管理系统中标签集合的服务。

表 6.692. 方法概述

名称	概述
add	向系统添加新标签。
list	列出系统中的标签。

6.228.1. 添加 POST

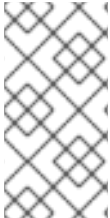
向系统添加新标签。

例如，要向系统添加名称为 **mytag** 的新标签，发送类似如下的请求：

POST /ovirt-engine/api/tags

这样的请求正文：

```
<tag>
  <name>mytag</name>
</tag>
```



注意

如果未指定父标签，**root** 标签是一个特殊的 **pseudo-tag**，它假定为默认的父亲标签。**root** 标签不能被删除，也不能分配父标签。

使用特定父标签创建新标签会发送请求正文，如下所示：

```
<tag>
  <name>mytag</name>
  <parent>
    <name>myparenttag</name>
  </parent>
</tag>
```

表 6.693. 参数摘要

名称	类型	方向	概述
tag	Tag	in/Out	添加的标签。

6.228.2. 列出 GET

列出系统中的标签。

例如，列出系统中标签的完整层次结构会发送如下请求：

GET /ovirt-engine/api/tags

```
<tags>
  <tag href="/ovirt-engine/api/tags/222" id="222">
    <name>root2</name>
    <description>root2</description>
    <parent href="/ovirt-engine/api/tags/111" id="111"/>
  </tag>
</tags>
```



```

</tag>
<tag href="/ovirt-engine/api/tags/333" id="333">
  <name>root3</name>
  <description>root3</description>
  <parent href="/ovirt-engine/api/tags/222" id="222"/>
</tag>
<tag href="/ovirt-engine/api/tags/111" id="111">
  <name>root</name>
  <description>root</description>
</tag>
</tags>

```

在前面的 XML 输出中，您可以看到标签的以下层次结构：

```

root:    (id: 111)
- root2  (id: 222)
- root3  (id: 333)

```

标签返回列表的顺序不受保证。

表 6.694. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大标签数。
tags	Tag[]	out	系统中的所有标签列表。

6.228.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.228.2.2. Max

设置要返回的最大标签数。如果未指定，则返回所有标签。

6.229. TEMPLATE

管理虚拟机模板和模板版本。

表 6.695. 方法概述

名称	概述
export	将模板导出到数据中心导出域。
get	返回有关此模板或模板版本的信息。
remove	删除虚拟机模板。
update	更新模板。

6.229.1. 导出 POST

将模板导出到数据中心导出域。

例如，发送以下请求：

```
POST /ovirt-engine/api/templates/123/export
```

这样的请求正文：

```
<action>
  <storage_domain id="456"/>
  <exclusive>true</exclusive>
</action>
```

表 6.696. 参数摘要

名称	类型	方向	概述
专用	布尔值	in	指明是否应该覆盖具有相同名称的现有模板。
storage_domain	StorageDomain	in	指定目标导出存储域。

6.229.1.1. 专用

指明是否应该覆盖具有相同名称的现有模板。

如果目标域中存在相同名称的模板，则导出操作会报告失败的操作。将此参数设置为 **true** 以更改此行

为并覆盖任何现有的模板。

6.229.2. 获取 GET

返回有关此模板或模板版本的信息。

表 6.697. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
模板	Template	out	有关模板或模板版本的信息。

6.229.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.229.3. 删除 DELETE

删除虚拟机模板。

```
DELETE /ovirt-engine/api/templates/123
```

表 6.698. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.229.4. 更新 PUT

更新模板。

`name`、`description`、`type`、`memory`、`cpu`、`topology`、`os`、`high_availability`、`display`、无状态、`usb` 和 `timezone` 元素可以在创建模板后更新。

例如，要更新模板，使其具有 1 GiB 内存发送请求，如下所示：

```
PUT /ovirt-engine/api/templates/123
```

使用以下请求正文：

```
<template>
  <memory>1073741824</memory>
</template>
```

`version_name` 属性是唯一可在用于模板 版本的 `version` 属性中更新的属性：

```
<template>
  <version>
    <version_name>mytemplate_2</version_name>
  </version>
</template>
```

表 6.699. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行更新。
模板	Template	in/Out	

6.230. TEMPLATECDROM

管理模板上 CD-ROM 设备的服务。

表 6.700. 方法概述

名称	概述
<code>get</code>	返回有关此 CD-ROM 设备的信息。

6.230.1. 获取 GET

返回有关此 CD-ROM 设备的信息。

例如，要获取有关模板 123 的 CD-ROM 设备的信息，请发送如下请求：

```
GET /ovirt-engine/api/templates/123/cdroms/
```

表 6.701. 参数摘要

名称	类型	方向	概述
cdrom	cdrom	out	有关 CD-ROM 设备的信息。
follow	字符串	in	指明应遵循哪些内联链接。

6.230.1.1. cdrom

有关 CD-ROM 设备的信息。

该信息由 `cdrom` 属性组成，其中包含对 CD-ROM 设备、模板和插入磁盘的引用。

如果插入了磁盘，则 `file` 属性将包含对 ISO 镜像的引用：

```
<cdrom href="..." id="000000000-0000-0000-0000-000000000000">
  <template href="/ovirt-engine/api/templates/123" id="123"/>
  <file id="mycd.iso"/>
</cdrom>
```

如果没有插入磁盘，则不会报告 `file` 属性：

```
<cdrom href="..." id="000000000-0000-0000-0000-000000000000">
  <template href="/ovirt-engine/api/templates/123" id="123"/>
</cdrom>
```

6.230.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.231. TEMPLATECDROMS

列出模板的 CD-ROM 设备。

表 6.702. 方法概述

名称	概述
list	返回模板的 CD-ROM 设备列表。

6.231.1. 列出 GET

返回模板的 CD-ROM 设备列表。

CD-ROM 设备返回列表的顺序不受保证。

表 6.703. 参数摘要

名称	类型	方向	概述
cdroms	Cdrom[]	out	模板的 CD-ROM 设备列表。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的最大 CD-ROM 数。

6.231.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.231.1.2. Max

设置要返回的最大 CD-ROM 数。如果没有指定，则返回所有 CD-ROM。

6.232. TEMPLATEDISK

表 6.704. 方法概述

名称	概述
copy	将附加到模板的指定磁盘复制到特定的存储域。
export	
get	

名称	概述
remove	

6.232.1. 复制 POST

将附加到模板的指定磁盘复制到特定的存储域。

表 6.705. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明复制应异步执行。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
storage_domain	StorageDomain	in	

6.232.2. 导出 POST

表 6.706. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行导出。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
storage_domain	StorageDomain	in	

6.232.3. 获取 GET

表 6.707. 参数摘要

名称	类型	方向	概述
disk	Disk	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.232.3.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.232.4. 删除 DELETE

表 6.708. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.233. TEMPLATEDISKATTACHMENT

此服务管理磁盘附加到模板。

表 6.709. 方法概述

名称	概述
get	返回附件的详细信息。
remove	从模板中删除磁盘。

6.233.1. 获取 GET

返回附件的详细信息。

表 6.710. 参数摘要

名称	类型	方向	概述
Attach	DiskAttachm ent	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.233.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.233.2. 删除 DELETE

从模板中删除磁盘。只有在其他存储域中存在磁盘的其他现有副本时，才会删除磁盘。

必须指定存储域来确定应删除哪些副本（模板磁盘可以在多个存储域上具有副本）。

```
DELETE /ovirt-engine/api/templates/{template:id}/diskattachments/{attachment:id}?
storage_domain=072fbaa1-08f3-4a40-9f34-a5ca22dd1d74
```

表 6.711. 参数摘要

名称	类型	方向	概述
force	布尔值	in	
storage_domain	字符串	in	指定要删除的镜像所在的存储域的标识符。

6.234. TEMPLATEDISKATTACHMENTS

此服务管理附加到模板的一组磁盘。每个附加的磁盘都由 [DiskAttachment](#) 表示。

表 6.712. 方法概述

名称	概述
list	列出附加到模板的磁盘。

6.234.1. 列出 GET

列出附加到模板的磁盘。

返回附件列表的顺序无法保证。

表 6.713. 参数摘要

名称	类型	方向	概述
附件	DiskAttachment[]	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.234.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.235. TEMPLATEDISKS

表 6.714. 方法概述

名称	概述
list	返回模板的磁盘列表。

6.235.1. 列出 GET

返回模板的磁盘列表。

无法保证返回的磁盘列表的顺序。

表 6.715. 参数摘要

名称	类型	方向	概述
磁盘	<code>disk[]</code>	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大磁盘数。

6.235.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.235.1.2. Max

设置要返回的最大磁盘数。如果没有指定，则返回所有磁盘。

6.236. TEMPLATEGRAPHICSCONSOLE

表 6.716. 方法概述

名称	概述
get	获取模板的图形控制台配置。
remove	从模板中删除 graphics 控制台。

6.236.1. 获取 GET

获取模板的图形控制台配置。

表 6.717. 参数摘要

名称	类型	方向	概述
console	GraphicsConsole	out	模板图形控制台的相关信息。
follow	字符串	in	指明应遵循哪些内联链接。

6.236.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.236.2. 删除 DELETE

从模板中删除 graphics 控制台。

表 6.718. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.237. TEMPLATEGRAPHICSCONSOLES

表 6.719. 方法概述

名称	概述
add	向模板添加新的图形控制台。
list	列出模板的所有配置的图形控制台。

6.237.1. 添加 POST

向模板添加新的图形控制台。

表 6.720. 参数摘要

名称	类型	方向	概述
console	GraphicsConsole	in/Out	

6.237.2. 列出 GET

列出模板的所有配置的图形控制台。

无法保证返回的图形控制台列表的顺序。

表 6.721. 参数摘要

名称	类型	方向	概述
consoles	GraphicsConsole[]	out	模板的图形控制台列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大控制台数。

6.237.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.237.2.2. Max

设置要返回的最大控制台数。如果没有指定，则返回所有控制台。

6.238. TEMPLATEMEDIATEDDEVICE

表 6.722. 方法概述

名称	概述
get	获取模板的介质设备配置。
remove	从模板创建介质设备。
update	更新有关介质设备的信息。

6.238.1. 获取 GET

获取模板的介质设备配置。

表 6.723. 参数摘要

名称	类型	方向	概述
device	VmMediated Device	out	有关模板的介质设备的信息。
follow	字符串	in	指明应遵循哪些内联链接。

6.238.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.238.2. 删除 DELETE

从模板创建介质设备。

表 6.724. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.238.3. 更新 PUT

更新有关介质设备的信息。

您可以使用 **specParams** 元素更新信息。

例如，要更新介质设备，请发送如下请求：

```
PUT /ovirt-engine/api/templates/123/mediateddevices/00000000-0000-0000-0000-000000000000
<vm_mediated_device>
  <spec_params>
    <property>
      <name>mdevType</name>
      <value>nvidia-11</value>
    </property>
  </spec_params>
</vm_mediated_device>
```

使用响应正文：

```
<vm_mediated_device href="/ovirt-engine/api/templates/123/mediateddevices/00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-000000000000">
  <template href="/ovirt-engine/api/templates/123" id="123"/>
  <spec_params>
    <property>
      <name>mdevType</name>
      <value>nvidia-11</value>
    </property>
  </spec_params>
</vm_mediated_device>
```

表 6.725. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
devices	VmMediated Device	in/Out	有关介质设备的信息。

6.238.3.1. devices

有关介质设备的信息。

请求数据必须包含 **specParams** 属性。响应数据包含有关更新的介质设备的完整信息。

6.239. TEMPLATEMEDIATEDDEVICES

管理模板的介质设备的服务。

表 6.726. 方法概述

名称	概述
add	向模板添加新的介质设备。
list	列出模板的所有配置的介质设备。

6.239.1. 添加 POST

向模板添加新的介质设备。

表 6.727. 参数摘要

名称	类型	方向	概述
device	VmMediated Device	in/Out	

6.239.2. 列出 GET

列出模板的所有配置的介质设备。

返回的介质设备列表的顺序不能保证。

表 6.728. 参数摘要

名称	类型	方向	概述
devices	VmMediated Device[]	out	模板的介质设备列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大介质设备数。

6.239.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.239.2.2. Max

设置要返回的最大介质设备数。如果没有指定所有介质设备，则会返回。

6.240. TEMPLATENIC

表 6.729. 方法概述

名称	概述
get	
remove	
update	更新附加到模板的指定网络接口卡。

6.240.1. 获取 GET

表 6.730. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
nic	NIC	out	

6.240.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.240.2. 删除 DELETE

表 6.731. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.240.3. 更新 PUT

更新附加到模板的指定网络接口卡。

表 6.732. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
nic	NIC	in/Out	

6.241. TEMPLATENICS

表 6.733. 方法概述

名称	概述
add	向模板添加新的网络接口卡。
list	返回模板的 NIC 列表。

6.241.1. 添加 POST

向模板添加新的网络接口卡。

表 6.734. 参数摘要

名称	类型	方向	概述
nic	NIC	in/Out	

6.241.2. 列出 GET

返回模板的 NIC 列表。

NIC 返回列表的顺序不受保证。

表 6.735. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。

名称	类型	方向	概述
max	整数	in	设置要返回的最大 NIC 数。
nics	Nic[]	out	

6.241.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.241.2.2. Max

设置要返回的最大 NIC 数。如果没有指定，则会返回所有 NIC。

6.242. TEMPLATEWATCHDOG

表 6.736. 方法概述

名称	概述
get	
remove	
update	为给定 ID 标识的模板更新 watchdog。

6.242.1. 获取 GET

表 6.737. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
watchdog	Watchdog	out	

6.242.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.242.2. 删除 DELETE

表 6.738. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.242.3. 更新 PUT

为给定 ID 标识的模板更新 watchdog。

表 6.739. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
watchdog	Watchdog	in/Out	

6.243. TEMPLATEWATCHDOGS

表 6.740. 方法概述

名称	概述
add	将 watchdog 添加到由给定 ID 标识的模板中。
list	返回 watchdogs 列表。

6.243.1. 添加 POST

将 watchdog 添加到由给定 ID 标识的模板中。

表 6.741. 参数摘要

名称	类型	方向	概述
watchdog	Watchdog	in/Out	

6.243.2. 列出 GET

返回 watchdogs 列表。

返回的 `watchdogs` 列表的顺序不受保证。

表 6.742. 参数摘要

名称	类型	方向	概述
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。
<code>max</code>	整数	in	设置要返回的最大 <code>watchdogs</code> 数。
<code>watchdogs</code>	<code>Watchdog[]</code>	out	

6.243.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.243.2.2. Max

设置要返回的最大 `watchdogs` 数。如果未指定，则返回所有 `watchdogs`。

6.244. 模板

此服务管理系统中可用的虚拟机模板。

表 6.743. 方法概述

名称	概述
<code>add</code>	创建新模板。
<code>list</code>	返回虚拟机模板列表。

6.244.1. 添加 POST

创建新模板。

这需要 `name` 和 `vm` 元素。若要识别虚拟机，可使用 `vm.id` 或 `vm.name` 属性。例如，要使用标识符 123 从虚拟机创建模板，请发送类似如下的请求：

POST /ovirt-engine/api/templates

这样的请求正文：

```
<template>
  <name>mytemplate</name>
  <vm id="123"/>
</template>
```

从版本 4.3 开始，为了从快照创建虚拟机模板，发送如下请求正文：

```
<template>
  <name>mytemplate</name>
  <vm id="123">
    <snapshots>
      <snapshot id="456"/>
    </snapshots>
  </vm>
</template>
```

可以自定义模板的磁盘，使其部分特征与原始虚拟机的磁盘不同。为此，可使用 `vm.disk_attachments` 属性，指定原始虚拟机的磁盘标识符和您要更改的特征。例如，如果原始虚拟机有一个标识符 456 的磁盘，并且对该磁盘要将格式更改为 `mydisk`，将其格式改为 *Copy On Write* 并将其 *稀疏*，发送一个请求正文，如下所示：

```
<template>
  <name>mytemplate</name>
  <vm id="123">
    <disk_attachments>
      <disk_attachment>
        <disk id="456">
          <name>mydisk</name>
          <format>cow</format>
          <sparse>true</sparse>
        </disk>
      </disk_attachment>
    </disk_attachments>
  </vm>
</template>
```

模板可以创建为现有模板的子版本。这需要新模板的名称和 `vm` 属性，以及新模板版本的 `base_template` 和 `version_name` 属性。`base_template` 和 `version_name` 属性必须在 `template` 部分括起的 `version` 部分中指定。使用 `id` 或 `name` 属性识别虚拟机。

```
<template>
  <name>mytemplate</name>
```

```

<vm id="123"/>
<version>
  <base_template id="456"/>
  <version_name>mytemplate_001</version_name>
</version>
</template>

```

模板的目标存储域可以通过以下两种方式之一来自定义：

1. 全局，在请求级别。请求必须列出要在存储域上创建所需的磁盘附件。如果没有列出磁盘附加，将忽略全局存储域参数。

```

<template>
  <name>mytemplate</name>
  <storage_domain id="123"/>
  <vm id="456">
    <disk_attachments>
      <disk_attachment>
        <disk id="789">
          <format>cow</format>
          <sparse>true</sparse>
        </disk>
      </disk_attachment>
    </disk_attachments>
  </vm>
</template>

```

2. 按每个磁盘附加。为每个磁盘附加指定所需的存储域。指定全局存储定义将根据磁盘附加规格覆盖存储域。

```

<template>
  <name>mytemplate</name>
  <vm id="123">
    <disk_attachments>
      <disk_attachment>
        <disk id="456">
          <format>cow</format>
          <sparse>true</sparse>
          <storage_domains>
            <storage_domain id="789"/>
          </storage_domains>
        </disk>
      </disk_attachment>
    </disk_attachments>
  </vm>
</template>

```

表 6.744. 参数摘要

名称	类型	方向	概述
clone_permissions	布尔值	in	指定虚拟机的权限是否应复制到模板。
seal	布尔值	in	密封模板。
模板	Template	in/Out	有关模板或模板版本的信息。

6.244.1.1. clone_permissions

指定虚拟机的权限是否应复制到模板。

如果提供了此可选参数，并且其值为 **true**，则虚拟机的权限（只有直接虚拟机的权限，而不是继承的参数）将复制到创建的模板。例如，要从 **myvm** 虚拟机创建模板复制其权限，请发送如下请求：

```
POST /ovirt-engine/api/templates?clone_permissions=true
```

这样的请求正文：

```
<template>
  <name>mytemplate</name>
  <vm>
    <name>myvm</name>
  </vm>
</template>
```

6.244.1.2. seal

密封模板。

如果提供了此可选参数，并且其值为 **true**，则模板将在创建后密封。

封装会从文件系统中删除所有特定于主机的配置：SSH 密钥、UDEV 规则、MAC 地址、系统 ID、主机名等，从而更加轻松地使用模板来创建多个虚拟机，无需人工干预。

目前，仅支持 Linux 操作系统的密封。

6.244.2. 列出 GET

返回虚拟机模板列表。

例如：

```
GET /ovirt-engine/api/templates
```

将返回虚拟机和虚拟机模板的列表。

无法保证返回的模板列表的顺序。

表 6.745. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的模板的最大数量。
search	字符串	in	用于限制返回的模板的查询字符串。
templates	template[]	out	虚拟机模板列表。

6.244.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 **search** 参数执行搜索。默认值为 **true**，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 **false**。

6.244.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.244.2.3. Max

设置要返回的模板的最大数量。如果未指定，则返回所有模板。

6.245. UNMANAGEDNETWORK

表 6.746. 方法概述

名称	概述
get	
remove	

6.245.1. 获取 GET

表 6.747. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
network	Unmanaged Network	out	

6.245.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.245.2. 删除 DELETE

表 6.748. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.246. UNMANAGEDNETWORKS

表 6.749. 方法概述

名称	概述
list	返回主机的非受管网络列表。

6.246.1. 列出 GET

返回主机的非受管网络列表。

网络返回列表的顺序无法保证。

表 6.750. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大网络数。
网络	Unmanaged Network[]	out	

6.246.1.1. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.246.1.2. Max

设置要返回的最大网络数。如果没有指定，则会返回所有网络。

6.247. 用户

在系统中管理系统的用户的服务。使用此服务获取用户详细信息或删除用户。要添加新用户，请使用第 6.250 节“用户”。

表 6.751. 方法概述

名称	概述
get	获取系统用户信息。
remove	删除系统用户。
update	更新用户的信息。

6.247.1. 获取 GET

获取系统用户信息。

使用方法：

GET /ovirt-engine/api/users/1234

将返回用户信息：

```
<user href="/ovirt-engine/api/users/1234" id="1234">
  <name>admin</name>
  <link href="/ovirt-engine/api/users/1234/sshpublickeys" rel="sshpublickeys"/>
  <link href="/ovirt-engine/api/users/1234/roles" rel="roles"/>
  <link href="/ovirt-engine/api/users/1234/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/users/1234/tags" rel="tags"/>
  <department></department>
  <domain_entry_id>23456</domain_entry_id>
  <email>user1@domain.com</email>
  <last_name>Lastname</last_name>
  <namespace>*</namespace>
  <principal>user1</principal>
  <user_name>user1@domain-authz</user_name>
  <domain href="/ovirt-engine/api/domains/45678" id="45678">
    <name>domain-authz</name>
  </domain>
</user>
```

表 6.752. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
user	用户	out	系统用户。

6.247.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.247.2. 删除 DELETE

删除系统用户。

使用方法：

DELETE /ovirt-engine/api/users/1234

表 6.753. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.247.3. 更新 PUT

更新用户的信息。

只有 `user_options` 字段才能更新。

例如，更新用户选项：

PUT /ovirt-engine/api/users/123

这样的请求正文：

```
<user>
  <user_options>
    <property>
      <name>test</name>
      <value>["any","JSON"]</value>
    </property>
  </user_options>
</user>
```

重要

自引擎版本 4.4.5 起，此操作已弃用，仅用于向后兼容。以后会将其移除。请改为使用 [选项端点](#)。

表 6.754. 参数摘要

名称	类型	方向	概述
user	用户	in/Out	

6.248. USEROPTION

表 6.755. 方法概述

名称	概述
get	返回 JSON 类型的用户配置文件属性。
remove	删除 JSON 类型的现有属性。

6.248.1. 获取 GET

返回 JSON 类型的用户配置文件属性。

请求示例（针对标识符为 123 和选项为 456 的用户）：

```
GET /ovirt-engine/api/users/123/options/456
```

其结果将是以下 XML 文档：

```
<user_option href="/ovirt-engine/api/users/123/options/456" id="456">
  <name>SomeName</name>
  <content>["any", "JSON"]</content>
  <user href="/ovirt-engine/api/users/123" id="123"/>
</user_option>
```

表 6.756. 参数摘要

名称	类型	方向	概述
选项	UserOption	out	

6.248.2. 删除 DELETE

删除 JSON 类型的现有属性。

请求示例（针对标识符为 **123** 和选项为 **456**的用户）：

```
DELETE /ovirt-engine/api/users/123/options/456
```

6.249. USEROPTIONS

表 6.757. 方法概述

名称	概述
add	添加类型为 JSON 的新用户配置文件属性。
list	返回 JSON 类型的用户配置文件属性列表。

6.249.1. 添加 POST

添加类型为 **JSON** 的新用户配置文件属性。

请求示例（针对标识符为 **123** 的用户）：

```
POST /ovirt-engine/api/users/123/options
```

payload:

```
<user_option>
  <name>SomeName</name>
  <content>["any", "JSON"]</content>
</user_option>
```

表 6.758. 参数摘要

名称	类型	方向	概述
选项	UserOption	in/Out	

6.249.2. 列出 GET

返回 **JSON** 类型的用户配置文件属性列表。

请求示例（针对标识符为 123 的用户）：

```
GET /ovirt-engine/api/users/123/options
```

其结果将是以下 XML 文档：

```
<user_options>
  <user_option href="/ovirt-engine/api/users/123/options/456" id="456">
    <name>SomeName</name>
    <content>["any", "JSON"]</content>
    <user href="/ovirt-engine/api/users/123" id="123"/>
  </user_option>
</user_options>
```

表 6.759. 参数摘要

名称	类型	方向	概述
选项	UserOption[]	out	

6.250. 用户

此服务用于管理系统中的用户。

表 6.760. 方法概述

名称	概述
add	从目录服务添加用户。
list	列出系统中的所有用户。

6.250.1. 添加 POST

从目录服务添加用户。

例如，要从 my extension-authz 授权供应商添加 my user 用户，发送类似如下的请求：

```
POST /ovirt-engine/api/users
```

这样的请求正文：

```
<user>
  <user_name>myuser@myextension-authz</user_name>
  <domain>
    <name>myextension-authz</name>
  </domain>
</user>
```

如果您正在使用 **Active Directory**，则必须将用户主体名称(UPN)传递为用户名，后跟授权提供程序名称。由于 [错误 1147900](#)，您需要提供用户 UPN 的 `principal` 参数。

例如，从 `myextension-authz` 授权供应商添加带有 UPN `myuser@mysubdomain.mydomain.com` 的用户，发送如下的请求正文：

```
<user>
  <principal>myuser@mysubdomain.mydomain.com</principal>
  <user_name>myuser@mysubdomain.mydomain.com@myextension-authz</user_name>
  <domain>
    <name>myextension-authz</name>
  </domain>
</user>
```

表 6.761. 参数摘要

名称	类型	方向	概述
<code>user</code>	用户	in/Out	

6.250.2. 列出 GET

列出系统中的所有用户。

使用方法：

GET `/ovirt-engine/api/users`

将返回用户列表：

```
<users>
  <user href="/ovirt-engine/api/users/1234" id="1234">
```



```

<name>admin</name>
<link href="/ovirt-engine/api/users/1234/sshpublickeys" rel="sshpublickeys"/>
<link href="/ovirt-engine/api/users/1234/roles" rel="roles"/>
<link href="/ovirt-engine/api/users/1234/permissions" rel="permissions"/>
<link href="/ovirt-engine/api/users/1234/tags" rel="tags"/>
<domain_entry_id>23456</domain_entry_id>
<namespace>*</namespace>
<principal>user1</principal>
<user_name>user1@domain-authz</user_name>
<domain href="/ovirt-engine/api/domains/45678" id="45678">
  <name>domain-authz</name>
</domain>
</user>
</users>

```

无法保证返回的用户列表的顺序。

表 6.762. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的最大用户数。
search	字符串	in	用于限制返回的用户的查询字符串。
用户	user[]	out	用户列表。

6.250.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 **search** 参数执行搜索。默认值为 **true**，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 **false**。

6.250.2.2. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。 [详情请查看此处。](#)

6.250.2.3. Max

设置要返回的最大用户数。如果未指定，则返回所有用户。

6.251. VIRTUALFUNCTIONALLOWEDNETWORK

表 6.763. 方法概述

名称	概述
get	
remove	

6.251.1. 获取 GET

表 6.764. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
network	Network	out	

6.251.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.251.2. 删除 DELETE

表 6.765. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.252. VIRTUALFUNCTIONALLOWEDNETWORKS

表 6.766. 方法概述

名称	概述
add	
list	返回网络列表。

6.252.1. 添加 POST

表 6.767. 参数摘要

名称	类型	方向	概述
network	Network	in/Out	

6.252.2. 列出 GET

返回网络列表。

网络返回列表的顺序无法保证。

表 6.768. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大网络数。
网络	network[]	out	

6.252.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.252.2.2. Max

设置要返回的最大网络数。如果没有指定，则会返回所有网络。

6.253. VM

表 6.769. 方法概述

名称	概述
autopincpuand numanodes	在虚拟机上应用自动 CPU 和 NUMA 配置。
cancelmigration	此操作将停止虚拟机到另一物理主机的任何迁移。
clone	

名称	概述
commitsnapshot	将虚拟机永久恢复为预览快照的状态。
detach	将虚拟机从池中分离。
export	导出虚拟机。
freezefilesystems	冻结虚拟机文件系统。
get	检索虚拟机的描述。
logon	启动自动用户登录，以从外部控制台访问虚拟机。
维护	在托管引擎虚拟机上设置全局维护模式。
migrate	将虚拟机迁移到另一物理主机。
previewsnapshot	将虚拟机临时恢复为快照的状态。
reboot	将重新引导请求发送到虚拟机。
remove	删除虚拟机，包括它所连接的虚拟磁盘。
reordermacaddresses	
reset	将重置请求发送到虚拟机。
截屏	捕获虚拟机当前状态的截图。
shutdown	此操作将关闭请求发送到虚拟机。
start	启动虚拟机。
stop	此操作会强制虚拟机关闭。
suspend	此操作会将虚拟机状态保存到磁盘并停止它。
thawfilesystems	thaws 虚拟机文件系统。
ticket	生成时间敏感身份验证令牌，用于访问虚拟机的显示。

名称	概述
undosnapshot	在预览快照前，将虚拟机恢复到其具有的状态。
update	更新系统中给定虚拟机 ID 的虚拟机。

6.253.1. autopincpuandnumanodes POST

在虚拟机上应用自动 CPU 和 NUMA 配置。



重要

由于引擎的 4.5 版已弃用，因此仅保留以向后兼容。以后会将其移除。改为使用 PUT，后跟 [更新操作](#)。

请求的示例：

```
POST /ovirt-engine/api/vms/123/autopincpuandnumanodes
```

这样的请求正文：

```
<action>
  <optimize_cpu_settings>true</optimize_cpu_settings>
</action>
```

表 6.770. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行分离操作。
optimize_cpu_settings	布尔值	in	指定如何应用自动 CPU 和 NUMA 配置。

6.253.1.1. optimize_cpu_settings

指定如何应用自动 CPU 和 NUMA 配置。如果设为 true，将调整 CPU 拓扑以适合虚拟机固定主机硬件。否则，它将使用 VM CPU 拓扑。

6.253.2. 取消迁移 POST

此操作将停止虚拟机到另一物理主机的任何迁移。

POST /ovirt-engine/api/vms/123/cancelmigration

取消迁移操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

表 6.771. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步取消迁移。

6.253.3. 克隆 POST

表 6.772. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行克隆。
discard_snapshots	布尔值	in	当虚拟机应折叠快照时，请使用 discard_snapshots 参数。
storage_domain	StorageDomain	in	虚拟机磁盘将要复制到的存储域。
vm	VM	in	

6.253.3.1. discard_snapshots

当虚拟机应折叠快照时，请使用 **discard_snapshots** 参数。默认值为 **true**。

6.253.4. commitsnapshot POST

将虚拟机永久恢复为预览快照的状态。

详情请查看 [preview_snapshot](#) 操作。

表 6.773. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步提交快照。

6.253.5. 分离 POST

将虚拟机从池中分离。

POST /ovirt-engine/api/vms/123/detach

detach 操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

表 6.774. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行分离操作。

6.253.6. 导出 POST

导出虚拟机。

虚拟机可以导出到导出域。例如，要将虚拟机 123 导出到导出域 **myexport**：

POST /ovirt-engine/api/vms/123/export

这样的请求正文：

```
<action>
  <storage_domain>
    <name>myexport</name>
  </storage_domain>
```

```
<exclusive>true</exclusive>
<discard_snapshots>true</discard_snapshots>
</action>
```

由于引擎的版本 4.2, 也可以将虚拟机导出为虚拟设备(OVA)。例如, 要将虚拟机 123 导出为名为 `myvm.ova` 的 OVA 文件, 该文件位于主机 `myhost` 上的 `/home/ovirt/` 目录中:

```
POST /ovirt-engine/api/vms/123/export
```

这样的请求正文:

```
<action>
  <host>
    <name>myhost</name>
  </host>
  <directory>/home/ovirt</directory>
  <filename>myvm.ova</filename>
</action>
```



注意

在尝试导出域中的任何操作之前, 确认导出操作已经完成。

表 6.775. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行导出。
<code>discard_snapshots</code>	布尔值	in	当虚拟机应折叠所有快照时, 请使用 <code>discard_snapshots</code> 参数。
专用	布尔值	in	当应导出虚拟机时, 请使用 <code>专用</code> 参数, 即使导出域中已存在另一个副本 (覆盖)。
<code>storage_domain</code>	StorageDomain	in	(导出) 要将虚拟机导出到的存储域。

6.253.7. freeze filesystems POST

冻结虚拟机文件系统。

此操作会在对正在运行的虚拟机执行实时快照时，使用 QEMU 客户机代理冻结虚拟机的文件系统。通常，由经理自动执行此操作，但是必须使用 OpenStack 卷(Cinder)磁盘的虚拟机 API 手动执行此操作。

例如：

```
POST /ovirt-engine/api/vms/123/freezefilesystems
```

```
<action/>
```

表 6.776. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行冻结。

6.253.8. 获取 GET

检索虚拟机的描述。

表 6.777. 参数摘要

名称	类型	方向	概述
all_content	布尔值	in	指明是否应将虚拟机的所有属性包含在响应中。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
next_run	布尔值	in	指示返回的结果是否将虚拟机描述为当前正在运行，或者是否使用已执行的修改描述虚拟机，但只有在虚拟机重启后才会生效。
ovf_as_ova	布尔值	in	指明结果是否应该公开显示在该虚拟机的 OVA 文件中的 OVF。
vm	VM	out	虚拟机的描述。

6.253.8.1. all_content

指明是否应将虚拟机的所有属性包含在响应中。

默认情况下排除以下属性：

- `console`
- `initial.configuration.data` - 描述虚拟机的 OVF 文档。
- `rng_source`
- `soundcard`
- `virtio_scsi`

例如，要检索虚拟机 '123' 的完整表示法：

```
GET /ovirt-engine/api/vms/123?all_content=true
```



注意

这些属性默认不包含在内，因为它们会降低性能。这些属性很少被使用，需要对数据库进行额外的查询。仅在需要时使用此参数，因为它会降低性能。

6.253.8.2. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.253.8.3. next_run

指示返回的结果是否将虚拟机描述为当前正在运行，或者是否使用已执行的修改描述虚拟机，但只有在虚拟机重启后才会生效。默认值为 `false`。

如果该参数包含在请求中，但没有值，则假设该值为 `true`。以下请求：

```
GET /vms/{vm:id};next_run
```

等同于使用值 `true` :

```
GET /vms/{vm:id};next_run=true
```

6.253.8.4. ovf_as_ova

指明结果是否应该公开显示在该虚拟机的 OVA 文件中时的 OVF。描述虚拟机的 OVF 文档。这个参数仅在设置了 `all_content=True` 时才有效。OVF 将显示在 `initial .configuration.data` 中。

例如 :

```
GET /vms/{vm:id}?all_content=true&ovf_as_ova=true
```

6.253.9. 登录 POST

启动自动用户登录，以从外部控制台访问虚拟机。

此操作要求安装 `ovirt-guest-agent-gdm-plugin` 和 `ovirt-guest-agent-pam-module` 软件包，并在虚拟机上运行 `ovirt-guest-agent` 服务。

用户需要相应的用户权限才能从外部控制台访问虚拟机。

例如 :

```
POST /ovirt-engine/api/vms/123/logon
```

请求正文 :

```
<action/>
```

表 6.778. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行登录。

6.253.10. 维护 POST

在托管引擎虚拟机上设置全局维护模式。

此操作对其他虚拟机没有影响。

例如：

```
POST /ovirt-engine/api/vms/123/maintenance
```

```
<action>
  <maintenance_enabled>true<maintenance_enabled/>
</action>
```

表 6.779. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行全局维护操作。
maintenance_enabled	布尔值	in	指明是否应启用或禁用全局维护。

6.253.11. 迁移 POST

将虚拟机迁移到另一物理主机。

例如：

```
POST /ovirt-engine/api/vms/123/migrate
```

要指定将虚拟机迁移到的特定主机：

```
<action>
  <host id="2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"/>
</action>
```

表 6.780. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行迁移。
cluster	Cluster	in	指定虚拟机应迁移到的集群。
force	布尔值	in	指定虚拟机应迁移，即使虚拟机被定义为不可迁移。
host	主机	in	指定虚拟机应迁移到的特定主机。
migrate_vms_in_affinity_closure	布尔值	in	还使用此虚拟机迁移正强制关联组中的所有其他虚拟机，这些虚拟机在同一主机上运行。

6.253.11.1. cluster

指定虚拟机应迁移到的集群。这是可选参数。默认情况下，虚拟机迁移到同一集群内的另一台主机上。



警告

不支持实时迁移到另一集群。在尝试迁移前，请强烈考虑目标集群的硬件架构和网络架构。

6.253.11.2. force

指定虚拟机应迁移，即使虚拟机被定义为不可迁移。这是可选参数。默认情况下，设置为 `false`。

6.253.11.3. 主机

指定虚拟机应迁移到的特定主机。这是可选参数。默认情况下，Red Hat Virtualization Manager 会自动选择用于在同一集群中迁移的默认主机。如果 API 用户需要特定的主机，用户可以使用 `id` 或 `name` 参数指定主机。

6.253.11.4. migrate_vms_in_affinity_closure

还使用此虚拟机迁移正强制关联组中的所有其他虚拟机，这些虚拟机在同一主机上运行。

默认值为 `false`。

6.253.12. previewsnapshot POST

将虚拟机临时恢复为快照的状态。

快照通过 `snapshot.id` 参数来指示。它会被临时恢复，以便可以检查内容。完成该检查后，可以使用 `commit_snapshot` 方法使虚拟机的状态变为永久，或使用 `undo_snapshot` 方法丢弃。

表 6.781. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行该预览。
<code>磁盘</code>	<code>disk[]</code>	in	指定快照预览中包含的磁盘。
<code>lease</code>	<code>StorageDomainLease</code>	in	指定要在快照预览中使用的租用存储域 ID。
<code>restore_memory</code>	布尔值	in	
<code>snapshot</code>	<code>snapshot</code>	in	
<code>vm</code>	VM	in	

6.253.12.1. 磁盘

指定快照预览中包含的磁盘。

对于每个磁盘参数，还需要指定其 `image_id`。

例如：要使用标识符 456 预览快照，其中包括一个标识符为 111 的磁盘，其 `image_id` 为 222，发送类似如下的请求：

```
POST /ovirt-engine/api/vms/123/previewsnapshot
```

请求正文：

```
<action>
  <disks>
    <disk id="111">
      <image_id>222</image_id>
    </disk>
  </disks>
  <snapshot id="456"/>
</action>
```

6.253.12.2. lease

指定要在快照预览中使用的租用存储域 ID。如果没有传递租期参数，则将使用预览的快照租用存储域。如果使用空存储域参数传递租期参数，则快照预览不会使用租用。如果租用参数与存储域参数传递，则存储域 ID 只能是属于其中一个虚拟机快照的租用域 ID 之一。这是一个可选参数，默认设置为 `null`

6.253.13. 重启 POST

将重新引导请求发送到虚拟机。

例如：

```
POST /ovirt-engine/api/vms/123/reboot
```

`reboot` 操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

要重新引导虚拟机，即使备份正在运行，该操作应包含“强制”元素。

例如，强制重新引导虚拟机 123:

```
POST /ovirt-engine/api/vms/123/reboot
```

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

表 6.782. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行重新引导。
force	布尔值	in	指明是否应强制重新引导虚拟机，即使它正在运行备份。

6.253.14. 删除 DELETE

删除虚拟机，包括它所连接的虚拟磁盘。

例如，要删除标识符为 123 的虚拟机：

```
DELETE /ovirt-engine/api/vms/123
```

表 6.783. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。
detach_only	布尔值	in	指明是否应先分离并保留所连接的虚拟磁盘，而不是被删除。
force	布尔值	in	指明是否应强制删除虚拟机。

6.253.14.1. force

指明是否应强制删除虚拟机。

如果没有将此标志设为 **true**，则无法删除带有锁定磁盘镜像的锁定虚拟机和虚拟机。

6.253.15. reordermacaddresses POST

表 6.784. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行该操作。

6.253.16. 重置 POST

将重置请求发送到虚拟机。

例如：

```
POST /ovirt-engine/api/vms/123/reset
```

重置操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

表 6.785. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行重置。

6.253.17. 截屏 POST

捕获虚拟机当前状态的截图。

例如：

```
POST /ovirt-engine/api/vms/123/screenshot
```

截屏操作不会采取任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

6.253.18. 关闭 POST

此操作将关闭请求发送到虚拟机。

例如：

```
POST /ovirt-engine/api/vms/123/shutdown
```

关闭操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

要关闭虚拟机，即使备份正在运行，该操作应包括"强制"元素。

例如，强制关闭虚拟机 123:

```
POST /ovirt-engine/api/vms/123/shutdown
```

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

表 6.786. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行关机。
force	布尔值	in	指明即使备份正在为其运行，是否应强制关闭虚拟机。
reason	字符串	in	虚拟机被停止的原因。

6.253.18.1. reason

虚拟机被停止的原因。关闭虚拟机时由用户设置（可选）。

6.253.19. 启动 POST

启动虚拟机。

如果虚拟环境已完成，并且虚拟机包含运行所需的所有组件，则可以启动它。

本例启动虚拟机：

POST /ovirt-engine/api/vms/123/start

通过请求正文：

<action/>

表 6.787. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行启动操作。
authorized_key	AuthorizedKey	in	
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
pause	布尔值	in	如果设为 true ，则以暂停模式启动虚拟机。
use_cloud_init	布尔值	in	如果设为 true ，初始化类型将设为 <i>cloud-init</i> 。
use_ignition	布尔值	in	如果设为 true ，初始化类型将设置为 <i>Ignition</i> 。
use_initialization	布尔值	in	如果设为 true ，则初始化类型由虚拟机的操作系统设置。
use_sysprep	布尔值	in	如果设置为 true ，初始化类型将设置为 <i>Sysprep</i> 。
vm	VM	in	此特定运行的虚拟机的定义。
volatile	布尔值	in	表示即使 guest 启动重启，此运行配置也会被丢弃。

6.253.19.1. pause

如果设为 **true**，则以暂停模式启动虚拟机。默认值为 **false**。

6.253.19.2. use_cloud_init

如果设为 **true**，初始化类型将设为 *cloud-init*。默认值为 **false**。[详情请查看此信息。](#)

6.253.19.3. use_ignition

如果设为 `true`，初始化类型将设置为 *Ignition*。默认值为 `false`。 [详情请查看此信息](#)。

6.253.19.4. use_initialization

如果设为 `true`，则初始化类型由虚拟机的操作系统设置。Windows 将设置为 *Sysprep*，Linux 设置为 *cloud-init*，并将红帽 CoreOS 设置为 *Ignition*。如果明确设置了任何初始化类型（`useCloudInit`、`useSysprep` 或 `useIgnition`），它们将获得优先权，并且忽略此标志。默认值为 `false`。

6.253.19.5. use_sysprep

如果设置为 `true`，初始化类型将设置为 *Sysprep*。默认值为 `false`。 [详情请查看此信息](#)。

6.253.19.6. vm

此特定运行的虚拟机的定义。

例如：

```
<action>
  <vm>
    <os>
      <boot>
        <devices>
          <device>cdrom</device>
        </devices>
      </boot>
    </os>
  </vm>
</action>
```

这将只针对这个特定启动将引导设备设置为 **CDROM**。虚拟机关闭后，该定义将被恢复。

6.253.19.7. 易失性

表示即使 `guest` 启动重启，此运行配置也会被丢弃。默认值为 `false`。

6.253.20. 停止 POST

此操作会强制虚拟机关闭。

例如：

```
POST /ovirt-engine/api/vms/123/stop
```

stop 操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

要停止虚拟机，即使备份正在运行，该操作应包含 **'force'** 元素。

例如，强制停止虚拟机 123:

```
POST /ovirt-engine/api/vms/123/stop
```

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

表 6.788. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应当异步执行停止操作。
force	布尔值	in	指明虚拟机是否应该被强制停止，即使它正在运行备份。
reason	字符串	in	虚拟机被停止的原因。

6.253.20.1. reason

虚拟机被停止的原因。关闭虚拟机时由用户设置（可选）。

6.253.21. 挂起 POST

此操作会将虚拟机状态保存到磁盘并停止它。启动暂停的虚拟机，并通过 **start** 操作恢复虚拟机状态。

例如：

```
POST /ovirt-engine/api/vms/123/suspend
```

挂起的操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

表 6.789. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行暂停操作。

6.253.22. thawfilesystems POST

thaws 虚拟机文件系统。

此操作会在对正在运行的虚拟机执行实时快照时，使用 QEMU 客户机代理修改虚拟机的文件系统。通常，由经理自动执行此操作，但是必须使用 OpenStack 卷(Cinder)磁盘的虚拟机 API 手动执行此操作。

例如：

```
POST /api/vms/123/thawfilesystems
```

```
<action/>
```

表 6.790. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应当异步执行 thaw 文件系统操作。

6.253.23. ticket POST

生成时间敏感身份验证令牌，用于访问虚拟机的显示。

例如：

```
POST /ovirt-engine/api/vms/123/ticket
```

客户端提供的操作可选包括所需的 `ticket` 值和/或到期时间（以秒为单位）。

`response` 指定使用的实际票据值和到期时间。

```
<action>
  <ticket>
    <value>abcd12345</value>
    <expiry>120</expiry>
  </ticket>
</action>
```

重要

如果虚拟机配置为仅支持一个图形协议，则生成的身份验证令牌将对该协议有效。但是，如果虚拟机配置为支持多个协议、VNC 和 SPICE，则身份验证令牌将仅对 SPICE 协议有效。

要获取特定协议（如 VNC）的身份验证令牌，请使用 [该服务](#) 的 `ticket` 方法，它通过发送请求来管理虚拟机的图形控制台：

```
POST /ovirt-engine/api/vms/123/graphicsconsoles/456/ticket
```

表 6.791. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行生成票据。
<code>ticket</code>	<code>ticket</code>	in/Out	

6.253.24. undosnapshot POST

在预览快照前，将虚拟机恢复到其具有的状态。

详情请查看 [preview_snapshot](#) 操作。

表 6.792. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行撤消快照操作。

6.253.25. 更新 PUT

更新系统中给定虚拟机 ID 的虚拟机。

表 6.793. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
next_run	布尔值	in	指明是否应立即将更新应用到虚拟机，或者是否应当在虚拟机重启后应用更新。
vm	VM	in/Out	

6.253.25.1. next_run

指明是否应立即将更新应用到虚拟机，或者是否应当在虚拟机重启后应用更新。默认值为 **false**，因此默认情况下会立即应用更改。

6.254. VMAPPLICATION

此服务提供有关虚拟机中安装的应用程序的信息。

表 6.794. 方法概述

名称	概述
get	返回有关应用的信息。

6.254.1. 获取 GET

返回有关应用的信息。

表 6.795. 参数摘要

名称	类型	方向	概述
应用	Application	out	有关应用的信息。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。

6.254.1.1. 应用

有关应用的信息。

该信息包含包含应用名称的 `name` 属性（这是任意字符串，可能还包含其他信息，如版本）和 `vm` 属性（标识虚拟机）。

例如，类似如下的请求：

```
GET /ovirt-engine/api/vms/123/applications/789
```

可返回如下信息：

```
<application href="/ovirt-engine/api/vms/123/applications/789" id="789">
  <name>ovirt-guest-agent-common-1.0.12-3.el7</name>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</application>
```

6.254.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.255. VMAPPLICATIONS

此服务提供有关虚拟机中安装的应用程序的信息。

表 6.796. 方法概述

名称	概述
list	返回虚拟机中安装的应用列表。

6.255.1. 列出 GET

返回虚拟机中安装的应用列表。

无法保证返回的应用程序列表的顺序。

表 6.797. 参数摘要

名称	类型	方向	概述
应用程序	application[]	out	虚拟机中安装的应用程序列表。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的最大应用程序数。

6.255.1.1. 应用程序

虚拟机中安装的应用程序列表。

例如，类似如下的请求：

```
GET /ovirt-engine/api/vms/123/applications/
```

可返回类似如下的列表：

```
<applications>
<application href="/ovirt-engine/api/vms/123/applications/456" id="456">
  <name>kernel-3.10.0-327.36.1.el7</name>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</application>
<application href="/ovirt-engine/api/vms/123/applications/789" id="789">
  <name>ovirt-guest-agent-common-1.0.12-3.el7</name>
```

```
<vm href="/ovirt-engine/api/vms/123" id="123"/>
</application>
</applications>
```

6.255.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.255.1.3. Max

设置要返回的最大应用程序数。如果没有指定，则会返回所有应用程序。

6.256. VMBACKUP

管理虚拟机的备份的服务。

表 6.798. 方法概述

名称	概述
Finish	完成虚拟机备份实体。
get	返回有关虚拟机备份的信息。

6.256.1. 最终 POST

完成虚拟机备份实体。

最终备份、解锁资源并执行清理。使用 id '123' 和 id '456' 发送请求（如下所示）的备份完成虚拟机：

```
POST /ovirt-engine/api/vms/123/backups/456/finalize
```

请求正文如下：

```
<action />
```

6.256.2. 获取 GET

返回有关虚拟机备份的信息。

表 6.799. 参数摘要

名称	类型	方向	概述
backup	Backup	out	有关虚拟机备份实体的信息。
follow	字符串	in	指明应遵循哪些内联链接。

6.256.2.1. backup

有关虚拟机备份实体的信息。

```
<backups>
  <backup id="backup-uuid">
    <from_checkpoint_id>previous-checkpoint-uuid</from_checkpoint_id>
    <link href="/ovirt-engine/api/vms/vm-uuid/backups/backup-uuid/disks" rel="disks"/>
    <status>initializing</status>
    <creation_date>
  </backup>
</backups>
```

6.256.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.257. VMBACKUPDISK

表 6.800. 方法概述

名称	概述
get	检索磁盘的描述。

6.257.1. 获取 GET

检索磁盘的描述。

表 6.801. 参数摘要

名称	类型	方向	概述
disk	Disk	out	磁盘的描述。
follow	字符串	in	指明应遵循哪些内联链接。

6.257.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.258. VMBACKUPDISKS

表 6.802. 方法概述

名称	概述
list	返回备份中的磁盘列表。

6.258.1. 列出 GET

返回备份中的磁盘列表。

表 6.803. 参数摘要

名称	类型	方向	概述
磁盘	disk[]	out	检索的磁盘列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大磁盘数。

6.258.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.258.1.2. Max

设置要返回的最大磁盘数。如果没有指定，则返回所有磁盘。

6.259. VMBACKUPS

列出虚拟机的备份。

表 6.804. 方法概述

名称	概述
add	将新的备份实体添加到虚拟机。
list	虚拟机备份列表。

6.259.1. 添加 POST

将新的备份实体添加到虚拟机。

例如，要从检查点 `id` 之前-`checkpoint-uuid` 开始为虚拟机启动新的增量备份，请发送类似如下的请求：

```
POST /ovirt-engine/api/vms/123/backups
```

这样的请求正文：

```
<backup>
  <from_checkpoint_id>previous-checkpoint-uuid</from_checkpoint_id>
  <disks>
    <disk id="disk-uuid" />
    ...
  </disks>
</backup>
```

响应正文：

```
<backup id="backup-uuid">
  <from_checkpoint_id>previous-checkpoint-uuid</from_checkpoint_id>
  <to_checkpoint_id>new-checkpoint-uuid</to_checkpoint_id>
  <disks>
    <disk id="disk-uuid" />
    ...
  </disks>
```

```
<status>initializing</status>
<creation_date>
</backup>
```

要提供所创建的备份的 ID，请发送类似如下的请求：

```
POST /ovirt-engine/api/vms/123/backups
```

这样的请求正文：

```
<backup id="backup-uuid">
  <from_checkpoint_id>previous-checkpoint-uuid</from_checkpoint_id>
  <disks>
    <disk id="disk-uuid" />
    ...
  </disks>
</backup>
```

表 6.805. 参数摘要

名称	类型	方向	概述
backup	Backup	in/Out	有关虚拟机备份实体的信息。
require_consistency	布尔值	in	指明如果虚拟机无法停止，备份是否会失败。
use_active	布尔值	in	指明是否使用活动卷来执行备份。

6.259.1.1. require_consistency

指明如果虚拟机无法停止，备份是否会失败。

如果 `requireConsistency=True` VM 备份失败（如果无法冻结虚拟机）。

REST API 调用应如下所示：

```
POST /ovirt-engine/api/vms/123/backups?require_consistency=true
```

`requireConsistency` 标志的默认值为 `false`。

6.259.1.2. use_active

指明是否使用活动卷来执行备份。

如果将 `useActive=False` 创建用于备份操作的快照。

REST API 调用应如下所示：

```
POST /ovirt-engine/api/vms/123/backups?use_active=false
```

`useActive` 标志的默认值为 `false`。

6.259.2. 列出 GET

虚拟机备份列表。

表 6.806. 参数摘要

名称	类型	方向	概述
Backups	<code>backup[]</code>	out	有关虚拟机备份实体的信息。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大虚拟机备份数。

6.259.2.1. Backups

有关虚拟机备份实体的信息。

```
<backups>
  <backup id="backup-uuid">
    <from_checkpoint_id>previous-checkpoint-uuid</from_checkpoint_id>
    <disks>
      <disk id="disk-uuid" />
      ...
      ...
    </disks>
    <status>initializing</status>
```



```
<creation_date>
</backup>
</backups>
```

6.259.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.259.2.3. Max

设置要返回的最大虚拟机备份数。如果未指定，则返回所有虚拟机备份。

6.260. VMCDROM

管理虚拟机的 CDROM 设备。

更改和弹出磁盘应始终使用 `update` 方法，以更改 `file` 属性的值。

表 6.807. 方法概述

名称	概述
<code>get</code>	返回有关此 CDROM 设备的信息。
<code>update</code>	更新有关此 CDROM 设备的信息。

6.260.1. 获取 GET

返回有关此 CDROM 设备的信息。

该信息由 `cdrom` 属性组成，其中包含对 CDROM 设备、虚拟机和插入磁盘的引用。

如果插入了磁盘，则 `file` 属性将包含对 ISO 镜像的引用：

```
<cdrom href="..." id="00000000-0000-0000-0000-000000000000">
  <file id="mycd.iso"/>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</cdrom>
```

如果没有插入磁盘，则不会报告 `file` 属性：

```
<cdrom href="..." id="00000000-0000-0000-0000-000000000000">
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</cdrom>
```

表 6.808. 参数摘要

名称	类型	方向	概述
<code>cdrom</code>	<code>cdrom</code>	out	有关 CDROM 设备的信息。
<code>current</code>	布尔值	in	指明操作是否应该返回当前正在运行的虚拟机的信息。
<code>follow</code>	字符串	in	指明应遵循哪些内联链接。

6.260.1.1. current

指明操作是否应该返回当前正在运行的虚拟机的信息。此参数是可选的，默认值为 `false`。

6.260.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.260.2. 更新 PUT

更新有关此 CDROM 设备的信息。

它可以通过更改 `file` 属性的值来更改或弹出磁盘。例如，要插入或更改磁盘发送如下请求：

```
PUT /ovirt-engine/api/vms/123/cdroms/00000000-0000-0000-0000-000000000000
```

正文应包含 `file` 属性的新值：

```
<cdrom>
  <file id="mycd.iso"/>
</cdrom>
```

本示例中 `id` 属性 `mycd.iso` 的值应当与附加 ISO 存储域中可用的文件对应。

要弹出磁盘，请使用带有空 `id` 的文件：

```
<cdrom>
  <file id=""/>
</cdrom>
```

默认情况下，上述操作会在下一次启动后永久更改虚拟机可见的磁盘，但它们对当前运行的虚拟机没有任何影响。如果要更改当前正在运行的虚拟机可见的磁盘，请添加 `current=true` 参数。例如，要弹出当前磁盘发送如下请求：

```
PUT /ovirt-engine/api/vms/123/cdroms/00000000-0000-0000-0000-000000000000?current=true
```

这样的请求正文：

```
<cdrom>
  <file id=""/>
</cdrom>
```



重要

使用 `current=true` 参数所做的更改永远不会保留，因此在虚拟机重启后它们不会有任何影响。

表 6.809. 参数摘要

名称	类型	方向	概述
<code>cdrom</code>	<code>cdrom</code>	in/Out	有关 CDROM 设备的信息。
<code>current</code>	布尔值	in	指明更新应应用到当前正在运行的虚拟机，还是在下次启动后应用到虚拟机。

6.260.2.1. `current`

指明更新应应用到当前正在运行的虚拟机，还是在下次启动后应用到虚拟机。这个参数是可选的，默认值为 `false`，这表示默认情况下，更新仅在下次引导后才会生效。

6.261. VMCDROMS

管理虚拟机的 CDROM 设备。

目前虚拟机只有一个 CDROM 设备。无法添加新设备，并且无法删除现有设备，因此无法添加或删除方法。更改和弹出 CDROM 磁盘，可使用管理 CDROM 设备的服务的更新方法。

表 6.810. 方法概述

名称	概述
add	将 cdrom 添加到由给定 ID 标识的虚拟机中。
list	返回虚拟机的 CDROM 设备列表。

6.261.1. 添加 POST

将 cdrom 添加到由给定 ID 标识的虚拟机中。

表 6.811. 参数摘要

名称	类型	方向	概述
cdrom	cdrom	in/Out	

6.261.2. 列出 GET

返回虚拟机的 CDROM 设备列表。

CD-ROM 设备返回列表的顺序不受保证。

表 6.812. 参数摘要

名称	类型	方向	概述
cdroms	Cdrom[]	out	虚拟机的 CDROM 设备列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大 CDROM 数。

6.261.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.261.2.2. Max

设置要返回的最大 CDROM 数。如果没有指定，则返回所有 CDROM。

6.262. VMCHECKPOINT

管理虚拟机的检查点的服务。

表 6.813. 方法概述

名称	概述
get	返回有关虚拟机检查点的信息。
remove	删除虚拟机检查点实体。

6.262.1. 获取 GET

返回有关虚拟机检查点的信息。

表 6.814. 参数摘要

名称	类型	方向	概述
checkpoint	checkpoint	out	有关虚拟机检查点实体的信息。
follow	字符串	in	指明应遵循哪些内联链接。

6.262.1.1. checkpoint

有关虚拟机检查点实体的信息。

```
<checkpoint id="checkpoint-uuid">
<link href="/ovirt-engine/api/vms/vm-uuid/checkpoints/checkpoint-uuid/disks" rel="disks"/>
<parent_id>parent-checkpoint-uuid</parent_id>
```

```
<creation_date>xxx</creation_date>
<vm href="/ovirt-engine/api/vms/vm-uuid" id="vm-uuid"/>
</checkpoint>
```

6.262.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.262.2. 删除 DELETE

删除虚拟机检查点实体。

从 libvirt 和数据库中删除检查点。

表 6.815. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.263. VMCHECKPOINTDISK

表 6.816. 方法概述

名称	概述
get	检索磁盘的描述。

6.263.1. 获取 GET

检索磁盘的描述。

表 6.817. 参数摘要

名称	类型	方向	概述
disk	Disk	out	磁盘的描述。
follow	字符串	in	指明应遵循哪些内联链接。

6.263.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.264. VMCHECKPOINTDISKS

表 6.818. 方法概述

名称	概述
list	返回检查点中的磁盘列表。

6.264.1. 列出 GET

返回检查点中的磁盘列表。

表 6.819. 参数摘要

名称	类型	方向	概述
磁盘	disk[]	out	检索的磁盘列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大磁盘数。

6.264.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.264.1.2. Max

设置要返回的最大磁盘数。如果没有指定，则返回所有磁盘。

6.265. VMCHECKPOINTS

列出虚拟机的检查点。

表 6.820. 方法概述

名称	概述
list	虚拟机检查点列表。

6.265.1. 列出 GET

虚拟机检查点列表。

要获取一个带有 id '123' 的虚拟机的检查点列表，请按如下所示发送请求：

```
GET /ovirt-engine/api/vms/123/checkpoints
```

表 6.821. 参数摘要

名称	类型	方向	概述
checkpoints	checkpoint[]	out	有关虚拟机检查点实体的信息。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大虚拟机检查点数。

6.265.1.1. checkpoints

有关虚拟机检查点实体的信息。

```
<checkpoints>
  <checkpoint id="checkpoint-uuid">
    <link href="/ovirt-engine/api/vms/vm-uuid/checkpoints/checkpoint-uuid/disks" rel="disks"/>
    <parent_id>parent-checkpoint-uuid</parent_id>
    <creation_date>xxx</creation_date>
    <vm href="/ovirt-engine/api/vm-uuid" id="vm-uuid"/>
  </checkpoint>
</checkpoints>
```

6.265.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.265.1.3. Max

设置要返回的最大虚拟机检查点数。如果没有指定，则返回所有虚拟机检查点。

6.266. VMDISK

表 6.822. 方法概述

名称	概述
激活	
取消激活	
export	
get	
Move	
减少	减少磁盘镜像的大小。
remove	将磁盘从虚拟机分离。
update	

6.266.1. 激活 POST

表 6.823. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行激活。

6.266.2. 取消激活 POST

表 6.824. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行取消激活。

6.266.3. 导出 POST

表 6.825. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行导出。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。

6.266.4. 获取 GET

表 6.826. 参数摘要

名称	类型	方向	概述
disk	Disk	out	
follow	字符串	in	指明应遵循哪些内联链接。

6.266.4.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.266.5. 移动 POST

表 6.827. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移动。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。

6.266.6. 减少 POST

减少磁盘镜像的大小。

在逻辑卷中调用 `less`（即，这仅适用于块存储域）。这适用于附加到未运行虚拟机的浮动磁盘和磁盘。无需指定大小，因为会自动计算最佳大小。

表 6.828. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.266.7. 删除 DELETE

将磁盘从虚拟机分离。



注意

在 API 版本 3 中，这曾用于从系统完全删除磁盘，但从版本 4 开始，它不会。如果您需要完全删除它，请使用[顶级磁盘服务的删除方法](#)。

表 6.829. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.266.8. 更新 PUT

表 6.830. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
disk	Disk	in/Out	

6.267. VMDISKS

表 6.831. 方法概述

名称	概述
add	
list	返回虚拟机的磁盘列表。

6.267.1. 添加 POST

表 6.832. 参数摘要

名称	类型	方向	概述
disk	Disk	in/Out	

6.267.2. 列出 GET

返回虚拟机的磁盘列表。

无法保证返回的磁盘列表的顺序。

表 6.833. 参数摘要

名称	类型	方向	概述
磁盘	disk[]	out	
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大磁盘数。

6.267.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.267.2.2. Max

设置要返回的最大磁盘数。如果没有指定，则返回所有磁盘。

6.268. VMGRAPHICSCONSOLE

表 6.834. 方法概述

名称	概述
get	检索虚拟机的图形控制台配置。
proxyticket	
remoteviewerconnectionfile	生成与 remote-viewer 客户端兼容的文件。
remove	从虚拟机中删除图形控制台。
ticket	生成时间敏感身份验证令牌，以访问此虚拟机的控制台。

6.268.1. 获取 GET

检索虚拟机的图形控制台配置。



重要

默认情况下，如果没有指定 **当前** 参数，返回的数据对应于下一次虚拟机执行。在系统的当前实施中，这意味着不会填充 **地址** 和 **端口** 属性，因为系统不知道将什么地址和端口用于下一次执行。由于在大多数情况下需要这些属性，因此强烈建议您明确包含 **当前** 参数，值为 **true**。

表 6.835. 参数摘要

名称	类型	方向	概述
console	GraphicsConsole	out	有关虚拟机的图形控制台的信息。
current	布尔值	in	指定返回的数据应当与下一次虚拟机执行或当前执行对应。
follow	字符串	in	指明应遵循哪些内联链接。

6.268.1.1. current

指定返回的数据应当与下一次虚拟机执行或当前执行对应。



重要

地址 和 **端口** 属性将不会填充，除非值为 **true**。

例如，要获取当前执行虚拟机的数据，包括 **地址** 和 **端口** 属性，请发送如下所示的请求：

```
GET /ovit-engine/api/vms/123/graphicsconsoles/456?current=true
```

默认值为 **false**。

6.268.1.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.268.2. proxyticket POST

表 6.836. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行生成票据。
proxy_ticket	ProxyTicket	out	

6.268.3. remotewriterconnectionfile POST

生成与 `remote-viewer` 客户端兼容的文件。

使用以下请求为图形控制台生成远程查看器连接文件：请注意，只有在虚拟机正在运行时，此操作才会生成文件。

```
POST /ovirt-engine/api/vms/123/graphicsconsoles/456/remotewriterconnectionfile
```

`remotewriterconnectionfile` 操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

响应中包含文件，可用于 `remote-viewer` 客户端。

```
<action>
  <remote_viewer_connection_file>
    [virt-viewer]
    type=spice
    host=192.168.1.101
    port=-1
    password=123456789
    delete-this-file=1
    fullscreen=0
    toggle-fullscreen=shift+f11
    release-cursor=shift+f12
    secure-attention=ctrl+alt+end
    tls-port=5900
    enable-smartcard=0
```

```

enable-usb-autoshare=0
usb-filter=null
tls-ciphers=DEFAULT
host-subject=O=local,CN=example.com
ca=...
</remote_viewer_connection_file>
</action>

```

例如，要获取远程查看器连接文件的内容并将其保存到临时文件中，用户可以按如下方式使用 oVirt Python SDK：

```

# Find the virtual machine:
vm = vms_service.list(search='name=myvm')[0]

# Locate the service that manages the virtual machine, as that is where
# the locators are defined:
vm_service = vms_service.vm_service(vm.id)

# Find the graphic console of the virtual machine:
graphics_consoles_service = vm_service.graphics_consoles_service()
graphics_console = graphics_consoles_service.list()[0]

# Generate the remote viewer connection file:
console_service = graphics_consoles_service.console_service(graphics_console.id)
remote_viewer_connection_file = console_service.remote_viewer_connection_file()

# Write the content to file "/tmp/remote_viewer_connection_file.vv"
path = "/tmp/remote_viewer_connection_file.vv"
with open(path, "w") as f:
    f.write(remote_viewer_connection_file)

```

当您创建远程 viewer 连接文件时，您可以连接到虚拟机图形控制台，如下所示：

```

#!/bin/sh -ex

remote-viewer --ovirt-ca-file=/etc/pki/ovirt-engine/ca.pem
/tmp/remote_viewer_connection_file.vv

```

表 6.837. 参数摘要

名称	类型	方向	概述
remote_viewer_connection_file	字符串	out	包含与 remote-viewer 客户端兼容的文件。

6.268.3.1. remote_viewer_connection_file

包含与 `remote-viewer` 客户端兼容的文件。

用户可以使用此属性的内容创建文件，该文件可以传递到 `remote-viewer` 客户端以连接到虚拟机图形控制台。

6.268.4. 删除 DELETE

从虚拟机中删除图形控制台。

表 6.838. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行移除。

6.268.5. ticket POST

生成时间敏感身份验证令牌，以访问此虚拟机的控制台。

```
POST /ovirt-engine/api/vms/123/graphicsconsoles/456/ticket
```

客户端提供的操作可选包括所需的 `ticket` 值和/或到期时间（以秒为单位）。

在所有情况下，响应都指定使用的实际票据值和到期时间。

```
<action>
  <ticket>
    <value>abcd12345</value>
    <expiry>120</expiry>
  </ticket>
</action>
```

表 6.839. 参数摘要

名称	类型	方向	概述
<code>ticket</code>	ticket	in/Out	生成可用于访问此控制台的票据。

6.269. VMGRAPHICSCONSOLES

表 6.840. 方法概述

名称	概述
add	向虚拟机添加新图形控制台。
list	列出虚拟机的所有配置的图形控制台。

6.269.1. 添加 POST

向虚拟机添加新图形控制台。

表 6.841. 参数摘要

名称	类型	方向	概述
console	GraphicsConsole	in/Out	

6.269.2. 列出 GET

列出虚拟机的所有配置的图形控制台。

**重要**

默认情况下，如果没有指定 **当前** 参数，返回的数据对应于下一次虚拟机执行。在系统的当前实施中，这意味着不会填充 **地址** 和 **端口** 属性，因为系统不知道将什么地址和端口用于下一次执行。由于在大多数情况下需要这些属性，因此强烈建议您明确包含 **当前** 参数，值为 **true**。

无法保证返回的图形控制台列表的顺序。

表 6.842. 参数摘要

名称	类型	方向	概述
consoles	GraphicsConsole[]	out	虚拟机的图形控制台列表。
current	布尔值	in	指定返回的数据应当与下一次虚拟机执行或当前执行对应。

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大控制台数。

6.269.2.1. current

指定返回的数据应当与下一次虚拟机执行或当前执行对应。



重要

地址和 端口 属性将不会填充，除非值为 **true**。

例如，要获取当前执行虚拟机的数据，包括 地址 和端口 属性，请发送如下所示的请求：

```
GET /ovirt-engine/api/vms/123/graphicsconsoles?current=true
```

默认值为 **false**。

6.269.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.269.2.3. Max

设置要返回的最大控制台数。如果没有指定，则返回所有控制台。

6.270. VMHOSTDEVICE

此服务用于管理附加到虚拟机的个别主机设备。

表 6.843. 方法概述

名称	概述
get	检索有关附加到给定虚拟机的特定主机设备的信息。
remove	从给定的虚拟机中删除此主机设备的附件。

6.270.1. 获取 GET

检索有关附加到给定虚拟机的特定主机设备的信息。

例如：

```
GET /ovirt-engine/api/vms/123/hostdevices/456
```

```
<host_device href="/ovirt-engine/api/hosts/543/devices/456" id="456">
  <name>pci_0000_04_00_0</name>
  <capability>pci</capability>
  <iommu_group>30</iommu_group>
  <placeholder>true</placeholder>
  <product id="0x13ba">
    <name>GM107GL [Quadro K2200]</name>
  </product>
  <vendor id="0x10de">
    <name>NVIDIA Corporation</name>
  </vendor>
  <host href="/ovirt-engine/api/hosts/543" id="543"/>
  <parent_device href="/ovirt-engine/api/hosts/543/devices/456" id="456">
    <name>pci_0000_00_03_0</name>
  </parent_device>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</host_device>
```

表 6.844. 参数摘要

名称	类型	方向	概述
device	HostDevice	out	检索与给定虚拟机连接的主机设备相关的信息。
follow	字符串	in	指明应遵循哪些内联链接。

6.270.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.270.2. 删除 DELETE

从给定的虚拟机中删除此主机设备的附件。



注意

如果此设备充当 IOMMU 占位符，则无法删除（删除只会导致 占位符 标记设为 `true`）。请注意，所有 IOMMU 占位符设备会在没有其他非占位符设备时自动删除（特定 IOMMU 组中的所有设备都将被分离）。

DELETE /ovirt-engine/api/vms/123/hostdevices/456

表 6.845. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行移除。

6.271. VMHOSTDEVICES

用于管理附加到虚拟机的主机设备的服务。

表 6.846. 方法概述

名称	概述
<code>add</code>	将目标设备附加到给定的虚拟机。
<code>list</code>	列出分配到给定虚拟机的主机设备。

6.271.1. 添加 POST

将目标设备附加到给定的虚拟机。

例如：

POST /ovirt-engine/api/vms/123/hostdevices

使用类型为 **HostDevice** 的请求正文，例如

```
<host_device id="123" />
```



注意

成功附加主机设备的一个必要条件是虚拟机必须固定到一个主机。然后相对于此主机获取设备 ID。



注意

附加作为较大 IOMMU 组一部分的 PCI 设备将导致从该 IOMMU 组附加其余设备作为“占位符”。然后，使用设置为 true 的 **HostDevice** 类型的占位符属性来识别这些设备。

如果您希望附加已充当 IOMMU 占位符的设备，只需为其发出明确的添加操作，并且其占位符标志将被清除，并且该设备可以被虚拟机访问。

表 6.847. 参数摘要

名称	类型	方向	概述
device	HostDevice	in/Out	要附加到给定虚拟机的主机设备。

6.271.2. 列出 GET

列出分配到给定虚拟机的主机设备。

无法保证返回的设备列表的顺序。

表 6.848. 参数摘要

名称	类型	方向	概述
device	HostDevice[]	out	检索到附加到给定虚拟机的主机设备列表。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的最大设备数。

6.271.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.271.2.2. Max

设置要返回的最大设备数。如果没有指定，则返回所有设备。

6.272. VMEDITEDDEVICE

表 6.849. 方法概述

名称	概述
get	检索虚拟机中介质设备的配置。
remove	从虚拟机中删除介质设备。
update	更新有关介质设备的信息。

6.272.1. 获取 GET

检索虚拟机中介质设备的配置。

表 6.850. 参数摘要

名称	类型	方向	概述
device	VmMediated Device	out	有关虚拟机的介质设备的信息。
follow	字符串	in	指明应遵循哪些内联链接。

6.272.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.272.2. 删除 DELETE

从虚拟机中删除介质设备。

表 6.851. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.272.3. 更新 PUT

更新有关介质设备的信息。

您可以使用 **specParams** 元素更新信息。

例如，要更新介质设备，请发送如下请求：

```
PUT /ovirt-engine/api/vms/123/mediateddevices/00000000-0000-0000-0000-000000000000
<vm_mediated_device>
  <spec_params>
    <property>
      <name>mdevType</name>
      <value>nvidia-11</value>
    </property>
  </spec_params>
</vm_mediated_device>
```

使用响应正文：

```
<vm_mediated_device href="/ovirt-engine/api/vms/123/mediateddevices/00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-000000000000">
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
  <spec_params>
    <property>
      <name>mdevType</name>
      <value>nvidia-11</value>
    </property>
  </spec_params>
</vm_mediated_device>
```

表 6.852. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。

名称	类型	方向	概述
device	VmMediated Device	in/Out	有关介质设备的信息。

6.272.3.1. device

有关介质设备的信息。

请求数据必须包含 **specParams** 属性。响应数据包含有关更新的介质设备的完整信息。

6.273. VM-MEDIATED DEVICES

此服务管理虚拟机的介质设备。

表 6.853. 方法概述

名称	概述
add	向虚拟机添加新的介质设备。
list	列出虚拟机的所有配置的介质设备。

6.273.1. 添加 POST

向虚拟机添加新的介质设备。

表 6.854. 参数摘要

名称	类型	方向	概述
device	VmMediated Device	in/Out	

6.273.2. 列出 GET

列出虚拟机的所有配置的介质设备。

未保证返回介质设备列表的顺序。

表 6.855. 参数摘要

名称	类型	方向	概述
devices	VmMediated Device[]	out	虚拟机的介质设备列表。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大介质设备数。

6.273.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.273.2.2. Max

设置要返回的最大介质设备数。如果没有指定所有介质设备，则会返回。

6.274. VMNIC

表 6.856. 方法概述

名称	概述
激活	
取消激活	
get	
remove	删除 NIC。
update	更新 NIC。

6.274.1. 激活 POST

表 6.857. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行激活。

6.274.2. 取消激活 POST

表 6.858. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行取消激活。

6.274.3. 获取 GET

表 6.859. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
nic	NIC	out	

6.274.3.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.274.4. 删除 DELETE

删除 NIC。

例如，要从 ID 123 的虚拟机中删除 ID 为 456 的 NIC，发送类似如下的请求：

```
DELETE /ovirt-engine/api/vms/123/nics/456
```

重要

热插拔功能仅支持带有热插拔操作的虚拟机操作系统。操作系统示例包括：

- **Red Hat Enterprise Linux 6**
- **Red Hat Enterprise Linux 5**
- **Windows Server 2008 和**
- **Windows Server 2003**

表 6.860. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.274.5. 更新 PUT

更新 NIC.

例如，更新具有 456 属于 ID 123 的虚拟机的 NIC 发送如下请求：

```
PUT /ovirt-engine/api/vms/123/nics/456
```

这样的请求正文：

```
<nic>
  <name>myNIC</name>
  <interface>e1000</interface>
  <vnic_profile id='789' />
</nic>
```

重要

热插拔功能仅支持带有热插拔操作的虚拟机操作系统。操作系统示例包括：

- **Red Hat Enterprise Linux 6**
- **Red Hat Enterprise Linux 5**
- **Windows Server 2008 和**
- **Windows Server 2003**

表 6.861. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
nic	NIC	in/Out	

6.275. VMNICS

表 6.862. 方法概述

名称	概述
add	将 NIC 添加到虚拟机。
list	返回虚拟机的 NIC 列表。

6.275.1. 添加 POST

将 NIC 添加到虚拟机。

以下示例使用 `virtio` 和 `NIC` 配置集 456 向虚拟机添加一个名为 `myNIC` 的网络接口。

```
POST /ovirt-engine/api/vms/123/nics
```

```
<nic>
  <name>myNIC</name>
  <interface>virtio</interface>
  <vnic_profile id="456"/>
</nic>
```

以下示例使用 `curl` 发送该请求：

```
curl \
--request POST \
--header "Version: 4" \
--header "Content-Type: application/xml" \
--header "Accept: application/xml" \
--user "admin@internal:mypassword" \
--cacert /etc/pki/ovirt-engine/ca.pem \
--data '
<nic>
  <name>myNIC</name>
  <interface>virtio</interface>
  <vnic_profile id="456"/>
</nic>
'\
https://myengine.example.com/ovirt-engine/api/vms/123/nics
```

重要

热插拔功能仅支持带有热插拔操作的虚拟机操作系统。操作系统示例包括：

- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 5
- Windows Server 2008 和
- Windows Server 2003

表 6.863. 参数摘要

名称	类型	方向	概述
nic	NIC	in/Out	

6.275.2. 列出 GET

返回虚拟机的 NIC 列表。

NIC 返回列表的顺序不受保证。

表 6.864. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大 NIC 数。
nics	Nic[]	out	

6.275.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.275.2.2. Max

设置要返回的最大 NIC 数。如果没有指定，则会返回所有 NIC。

6.276. VMNUMANODE

表 6.865. 方法概述

名称	概述
get	
remove	删除虚拟 NUMA 节点。
update	更新虚拟 NUMA 节点。

6.276.1. 获取 GET

表 6.866. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
node	VirtualNuma Node	out	

6.276.1.1. follow

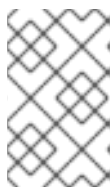
指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.276.2. 删除 DELETE

删除虚拟 NUMA 节点。

删除虚拟 NUMA 节点的示例：

```
DELETE /ovirt-engine/api/vms/123/numanodes/456
```



注意

首先需要从最高索引中删除 numa 节点。

表 6.867. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.276.3. 更新 PUT

更新虚拟 NUMA 节点。

将虚拟 NUMA 节点固定到主机上的物理 NUMA 节点的示例：

```
PUT /ovirt-engine/api/vms/123/numanodes/456
```

请求正文应包含以下内容：

```
<vm_numa_node>
  <numa_node_pins>
    <numa_node_pin>
      <index>0</index>
    </numa_node_pin>
  </numa_node_pins>
</vm_numa_node>
```

表 6.868. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
node	VirtualNuma Node	in/Out	

6.277. VMNUMANODES

表 6.869. 方法概述

名称	概述
add	为虚拟机创建新的虚拟 NUMA 节点。
list	列出虚拟机的虚拟 NUMA 节点。

6.277.1. 添加 POST

为虚拟机创建新的虚拟 NUMA 节点。

创建 NUMA 节点的示例：

```
POST /ovirt-engine/api/vms/c7ecd2dc/numanodes
Accept: application/xml
Content-type: application/xml
```

请求正文可包含以下内容：

```
<vm_numa_node>
```



```

<cpu>
  <cores>
    <core>
      <index>0</index>
    </core>
  </cores>
</cpu>
<index>0</index>
<memory>1024</memory>
<numa_tune_mode>strict</numa_tune_mode>
</vm_numa_node>

```

表 6.870. 参数摘要

名称	类型	方向	概述
node	VirtualNuma Node	in/Out	

6.277.2. 列出 GET

列出虚拟机的虚拟 NUMA 节点。

NUMA 节点返回列表的顺序不受保证。

表 6.871. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大节点数。
节点	VirtualNuma Node[]	out	

6.277.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.277.2.2. Max

设置要返回的最大节点数。如果没有指定，则返回所有节点。

6.278. VMPOOL

用于管理虚拟机池的服务。

表 6.872. 方法概述

名称	概述
allocatevm	此操作将在虚拟机池中分配虚拟机。
get	获取虚拟机池。
remove	删除虚拟机池。
update	更新虚拟机池。

6.278.1. assignvm POST

此操作将在虚拟机池中分配虚拟机。

POST /ovirt-engine/api/vmpools/123/allocatevm

allocate 虚拟机操作不执行任何特定于操作的参数，因此请求正文应包含空操作：

```
<action/>
```

表 6.873. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行分配。

6.278.2. 获取 GET

获取虚拟机池。

GET /ovirt-engine/api/vmpools/123

您将获得如下 XML 响应：

```

<vm_pool id="123">
  <actions>...</actions>
  <name>MyVmPool</name>
  <description>MyVmPool description</description>
  <link href="/ovirt-engine/api/vmpools/123/permissions" rel="permissions"/>
  <max_user_vms>1</max_user_vms>
  <prestarted_vms>0</prestarted_vms>
  <size>100</size>
  <stateful>false</stateful>
  <type>automatic</type>
  <use_latest_template_version>false</use_latest_template_version>
  <cluster id="123"/>
  <template id="123"/>
  <vm id="123">...</vm>
  ...
</vm_pool>

```

表 6.874. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
池	VmPool	out	检索到的虚拟机池。

6.278.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.278.3. 删除 DELETE

删除虚拟机池。

```
DELETE /ovirt-engine/api/vmpools/123
```

表 6.875. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.278.4. 更新 PUT

更新虚拟机池。

```
PUT /ovirt-engine/api/vmpools/123
```

创建虚拟机池后，可以更新 `name`、`description`、`size`、`prestarted_vms` 和 `max_user_vms` 属性。

```
<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>
```

表 6.876. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行更新。
<code>池</code>	VmPool	in/Out	正在更新的虚拟机池。
<code>seal</code>	布尔值	in	指定在创建池之后为池创建的虚拟机应密封。

6.278.4.1. seal

指定在创建池之后为池创建的虚拟机应密封。

如果提供此可选参数，且其值为 `true`，则为池创建的虚拟机将在创建后密封。如果值为 `'false'`，则不会密封虚拟机。如果没有提供该参数，则虚拟机将被密封，只有在从封装的模板中创建它们时，其客户机操作系统不会设置为 `Windows`。这个参数只影响更新池时创建的虚拟机。

例如，要更新虚拟机池并密封创建的其他虚拟机，请发送如下请求：

```
PUT /ovirt-engine/api/vmpools/123?seal=true
```

使用以下机构：

```
<vmpool>
  <name>VM_Pool_B</name>
```

```
<description>Virtual Machine Pool B</description>
<size>7</size>
</vmpool>
```

6.279. VMPOOLS

提供对虚拟机池的读写访问权限。

表 6.877. 方法概述

名称	概述
add	创建新的虚拟机池。
list	获取可用虚拟机池的列表。

6.279.1. 添加 POST

创建新的虚拟机池。

新池需要 **name**、**cluster**和 **template** 属性。使用 **id** 或 **name** 嵌套属性识别集群和模板：

```
POST /ovirt-engine/api/vmpools
```

使用以下机构：

```
<vmpool>
  <name>mypool</name>
  <cluster id="123"/>
  <template id="456"/>
</vmpool>
```

表 6.878. 参数摘要

名称	类型	方向	概述
池	VmPool	in/Out	要添加的池.
seal	布尔值	in	指定在创建池之后为池创建的虚拟机应密封。

6.279.1.1. seal

指定在创建池之后为池创建的虚拟机应密封。

如果提供此可选参数，且其值为 `true`，则为池创建的虚拟机将在创建后密封。如果值为 `false`，则不会密封虚拟机。如果没有提供该参数，则虚拟机将被密封，只有在从封装的模板中创建它们时，其客户机操作系统不会设置为 `Windows`。这个参数只影响创建池时创建的虚拟机。

例如，要创建包含 5 个虚拟机的虚拟机池并密封虚拟机，请发送如下请求：

```
POST /ovirt-engine/api/vmpools?seal=true
```

使用以下机构：

```
<vmpool>
  <name>mypool</name>
  <cluster id="123"/>
  <template id="456"/>
  <size>5</size>
</vmpool>
```

6.279.2. 列出 GET

获取可用虚拟机池的列表。

```
GET /ovirt-engine/api/vmpools
```

您将收到以下响应：

```
<vm_pools>
  <vm_pool id="123">
    ...
  </vm_pool>
  ...
</vm_pools>
```

只有 `search` 参数中包含 `sortby` 子句时，才会保证池返回列表的顺序。

表 6.879. 参数摘要

名称	类型	方向	概述
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	设置要返回的最大池数。
pools	VmPool[]	out	检索的池。
search	字符串	in	用于限制返回的池的查询字符串。

6.279.2.1. case_sensitive

指明是否应该在考虑大小写的情况下使用 **search** 参数执行搜索。默认值为 **true**，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 **false**。

6.279.2.2. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。详情请查看[此处](#)。

6.279.2.3. Max

设置要返回的最大池数。如果未指定此值，则返回所有池。

6.280. VMREPORTEDDEVICE

表 6.880. 方法概述

名称	概述
get	

6.280.1. 获取 GET

表 6.881. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
reported_device	ReportedDevice	out	

6.280.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.281. VMREPORTEDDEVICES

表 6.882. 方法概述

名称	概述
list	返回虚拟机报告的设备列表。

6.281.1. 列出 GET

返回虚拟机报告的设备列表。

无法保证返回的设备列表的顺序。

表 6.883. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大设备数。
reported_device	ReportedDevice[]	out	

6.281.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.281.1.2. Max

设置要返回的最大设备数。如果没有指定，则返回所有设备。

6.282. VMSESSION

表 6.884. 方法概述

名称	概述
get	

6.282.1. 获取 GET

表 6.885. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
会话	会话	out	

6.282.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.283. VMSESSIONS

提供有关虚拟机用户会话的信息。

表 6.886. 方法概述

名称	概述
list	列出此虚拟机的所有用户会话。

6.283.1. 列出 GET

列出此虚拟机的所有用户会话。

例如，要检索虚拟机的会话信息 123 发送如下请求：

```
GET /ovirt-engine/api/vms/123/sessions
```

响应正文将包含类似如下的内容：

```
<sessions>
  <session href="/ovirt-engine/api/vms/123/sessions/456" id="456">
    <console_user>true</console_user>
    <ip>
      <address>192.168.122.1</address>
    </ip>
    <user href="/ovirt-engine/api/users/789" id="789"/>
    <vm href="/ovirt-engine/api/vms/123" id="123"/>
  </session>
  ...
</sessions>
```

返回的会话列表的顺序不受保证。

表 6.887. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置返回的最大会话数。
会话	<code>session[]</code>	out	

6.283.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.283.1.2. Max

设置返回的最大会话数。如果未指定，将返回所有会话。

6.284. VMWATCHDOG

管理虚拟机上监视的服务。

表 6.888. 方法概述

名称	概述
get	返回有关 watchdog 的信息。
remove	从虚拟机中删除 watchdog。
update	更新有关 watchdog 的信息。

6.284.1. 获取 GET

返回有关 watchdog 的信息。

表 6.889. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
watchdog	Watchdog	out	有关 watchdog 的信息。

6.284.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.284.1.2. watchdog

有关 watchdog 的信息。

这些信息包括 模型 元素、操作 元素以及对虚拟机的引用。它可能类似如下：

```
<watchdogs>
  <watchdog href="/ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000"
id="00000000-0000-0000-0000-000000000000">
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
  <action>poweroff</action>
```

```
<model>i6300esb</model>
</watchdog>
</watchdogs>
```

6.284.2. 删除 DELETE

从虚拟机中删除 `watchdog`。

例如，要从虚拟机中删除 `watchdog`，请发送如下请求：

```
DELETE /ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000
```

表 6.890. 参数摘要

名称	类型	方向	概述
<code>async</code>	布尔值	in	指明是否应异步执行移除。

6.284.3. 更新 PUT

更新有关 `watchdog` 的信息。

您可以使用 `操作` 和 `模型` 元素来更新信息。

例如，要更新 `watchdog`，请发送类似如下的请求：

```
PUT /ovirt-engine/api/vms/123/watchdogs
<watchdog>
  <action>reset</action>
</watchdog>
```

使用响应正文：

```
<watchdog href="/ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000"
id="00000000-0000-0000-0000-000000000000">
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
  <action>reset</action>
  <model>i6300esb</model>
</watchdog>
```

表 6.891. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
watchdog	Watchdog	in/Out	有关 watchdog 的信息。

6.284.3.1. watchdog

有关 watchdog 的信息。

请求数据必须至少包含 **模型** 和 **操作元素** 之一。响应数据包含有关更新的 **watchdog** 的完整信息。

6.285. VMWATCHDOGS

列出虚拟机的监视。

表 6.892. 方法概述

名称	概述
add	将新的 watchdog 添加到虚拟机。
list	虚拟机的 watchdogs 列表。

6.285.1. 添加 POST

将新的 watchdog 添加到虚拟机。

例如，要将 **watchdog** 添加到虚拟机中，请发送如下请求：

```
POST /ovirt-engine/api/vms/123/watchdogs
<watchdog>
  <action>poweroff</action>
  <model>i6300esb</model>
</watchdog>
```

使用响应正文：

```
<watchdog href="/ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000"
id="00000000-0000-0000-0000-000000000000">
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
  <action>poweroff</action>
  <model>i6300esb</model>
</watchdog>
```

表 6.893. 参数摘要

名称	类型	方向	概述
watchdog	Watchdog	in/Out	有关 watchdog 的信息。

6.285.1.1. watchdog

有关 watchdog 的信息。

请求数据必须包含模型元素（如 i6300esb）和 action 元素（none、re
reset、poweroff、dump、pause）。响应数据还包含对添加的 watchdog 和虚拟机的引用。

6.285.2. 列出 GET

虚拟机的 watchdogs 列表。

返回的 watchdogs 列表的顺序不受保证。

表 6.894. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大 watchdogs 数。
watchdogs	Watchdog[]	out	有关 watchdog 的信息。

6.285.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.285.2.2. Max

设置要返回的最大 `watchdogs` 数。如果未指定，则返回所有 `watchdogs`。

6.285.2.3. watchdogs

有关 `watchdog` 的信息。

这些信息包括 `模型` 元素、`操作` 元素以及对虚拟机的引用。它可能类似如下：

```
<watchdogs>
  <watchdog href="/ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000"
id="00000000-0000-0000-0000-000000000000">
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
  <action>poweroff</action>
  <model>i6300esb</model>
</watchdog>
</watchdogs>
```

6.286. VM

表 6.895. 方法概述

名称	概述
<code>add</code>	创建新虚拟机。
<code>list</code>	返回系统虚拟机的列表。

6.286.1. 添加 POST

创建新虚拟机。

可以通过不同的方式创建虚拟机：

- 从模板.在这种情况下，必须提供模板的标识符或名称。例如，使用普通 `shell` 脚本和 `XML`：

```
#!/bin/sh -x

url="https://engine.example.com/ovirt-engine/api"
user="admin@internal"
password="..."
curl \
--verbose \
--cacert /etc/pki/ovirt-engine/ca.pem \
--user "${user}:${password}" \
--request POST \
--header "Version: 4" \
--header "Content-Type: application/xml" \
--header "Accept: application/xml" \
--data '
<vm>
  <name>myvm</name>
  <template>
    <name>Blank</name>
  </template>
  <cluster>
    <name>mycluster</name>
  </cluster>
</vm>
'\
"${url}/vms"
```

-

从快照.在这种情况下, 必须提供快照的标识符。例如, 使用普通的 shell 脚本和 XML :

```
#!/bin/sh -x

url="https://engine.example.com/ovirt-engine/api"
user="admin@internal"
password="..."
curl \
--verbose \
--cacert /etc/pki/ovirt-engine/ca.pem \
--user "${user}:${password}" \
--request POST \
--header "Content-Type: application/xml" \
--header "Accept: application/xml" \
--data '
<vm>
  <name>myvm</name>
  <snapshots>
    <snapshot id="266742a5-6a65-483c-816d-d2ce49746680"/>
  </snapshots>
  <cluster>
    <name>mycluster</name>
  </cluster>
</vm>
'\
"${url}/vms"
```


从模板或快照创建虚拟机时，明确指明在哪个存储域中为虚拟机创建磁盘通常很有用。如果从模板创建虚拟机，这可以通过一组代表映射的 `disk_attachment` 元素实现：

```
<vm>
...
<disk_attachments>
  <disk_attachment>
    <disk id="8d4bd566-6c86-4592-a4a7-912dbf93c298">
      <storage_domains>
        <storage_domain id="9cb6cb0a-cf1d-41c2-92ca-5a6d665649c9"/>
      </storage_domains>
    </disk>
  <disk_attachment>
</disk_attachments>
</vm>
```

从快照创建虚拟机时，此磁盘集合略有不同，它使用 `image_id` 属性而不是 `id`。

```
<vm>
...
<disk_attachments>
  <disk_attachment>
    <disk>
      <image_id>8d4bd566-6c86-4592-a4a7-912dbf93c298</image_id>
      <storage_domains>
        <storage_domain id="9cb6cb0a-cf1d-41c2-92ca-5a6d665649c9"/>
      </storage_domains>
    </disk>
  <disk_attachment>
</disk_attachments>
</vm>
```

可以在 XML 描述中指定额外的虚拟机参数，例如，桌面类型的虚拟机，RAM 为 2 GiB，其他描述可以添加发送请求正文，如下所示：

```
<vm>
  <name>myvm</name>
  <description>My Desktop Virtual Machine</description>
  <type>desktop</type>
  <memory>2147483648</memory>
...
</vm>
```

可引导 CDROM 设备可以设置为如下：

```
<vm>
...
```

```

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

```

要从 CDROM 启动，您首先需要插入磁盘，如 [CDROM 服务中所述](#)。然后，可以使用 `the os.boot.devices` 属性从该 CDROM 引导：

```

<vm>
  ...
  <os>
    <boot>
      <devices>
        <device>cdrom</device>
      </devices>
    </boot>
  </os>
</vm>

```

在所有情况下，创建虚拟机的集群的名称或标识符都是必须的。

表 6.896. 参数摘要

名称	类型	方向	概述
auto_pinning_policy	AutoPinning Policy	in	指定是否以及如何应用自动 CPU 和 NUMA 配置。
clone	布尔值	in	指定虚拟机是否应该独立于模板。
clone_permissions	布尔值	in	指定模板的权限是否应复制到虚拟机。
filter	布尔值	in	仅与管理员用户相关。
seal	布尔值	in	指定在创建虚拟机后是否应密封虚拟机。
vm	VM	in/Out	

6.286.1.1. auto_pinning_policy

指定是否以及如何应用自动 CPU 和 NUMA 配置。



重要

由于引擎的 4.5 版已弃用，因此仅保留以向后兼容。以后会将其移除。请使用 POST 后跟 [添加操作](#)。

请求的示例：

```
POST /ovirt-engine/api/vms?auto_pinning_policy=existing/adjust
```

这样的请求正文：

```
<vm>
  <name>myvm</name>
  <template>
    <name>mytemplate</name>
  </template>
  <cluster>
    <name>mycluster</name>
  </cluster>
  <placement_policy>
    <hosts>
      <host>
        <name>myhost</name>
      </host>
    </hosts>
  </placement_policy>
</vm>
```

6.286.1.2. clone

指定虚拟机是否应该独立于模板。

*默认情况下，从模板创建虚拟机时，虚拟机的磁盘取决于模板的磁盘，它们使用写入机制时复制，以便仅模板的不同占用实际存储空间。如果指定此参数且值为 **true**，则创建的虚拟机的磁盘将被克隆并独立于模板。例如，要创建独立虚拟机，请发送如下请求：*

```
POST /ovirt-engine/vms?clone=true
```

这样的请求正文：

```
<vm>
  <name>myvm</name>
```

```

<template>
  <name>mytemplate</name>
</template>
<cluster>
  <name>mycluster</name>
</cluster>
</vm>

```



注意

此参数为 `true` 时，也会复制模板的权限，如使用 `clone_permissions=true` 时。

6.286.1.3. clone_permissions

指定模板的权限是否应复制到虚拟机。

如果提供了此可选参数，其值为 `true`，则模板的权限（仅直接权限，而不是继承的值）将复制到创建的虚拟机。例如，要从 `mytemplate` 模板创建虚拟机复制其权限，请发送类似如下的请求：

```
POST /ovirt-engine/api/vms?clone_permissions=true
```

这样的请求正文：

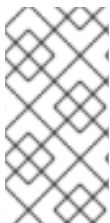
```

<vm>
  <name>myvm</name>
  <template>
    <name>mytemplate</name>
  </template>
  <cluster>
    <name>mycluster</name>
  </cluster>
</vm>

```

6.286.1.4. filter

仅与管理员用户相关。指明是否为此用户在创建的虚拟机上分配 `UserVmManager` 角色。这将允许用户稍后访问虚拟机，就像他是非管理员用户一样，增强其管理员权限（通过提供 `filter=true`）。



注意

admin-as-user（这意味着在现有虚拟机上提供 `filter=true`）POST 请求将失败，除非管理员之前以用户的身份创建了虚拟机（源自 `filter=true`）。

6.286.1.5. seal

指定在创建虚拟机后是否应密封虚拟机。

如果提供此可选参数，其值为 `true`，则虚拟机将在创建后密封。如果值为 `'false'`，则虚拟机不会被密封。如果没有提供该参数，则虚拟机将被密封，只有在从封装的模板中创建它时，其客户机操作系统不会设置为 **Windows**。

例如，要从 `mytemplate` 模板创建虚拟机并封装它，请发送如下请求：

```
POST /ovirt-engine/api/vms?seal=true
```

这样的请求正文：

```
<vm>
  <name>myvm</name>
  <template>
    <name>mytemplate</name>
  </template>
  <cluster>
    <name>mycluster</name>
  </cluster>
</vm>
```

6.286.2. 列出 GET

返回系统虚拟机的列表。

只有 `search` 参数中包含 `sortby` 子句时，才会保证虚拟机返回列表的顺序。

表 6.897. 参数摘要

名称	类型	方向	概述
all_content	布尔值	in	指明是否应将虚拟机的所有属性包含在响应中。
case_sensitive	布尔值	in	指明是否应该在考虑大小写的情况下使用 search 参数执行搜索。
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	<i>指明应遵循哪些内联链接。</i>
max	整数	in	要返回的最大结果数。
ovf_as_ova	布尔值	in	指明结果是否应该公开显示在该虚拟机的 OVA 文件中的 OVF。
search	字符串	in	用于限制返回的虚拟机的查询字符串。
vms	Vm[]	out	

6.286.2.1. all_content

指明是否应将虚拟机的所有属性包含在响应中。

默认情况下排除以下属性：

- **console**
- **initial.configuration.data** - 描述虚拟机的 OVF 文档。
- **rng_source**
- **soundcard**
- **virtio_scsi**

例如，要检索虚拟机的完整表示法，发送如下请求：

```
GET /ovirt-engine/api/vms?all_content=true
```



注意

不包含这些属性的原因是性能：它们很少被使用，需要额外查询数据库。因此，仅当真正需要时才尝试使用此参数。

6.286.2.2. case_sensitive

指明是否应该在考虑大小写的情况下使用 `search` 参数执行搜索。默认值为 `true`，这意味着会考虑这个情况。如果要搜索忽略大小写的操作，将其设置为 `false`。

6.286.2.3. follow

指明应遵循哪些内联链接。 这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.286.2.4. ovf_as_ova

指明结果是否应该公开显示在该虚拟机的 OVA 文件中时的 OVF。描述虚拟机的 OVF 文档。这个参数仅在设置了 `all_content=True` 时才有效。OVF 将显示在 `initial .configuration.data` 中。

例如：

```
GET /vms?all_content=true&ovf_as_ova=true
```

6.287. VNICPROFILE

此服务管理 vNIC 配置集。

表 6.898. 方法概述

名称	概述
get	检索 vNIC 配置集的详细信息。

名称	概述
remove	删除 vNIC 配置集。
update	更新 vNIC 配置集的详情。

6.287.1. 获取 GET

检索 vNIC 配置集的详细信息。

表 6.899. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
profile	VnicProfile	out	

6.287.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.287.2. 删除 DELETE

删除 vNIC 配置集。

表 6.900. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.287.3. 更新 PUT

更新 vNIC 配置集的详情。

表 6.901. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行更新。
profile	VnicProfile	in/Out	正在更新的 vNIC 配置集。

6.288. VNICPROFILES

此服务管理所有 vNIC 配置集的集合。

表 6.902. 方法概述

名称	概述
add	添加 vNIC 配置集。
list	列出所有 vNIC 配置文件。

6.288.1. 添加 POST

添加 vNIC 配置集。

例如，将 vNIC 配置集 123 添加到网络 456 将请求发送到：

```
POST /ovirt-engine/api/networks/456/vnicprofiles
```

使用以下机构：

```
<vnic_profile id="123">
  <name>new_vNIC_name</name>
  <pass_through>
    <mode>disabled</mode>
  </pass_through>
  <port_mirroring>false</port_mirroring>
</vnic_profile>
```

请注意，每个 VNIC 配置集都有一个默认的网络过滤器。有关如何计算默认网络过滤器的更多详细信息，请参阅 [NetworkFilters](#) 中的文档。

**注意**

为外部网络自动创建的 vNIC 配置集将不使用网络过滤器。

创建新 vNIC 配置集的输出取决于给定的正文参数。如果未指定网络过滤器，则将配置默认网络过滤器。例如：

```
<vnic_profile href="/ovirt-engine/api/vnicprofiles/123" id="123">
  <name>new_vNIC_name</name>
  <link href="/ovirt-engine/api/vnicprofiles/123/permissions" rel="permissions"/>
  <pass_through>
    <mode>disabled</mode>
  </pass_through>
  <port_mirroring>false</port_mirroring>
  <network href="/ovirt-engine/api/networks/456" id="456"/>
  <network_filter href="/ovirt-engine/api/networkfilters/789" id="789"/>
</vnic_profile>
```

如果给出了空的网络过滤器，则不会为特定的 vNIC 配置集配置网络过滤器，而不考虑 vNIC 配置集的默认网络过滤器。例如：

```
<vnic_profile>
  <name>no_network_filter</name>
  <network_filter/>
</vnic_profile>
```

如果给出了特定的有效网络过滤器 ID，则 vNIC 配置集将使用给定网络过滤器进行配置，而不考虑 vNIC 配置集的默认网络过滤器。例如：

```
<vnic_profile>
  <name>user_choice_network_filter</name>
  <network_filter id="0000001b-001b-001b-001b-0000000001d5"/>
</vnic_profile>
```

表 6.903. 参数摘要

名称	类型	方向	概述
profile	VnicProfile	in/Out	正在添加的 vNIC 配置集。

6.288.2. 列出 GET

列出所有 vNIC 配置文件。

vNIC 配置集返回列表的顺序不受保证。

表 6.904. 参数摘要

名称	类型	方向	概述
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置要返回的最大配置集数。
profiles	VnicProfile[]	out	所有 vNIC 配置集列表。

6.288.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处](#)。

6.288.2.2. Max

设置要返回的最大配置集数。如果没有指定，则返回所有配置集。

6.289. WEIGHT

表 6.905. 方法概述

名称	概述
get	
remove	

6.289.1. 获取 GET

表 6.906. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
weight	weight	out	

6.289.1.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.289.2. 删除 DELETE

表 6.907. 参数摘要

名称	类型	方向	概述
async	布尔值	in	指明是否应异步执行移除。

6.290. WEIGHT

表 6.908. 方法概述

名称	概述
add	向指定用户定义的调度策略添加权重。
list	返回权重列表。

6.290.1. 添加 POST

向指定用户定义的调度策略添加权重。

表 6.909. 参数摘要

名称	类型	方向	概述
weight	weight	in/Out	

6.290.2. 列出 GET

返回权重列表。

返回的权重列表的顺序不保证。

表 6.910. 参数摘要

名称	类型	方向	概述
filter	布尔值	in	指明是否应根据用户的权限过滤结果。
follow	字符串	in	指明应遵循哪些内联链接。
max	整数	in	设置返回的最大权重数。
weights	Weight[]	out	

6.290.2.1. follow

指明应遵循哪些内联链接。这些链接引用的对象将在当前请求中获取。[详情请查看此处。](#)

6.290.2.2. Max

设置返回的最大权重数。如果没有指定，则返回所有权重。

第 7 章 类型

本节枚举 API 中所有可用的数据类型。

7.1. ACCESSPROTOCOL 枚举

表示 Gluster 卷支持的访问协议。Gluster 和 nfs 默认启用。

表 7.1. 值概述

名称	概述
cifs	CIFS 访问协议.
Gluster	Gluster 访问协议.
nfs	NFS 访问协议.

7.2. ACTION STRUCT

表 7.2. 属性概述

名称	类型	概述
激活	布尔值	
allow_partial_import	布尔值	
async	布尔值	
Attach	DiskAttachment	
authorized_key	AuthorizedKey	
auto_pinning_policy	AutoPinningPolicy	
brick	GlusterBrick[]	
证书	certificate[]	
check_connectivity	布尔值	

名称	类型	概述
clone	布尔值	
clone_permissions	布尔值	
cluster	Cluster	
collapse_snapshots	布尔值	
comment	字符串	包含有关此对象的注释的免费文本。
commit_on_success	布尔值	
连接	StorageConnection	
connectivity_timeout	整数	
correlation_id	字符串	
data_center	DataCenter	
deploy_hosted_engine	布尔值	
description	字符串	以纯文本格式人类可读的描述。
详情	GlusterVolumeProfileDetails	
directory	字符串	
discard_snapshots	布尔值	
discovered_targets	IscsiDetails[]	
disk	Disk	
disk_profile	DiskProfile	
磁盘	disk[]	

名称	类型	概述
专用	布尔值	
fault	故障	
fence_type	字符串	
filename	字符串	
filter	布尔值	
fix_layout	布尔值	
follow	字符串	
force	布尔值	
grace_period	GracePeriod	
host	主机	
id	字符串	唯一标识符。
image	字符串	
image_transfer	ImageTransfer	
import_as_template	布尔值	
is_attached	布尔值	
iscsi	iscsiDetails	
iscsi_targets	string[]	
job	Job	
lease	StorageDomainLease	
logical_units	LogicalUnit[]	
maintenance_after_restart	布尔值	

名称	类型	概述
maintenance_enabled	布尔值	
migrate_vms_in_affinity_closure	布尔值	
modified_bonds	HostNic[]	
modified_labels	NetworkLabel[]	
modified_network_attachments	NetworkAttachment[]	
name	字符串	纯文本中的人类可读名称。
optimize_cpu_settings	布尔值	
选项	选项	
pause	布尔值	
权限	权限	
power_management	PowerManagement	
proxy_ticket	ProxyTicket	
quota	Quota	
reason	字符串	
reassign_bad_macs	布尔值	
reboot	布尔值	
registration_configuration	RegistrationConfiguration	
remote_viewer_connection_file	字符串	
removed_bonds	HostNic[]	

名称	类型	概述
removed_labels	NetworkLabel[]	
removed_network_attachments	NetworkAttachment[]	
resolution_type	字符串	
restore_memory	布尔值	
root_password	字符串	
seal	布尔值	
snapshot	snapshot	
source_host	主机	
ssh	ssh	
status	字符串	
stop_gluster_service	布尔值	
storage_domain	StorageDomain	
storage_domains	StorageDomain[]	
succeeded	布尔值	
synchronized_network_attachments	NetworkAttachment[]	
模板	Template	
ticket	ticket	
timeout	整数	
undeploy_hosted_engine	布尔值	

名称	类型	概述
upgrade_action	ClusterUpgradeAction	
upgrade_percent_complete	整数	
use_cloud_init	布尔值	
use_ignition	布尔值	
use_initialization	布尔值	
use_sysprep	布尔值	
virtual_functions_configuration	HostNicVirtualFunctionsConfiguration	
vm	VM	
vnic_profile_mappings	VnicProfileMapping[]	
volatile	布尔值	

7.3. AFFINITYGROUP STRUCT

关联性组表示一组具有定义关系的虚拟机。

表 7.3. 属性概述

名称	类型	概述
broken	布尔值	指定关联性组是否中断。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
enforcing	布尔值	指定 affinity 组是使用硬的还是软强制来应用到属于该关联性组成员的虚拟机。

名称	类型	概述
hosts_rule	AffinityRule	指定在属于此关联性组成员的虚拟机和主机之间应用的关联性规则。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
positive	布尔值	指定关联性组是对属于该关联性组的成员的虚拟机应用正关联性还是负关联性。
priority	十进制	关联性组的优先级。
vms_rule	AffinityRule	指定应用于属于此关联性组成员的虚拟机的关联性规则。

7.3.1. broken

指定关联性组是否中断。当其任何规则不满足时，关联性组被视为被破坏。**broken** 字段是引擎中的 **computed** 字段。因此，此字段仅适用于 **GET** 请求。

7.3.2. Enforcing

指定 **affinity** 组是使用硬的还是软强制来应用到属于该关联性组成员的虚拟机。



警告

请注意，此属性自引擎版本 4.1 起已弃用，并将在以后的中删除。使用来自现在的 **vms_rule** 属性。

7.3.3. 正

指定关联性组是对属于该关联性组的成员的虚拟机应用正关联性还是负关联性。

**警告**

请注意，此属性自引擎版本 4.1 起已弃用，并将在以后的中删除。使用来自现在的 `vms_rule` 属性。

表 7.4. 链接摘要

名称	类型	概述
<code>cluster</code>	Cluster	对关联性组所应用的群集的引用。
<code>host_labels</code>	AffinityLabel[]	分配给此关联性组的所有主机标签的列表。
主机	Host[]	分配给此关联性组的所有主机的列表。
<code>vm_labels</code>	AffinityLabel[]	分配给此关联性组的所有虚拟机标签的列表。
<code>vms</code>	Vm[]	分配给此关联性组的所有虚拟机的列表。

7.4. AFFINITYLABELSTRUCT

关联性标签可能会影响虚拟机调度。它最常用于从可用的主机创建子集群。

表 7.5. 属性概述

名称	类型	概述
<code>comment</code>	字符串	包含有关此对象的注释的免费文本。
<code>description</code>	字符串	以纯文本格式人类可读的描述。
<code>has_implicit_affinity_group</code>	布尔值	此属性启用标签的旧行为。
<code>id</code>	字符串	唯一标识符。
<code>name</code>	字符串	纯文本中的人类可读名称。
<code>read_only</code>	布尔值	<code>read_only</code> 属性标记无法修改的标签。

7.4.1. has_implicit_affinity_group

此属性启用标签的旧行为。如果为 **true**，该标签也会充当一个正强制虚拟机到主机关联性组。

此参数仅用于兼容版本 4.3 或更低的集群。

7.4.2. read_only

read_only 属性标记无法修改的标签。在列出内部生成的标签时，通常会出现这种情况。

表 7.6. 链接摘要

名称	类型	概述
主机	Host[]	已使用此调度标签标记的主机列表。
vms	Vm[]	已使用此调度标签标记的虚拟机列表。

7.5. AFFINITYRULE STRUCT

关联性组的通用规则定义。每种支持的资源类型（虚拟机、主机）都由单独的规则控制。这允许表达如下规则：定义的虚拟机之间没有关联性，但定义的虚拟机和虚拟主机之间没有紧密关联。

表 7.7. 属性概述

名称	类型	概述
enabled	布尔值	指定关联性组是否使用此规则。
enforcing	布尔值	指定 affinity 组是使用硬的还是软强制机制，应用到此规则控制的资源。
positive	布尔值	指定关联性组是对此规则控制的资源应用正关联性还是负关联。

7.5.1. enabled

指定关联性组是否使用此规则。此属性在创建过程中是可选的，在未提供时被视为 **true**。如果此属性未提供给更新操作，则如果也设置了 **AffinityGroup** 正属性，则被视为 **true**。当缺少已启用和正属性时，启用后端的值将会保留。

7.5.2. Enforcing

指定 **affinity** 组是使用硬的还是软强制机制，应用到此规则控制的资源。如果启用了规则，并且在规则禁用时忽略该参数，则此参数是必需的。

7.5.3. 正

指定关联性组是对此规则控制的资源应用正关联性还是负关联。如果启用了规则，并且在规则禁用时忽略该参数，则此参数是必需的。

7.6. AGENT STRUCT

键入代表隔离代理。

表 7.8. 属性概述

名称	类型	概述
地址	字符串	隔离代理地址。
comment	字符串	包含有关此对象的注释的免费文本。
并发	布尔值	指定代理应同时还是按顺序使用。
description	字符串	以纯文本格式人类可读的描述。
encrypt_options	布尔值	指定是否应加密选项。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
选项	选项[]	隔离代理选项（以键值对分隔的组合）。
订单	整数	此代理的顺序（如果与其他代理一起使用）。
password	字符串	隔离代理密码。
port	整数	隔离代理端口。
type	字符串	隔离代理类型。
username	字符串	隔离代理用户名。

表 7.9. 链接摘要

名称	类型	概述
host	主机	对主机服务的引用.

7.6.1. 主机

对主机服务的引用.每个隔离代理都属于一个主机。

7.7. AGENTCONFIGURATION STRUCT

弃用的代理配置设置.

忽略，因为自 Red Hat Virtualization 4.4.0 起，OpenStack Neutron 代理的部署已被丢弃。OpenStack 主机的部署可以由 Red Hat OpenStack Platform Director 或 TripleO 进行。

表 7.10. 属性概述

名称	类型	概述
地址	字符串	
broker_type	MessageBrokerType	
network_mappings	字符串	不建议使用，因为 Open vSwitch 接口映射由红帽虚拟化 4 起 VDSM 管理。
password	字符串	
port	整数	
username	字符串	

7.7.1. network_mappings

不建议使用，因为 Open vSwitch 接口映射由 Red Hat Virtualization 4.2.0 之后的 VDSM 管理。

7.8. API STRUCT

此类型包含 API 根服务返回的信息。

要获取该信息，请发送类似如下的请求：

GET /ovirt-engine/api

结果如下：

```
<api>
  <link rel="hosts" href="/ovirt-engine/api/hosts"/>
  <link rel="vms" href="/ovirt-engine/api/vms"/>
  ...
  <product_info>
    <name>oVirt Engine</name>
    <vendor>ovirt.org</vendor>
    <version>
      <build>0</build>
      <full_version>4.1.0_master</full_version>
      <major>4</major>
      <minor>1</minor>
      <revision>0</revision>
    </version>
  </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>
  <time>2016-12-12T12:22:25.866+01:00</time>
</api>
```

表 7.11. 属性概述

名称	类型	概述
product_info	ProductInfo	有关产品的信息，如产品名称、供应商名称和版本。
special_objects	SpecialObjects	引用特殊对象，如空白模板和标签层次结构的根目录。
summary	ApiSummary	包含相关对象总数的摘要，如虚拟机、主机和存储域。
time	Date	生成此信息的日期与时间。

表 7.12. 链接摘要

名称	类型	概述
authenticated_user	用户	对经过身份验证的用户的引用。
effective_user	用户	引用有效用户。

7.8.1. authenticated_user

对经过身份验证的用户的引用。

已验证身份的用户是验证其凭据以接受当前请求的用户。在系统的当前版本中，经过身份验证的用户和有效用户始终相同。在未来，当引入对用户模仿的支持时，它们可能会有所不同。

7.8.2. effective_user

引用有效用户。

有效的用户是在当前请求期间应用其权限的用户。在系统的当前版本中，经过身份验证的用户和有效用户始终相同。在未来，当引入对用户模仿的支持时，它们可能会有所不同。

7.9. APISUMMARY STRUCT

包含相关对象总数的摘要，如虚拟机、主机和存储域。

表 7.13. 属性概述

名称	类型	概述
主机	ApiSummaryItem	主机摘要。
storage_domains	ApiSummaryItem	存储域摘要。
用户	ApiSummaryItem	用户摘要。
vms	ApiSummaryItem	虚拟机摘要。

7.10. APISUMMARYITEM STRUCT

此类型包含 API 概述的项。每个项目包含某种对象的总数和活动数量。

表 7.14. 属性概述

名称	类型	概述
Active	整数	活动对象的总数。
总计	整数	对象的总数。

7.11. APPLICATION STRUCT

代表虚拟机上安装的应用。如果您在虚拟机操作系统上部署应用程序，客户机代理会报告应用程序。

要获取该信息，请发送类似如下的请求：

```
GET /ovirt-engine/api/vms/123/applications/456
```

结果如下：

```
<application href="/ovirt-engine/api/vms/123/applications/456" id="456">
  <name>application-test-1.0.0-0.el7</name>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</application>
```

表 7.15. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.16. 链接摘要

名称	类型	概述
vm	VM	对安装应用程序的虚拟机的引用。

7.12. 架构 枚举

表 7.17. 值概述

名称	概述
aarch64	AARCH64 CPU 架构。
ppc64	
s390x	IBM S390X CPU 架构。
undefined	
x86_64	

7.12.1. s390x

IBM S390X CPU 架构。

需要为在 S390X 架构中运行的虚拟机和群集指定。

请注意，S390 通常用于明确描述一般计算机架构或其 31 位变体。S390X 专门用于 64 位体系结构，它与其他架构一致，如 X86_64 或 PPC64。

7.13. AUTHORIZEDKEY STRUCT

表 7.18. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
key	字符串	
name	字符串	纯文本中的人类可读名称。

表 7.19. 链接摘要

名称	类型	概述
user	用户	

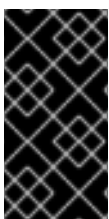
7.14. AUTONUMASTATUS ENUM

表 7.20. 值概述

名称	概述
disable	
启用	
Unknown	

7.15. AUTOPINNINGPOLICY ENUM

输入，键入 代表 CPU 和 NUMA 固定策略是什么。

**重要**

由于引擎的 4.5 版已弃用，因此仅保留以向后兼容。以后会将其移除。请使用 `CpuPinningPolicy`。

表 7.21. 值概述

名称	概述
adjust	CPU 和 NUMA 固定将由专用主机配置。
disabled	不计算 CPU 和 NUMA 固定。
existing	CPU 和 NUMA 固定将由虚拟机当前状态配置。

7.15.1. 调整

CPU 和 NUMA 固定将由专用主机配置。

目前，其含义在于 **CPU 和 NUMA 固定** 将使用专用的主机 CPU 拓扑。虚拟机配置将自动设置为适合主机以获得最高性能。

7.15.2. disabled

不计算 CPU 和 NUMA 固定。

目前，其含义是 **CPU 和 NUMA 固定** 不会计算到当前的虚拟机配置。默认情况下，设置了 1 个套接字、1 个核心和 1 个线程的虚拟机拓扑。

7.15.3. existing

CPU 和 NUMA 固定将由虚拟机当前状态配置。

目前，其含义在于 **CPU 和 NUMA 固定** 将使用提供的虚拟机 CPU 拓扑。如果没有指定 CPU 拓扑，它将使用引擎默认值（带有 1 个插槽、1 个核心和 1 线程的虚拟机拓扑）。

7.16. BACKUP STRUCT

表 7.22. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。

名称	类型	概述
<code>creation_date</code>	Date	备份创建日期。
<code>description</code>	字符串	以纯文本格式人类可读的描述。
<code>from_checkpoint_id</code>	字符串	要启动增量备份的检查点 id。
<code>id</code>	字符串	唯一标识符。
<code>modification_date</code>	Date	备份修改日期。
<code>name</code>	字符串	纯文本中的人类可读名称。
<code>phase</code>	BackupPhase	备份操作的阶段。
<code>to_checkpoint_id</code>	字符串	此备份操作创建的检查点 ID。

7.16.1. to_checkpoint_id

此备份操作创建的检查点 ID。在下一个增量备份中，这个 id 可用作 `fromCheckpointId`。

表 7.23. 链接摘要

名称	类型	概述
磁盘	disk[]	虚拟机备份中包含的磁盘列表。
host	主机	用于启动备份的主机。
snapshot	snapshot	如果备份在使用快照，则引用创建的快照。
vm	VM	对与备份关联的虚拟机的引用。

7.17. BACKUPPHASE ENUM

表 7.24. 值概述

名称	概述
Failed	最后一个阶段，表示备份失败。

名称	概述
挪威	在此阶段，备份调用 'stop_backup' 操作，以完成备份并解锁相关磁盘。
initializing	备份的初始阶段。
Ready	该阶段表示相关磁盘的备份 URL 已就绪，可使用映像传输进行下载。
Starting	该阶段是在 vdsd/libvirt 中调用 'start_backup' 操作前设置的（这意味着应该调用 'stop_backup' 以完成流）。
succeeded	最后一个阶段，表示备份已成功完成。

7.17.1. 初始化

备份的初始阶段。它在实体创建时设置。

7.18. BALANCE STRUCT

表 7.25. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.26. 链接摘要

名称	类型	概述
scheduling_policy	SchedulingPolicy	
scheduling_policy_unit	SchedulingPolicyUnit	

7.19. BIOS STRUCT

表 7.27. 属性概述

名称	类型	概述
<code>boot_menu</code>	<code>BootMenu</code>	
<code>type</code>	<code>BiosType</code>	芯片组和 BIOS 类型组合。

7.20. BIOSTYPE ENUM

键入代表芯片组和 BIOS 类型组合。

表 7.28. 值概述

名称	概述
<code>cluster_default</code>	使用集群范围的默认值。
<code>i440fx_sea_bios</code>	使用 SeaBIOS 的 i440fx 芯片组。
<code>q35_ovmf</code>	使用 OVMF(UEFI)BIOS 的 q35 芯片组。
<code>q35_sea_bios</code>	使用 SeaBIOS 的 q35 芯片组。
<code>q35_secure_boot</code>	启用了 SecureBoot 的 OVMF(UEFI)BIOS 的 q35 芯片组。

7.20.1. cluster_default

使用集群范围的默认值。

这个值不能用于集群。

7.20.2. i440fx_sea_bios

使用 SeaBIOS 的 i440fx 芯片组。

对于非 x86 架构，这是唯一允许的非默认值。

7.21. BLOCKSTATISTICSTRUCT

表 7.29. 属性概述

名称	类型	概述
statistics	Statistic[]	

7.22. BONDING STRUCT

表示网络接口绑定。

表 7.30. 属性概述

名称	类型	概述
ad_partner_mac	MAC	在模式 4 中合作伙伴绑定的 ad_partner_mac 属性。
选项	选项[]	绑定接口的选项元素列表。
slaves	HostNic[]	绑定接口的从 NIC 列表。

7.22.1. ad_partner_mac

在模式 4 中合作伙伴绑定的 **ad_partner_mac** 属性。绑定模式 4 是 802.3ad 标准，也称为动态链路聚合。如需更多信息，请参阅[维基百科](#)和[演示](#)。**ad_partner_mac** 是绑定的另一端系统(**switch**)的 MAC 地址。此参数为只读。设置它对绑定没有影响。它从绑定所在的系统上的 `/sys/class/net/bondX/bonding/ad_partner_mac` 文件检索。

7.22.2. 选项

绑定接口的选项元素列表。每一选项包含属性名称和值属性。仅在添加绑定接口时才需要。

7.22.3. slaves

绑定接口的从 NIC 列表。仅在添加绑定接口时才需要。

表 7.31. 链接摘要

名称	类型	概述
active_slave	HostNic	绑定 (active-backup、 slave 和 balance-tlb) 的 active_slave 属性。

7.22.4. active_slave

绑定 (`active-backup`、`slave` 和 `balance-tlb`) 的 `active_slave` 属性。详情请查看 [Linux 文档](#)。此参数为只读。设置它对绑定没有影响。它从绑定所在的系统中的 `/sys/class/net/bondX/bonding/active_slave` 文件中检索。

例如：

```
GET /ovirt-engine/api/hosts/123/nics/321
```

将响应：

```
<host_nic href="/ovirt-engine/api/hosts/123/nics/321" id="321">
...
<bonding>
  <slaves>
    <host_nic href="/ovirt-engine/api/hosts/123/nics/456" id="456" />
    ...
  </slaves>
  <active_slave href="/ovirt-engine/api/hosts/123/nics/456" id="456" />
</bonding>
...
</host_nic>
```

7.23. BOOKMARK STRUCT

代表系统中的书签。

表 7.32. 属性概述

名称	类型	概述
<code>comment</code>	字符串	包含有关此对象的注释的免费文本。
<code>description</code>	字符串	以纯文本格式人类可读的描述。
<code>id</code>	字符串	唯一标识符。
<code>name</code>	字符串	纯文本中的人类可读名称。
<code>value</code>	字符串	书签值，代表引擎中的搜索。

7.24. BOOT STRUCT

配置虚拟机的启动序列。

表 7.33. 属性概述

名称	类型	概述
devices	BootDevice[]	引导设备的顺序列表。

7.24.1. devices

引导设备的顺序列表。虚拟机将尝试以给定的顺序从给定启动设备启动。

7.25. BOOTDEVICE ENUM

表示虚拟机可从其启动的设备类型。

表 7.34. 值概述

名称	概述
cdrom	从 CD-ROM 启动。
hd	从硬盘驱动器引导。
network	使用 PXE 从网络引导。

7.25.1. cdrom

从 CD-ROM 启动。CD-ROM 可以从 ISO 域中可用的 ISO 文件列表中选择，附加到虚拟机所属的 ata 中心。

7.25.2. network

使用 PXE 从网络引导。必须在虚拟机所连接的网络上配置 [PXE](#)。

7.26. BOOTMENU STRUCT

表示虚拟机的引导菜单配置。

表 7.35. 属性概述

名称	类型	概述
<code>enabled</code>	布尔值	是否为此虚拟机（或模板）启用了引导菜单。

7.27. BOOTPROTOCOL ENUM

定义 NIC 的 IP 地址分配方法的选项。

表 7.36. 值概述

名称	概述
<code>autoconf</code>	无状态地址自动配置.
<code>dhcp</code>	动态主机配置协议.
<code>none</code>	没有地址配置。
<code>poly_dhcp_autoconf</code>	DHCP 和无状态地址自动配置(SLAAC)。
<code>static</code>	静态定义的地址、掩码和网关.

7.27.1. autoconf

无状态地址自动配置.

机制由 [RFC 4862](#) 定义。有关详细信息，请参阅此[维基百科文章](#)。



注意

该值仅适用于 IPv6 地址。

7.27.2. dhcp

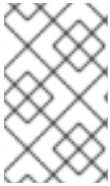
动态主机配置协议.

有关详细信息，请参阅此[维基百科文章](#)。

7.27.3. poly_dhcp_autoconf

DHCP 和无状态地址自动配置(SLAAC)。

SLAAC 机制由 [RFC 4862](#) 定义。如需更多信息，请参阅[无状态地址自动配置](#) 文章和 [DHCP](#) 文章。



注意

该值仅适用于 IPv6 地址。

7.28. BRICKPROFILEDETAIL STRUCT

表 7.37. 属性概述

名称	类型	概述
profile_details	ProfileDetail[]	

表 7.38. 链接摘要

名称	类型	概述
brick	GlusterBrick	

7.29. CDROM STRUCT

表 7.39. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
file	File	
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

名称	类型	概述
----	----	----

表 7.40. 链接摘要

名称	类型	概述
<code>instance_type</code>	<code>InstanceType</code>	(可选) 引用设备使用的实例类型。
模板	<code>Template</code>	(可选) 引用设备使用的模板。
<code>vm</code>	<code>VM</code>	不要使用此元素，而是使用 <code>vms</code> 。
<code>vms</code>	<code>Vm[]</code>	引用使用此设备的虚拟机。

7.29.1. vms

引用使用此设备的虚拟机。个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.30. 证书 结构

表 7.41. 属性概述

名称	类型	概述
<code>comment</code>	字符串	包含有关此对象的注释的免费文本。
<code>content</code>	字符串	
<code>description</code>	字符串	以纯文本格式人类可读的描述。
<code>id</code>	字符串	唯一标识符。
<code>name</code>	字符串	纯文本中的人类可读名称。
机构	字符串	
<code>subject</code>	字符串	

7.31. CHECKPOINT STRUCT

表 7.42. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
creation_date	Date	检查点创建日期。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
parent_id	字符串	父检查点 ID。
状态	CheckpointState	检查点的状态。

表 7.43. 链接摘要

名称	类型	概述
磁盘	disk[]	备份检查点中包含的磁盘列表。
vm	VM	对与检查点关联的虚拟机的引用。

7.32. CHECKPOINTSTATE ENUM

表 7.44. 值概述

名称	概述
Created	检查点的初始状态。
invalid	当检查点无法再用于增量备份并且应删除（例如，提交至旧虚拟机快照后）时，设置的 INVALID 状态。

7.32.1. Created

检查点的初始状态。它在实体创建时设置。

7.33. CLOUDINIT STRUCT

弃用了类型以指定 *cloud-init* 配置。

此类型已弃用，并由 [Initialization](#) 类型内的替代属性替代。详情请查看 [cloud_init](#) 属性文档。

表 7.45. 属性概述

名称	类型	概述
<code>authorized_keys</code>	AuthorizedKey[]	
文件	file[]	
<code>host</code>	主机	
<code>network_configuration</code>	NetworkConfiguration	
<code>regenerate_ssh_keys</code>	布尔值	
<code>timezone</code>	字符串	
用户	user[]	

7.34. CLOUDINITNETWORKPROTOCOL ENUM

定义 cloud-init 协议的值。此协议决定 cloud-init 网络参数在传递到虚拟机之前如何格式化，以便 cloud-init 处理。

支持的协议取决于 cloud-init 版本。[如需更多信息，请参阅网络配置源](#)

表 7.46. 值概述

名称	概述
<code>Eni</code>	传统协议。
<code>openstack_metadata</code>	ENI 协议的后续，支持 IPv6 等等。

7.34.1. Eni

传统协议不支持 IPv6。[如需更多信息，请参阅网络配置 ENI\(Legacy\)](#)

7.34.2. openstack_metadata

ENI 协议的后续, 支持 IPv6 等等。这是默认值。如需更多信息, 请参阅 [API: Proxy neutron 配置到客户机实例](#)

7.35. CLUSTER STRUCT

集群的类型表示法。

集群的 JSON 表示 :

```
{
  "cluster" : [ {
    "ballooning_enabled" : "false",
    "cpu" : {
      "architecture" : "x86_64",
      "type" : "Intel SandyBridge Family"
    },
    "custom_scheduling_policy_properties" : {
      "property" : [ {
        "name" : "HighUtilization",
        "value" : "80"
      }, {
        "name" : "CpuOverCommitDurationMinutes",
        "value" : "2"
      } ]
    },
    "error_handling" : {
      "on_error" : "migrate"
    },
    "fencing_policy" : {
      "enabled" : "true",
      "skip_if_connectivity_broken" : {
        "enabled" : "false",
        "threshold" : "50"
      },
      "skip_if_gluster_bricks_up" : "false",
      "skip_if_gluster_quorum_not_met" : "false",
      "skip_if_sd_active" : {
        "enabled" : "false"
      }
    },
    "gluster_service" : "false",
    "firewall_type" : "iptables",
    "ha_reservation" : "false",
    "ksm" : {
      "enabled" : "true",
      "merge_across_nodes" : "true"
    }
  },
  ],
}
```

```

"memory_policy" : {
  "over_commit" : {
    "percent" : "100"
  },
  "transparent_hugepages" : {
    "enabled" : "true"
  }
},
"migration" : {
  "auto_converge" : "inherit",
  "bandwidth" : {
    "assignment_method" : "auto"
  },
  "compressed" : "inherit",
  "policy" : {
    "id" : "00000000-0000-0000-0000-000000000000"
  }
},
"required_rng_sources" : {
  "required_rng_source" : [ "random" ]
},
"switch_type" : "legacy",
"threads_as_cores" : "false",
"trusted_service" : "false",
"tunnel_migration" : "false",
"version" : {
  "major" : "4",
  "minor" : "1"
},
"virt_service" : "true",
"data_center" : {
  "href" : "/ovirt-engine/api/datacenters/123",
  "id" : "123"
},
"mac_pool" : {
  "href" : "/ovirt-engine/api/macpools/456",
  "id" : "456"
},
"scheduling_policy" : {
  "href" : "/ovirt-engine/api/schedulingpolicies/789",
  "id" : "789"
},
"actions" : {
  "link" : [ {
    "href" : "/ovirt-engine/api/clusters/234/resetemulatedmachine",
    "rel" : "resetemulatedmachine"
  } ]
},
"name" : "Default",
"description" : "The default server cluster",
"href" : "/ovirt-engine/api/clusters/234",
"id" : "234",
"link" : [ {
  "href" : "/ovirt-engine/api/clusters/234/permissions",
  "rel" : "permissions"
} ],
{

```

```

    "href" : "/ovirt-engine/api/clusters/234/cpuprofiles",
    "rel" : "cpuprofiles"
  }, {
    "href" : "/ovirt-engine/api/clusters/234/networkfilters",
    "rel" : "networkfilters"
  }, {
    "href" : "/ovirt-engine/api/clusters/234/networks",
    "rel" : "networks"
  }, {
    "href" : "/ovirt-engine/api/clusters/234/affinitygroups",
    "rel" : "affinitygroups"
  }, {
    "href" : "/ovirt-engine/api/clusters/234/glusterhooks",
    "rel" : "glusterhooks"
  }, {
    "href" : "/ovirt-engine/api/clusters/234/glustervolumes",
    "rel" : "glustervolumes"
  }, {
    "href" : "/ovirt-engine/api/clusters/234/enabledfeatures",
    "rel" : "enabledfeatures"
  }, {
    "href" : "/ovirt-engine/api/clusters/234/externalnetworkproviders",
    "rel" : "externalnetworkproviders"
  }
]
}

```

表 7.47. 属性概述

名称	类型	概述
ballooning_enabled	布尔值	
bios_type	BiosType	芯片组和 BIOS 类型组合。
comment	字符串	包含有关此对象的注释的免费文本。
cpu	cpu	
custom_scheduling_policy_properties	attribute[]	集群的自定义调度策略属性。
description	字符串	以纯文本格式人类可读的描述。
display	显示	
error_handling	ErrorHandling	
fencing_policy	FencingPolicy	可以为群集定义自定义隔离策略。

名称	类型	概述
fips_mode	FipsMode	集群的 FIPS 模式。
firewall_type	FirewallType	此群集中的主机上要使用的防火墙类型。
gluster_service	布尔值	
gluster_tuned_profile	字符串	调优配置集的名称。
ha_reservation	布尔值	
id	字符串	唯一标识符。
ksm	km	
log_max_memory_used_threshold	整数	日志记录审计日志事件的内存消耗阈值。
log_max_memory_used_threshold_type	LogMaxMemoryUsedThresholdType	日志记录审计日志事件的内存消耗阈值类型。
maintenance_reason_required	布尔值	此属性不再相关，并且已弃用。
memory_policy	MemoryPolicy	
迁移	MigrationOptions	
name	字符串	纯文本中的人类可读名称。
optional_reason	布尔值	此属性不再相关，并且已弃用。
required_rng_sources	RngSource[]	来自集群中每个主机的随机数字生成器(RNG)源。
serial_number	SerialNumber	
supported_versions	版本[]	
switch_type	SwitchType	给定集群中所有网络使用的交换机类型。
threads_as_cores	布尔值	

名称	类型	概述
<code>trusted_service</code>	布尔值	
<code>tunnel_migration</code>	布尔值	
<code>upgrade_correlation_id</code>	字符串	升级关联标识符。
<code>upgrade_in_progress</code>	布尔值	表示升级是否已为集群启动。
<code>upgrade_percent_complete</code>	整数	如果升级正在进行，则升级报告为百分比。
<code>version</code>	版本	集群的兼容性版本。
<code>virt_service</code>	布尔值	
<code>vnc_encryption</code>	布尔值	启用 VNC 加密。

7.35.1. bios_type

芯片组和 BIOS 类型组合。

该值用作集群中 `biosType` 设置为 `CLUSTER_DEFAULT` 的所有虚拟机的默认值。

7.35.2. custom_scheduling_policy_properties

集群的自定义调度策略属性。这些可选属性覆盖由 `scheduling_policy` 链接指定的调度策略的属性，并且仅适用于此特定集群。

例如，要更新集群的自定义属性，请发送请求：

```
PUT /ovirt-engine/api/clusters/123
```

通过请求正文：

```
<cluster>
```

```

<custom_scheduling_policy_properties>
  <property>
    <name>HighUtilization</name>
    <value>70</value>
  </property>
</custom_scheduling_policy_properties>
</cluster>

```

使用 `custom_scheduling_policy_properties` 属性更新操作不会更新 `scheduling_policy` 链接指定的调度策略属性，它们只会反映在这个特定的集群中。

7.35.3. fencing_policy

可以为群集定义自定义隔离策略。

例如：

```
PUT /ovirt-engine/api/cluster/123
```

借助此类请求正文：

```

<cluster>
  <fencing_policy>
    <enabled>true</enabled>
    <skip_if_sd_active>
      <enabled>false</enabled>
    </skip_if_sd_active>
    <skip_if_connectivity_broken>
      <enabled>false</enabled>
      <threshold>50</threshold>
    </skip_if_connectivity_broken>
  </fencing_policy>
</cluster>

```

7.35.4. fips_mode

集群的 FIPS 模式。

FIPS 模式代表集群的策略对主机的策略。将检查添加到集群中的主机，以满足集群的 FIPS 模式，使其无法运行（如果它们无法运行）。除非明确提供值，否则默认将新集群初始化为 `UNDEFINED`。这个值会自动更改为添加到集群中的第一个主机的 FIPS 模式。

7.35.5. gluster_tuned_profile

调优配置集的名称。

[在集群中的所有主机上设置的调优配置集](#)。这不是强制要求，并且仅与带有 Gluster 服务的集群相关。

7.35.6. log_max_memory_used_threshold

日志记录审计日志事件的内存消耗阈值。

对于百分比，如果已用内存超过指定的值，则会记录审计日志事件。为绝对值，当可用内存低于以 MB 为单位指定的值时，会记录审计日志事件。

7.35.7. log_max_memory_used_threshold_type

日志记录审计日志事件的内存消耗阈值类型。

您可以在 'percentage' 和 'absolute_value_in_mb' 之间进行选择。

7.35.8. maintenance_reason_required

此属性不再相关，并且已弃用。默认值为 true，

7.35.9. optional_reason

此属性不再相关，并且已弃用。默认值为 true。

7.35.10. required_rng_sources

来自集群中每个主机的随机数字生成器(RNG)源。

读取时，它会返回隐式 urandom（用于集群版本 4.1 及更高版本）或随机值（用于集群版本 4.0 及更低）以及其他所选 RNG 源。编写时，无法删除隐式 RNG 源和随机 RNG 源。

**重要**

在引擎版本 4.1 之前，管理员完全可以控制所需的随机数字生成器；任何源都可以添加或删除，包括随机源。但从版本 4.1 开始，urandom 和 random source 始终是这个集合的一部分，且无法被删除。

**重要**

引擎版本 4.1 引入了一个新的 RNG source urandom，它在兼容版本 4.1 或更高版本的集群中替换随机 RNG 源。

7.35.11. upgrade_correlation_id

升级关联标识符。使用 `upgrade_correlation_id` 将详细说明集群升级的事件与升级本身相关联。

7.35.12. version

集群的兼容性版本。

此集群中的所有主机必须至少支持此兼容性版本。

例如：

```
GET /ovirt-engine/api/clusters/123
```

将通过以下方式做出响应：

```
<cluster>
...
<version>
  <major>4</major>
  <minor>0</minor>
</version>
...
</cluster>
```

要更新兼容性版本，请使用：

PUT /ovirt-engine/api/clusters/123

这样的请求正文：

```
<cluster>
  <version>
    <major>4</major>
    <minor>1</minor>
  </version>
</cluster>
```

要更新集群兼容性版本，集群中的所有主机都必须支持新的兼容性版本。

7.35.13. vnc_encryption

启用 VNC 加密。此属性的默认值为 **false**。

表 7.48. 链接摘要

名称	类型	概述
affinity_groups	AffinityGroup[]	
cpu_profiles	CpuProfile[]	
data_center	DataCenter	
enabled_features	ClusterFeature[]	为集群启用的自定义功能。
external_network_providers	ExternalProvider[]	对集群中可用的外部网络供应商的引用。
gluster_hooks	GlusterHook[]	
gluster_volumes	GlusterVolume[]	
mac_pool	MacPool	对集群使用的 MAC 池的引用。
management_network	Network	
network_filters	NetworkFilter[]	

名称	类型	概述
网络	<code>network[]</code>	
权限	<code>permission[]</code>	
<code>scheduling_policy</code>	<code>SchedulingPolicy</code>	引用此集群使用的默认调度策略。

7.35.14. external_network_providers

对集群中可用的外部网络供应商的引用。

如果支持自动部署外部网络供应商，则引用网络供应商的网络将在集群中的每一主机上可用。只能在添加集群时设置集群的外部网络供应商。在添加主机期间，可以为各个主机覆盖此值。

7.35.15. scheduling_policy

引用此集群使用的默认调度策略。



注意

调度策略属性默认从引用的调度策略获取，但会被此集群的 `custom_scheduling_policy_properties` 属性中指定的属性覆盖。

7.36. CLUSTERFEATURE STRUCT

`type` 代表一个集群级别可用的额外功能。

表 7.49. 属性概述

名称	类型	概述
<code>comment</code>	字符串	包含有关此对象的注释的免费文本。
<code>description</code>	字符串	以纯文本格式人类可读的描述。
<code>id</code>	字符串	唯一标识符。

名称	类型	概述
name	字符串	纯文本中的人类可读名称。

表 7.50. 链接摘要

名称	类型	概述
cluster_level	ClusterLevel	引用集群级别。

7.37. CLUSTERLEVEL STRUCT

描述特定集群级别支持的功能。

表 7.51. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
cpu_types	CpuType[]	此集群级别支持的 CPU 类型。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
Allow	Permit[]	此集群级别的允许。

表 7.52. 链接摘要

名称	类型	概述
cluster_features	ClusterFeature[]	此集群级别支持的其他功能。

7.38. CLUSTERUPGRADEACTION ENUM

集群升级操作的操作类型。

表 7.53. 值概述

名称	概述
finish	通过将集群的 <code>upgrade_running</code> 标记标记为 <code>false</code> 来传递升级操作来完成集群升级过程。
start	通过将集群的 <code>upgrade_running</code> 标记标记为 <code>true</code> 来传递用于启动集群升级过程的升级操作。
update_progress	要传递的升级操作来更新集群升级进度。

7.38.1. finish

通过将集群的 `upgrade_running` 标记标记为 `false` 来传递升级操作来完成集群升级过程。这应该在集群升级过程结束时使用。

7.38.2. start

通过将集群的 `upgrade_running` 标记标记为 `true` 来传递用于启动集群升级过程的升级操作。这应该在集群升级过程开始时使用。

7.38.3. update_progress

要传递的升级操作来更新集群升级进度。这应该象升级一样使用。

7.39. CONFIGURATION STRUCT

表 7.54. 属性概述

名称	类型	概述
data	字符串	描述虚拟机的文档。
type	<code>ConfigurationType</code>	

7.39.1. data

描述虚拟机的文档。

OVF 文档示例：

```

<?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>fd6c627c-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>
    <IsInitialized>false</IsInitialized>
    <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 Memory</Info>
      <System>
        <vssd:VirtualSystemType>ENGINE 3.5.0.0</vssd:VirtualSystemType>
      </System>
    </Item>
  </Content>
</Envelope>

```

```

<rasd:Caption>1 virtual cpu</rasd:Caption>
<rasd:Description>Number of virtual CPU</rasd:Description>
<rasd:Instanceld>1</rasd:Instanceld>
<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:Instanceld>2</rasd:Instanceld>
<rasd:ResourceType>4</rasd:ResourceType>
<rasd:AllocationUnits>MegaBytes</rasd:AllocationUnits>
<rasd:VirtualQuantity>1024</rasd:VirtualQuantity>
</Item>
<Item>
<rasd:Caption>USB Controller</rasd:Caption>
<rasd:Instanceld>3</rasd:Instanceld>
<rasd:ResourceType>23</rasd:ResourceType>
<rasd:UsbPolicy>DISABLED</rasd:UsbPolicy>
</Item>
</Section>
</Content>
</ovf:Envelope>

```

7.40. CONFIGURATIONTYPE ENUM

配置格式类型.

表 7.55. 值概述

名称	概述
OVA	标准 OVF 类型的配置类型.
ovf	兼容 oVirt 的 OVF 类型的 ConfigurationType.

7.40.1. OVA

标准 OVF 类型的配置类型.

提供的虚拟机配置符合开放虚拟化格式(OVF)标准。此值应该用于 OVF 配置，该配置从 oVirt 或其他供应商生成的 Open Virtual Appliance(OVA)中提取。[请参阅此处了解 OVF 规格。](#)

7.40.2. obf

兼容 oVirt 的 OVF 类型的 ConfigurationType.

提供的虚拟机配置符合 Open Virtualization Format(OVF)的 oVirt 兼容形式。请注意，OVF 的 oVirt 兼容形式可能与其他供应商使用的 OVF 标准有所不同。此值应当用于从存储域获取的 OVF 配置。

7.41. CONSOLE STRUCT

串行控制台设备的表示。

表 7.56. 属性概述

名称	类型	概述
enabled	布尔值	启用/禁用串行控制台设备。

7.42. CORE STRUCT

表 7.57. 属性概述

名称	类型	概述
index	整数	
socket	整数	

7.43. CPU STRUCT

表 7.58. 属性概述

名称	类型	概述
架构	构架	
内核	core[]	
cpu_tune	CpuTune	
level	整数	
mode	CpuMode	
name	字符串	

名称	类型	概述
速度	十进制	
topology	CpuTopology	
type	字符串	

7.44. CPUMODE ENUM

表 7.59. 值概述

名称	概述
custom	
host_model	
host_passthrough	

7.45. CPUPINNINGPOLICY ENUM

类型代表 CPU 和 NUMA 固定策略。

表 7.60. 值概述

名称	概述
专用	虚拟机启动时，引擎将自动计算 CPU 固定，并在虚拟机停止时丢弃。
Manual	CPU 固定将被手动配置。
none	不会配置 CPU 固定。
resize_and_pin_numa	CPU 和 NUMA 固定将由专用主机配置。

7.45.1. 专用

虚拟机启动时，引擎将自动计算 CPU 固定，并在虚拟机停止时丢弃。

固定是独占的，意味着其他任何虚拟机都无法使用固定物理 CPU。

7.45.2. Manual

CPU 固定将被手动配置。

目前，这意味着 CPU 固定将手动配置至当前虚拟机配置。虚拟机需要固定到至少一个主机。Pinning 在 CPU 配置中提供，使用 CpuTune。

7.45.3. none

不会配置 CPU 固定。

目前，这意味着 CPU 固定不会配置为当前虚拟机配置。默认情况下，虚拟机拓扑被设置为 1 个插槽、1 核和 1 Thread。

7.45.4. resize_and_pin_numa

CPU 和 NUMA 固定将由专用主机配置。

CPU 和 NUMA 固定将使用专用的主机 CPU 拓扑。虚拟机配置将自动设置为适合主机以获得最高性能。

7.46. CPUPROFILE STRUCT

表 7.61. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.62. 链接摘要

名称	类型	概述
cluster	Cluster	
权限	permission[]	
qos	QoS	

7.47. CPUTOPOLOGY STRUCT

表 7.63. 属性概述

名称	类型	概述
内核	整数	
套接字	整数	
线程	整数	

7.48. CPUTUNE STRUCT

表 7.64. 属性概述

名称	类型	概述
vcpu_pins	VcpuPin[]	

7.49. CPUTYPE STRUCT

描述支持的 CPU 类型。

表 7.65. 属性概述

名称	类型	概述
架构	构架	CPU 架构。
level	整数	CPU 类型的级别。
name	字符串	CPU 类型的名称，如 Intel Nehalem Family 。

7.50. CREATIONSTATUS ENUM

表 7.66. 值概述

名称	概述
complete	
Failed	
in_progress	
待处理	

7.51. CUSTOMPROPERTY STRUCT

自定义属性表示。

表 7.67. 属性概述

名称	类型	概述
name	字符串	属性名称。
regexp	字符串	定义自定义属性可获得的可用值的正则表达式。
value	字符串	属性值。

7.52. DATACENTER STRUCT

表 7.68. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
local	布尔值	
name	字符串	纯文本中的人类可读名称。
quota_mode	QuotaModeType	
status	DataCenterStatus	

名称	类型	概述
storage_format	StorageFormat	
supported_versions	版本[]	
version	版本	数据中心的兼容性版本。

7.52.1. version

数据中心的兼容性版本。

此数据中心中的所有集群必须已设置为至少此兼容性版本。

例如：

```
GET /ovirt-engine/api/datacenters/123
```

将响应：

```
<data_center>
...
<version>
  <major>4</major>
  <minor>0</minor>
</version>
...
</data_center>
```

要更新兼容性版本，请使用：

```
PUT /ovirt-engine/api/datacenters/123
```

通过请求正文：

```
<data_center>
<version>
  <major>4</major>
```

```

<minor>1</minor>
</version>
</data_center>

```

表 7.69. 链接摘要

名称	类型	概述
clusters	cluster[]	引用此数据中心内的集群。
iscsi_bonds	IscsiBond[]	参考此数据中心使用的 iSCSI 绑定。
mac_pool	MacPool	引用此数据中心使用的 MAC 池。
网络	network[]	参考附加到此数据中心的网络。
权限	permission[]	引用分配给此数据中心的权限。
qoss	Qos[]	参考此数据中心所用服务质量。
quotas	Quota[]	引用分配到此数据中心的配额。
storage_domains	StorageDomain[]	引用附加到此数据中心的存储域。

7.53. DATACENTERSTATUS ENUM

表 7.70. 值概述

名称	概述
contend	
维护	
not_operational	
problematic	
未初始化	
up	

7.54. DEVICE STRUCT

设备将链接到设备的潜在父项。

表 7.71. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.72. 链接摘要

名称	类型	概述
instance_type	InstanceType	(可选) 引用设备使用的实例类型。
模板	Template	(可选) 引用设备使用的模板。
vm	VM	不要使用此元素，而是使用 vms 。
vms	Vm[]	引用使用此设备的虚拟机。

7.54.1. vms

引用使用此设备的虚拟机。个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.55. DISK STRUCT

表示虚拟磁盘设备。

表 7.73. 属性概述

名称	类型	概述
Active	布尔值	指示该磁盘是否对虚拟机可见。
actual_size	整数	磁盘的实际大小，以字节为单位。
Alias	字符串	
backup	DiskBackup	磁盘支持的备份行为。

名称	类型	概述
backup_mode	DiskBackupMode	磁盘备份的类型（full/递增）仅在磁盘备份进行时可见。
bootable	布尔值	表示磁盘是否已标记为可引导。
comment	字符串	包含有关此对象的注释的免费文本。
content_type	DiskContentType	指明磁盘上的实际内容。
description	字符串	以纯文本格式人类可读的描述。
external_disk	字符串	使用外部磁盘。
format	DiskFormat	底层存储格式。
id	字符串	唯一标识符。
image_id	字符串	
initial_size	整数	在块存储上创建的稀疏镜像磁盘的初始大小，以字节为单位。
Interface	DiskInterface	用于将磁盘设备连接到虚拟机的接口驱动程序类型。
logical_name	字符串	
lun_storage	HostStorage	
name	字符串	纯文本中的人类可读名称。
propagate_errors	布尔值	指明磁盘错误是否应该导致虚拟机暂停，或者是否应将磁盘错误传播到客户机操作系统。
provisioned_size	整数	磁盘的虚拟大小，以字节为单位。
qcow_version	QcowVersion	QCOW 卷的底层 QCOW 版本。
read_only	布尔值	指明磁盘是否处于只读模式。
sgio	ScsiGenericIO	指明 SCSI passthrough 是否启用及其策略。
共享	布尔值	指明该磁盘是否可以附加到多个虚拟机。
稀疏	布尔值	指明是否应预分配磁盘的物理存储。
status	DiskStatus	磁盘设备的状态。

名称	类型	概述
<code>storage_type</code>	<code>DiskStorageType</code>	
<code>total_size</code>	整数	包括所有快照的磁盘总大小，以字节为单位。
<code>uses_scsi_reservation</code>	布尔值	
<code>wipe_after_delete</code>	布尔值	表示磁盘的块是否会在删除后以零形式读取： - 在块存储上，磁盘将被零，然后仅被删除。

7.55.1. Active

指示该磁盘是否对虚拟机可见。



重要

在向虚拟机添加磁盘附加时，如果服务器接受不包含此属性的请求，则结果未定义。在某些情况下，磁盘将被自动激活，在其他情况下则不会激活。为避免出现问题，强烈建议始终使用所需的值包含此属性。

7.55.2. actual_size

磁盘的实际大小，以字节为单位。

实际大小是磁盘实际使用的字节数。它将小于使用 `cow` 格式的磁盘的调配大小。

7.55.3. Bootable

表示磁盘是否已标记为可引导。



重要

此属性仅适用于实际连接到虚拟机的磁盘，在 API 版本 4 中，它已移至 `DiskAttachment` 类型。它保留在此仅用于向后兼容，将来会删除。

7.55.4. external_disk

使用外部磁盘。

外部磁盘可以是到本地文件或块设备的路径，也可以是 QEMU 支持的 URL，例如：

- `nbd:<host>:<port>[:exportname=<export>]`
- `nbd:unix:</path>[:exportname=<export>]`
- `http://[<username>[:<password>]@]<host>/<path>`
- `https://[<username>[:<password>]@]<host>/<path>`
- `ftp://[<username>[:<password>]@]<host>/<path>`
- `ftps://[<username>[:<password>]@]<host>/<path>`

有关其他支持的协议和更多信息，请参阅 QEMU 手册。

7.55.5. initial_size

在块存储上创建的稀疏镜像磁盘的初始大小，以字节为单位。

初始大小是在块存储上创建时首先使用分配稀疏磁盘的字节数。初始大小将小于置备的大小。如果没有指定，系统将使用的默认初始大小将被分配。

7.55.6. Interface

用于将磁盘设备连接到虚拟机的接口驱动程序类型。



重要

此属性仅适用于实际连接到虚拟机的磁盘，在 API 版本 4 中，它已移至 **DiskAttachment** 类型。它保留在此仅用于向后兼容，将来会删除。

7.55.7. provisioned_size

磁盘的虚拟大小，以字节为单位。

此属性在创建新磁盘时是必需的。

7.55.8. qcow_version

QCOW 卷的底层 QCOW 版本。QCOW 版本将卷支持的 qemu 版本指定为 qemu 版本。该字段可使用更新 API 更新，并且仅针对 QCOW 卷报告。由在其上创建磁盘的存储域版本决定。版本低于 V4 的存储域支持 QCOW2 卷。V4 存储域也支持 QCOW2v3。有关不同 QCOW 版本功能的更多信息，请参阅[这里](#)。

7.55.9. read_only

指明磁盘是否处于只读模式。

由于 API 中未显示此属性，并且已移到 **DiskAttachment**。

自 Red Hat Virtualization Manager 版本 4.1.2 起，此属性已弃用，并将在以后的中删除。若要以只读模式附加磁盘，请使用 **DiskAttachment** 类型的 `read_only` 属性。例如：

```
POST /ovirt-engine/api/vms/123/diskattachments
```

```
<disk_attachment>
  <read_only>true</read_only>
  ...
</disk_attachment>
```

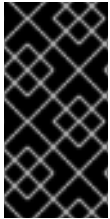
7.55.10. sgio

指明 SCSI passthrough 是否启用及其策略。

设置过滤的 `/unfiltered` 的值将为带有非特权/特权 SCSI I/O 的 LUN 磁盘启用 SCSI 直通。要禁用 SCSI passthrough, 该值应设置为禁用

7.55.11. 共享

指明该磁盘是否可以附加到多个虚拟机。



重要

当磁盘附加到多个虚拟机时, 这些虚拟机的客户机操作系统负责协调对其的访问, 以避免数据损坏, 例如使用诸如 [GlusterFS](#) 或 [GFS](#) 的共享文件系统。

7.55.12. total_size

包括所有快照的磁盘总大小, 以字节为单位。

总大小是磁盘实际使用的字节数加上其快照的大小。它不会被填充为直接 LUN 和 Cinder 磁盘。对于没有快照的磁盘, 其总大小等于实际大小。

7.55.13. wipe_after_delete

表示磁盘的块是否会在删除后以零形式读取：

- 在块存储上, 磁盘将被零, 然后仅被删除。
- 在文件存储上, 由于文件系统已保证之前删除的块被写回为零, 因此该磁盘将立即被删除。

表 7.74. 链接摘要

名称	类型	概述
<code>disk_profile</code>	DiskProfile	
<code>disk_snapshots</code>	DiskSnapshot[]	
<code>instance_type</code>	InstanceType	(可选) 引用设备使用的实例类型。

名称	类型	概述
openstack_volume_type	OpenStackVolumeType	
权限	permission[]	
quota	Quota	
snapshot	snapshot	
statistics	Statistic[]	磁盘公开的统计数据。
storage_domain	StorageDomain	
storage_domains	StorageDomain[]	与此磁盘关联的存储域。
模板	Template	(可选) 引用设备使用的模板。
vm	VM	不要使用此元素，而是使用 vms 。
vms	Vm[]	引用使用此设备的虚拟机。

7.55.14. statistics

磁盘公开的统计数据。例如：

```
<statistics>
  <statistic href="/ovirt-engine/api/disks/123/statistics/456" id="456">
    <name>data.current.read</name>
    <description>Read data rate</description>
    <kind>gauge</kind>
    <type>decimal</type>
    <unit>bytes_per_second</unit>
    <values>
      <value>
        <datum>1052</datum>
      </value>
    </values>
    <disk href="/ovirt-engine/api/disks/123" id="123"/>
  </statistic>
  ...
</statistics>
```

这些统计数据在检索磁盘时不直接包含，而仅包含一个链接。要获取统计数据，请使用包括的链接：

GET /ovirt-engine/api/disks/123/statistics

7.55.15. storage_domains

与此磁盘关联的存储域。



注意

仅当第一个磁盘添加到本身不是从模板创建的虚拟机时才需要。

7.55.16. vms

引用使用此设备的虚拟机。个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.56. DISKATTACHMENT STRUCT

描述如何将磁盘附加到虚拟机。

表 7.75. 属性概述

名称	类型	概述
Active	布尔值	定义磁盘是否在其所附加的虚拟机中处于活动状态。
bootable	布尔值	定义磁盘是否可引导。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
Interface	DiskInterface	用于将磁盘设备连接到虚拟机的接口驱动程序类型。
logical_name	字符串	虚拟机磁盘的逻辑名称，如虚拟机内部所示。
name	字符串	纯文本中的人类可读名称。

名称	类型	概述
<code>pass_discard</code>	布尔值	定义虚拟机是否将丢弃命令传递给存储。
<code>read_only</code>	布尔值	指明磁盘是否以只读方式连接到虚拟机。
<code>uses_scsi_reservation</code>	布尔值	定义是否为此磁盘启用了 SCSI 保留。

7.56.1. Active

定义磁盘是否在其所附加的虚拟机中处于活动状态。

附加到处于活动状态的虚拟机的磁盘在运行时连接到虚拟机，并可使用。

7.56.2. logical_name

虚拟机磁盘的逻辑名称，如虚拟机内部所示。

只有在虚拟机中安装客户机代理并运行时，才会报告磁盘的逻辑名称。

例如，如果客户机操作系统是 Linux 且磁盘通过 VirtIO 接口连接，则逻辑名称将报告为 `/dev/vda`：

```
<disk_attachment>
...
<logical_name>/dev/vda</logical_name>
</disk_attachment>
```

如果客户机操作系统为 Windows，则逻辑名称将报告为 `\\.\PHYSICALDRIVE0`。

7.56.3. read_only

指明磁盘是否以只读方式连接到虚拟机。

添加新磁盘附加时，默认值为 `false`。

```
<disk_attachment>
...
<read_only>true</read_only>
</disk_attachment>
```

7.56.4. uses_scsi_reservation

定义是否为此磁盘启用了 SCSI 保留。

启用 VIRTIO-SCSI 直通的虚拟机可以在磁盘上设置持久的 SCSI 预留。如果他们设置了持久的 SCSI 预留，则这些虚拟机无法迁移到其他主机，因为它们将失去对磁盘的访问权限，因为 SCSI 保留特定于 SCSI 启动器，因此也特定于主机。无法自动检测到这种情况。为避免迁移这些虚拟机，用户可以将此属性设置为 `true`，以指示虚拟机正在使用 SCSI 预留。

表 7.76. 链接摘要

名称	类型	概述
<code>disk</code>	Disk	对磁盘的引用。
模板	Template	对模板的引用。
<code>vm</code>	VM	对虚拟机的引用。

7.57. DISKBACKUP 枚举

表示磁盘上启用的备份机制的枚举。

表 7.77. 值概述

名称	概述
<code>incremental</code>	增量备份支持。
<code>none</code>	无备份支持。

7.58. DISKBACKUPMODE ENUM

表示备份模式的枚举

表 7.78. 值概述

名称	概述
full	此磁盘支持完整备份。
incremental	此磁盘支持增量备份。

7.58.1. full

此磁盘支持完整备份。您可以查询零扩展并下载所有磁盘数据。

7.58.2. incremental

此磁盘支持增量备份。您可以查询脏扩展并下载更改的块。

7.59. DISKCONTENTTYPE ENUM

驻留于磁盘上的实际内容。

表 7.79. 值概述

名称	概述
backup_scratch	磁盘包含受保护的虚拟机备份数据。
data	磁盘包含数据。
hosted_engine	磁盘包含托管引擎虚拟机磁盘。
hosted_engine_configuration	磁盘包含托管引擎配置磁盘。
hosted_engine_metadata	磁盘包含托管引擎元数据磁盘。
hosted_engine_sanlock	磁盘包含托管引擎 Sanlock 磁盘。
iso	磁盘包含一个 ISO 映像，供 CDROM 设备使用。
memory_dump_volume	磁盘包含来自实时快照的内存转储。

名称	概述
memory_metad ata_volume	磁盘包含来自实时快照的内存元数据。
ovf_store	磁盘是 OVF 存储。

7.60. DISKFORMAT ENUM

磁盘的底层存储格式。

表 7.80. 值概述

名称	概述
cow	Copy On Write 格式允许快照，其性能开销较小。
Raw	raw 格式不允许快照，但提供了更高的性能。

7.61. DISKINTERFACE ENUM

磁盘与控制器通信的底层存储接口。

表 7.81. 值概述

名称	概述
IDE	传统控制器设备。
SATA	SATA 控制器节点。
spapr_vscsi	IBM pSeries 系列计算机支持的半虚拟化设备，使用 SCSI 协议。
VirtIO	虚拟化接口，只有虚拟客户机的设备驱动程序知道该接口正在虚拟环境中运行。
virtio_scsi	半虚拟化 SCSI 控制器节点。

7.61.1. IDE

传统控制器设备几乎与所有 **guest** 操作系统配合使用，因此非常适合兼容性。性能低于其它替代方案。

7.61.2. VirtIO

虚拟化接口，只有虚拟客户机的设备驱动程序知道该接口正在虚拟环境中运行。使虚拟机能够获取高性能的磁盘操作。

7.61.3. virtio_scsi

半虚拟化 SCSI 控制器节点.使用 SCSI 协议通过直接物理存储设备地址快速与客户机接口。

7.62. DISKPROFILE STRUCT

表 7.82. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.83. 链接摘要

名称	类型	概述
权限	permission[]	
qos	QoS	
storage_domain	StorageDomain	

7.63. DISKSNAPSHOT STRUCT

表 7.84. 属性概述

名称	类型	概述
Active	布尔值	指示该磁盘是否对虚拟机可见。

名称	类型	概述
actual_size	整数	磁盘的实际大小，以字节为单位。
Alias	字符串	
backup	DiskBackup	磁盘支持的备份行为。
backup_mode	DiskBackupMode	磁盘备份的类型（full/递增）仅在磁盘备份进行时可见。
bootable	布尔值	表示磁盘是否已标记为可引导。
comment	字符串	包含有关此对象的注释的免费文本。
content_type	DiskContentType	指明磁盘上的实际内容。
description	字符串	以纯文本格式人类可读的描述。
external_disk	字符串	使用外部磁盘。
format	DiskFormat	底层存储格式。
id	字符串	唯一标识符。
image_id	字符串	
initial_size	整数	在块存储上创建的稀疏镜像磁盘的初始大小，以字节为单位。
Interface	DiskInterface	用于将磁盘设备连接到虚拟机的接口驱动程序类型。
logical_name	字符串	
lun_storage	HostStorage	
name	字符串	纯文本中的人类可读名称。
propagate_errors	布尔值	指明磁盘错误是否应该导致虚拟机暂停，或者是否应将磁盘错误传播到客户机操作系统。
provisioned_size	整数	磁盘的虚拟大小，以字节为单位。
qcow_version	QcowVersion	QCOW 卷的底层 QCOW 版本。
read_only	布尔值	指明磁盘是否处于只读模式。
sgio	ScsiGenericIO	指明 SCSI passthrough 是否启用及其策略。

名称	类型	概述
共享	布尔值	指明该磁盘是否可以附加到多个虚拟机。
稀疏	布尔值	指明是否应预分配磁盘的物理存储。
status	DiskStatus	磁盘设备的状态。
storage_type	DiskStorageType	
total_size	整数	包括所有快照的磁盘总大小，以字节为单位。
uses_scsi_reservation	布尔值	
wipe_after_delete	布尔值	表示磁盘的块是否会在删除后以零形式读取： - 在块存储上，磁盘将被零，然后仅被删除。

7.63.1. Active

指示该磁盘是否对虚拟机可见。



重要

在向虚拟机添加磁盘附加时，如果服务器接受不包含此属性的请求，则结果未定义。在某些情况下，磁盘将被自动激活，在其他情况下则不会激活。为避免出现问题，强烈建议始终使用所需的值包含此属性。

7.63.2. actual_size

磁盘的实际大小，以字节为单位。

实际大小是磁盘实际使用的字节数。它将小于使用 **cow** 格式的磁盘的调配大小。

7.63.3. Bootable

表示磁盘是否已标记为可引导。

**重要**

此属性仅适用于实际连接到虚拟机的磁盘，在 API 版本 4 中，它已移至 **DiskAttachment** 类型。它保留在此仅用于向后兼容，将来会删除。

7.63.4. external_disk

使用外部磁盘。

外部磁盘可以是到本地文件或块设备的路径，也可以是 QEMU 支持的 URL，例如：

- `nbd:<host>:<port>[:exportname=<export>]`
- `nbd:unix:</path>[:exportname=<export>]`
- `http://[<username>[:<password>]@]<host>/<path>`
- `https://[<username>[:<password>]@]<host>/<path>`
- `ftp://[<username>[:<password>]@]<host>/<path>`
- `ftps://[<username>[:<password>]@]<host>/<path>`

有关其他支持的协议和更多信息，请参阅 QEMU 手册。

7.63.5. initial_size

在块存储上创建的稀疏镜像磁盘的初始大小，以字节为单位。

初始大小是在块存储上创建时首先使用分配稀疏磁盘的字节数。初始大小将小于置备的大小。如果没有指定，系统将使用的默认初始大小将被分配。

7.63.6. Interface

用于将磁盘设备连接到虚拟机的接口驱动程序类型。



重要

此属性仅适用于实际连接到虚拟机的磁盘，在 API 版本 4 中，它已移至 **DiskAttachment** 类型。它保留在此仅用于向后兼容，将来会删除。

7.63.7. provisioned_size

磁盘的虚拟大小，以字节为单位。

此属性在创建新磁盘时是必需的。

7.63.8. qcow_version

QCOW 卷的底层 QCOW 版本。QCOW 版本将卷支持的 qemu 版本指定为 qemu 版本。该字段可使用更新 API 更新，并且仅针对 QCOW 卷报告。由在其上创建磁盘的存储域版本决定。版本低于 V4 的存储域支持 QCOW2 卷。V4 存储域也支持 QCOW2v3。有关不同 QCOW 版本功能的更多信息，请参阅[这里](#)。

7.63.9. read_only

指明磁盘是否处于只读模式。

由于 API 中未显示此属性，并且已移到 **DiskAttachment**。

自 Red Hat Virtualization Manager 版本 4.1.2 起，此属性已弃用，并将在以后的中删除。若要以只读模式附加磁盘，请使用 **DiskAttachment** 类型的 **read_only** 属性。例如：

```
POST /ovirt-engine/api/vms/123/diskattachments
```

```
<disk_attachment>
  <read_only>true</read_only>
  ...
</disk_attachment>
```

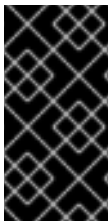
7.63.10. sgio

指明 SCSI passthrough 是否启用及其策略。

设置过滤的 `/unfiltered` 的值将为带有非特权/特权 SCSI I/O 的 LUN 磁盘启用 SCSI 直通。要禁用 SCSI passthrough, 该值应设置为禁用

7.63.11. 共享

指明该磁盘是否可以附加到多个虚拟机。



重要

当磁盘附加到多个虚拟机时, 这些虚拟机的客户机操作系统负责协调对其的访问, 以避免数据损坏, 例如使用诸如 [GlusterFS](#) 或 [GFS](#) 的共享文件系统。

7.63.12. total_size

包括所有快照的磁盘总大小, 以字节为单位。

总大小是磁盘实际使用的字节数加上其快照的大小。它不会被填充为直接 LUN 和 Cinder 磁盘。对于没有快照的磁盘, 其总大小等于实际大小。

7.63.13. wipe_after_delete

表示磁盘的块是否会在删除后以零形式读取:

- 在块存储上, 磁盘将被零, 然后仅被删除。
- 在文件存储上, 由于文件系统已保证之前删除的块被写回为零, 因此该磁盘将立即被删除。

表 7.85. 链接摘要

名称	类型	概述
disk	Disk	
disk_profile	DiskProfile	
disk_snapshots	DiskSnapshot[]	
instance_type	InstanceType	(可选) 引用设备使用的实例类型。
openstack_volume_type	OpenStackVolumeType	
父级	DiskSnapshot	父磁盘快照。
权限	permission[]	
quota	Quota	
snapshot	snapshot	
statistics	Statistic[]	磁盘公开的统计数据。
storage_domain	StorageDomain	
storage_domains	StorageDomain[]	与此磁盘关联的存储域。
模板	Template	(可选) 引用设备使用的模板。
vm	VM	不要使用此元素，而是使用 vms 。
vms	Vm[]	引用使用此设备的虚拟机。

7.63.14. statistics

磁盘公开的统计数据。例如：

```
<statistics>
  <statistic href="/ovirt-engine/api/disks/123/statistics/456" id="456">
    <name>data.current.read</name>
    <description>Read data rate</description>
    <kind>gauge</kind>
    <type>decimal</type>
    <unit>bytes_per_second</unit>
    <values>
      <value>
```

```

    <datum>1052</datum>
  </value>
</values>
  <disk href="/ovirt-engine/api/disks/123" id="123"/>
</statistic>
...
</statistics>

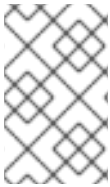
```

这些统计数据在检索磁盘时不直接包含，而仅包含一个链接。要获取统计数据，请使用包括的链接：

GET /ovirt-engine/api/disks/123/statistics

7.63.15. storage_domains

与此磁盘关联的存储域。



注意

仅当第一个磁盘添加到本身不是从模板创建的虚拟机时才需要。

7.63.16. vms

引用使用此设备的虚拟机。个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.64. DISKSTATUS ENUM

磁盘的当前状态表示法。

表 7.86. 值概述

名称	概述
illegal	虚拟机无法访问磁盘，用户也需要采取措施来解决问题。
Locked	该磁盘正被系统使用，因此此时虚拟机无法访问它。
确定	磁盘状态为正常，可由虚拟机访问。

7.64.1. Locked

该磁盘正被系统使用，因此此时虚拟机无法访问它。这通常是临时状态，直到磁盘被释放为止。

7.65. DISKSTORAGETYPE ENUM

表 7.87. 值概述

名称	概述
cinder	
image	
lun	
managed_block_storage	存储类型，用于使用 cinderlib 驱动程序创建的存储域。

7.66. DISKTYPE ENUM

表 7.88. 值概述

名称	概述
data	
system	

7.67. 显示 结构

表示图形控制台配置。

表 7.89. 属性概述

名称	类型	概述
地址	字符串	要连接图形控制台客户端的客户机的 IP 地址。
allow_override	布尔值	表示是否覆盖每个主机的显示地址。
certificate	证书	TLS 证书，以防出现 TLS 连接。
copy_paste_enabled	布尔值	指明用户是否能够从外部主机复制内容并将其粘贴到图形控制台中。

名称	类型	概述
disconnect_action	字符串	返回图形控制台断开连接时将执行的操作。
disconnect_action_delay	整数	在图形控制台断开连接操作执行前的延迟（分钟）。
file_transfer_enabled	布尔值	指明用户是否能够将文件从外部主机拖放到图形控制台中。
keyboard_layout	字符串	用于此图形控制台的键盘布局。
monitor	整数	为此图形控制台打开的显示器数量。
port	整数	guest 上的端口地址，用于连接图形控制台客户端。
proxy	字符串	图形控制台客户端将用于连接 guest 的代理 IP。
secure_port	整数	如果使用 TLS，客户机上的安全端口地址将图形控制台客户端连接到。
single_qxl_pci	布尔值	引擎现在根据操作系统自动设置它。
smartcard_enabled	布尔值	指明是否使用智能卡验证。
type	DisplayType	图形控制台协议类型。

7.67.1. allow_override

表示是否覆盖每个主机的显示地址。仅与 **Host.display** 属性相关。如果设置，虚拟机的图形控制台地址将被指定的显示地址覆盖。如果没有设置，则虚拟机的图形控制台地址不会被覆盖。

7.67.2. certificate

TLS 证书，以防出现 TLS 连接。如果未启用 TLS，则不会报告它。

7.67.3. copy_paste_enabled

指明用户是否能够从外部主机复制内容并将其粘贴到图形控制台中。此选项仅适用于 **SPICE** 控制台类型。

7.67.4. disconnect_action

返回图形控制台断开连接时将执行的操作。这些选项是：

none

未采取行动。

lock_screen

锁定当前活动的用户会话。

logout

注销当前活动的用户会话。

reboot

启动正常虚拟机重新引导。

shutdown

启动正常虚拟机关闭。

此选项仅适用于 **SPICE** 控制台类型。

7.67.5. disconnect_action_delay

在图形控制台断开连接操作执行前的延迟（分钟）。此选项仅适用于 **Shutdown disconnect** 操作。

7.67.6. file_transfer_enabled

指明用户是否能够将文件从外部主机拖放到图形控制台中。此选项仅适用于 **SPICE** 控制台类型。

7.67.7. keyboard_layout

用于此图形控制台的键盘布局。这个选项只适用于 **VNC** 控制台类型。如果没有启用键盘，则不会报告该键盘。

7.67.8. monitor

为此图形控制台打开的显示器数量。此选项仅适用于 **SPICE** 控制台类型。可能的值有 1、2 或 4。

7.67.9. proxy

图形控制台客户端将用于连接 **guest** 的代理 IP。当客户端位于 **guest** 的网络之外时，此功能很有用。此选项仅适用于 **SPICE** 控制台类型。此代理可以在全局配置、集群级别、虚拟机池级别或在每个虚拟机中禁用。如果在上述任意位置设置了代理，并且未为虚拟机禁用代理，则此方法将返回该代理。如果未设置代理，则不会报告任何内容。

7.67.10. secure_port

如果使用 **TLS**，客户机上的安全端口地址将图形控制台客户端连接到。如果未启用 **TLS**，则不会报告它。

7.67.11. single_qxl_pci

引擎现在根据操作系统自动设置它。因此，它自 4.4.5 起已弃用。指明是否为每个 **monitor** 使用一个 **PCI** 插槽，或者将单个 **PCI** 通道用于所有多个监视器。此选项仅适用于 **SPICE** 控制台类型，仅适用于连接基于 **guest Linux** 的操作系统。

7.67.12. smartcard_enabled

指明是否使用智能卡验证。此选项仅适用于 **SPICE** 控制台类型。

7.68. DISPLAYTYPE ENUM

表示用于连接到虚拟机的图形控制台的协议的枚举。

表 7.90. 值概述

名称	概述
spice	显示 SPICE 类型。
vnc	显示 VNC 类型。

7.68.1. spice

显示 SPICE 类型.详情请查看 <https://www.spice-space.org>。

7.68.2. vnc

显示 VNC 类型.VNC 代表虚拟网络计算,它是图形桌面共享系统,使用 RFB (远程帧缓冲) 协议远程控制另一台计算机。

7.69. DNS STRUCT

表示 DNS 解析器配置。

表 7.91. 属性概述

名称	类型	概述
search_domains	Host[]	作为搜索域的主机数组。
服务器	Host[]	充当 DNS 服务器的主机数组。

7.70. DNSRESOLVERCONFIGURATION STRUCT

表示 DNS 解析器配置。

表 7.92. 属性概述

名称	类型	概述
name_servers	string[]	名称服务器的地址数组。

7.70.1. name_servers

名称服务器的地址数组.可以指定 IPv4 或 IPv6 地址。

7.71. DOMAIN STRUCT

此类型表示目录服务域。

表 7.93. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
user	用户	

表 7.94. 链接摘要

名称	类型	概述
groups	group[]	对目录服务中所有组的引用。
用户	user[]	对 directory 服务中所有用户的列表的引用。

7.71.1. 用户

对 **directory** 服务中所有用户的列表的引用。这些信息用于在 Red Hat Virtualization 环境中添加新用户。

7.72. DYNAMICCPU STRUCT

配置虚拟机的动态 CPU。

表 7.95. 属性概述

名称	类型	概述
cpu_tune	CpuTune	
topology	CpuTopology	

7.73. ENTITYEXTERNALSTATUS ENUM

键入代表外部实体状态。

表 7.96. 值概述

名称	概述
错误	外部实体状态错误。
失败	外部实体存在导致失败的问题。
info	有外部实体状态正常，但有些信息可能相关。
确定	外部实体状态正常。
warning	外部实体状态正常，但存在可能需要注意的问题。

7.73.1. 错误

外部实体状态错误。这可能需要中等程度的关注。

7.73.2. 失败

外部实体存在导致失败的问题。这可能需要立即关注。

7.74. ENTITYPROFILEDETAIL STRUCT

表 7.97. 属性概述

名称	类型	概述
profile_details	ProfileDetail[]	

7.75. ERRORHANDLING STRUCT

表 7.98. 属性概述

名称	类型	概述
on_error	MigrateOnError	

7.76. EVENT STRUCT

键入代表事件。

表 7.99. 属性概述

名称	类型	概述
code	整数	事件代码。
comment	字符串	包含有关此对象的注释的免费文本。
correlation_id	字符串	事件相关性标识符。
custom_data	字符串	代表自定义事件数据的免费文本。
custom_id	整数	自定义事件标识符。
description	字符串	以纯文本格式人类可读的描述。
flood_rate	整数	定义填充率。
id	字符串	唯一标识符。
index	整数	此事件的数字索引。
log_on_host	布尔值	指定是否也应将事件写入 <code>hypervisor</code> 。
name	字符串	纯文本中的人类可读名称。
Origin	字符串	确定事件来源的免费文本。
严重性	LogSeverity	事件严重性。
time	Date	事件时间。

7.76.1. correlation_id

事件相关性标识符。用于将多个事件关联在一起。

7.76.2. flood_rate

定义填充率。如果事件出现在定义的速率中多次，这可防止填充。默认值为 **30 秒**。

7.76.3. index

此事件的数字索引。事件索引始终增加，因此索引较高的事件可以保证比索引较低的事件旧。



重要

在当前实施引擎时，`id` 属性的值与此索引属性相同。这是 API 用户不应依赖的实施细节。将来，`id` 属性可以更改为任意字符串，其包含非数字字符和隐式顺序。另一方面，这个索引属性保证保持为整数并按顺序排列。

7.76.4. log_on_host

指定是否还应将事件写入 `${hypervisor.name}` 日志。如果未指定主机，则事件描述将写入到所有主机。默认值为 `false`。

表 7.100. 链接摘要

名称	类型	概述
<code>cluster</code>	Cluster	对群集服务的引用。
<code>data_center</code>	DataCenter	数据中心服务的引用。
<code>host</code>	主机	对主机服务的引用。
<code>storage_domain</code>	StorageDomain	引用存储域服务。
模板	Template	引用模板服务。
<code>user</code>	用户	用户服务的引用。
<code>vm</code>	VM	引用虚拟机服务。

7.76.5. cluster

对群集服务的引用。事件可以与集群关联。

7.76.6. data_center

数据中心服务的引用。事件可以与数据中心关联。

7.76.7. 主机

对主机服务的引用。事件可以与主机关联。

7.76.8. storage_domain

引用存储域服务.事件可以与存储域关联。

7.76.9. 模板

引用模板服务.事件可以与模板关联。

7.76.10. user

用户服务的引用.事件可以与用户关联。

7.76.11. vm

引用虚拟机服务。事件可以与虚拟机关联。

7.77. EVENTSUBSCRIPTION STRUCT

表 7.101. 属性概述

名称	类型	概述
地址	字符串	通知应发送到的电子邮件地址。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
event	NotifiableEvent	订阅的活动。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
notification_method	NotificationMethod	通知方法：SMTP 或 SNMP。
user	用户	订阅用户。

7.77.1. 地址

通知应发送到的电子邮件地址。

如果未提供，则会将通知发送到用户的电子邮件。目前，每个用户只支持一个地址。如果在现有订阅中添加了具有不同电子邮件地址的订阅，则会返回 409(CONFLICT)状态，说明提供的地址与该用户的事件订阅的现有地址冲突。

此字段将来可能会被弃用，通知将始终在用户的电子邮件地址上发送。

7.77.2. event

订阅的活动。

(与用户组合，唯一标识 event-subscription)。

7.77.3. notification_method

通知方法：SMTP 或 SNMP。

目前仅支持 API 支持的 SMTP。未来将增加对 SNMP 的支持。

7.77.4. user

订阅用户。

与 event-name 结合使用，唯一标识 event-subscription。

7.78. EXTERNALCOMPUTERESOURCE STRUCT

表 7.102. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。

名称	类型	概述
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
provider	字符串	
url	字符串	
user	字符串	

表 7.103. 链接摘要

名称	类型	概述
external_host_provider	ExternalHostProvider	

7.79. EXTERNALDISCOVEREDHOST STRUCT

表 7.104. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
ip	字符串	
last_report	字符串	
MAC	字符串	
name	字符串	纯文本中的人类可读名称。
subnet_name	字符串	

表 7.105. 链接摘要

名称	类型	概述
external_host_provider	ExternalHostProvider	

7.80. EXTERNALHOST STRUCT

代表由主机提供程序（如 Foreman/Satellite）调配的主机。

有关 Foreman 的更多详细信息，请参见 <https://www.theforeman.org/>。有关 Red Hat Satellite 的详情，请查看 <https://access.redhat.com/products/red-hat-satellite>。

表 7.106. 属性概述

名称	类型	概述
地址	字符串	主机的地址，即 FQDN 的 IP 地址（完全限定域名）。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.107. 链接摘要

名称	类型	概述
external_host_provider	ExternalHostProvider	对主机所管理的外部主机提供程序的引用。

7.81. EXTERNALHOSTGROUP STRUCT

表 7.108. 属性概述

名称	类型	概述
architecture_name	字符串	

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
domain_name	字符串	
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
operating_system_name	字符串	
subnet_name	字符串	

表 7.109. 链接摘要

名称	类型	概述
external_host_provider	ExternalHostProvider	

7.82. EXTERNALHOSTPROVIDER STRUCT

代表外部主机提供程序，如 Foreman 或卫星。

有关 Foreman 的更多详细信息，请参见 <https://www.theforeman.org/>。有关 Red Hat Satellite 的详情，请查看 <https://access.redhat.com/products/red-hat-satellite>。

表 7.110. 属性概述

名称	类型	概述
authentication_url	字符串	定义外部提供程序身份验证 URL 地址。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。

名称	类型	概述
name	字符串	纯文本中的人类可读名称。
password	字符串	定义用户在身份验证过程中的密码。
属性	attribute[]	提供程序名称/值属性数组。
required_authentication	布尔值	定义是否需要供应商身份验证。
url	字符串	定义外部提供程序的 URL 地址。
username	字符串	定义要在身份验证过程中使用的用户名。

7.82.1. required_authentication

定义是否需要供应商身份验证。

如果需要身份验证，则身份验证期间将使用 **用户名和密码** 属性。

表 7.111. 链接摘要

名称	类型	概述
证书	certificate[]	引用引擎为此供应商支持的证书。
compute_resources	ExternalComputeResource[]	对主机提供程序中所指示的计算资源的引用。
discovered_hosts	ExternalDiscoveredHost[]	对主机提供程序中发现的主机的引用。
host_groups	ExternalHostGroup[]	对主机提供程序中的主机组的引用。
主机	Host[]	对主机提供程序调配的主机的引用。

7.82.2. compute_resources

对主机提供程序中所指示的计算资源的引用。每一主机提供程序有可选的引擎定义为计算资源，允许在引擎中创建虚拟机。此计算资源详细信息在裸机恢复调配用例中使用，以部署虚拟机监控程序。

7.82.3. discovered_hosts

对主机提供程序中发现的主机的引用。发现的主机是尚未调配的主机。

7.82.4. host_groups

对主机提供程序中的主机组的引用。主机组包含不同的属性，供主机提供程序应用到属于此组的所有主机上。例如安装的软件、系统定义、密码等。

7.83. EXTERNALNETWORKPROVIDERCONFIGURATION STRUCT

描述主机上如何调配外部网络提供程序。

表 7.112. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.113. 链接摘要

名称	类型	概述
external_network_provider	ExternalProvider	链接到外部网络提供程序。
host	主机	到主机的链接。

7.84. EXTERNALPROVIDER STRUCT

代表外部提供程序。

表 7.114. 属性概述

名称	类型	概述
authentication_url	字符串	定义外部提供程序身份验证 URL 地址。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
password	字符串	定义用户在身份验证过程中的密码。
属性	attribute[]	提供程序名称/值属性数组。
required_authentication	布尔值	定义是否需要供应商身份验证。
url	字符串	定义外部提供程序的 URL 地址。
username	字符串	定义要在身份验证过程中使用的用户名。

7.84.1. required_authentication

定义是否需要供应商身份验证。

如果需要身份验证，则身份验证期间将使用 用户名和密码 属性。

7.85. EXTERNALSTATUS ENUM

表示外部状态。此状态目前用于主机和 存储域，并且允许外部系统更新它所识别的对象的状态。

表 7.115. 值概述

名称	概述
错误	错误状态。
失败	故障状态。

名称	概述
info	信息状态.
确定	确定状态.
warning	Warning 状态.

7.85.1. 错误

错误状态.相关对象中存在某种错误。

7.85.2. 失败

故障状态.相关对象失败。

7.85.3. info

信息状态.相关对象处于 OK 状态，但也有与管理员相关的可用信息。

7.85.4. 确定

确定状态.相关对象运行良好。

7.85.5. warning

Warning 状态.相关对象运行良好，但有一些警告可能与管理员相关。

7.86. EXTERNALSYSTEMTYPE ENUM

表示与该 步骤 关联的外部系统的类型。

表 7.116. 值概述

名称	概述
Gluster	将 Gluster 表示与 步骤 关联的外部系统。

名称	概述
vdsm	表示 VDSM 作为与该步骤 关联的外部系统。

7.87. EXTERNALTEMPLATEIMPORT STRUCT

描述从外部系统模板导入操作的参数。目前仅支持 OVA。

表 7.117. 属性概述

名称	类型	概述
clone	布尔值	可选。
url	字符串	要传递给引擎的 URL。

7.87.1. clone

可选。表示是否应该重新生成导入模板的标识符。

默认情况下，在导入模板时，标识符会被保留。这意味着同一模板无法多次导入，因为该标识符需要是唯一的。要允许多次导入同一模板，将此参数设置为 `true`，因为默认值为 `false`。

7.87.2. url

要传递给引擎的 URL。

例如：

```
ova:///mnt/ova/ova_file.ova
```

表 7.118. 链接摘要

名称	类型	概述
cluster	Cluster	为生成的模板指定目标集群。
cpu_profile	CpuProfile	可选。

名称	类型	概述
host	主机	指定存在 OVA 文件的主机。
quota	Quota	可选。
storage_domain	StorageDomain	指定磁盘的目标存储域。
模板	Template	用于指定新创建的模板名称的模板实体。

7.87.3. cpu_profile

可选。指定生成的模板的 CPU 配置集。

7.87.4. quota

可选。指定应用到生成的模板的配额。

7.87.5. 模板

用于指定新创建的模板名称的模板实体。

如果未指定名称，则使用源模板名称。

7.88. EXTERNALVMIMPORT STRUCT

描述从外部系统虚拟机导入操作的参数。

表 7.119. 属性概述

名称	类型	概述
name	字符串	要导入的虚拟机的名称，如外部系统中所定义。
password	字符串	用于对外部虚拟机监控程序系统进行身份验证的密码。
provider	ExternalVmProviderType	外部虚拟机提供程序的类型。

名称	类型	概述
稀疏	布尔值	可选。
url	字符串	传递给 virt-v2v 工具以进行转换的 URL。
username	字符串	要针对外部虚拟机监控程序系统进行身份验证的用户名。

7.88.1. sparse

可选。指定生成的虚拟机的磁盘分配策略：**true** 用于稀疏，对于预分配为 **false**。

如果没有指定：**-** 在导入由 oVirt 生成的 OVA 时，它将根据 OVF 中的磁盘配置决定。**-** 否则将设置为 **true**。

7.88.2. url

传递给 **virt-v2v** 工具以进行转换的 URL。

例如：

```
vpix://vmware_user@vcenter-host/DataCenter/Cluster/esxi-host?no_verify=1
```

如需更多示例，请访问

表 7.120. 链接摘要

名称	类型	概述
cluster	Cluster	为生成的虚拟机指定目标集群。
cpu_profile	CpuProfile	可选。
drivers_iso	File	可选。
host	主机	可选。
quota	Quota	可选。
storage_domain	StorageDomain	为转换的磁盘指定目标存储域。

名称	类型	概述
vm	VM	用于为新创建的虚拟机指定名称的虚拟机实体。

7.88.3. cpu_profile

可选。指定生成的虚拟机的 CPU 配置集。

7.88.4. drivers_iso

可选。包含可在 virt-v2v 转换过程中使用的驱动程序的 ISO 名称。

7.88.5. 主机

可选。指定用于转换过程的主机（使用主机的 ID）。如果没有指定，则会自动选择一个。

7.88.6. quota

可选。指定应用到生成的虚拟机的配额。

7.88.7. vm

用于为新创建的虚拟机指定名称的虚拟机实体。

如果未指定名称，则将使用源虚拟机名称。

7.89. EXTERNALVMPROVIDERTYPE ENUM

描述外部系统管理程序系统的类型。

表 7.121. 值概述

名称	概述
kvm	
vmware	
Xen	

7.90. FAULT STRUCT

表 7.122. 属性概述

名称	类型	概述
详情	字符串	
reason	字符串	

7.91. FENCETYPE ENUM

键入 代表隔离操作的类型。

表 7.123. 值概述

名称	概述
Manual	通过电源管理手动主机隔离。
restart	通过电源管理重新启动主机。
start	通过电源管理启动主机。
status	通过电源管理检查主机电源状态。
stop	通过电源管理来停止主机。

7.92. FENCINGPOLICY STRUCT

键入代表群集隔离策略。

表 7.124. 属性概述

名称	类型	概述
enabled	布尔值	在此群集上启用或禁用隔离。
skip_if_connectivity_broken	SkipIfConnectivityBroken	如果启用，我们将不会在群集中主机超过连接丢失百分比时隔离主机。
skip_if_gluster_bricks_up	布尔值	指示是否应在被隔离的主机中启动并运行 Gluster brick 时是否应该跳过隔离的标志。
skip_if_gluster_quorum_not_met	布尔值	指示是否应跳过隔离的标志，如果 Gluster brick 已启动并在运行，并且没有这些 brick 就无法满足 Gluster 仲裁。
skip_if_sd_active	SkipIfSdActive	如果启用，我们将在主机在存储中维持其租用时跳过隔离。

7.92.1. skip_if_connectivity_broken

如果启用，我们将不会在群集中主机超过连接丢失百分比时隔离主机。这可用于防止在群集中存在全局网络问题时出现隔离风暴。

7.92.2. skip_if_gluster_bricks_up

指示是否应在被隔离的主机中启动并运行 Gluster brick 时是否应该跳过隔离的标志。这个标志是可选的，默认值为 `false`。

7.92.3. skip_if_gluster_quorum_not_met

指示是否应跳过隔离的标志，如果 Gluster brick 已启动并在运行，并且没有这些 brick 就无法满足 Gluster 仲裁。这个标志是可选的，默认值为 `false`。

7.92.4. skip_if_sd_active

如果启用，我们将在主机在存储中维持其租用时跳过隔离。这意味着，如果主机仍具有存储访问权限，则它不会被隔离。

7.93. FILE STRUCT

表 7.125. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
content	字符串	
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
type	字符串	

表 7.126. 链接摘要

名称	类型	概述
storage_domain	StorageDomain	

7.94. FILTER STRUCT

表 7.127. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
位置	整数	

表 7.128. 链接摘要

名称	类型	概述
scheduling_policy_unit	SchedulingPolicyUnit	

7.95. FIPSMODE ENUM

将 FIPS 模式表示到集群。

表 7.129. 值概述

名称	概述
disabled	禁用 FIPS 模式。
enabled	启用了 FIPS 模式。
undefined	FIPS 模式还没有被评估。

7.95.1. disabled

禁用 FIPS 模式。

其含义是禁用 FIPS 模式，并且 中的主机应该禁用 FIPS 模式，否则它们将无法运行。

7.95.2. enabled

启用了 FIPS 模式。

其含义是启用了 FIPS 模式，且 中的主机应启用 FIPS 模式，否则不应运行。

7.95.3. 未定义

FIPS 模式还没有被评估。

目前，其含义是未确定 FIPS 模式。添加主机后，该值将根据主机设置进行切换。

7.96. FIREWALLTYPE ENUM

描述系统支持的所有防火墙类型。

表 7.130. 值概述

名称	概述
firewalld	firewalld 防火墙类型.
iptables	iptables 防火墙类型.

7.96.1. firewalld

firewalld 防火墙类型.

当群集的防火墙类型设置为 **firewalld** 时，将使用 **firewalld** 配置群集中所有主机的防火墙。**firewalld** 在版本 4.2 中替换了 **IPTables**。它使用命令程序和动态配置来简化配置。

7.96.2. iptables

iptables 防火墙类型.

iptables 已被弃用。

7.97. FLOPPY STRUCT

软盘文件的基本表示。

表 7.131. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
file	File	代表软盘设备内容及其类型的文件对象。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.132. 链接摘要

名称	类型	概述
instance_type	InstanceType	(可选) 引用设备使用的实例类型。
模板	Template	(可选) 引用设备使用的模板。
vm	VM	不要使用此元素，而是使用 vms 。
vms	Vm[]	引用使用此设备的虚拟机。

7.97.1. vms

引用使用此设备的虚拟机。个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.98. FOPSTATISTICSTRUCT

表 7.133. 属性概述

名称	类型	概述
name	字符串	
statistics	Statistic[]	

7.99. GLUSTERBRICK STRUCT

表 7.134. 属性概述

名称	类型	概述
brick_dir	字符串	
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
device	字符串	
fs_name	字符串	
gluster_clients	GlusterClient[]	
id	字符串	唯一标识符。

名称	类型	概述
memory_pools	GlusterMemoryPool[]	
mnt_options	字符串	
name	字符串	纯文本中的人类可读名称。
pid	整数	
port	整数	
server_id	字符串	
status	GlusterBrickStatus	

表 7.135. 链接摘要

名称	类型	概述
gluster_volume	GlusterVolume	
instance_type	InstanceType	(可选) 引用设备使用的实例类型。
statistics	Statistic[]	
模板	Template	(可选) 引用设备使用的模板。
vm	VM	不要使用此元素，而是使用 vms 。
vms	Vm[]	引用使用此设备的虚拟机。

7.99.1. vms

引用使用此设备的虚拟机。个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.100. GLUSTERBRICKADVANCEDDETAILS STRUCT

表 7.136. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
device	字符串	
fs_name	字符串	
gluster_clients	GlusterClient[]	
id	字符串	唯一标识符。
memory_pools	GlusterMemoryPool[]	
mnt_options	字符串	
name	字符串	纯文本中的人类可读名称。
pid	整数	
port	整数	

表 7.137. 链接摘要

名称	类型	概述
instance_type	InstanceType	(可选) 引用设备使用的实例类型。
模板	Template	(可选) 引用设备使用的模板。
vm	VM	不要使用此元素，而是使用 vms 。
vms	Vm[]	引用使用此设备的虚拟机。

7.100.1. vms

引用使用此设备的虚拟机。个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.101. GLUSTERBRICKMEMORYINFO STRUCT

表 7.138. 属性概述

名称	类型	概述
memory_pools	GlusterMemoryPool[]	

7.102. GLUSTERBRICKSTATUS ENUM

表 7.139. 值概述

名称	概述
down	Brick 处于 down 状态，无法从中存储或检索数据。
Unknown	当由于主机不响应而无法确定状态时。
up	Brick 处于 up 状态，数据可以存储或从中检索。

7.103. GLUSTERCLIENT STRUCT

表 7.140. 属性概述

名称	类型	概述
bytes_read	整数	
bytes_written	整数	
client_port	整数	
host_name	字符串	

7.104. GLUSTERHOOK STRUCT

表 7.141. 属性概述

名称	类型	概述
checksum	字符串	
comment	字符串	包含有关此对象的注释的免费文本。
conflict_status	整数	
Conflicts	字符串	

名称	类型	概述
content	字符串	
content_type	HookContentType	
description	字符串	以纯文本格式人类可读的描述。
gluster_command	字符串	
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
stage	HookStage	
status	GlusterHookStatus	

表 7.142. 链接摘要

名称	类型	概述
cluster	Cluster	
server_hooks	GlusterServerHook[]	

7.105. GLUSTERHOOKSTATUS ENUM

表 7.143. 值概述

名称	概述
disabled	集群中禁用 hook。
enabled	集群中启用 hook。
缺少	未知/缺少 hook 状态。

7.106. GLUSTERMEMORYPOOL STRUCT

表 7.144. 属性概述

名称	类型	概述
alloc_count	整数	
cold_count	整数	
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
hot_count	整数	
id	字符串	唯一标识符。
max_alloc	整数	
max_stdalloc	整数	
name	字符串	纯文本中的人类可读名称。
padded_size	整数	
pool_misses	整数	
type	字符串	

7.107. GLUSTERSEVERHOOK STRUCT

表 7.145. 属性概述

名称	类型	概述
checksum	字符串	
comment	字符串	包含有关此对象的注释的免费文本。
content_type	HookContentType	
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
status	GlusterHookStatus	

表 7.146. 链接摘要

名称	类型	概述
host	主机	

7.108. GLUSTERSTATE ENUM

表 7.147. 值概述

名称	概述
down	
Unknown	
up	

7.109. GLUSTERVOLUME STRUCT

表 7.148. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
disperse_count	整数	
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
选项	选项[]	
redundancy_count	整数	
replica_count	整数	
status	GlusterVolumeStatus	
stripe_count	整数	
transport_types	TransportType[]	

名称	类型	概述
volume_type	GlusterVolumeType	

表 7.149. 链接摘要

名称	类型	概述
brick	GlusterBrick[]	
cluster	Cluster	
statistics	Statistic[]	

7.110. GLUSTERVOLUMEPROFILEDETAILS STRUCT

表 7.150. 属性概述

名称	类型	概述
brick_profile_details	BrickProfileDetail[]	
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
nfs_profile_details	NfsProfileDetail[]	

7.111. GLUSTERVOLUMESTATUS ENUM

表 7.151. 值概述

名称	概述
down	需要启动卷，以使客户端能够挂载和使用它。
Unknown	当由于主机不响应而无法确定状态时。

名称	概述
up	卷已启动，客户端可以挂载和使用卷。

7.112. GLUSTERVOLUMETYPE ENUM

类型代表 Gluster 卷的类型。

表 7.152. 值概述

名称	概述
disperse	离散式卷基于纠删代码，提供有效空间保护，防止磁盘或服务器故障。
distribution	分布式卷在卷中的 brick 之间分发文件。
distributed_disperse	分布式离散式卷在离散子卷之间分发文件。
distributed_replicate	分布式复制卷在卷中的复制 brick 之间分发文件。
distributed_stripe	分布式分条卷在集群中的两个或多个节点之间条状分布数据。
distributed_striped_replicate	分布式复制卷在集群中复制 brick 之间分发分条数据。
复制	复制卷在卷中跨 brick 复制文件。
stripe	条状卷将数据分条卷中 brick 之间的数据。
striped_replicate	分条复制卷在集群中复制 brick 之间分条数据。

7.112.1. disperse

离散式卷基于纠删代码，提供有效空间保护，防止磁盘或服务器故障。

将原始文件的一个编码片段分散到每个 brick 时，只需将片段的一个子集才能恢复原始文件。管理员在卷创建时间上配置在不丢失数据访问权限的情况下丢失的 brick 数量。

7.112.2. distribution

分布式卷在卷中的 **brick** 之间分发文件。

当要求缩放存储且冗余不重要，或者由其他硬件/软件层提供时，可以使用分布式卷。

7.112.3. distributed_disperse

分布式离散式卷在离散子卷之间分发文件。

这具有相同的优势，可以分发复制卷，但使用 **disperse** 将数据存储到 **brick** 中。

7.112.4. distributed_replicate

分布式复制卷在卷中的复制 **brick** 之间分发文件。

分布式复制卷可在要求扩展存储和高可靠性至关重要的环境中使用。分布式复制卷还可在大多数环境中提供更高的读取性能。

7.112.5. distributed_stripe

分布式分条卷在集群中的两个或多个节点之间条状分布数据。

在要求扩展存储和访问非常大文件的高并发环境中，应使用分布式条状卷。

注：在 **Glusterfs 3.7** 版本中引入 **Sharding** 时，不建议使用条带卷，它将在以后的发行版中删除。

7.112.6. distributed_stripped_replicate

分布式复制卷在集群中复制 **brick** 之间分发分条数据。

为获得最佳效果，分布式复制卷应在高度并发的环境中使用，其中并行访问非常大的文件和性能至关重要。

注：在 Glusterfs 3.7 版本中引入 Sharding 时，不建议使用条带卷，它将在以后的发行版中删除。

7.112.7. 复制

复制卷在卷中跨 brick 复制文件。

复制卷可用于高可用性和高可靠性至关重要的环境中。

7.112.8. stripe

条状卷将数据分条卷中 brick 之间的数据。

为获得最佳效果，条带卷应该仅在访问非常大文件的高并发环境中。

注：在 Glusterfs 3.7 版本中引入 Sharding 时，不建议使用条带卷，它将在以后的发行版中删除。

7.112.9. striped_replicate

分条复制卷在集群中复制 brick 之间分条数据。

为获得最佳效果，分条复制卷应在高度并发环境中使用，其中可以并行访问非常大的文件，性能至关重要。

注：在 Glusterfs 3.7 版本中引入 Sharding 时，不建议使用条带卷，它将在以后的发行版中删除。

7.113. GRACEPERIOD STRUCT

表 7.153. 属性概述

名称	类型	概述
到期	整数	

7.114. GRAPHICSCONSOLE STRUCT

表 7.154. 属性概述

名称	类型	概述
地址	字符串	
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
port	整数	
protocol	GraphicsType	
tls_port	整数	

表 7.155. 链接摘要

名称	类型	概述
instance_type	InstanceType	
模板	Template	
vm	VM	

7.115. GRAPHICSTYPE ENUM

用于连接到图形控制台的图形协议。

表 7.156. 值概述

名称	概述
spice	SPICE 类型的图形协议。
vnc	VNC 类型的图形协议。

7.115.1. spice

SPICE 类型的图形协议.详情请查看 <https://www.spice-space.org>。

7.115.2. vnc

VNC 类型的图形协议.VNC 代表虚拟网络计算,它是图形桌面共享系统,使用 RFB (远程帧缓冲) 协议远程控制另一台计算机。

7.116. GROUP STRUCT

此类型表示目录服务中的所有组。

表 7.157. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
domain_entry_id	字符串	包含目录服务域 ID。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
namespace	字符串	组驻留的命名空间。

表 7.158. 链接摘要

名称	类型	概述
domain	域	包含该组的域的链接。
权限	permission[]	权限子集合链接,用于关联至该组的权限。
角色	role[]	与该组关联的角色子集合的链接。
tags	Tag[]	附加到此组的标签子集合的链接。

7.116.1. 角色

与该组关联的角色子集合的链接。

仅用于表示新组的初始角色分配；因此，角色分配的修改仅通过角色子集合支持。

7.117. GUESTOPERATINGSYSTEM STRUCT

代表虚拟机上安装的操作系统。

要获取该信息，请发送类似如下的请求：

```
GET /ovirt-engine/api/vms/123
```

结果如下：

```
<vm href="/ovirt-engine/api/vms/123" id="123">
...
<guest_operating_system>
  <architecture>x86_64</architecture>
  <codename>Maipo</codename>
  <distribution>Red Hat Enterprise Linux Server</distribution>
  <family>Linux</family>
  <kernel>
    <version>
      <build>0</build>
      <full_version>3.10.0-514.10.2.el7.x86_64</full_version>
      <major>3</major>
      <minor>10</minor>
      <revision>514</revision>
    </version>
  </kernel>
  <version>
    <full_version>7.3</full_version>
    <major>7</major>
    <minor>3</minor>
  </version>
</guest_operating_system>
</vm>
```

表 7.159. 属性概述

名称	类型	概述
架构	字符串	操作系统的架构，如 x86_64。

名称	类型	概述
codename	字符串	操作系统的代码名称，如 Maipo 。
distribution	字符串	操作系统分发的全名。
产品线	字符串	操作系统系列，如 Linux 。
kernel	内核	操作系统的内核版本。
version	版本	安装的操作系统的版本。

7.118. HARDWAREINFORMATION STRUCT

表示主机的硬件信息。

要获取该信息，请发送类似如下的请求：

```
GET /ovirt-engine/api/hosts/123
```

结果如下：

```
<host href="/ovirt-engine/api/hosts/123" id="123">
...
<hardware_information>
  <family>Red Hat Enterprise Linux</family>
  <manufacturer>Red Hat</manufacturer>
  <product_name>RHEV Hypervisor</product_name>
  <serial_number>01234567-89AB-CDEF-0123-456789ABCDEF</serial_number>
  <supported_rng_sources>
    <supported_rng_source>random</supported_rng_source>
  </supported_rng_sources>
  <uuid>12345678-9ABC-DEF0-1234-56789ABCDEF0</uuid>
  <version>1.2-34.5.el7ev</version>
</hardware_information>
...
</application>
```

表 7.160. 属性概述

名称	类型	概述
产品线	字符串	主机的 CPU 类型。

名称	类型	概述
制造商	字符串	主机的机器和硬件供应商的制造商。
product_name	字符串	主机的产品名称（如 RHEV 系统管理程序 ）。
serial_number	字符串	主机机箱的唯一 ID。
supported_rng_sources	RngSource[]	随机数字生成器的支持源。
uuid	字符串	每个主机的唯一 ID。
version	字符串	每个制造商的唯一名称。

7.119. HIGHAVAILABILITY STRUCT

键入代表虚拟机的高可用性。

表 7.161. 属性概述

名称	类型	概述
enabled	布尔值	定义虚拟机是否被视为高度可用。
priority	整数	表示运行和迁移队列中虚拟机的优先级。

7.119.1. enabled

定义虚拟机是否被视为高度可用。强烈建议配置虚拟机租用（请参阅该部分）以防止出现脑裂情况。使用引导磁盘的 `storage-domain` 或任何其他活跃的 `storage-domain`。

7.119.2. priority

表示运行和迁移队列中虚拟机的优先级。

优先级较高的虚拟机将在优先级较低的虚拟机之前启动和迁移。

该数值为 0 到 100 之间的整数。数值越大，优先级越高。

图形用户界面(GUI)不允许指定所有可能的值，而是只允许您选择 *Low*、*Medium* 或 *High*。使用 API 设置值时，GUI 将设置标签，如下所示：

API 值	GUI 标签
0 - 25	低
26 - 74	中
75 - 100	high

使用 GUI 选择标签时，API 中的值将设置如下：

GUI 标签	API 值
低	1
中	50
high	100

7.120. HOOK STRUCT

表示 *hook*。

表 7.162. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
event_name	字符串	要对其执行 hook 的事件的名称。
id	字符串	唯一标识符。
md5	字符串	hook 的校验和。
name	字符串	纯文本中的人类可读名称。

表 7.163. 链接摘要

名称	类型	概述
host	主机	对 hook 所属主机的引用。

7.121. HOOKCONTENTTYPE ENUM

表示 hook 脚本的内容类型。

表 7.164. 值概述

名称	概述
二进制	hook 的二进制内容类型。
text	hook 的文本内容类型。

7.122. HOOKSTAGE ENUM

type 表示 hook 在其中执行的卷事件阶段。

表 7.165. 值概述

名称	概述
post	卷启动后的阶段。
pre	卷启动前的阶段。

7.123. HOOKSTATUS ENUM

type 表示 hook 的状态。

表 7.166. 值概述

名称	概述
disabled	Hook 已被禁用。
enabled	Hook 已启用。

名称	概述
缺少	缺少 hook。

7.124. HOST STRUCT

键入代表主机。

表 7.167. 属性概述

名称	类型	概述
地址	字符串	主机地址(FQDN/IP)。
auto_numa_status	AutoNumaStatus	主机自动非统一内存访问(NUMA)状态。
certificate	证书	主机证书。
comment	字符串	包含有关此对象的注释的免费文本。
cpu	cpu	此主机的 CPU 类型。
description	字符串	以纯文本格式人类可读的描述。
device_passthrough	HostDevicePassthrough	指定是否在此主机上启用主机设备直通。
display	显示	(可选) 明确指定此主机的显示地址。
external_status	ExternalStatus	主机外部状态。
hardware_information	HardwareInformation	主机硬件信息。
hosted_engine	HostedEngine	此主机的自托管引擎状态。
id	字符串	唯一标识符。
iscsi	IscsiDetails	主机 iSCSI 详细信息。
kdump_status	KdumpStatus	主机 KDUMP 状态。
ksm	km	内核同页合并(KSM)减少了从多个相同页面到单个页面参考的内存页的引用。

名称	类型	概述
libvirt_version	版本	主机 libvirt 版本。
max_scheduling_memory	整数	此主机上以字节为单位的最大调度内存。
内存	整数	此主机上的物理内存量（以字节为单位）。
name	字符串	纯文本中的人类可读名称。
network_operation_in_progress	布尔值	指定当前是否在此主机上执行与网络相关的操作，如 'setup network'、'sync network' 或 'refresh capabilities'。
numa_supported	布尔值	<i>指定此主机上是否支持非统一内存访问(NUMA)。</i>
OS	OperatingSystem	此主机上的操作系统。
override_ip_tables	布尔值	指定是否应覆盖防火墙定义。
ovn_configured	布尔值	表示主机是否已正确配置 OVN。
port	整数	主机端口。
power_management	PowerManagement	主机电源管理定义。
protocol	HostProtocol	引擎用于与主机通信的协议。
reinstallation_required	布尔值	指定是否应重新安装主机。
root_password	字符串	在创建新主机时，如果选择了密码身份验证方法，则需要 root 密码，但这随后不包含在表示法中。
se_linux	SeLinux	主机 SELinux 状态。
spm	SPM	<i>主机存储池管理器 (SPM) 状态和定义。</i>
ssh	ssh	SSH 定义。
status	HostStatus	主机状态。
status_detail	字符串	主机状态详细信息。

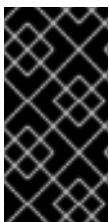
名称	类型	概述
summary	VmSummary	虚拟机摘要 - 有多少处于活动状态、迁移和总计。
transparent_huge_pages	TransparentHugePages	透明大内存页支持扩展内存页面大小，超过标准的 4 KiB 限制。
type	HostType	指明主机是否包含操作系统的完整安装，或者仅用于托管虚拟机的缩减版本。
update_available	布尔值	指定此主机上是否存在与 oVirt 相关的更新。
version	版本	VDSM 的版本。
vgpu_placement	VgpuPlacement	指定 vGPU 放置策略。

7.124.1. external_status

主机外部状态。 这可供第三方软件用于在出现问题时更改主机外部状态。这不会影响主机生命周期，除非第三方软件检查此状态并进行相应操作。

7.124.2. hosted_engine

此主机的自托管引擎状态。



重要

检索主机或主机集合时，不包含此属性，除非操作的 `all_content` 参数明确设置为 `true`。有关详细信息，请参阅检索一个或多个主机的操作文档。

7.124.3. kdump_status

主机 KDUMP 状态。 当主机内核崩溃且现在已执行内存转储时，KDUMP 会发生。

7.124.4. ksm

内核同页合并(KSM)减少了从多个相同页面到单个页面引用的内存页的引用。这有助于优化内存密度。

例如，要为主机 123 启用 KSM，请发送如下请求：

```
PUT /ovirt-engine/api/hosts/123
```

这样的请求正文：

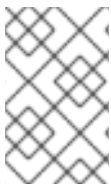
```
<host>
  <ksm>
    <enabled>true</enabled>
  </ksm>
</host>
```

7.124.5. libvirt_version

主机 libvirt 版本.有关 libvirt 的更多信息，请转至 [libvirt](#)。

7.124.6. network_operation_in_progress

指定当前是否在此主机上执行与网络相关的操作，如 'setup network'、'sync network' 或 'refresh capabilities'。



注意

必须将标头 `All-Content:true` 添加到请求中，才能将此属性包含在响应中。

7.124.7. override_iptables

指定是否应覆盖防火墙定义。这只在安装或重新安装主机时才适用。

7.124.8. 协议

引擎用于与主机通信的协议。

**警告**

自引擎版本 4.1 起，自 `xml` 删除后，协议始终设置为 `stomp`。

7.124.9. se_linux

主机 SELinux 状态。 `Security-Enhanced Linux (SELinux)` 是 Linux 内核中的一个组件，为支持访问控制安全策略提供了一种机制。

7.124.10. spm

主机存储池管理器 (SPM) 状态和定义。 使用它来设置此主机的 SPM 优先级，并查看这是否为当前 SPM。

7.124.11. status_detail

主机状态详细信息。 与 Gluster 主机相关。

7.124.12. transparent_huge_pages

透明大内存页支持扩展内存页面大小，超过标准的 4 KiB 限制。 这可减少内存消耗并提升主机性能。

例如，要为主机 123 启用透明巨页支持，请发送如下请求：

```
PUT /ovirt-engine/api/hosts/123
```

这样的请求正文：

```
<host>
  <transparent_hugepages>
    <enabled>true</enabled>
  </transparent_hugepages>
</host>
```

7.124.13. version

VDSM 的版本。

例如：

```
GET /ovirt-engine/api/hosts/123
```

此 GET 请求 将返回以下输出：

```
<host>
...
<version>
  <build>999</build>
  <full_version>vdsm-4.18.999-419.gitcf06367.el7</full_version>
  <major>4</major>
  <minor>18</minor>
  <revision>0</revision>
</version>
...
</host>
```

表 7.168. 链接摘要

名称	类型	概述
affinity_labels	AffinityLabel[]	
代理	agent[]	
cluster	Cluster	
cpu_units	HostCpuUnit[]	所有主机的 CPU 列表，其中包含关于拓扑（套接字、核心）以及当前 CPU 固定信息的详细信息。
devices	HostDevice[]	
external_host_provider	ExternalHostProvider	
external_network_provider_configurations	ExternalNetworkProviderConfiguration[]	主机上调配的外部网络提供程序。
Hook	Hook[]	

名称	类型	概述
katello_errata	KatelloErratum[]	列出分配到该主机的所有 Katello 勘误表。
network_attachments	NetworkAttachment[]	
nics	HostNic[]	
numa_nodes	NumaNode[]	
权限	permission[]	
statistics	Statistic[]	每一主机资源公开一个统计信息子集合，用于特定于主机的统计信息。
storage_connection_extensions	StorageConnectionExtension[]	
Storages	HostStorage[]	
tags	Tag[]	
unmanaged_networks	UnmanagedNetwork[]	

7.124.14. cpu_units

所有主机的 CPU 列表，其中包含关于拓扑（套接字、核心）以及当前 CPU 固定信息的详细信息。

`GET /ovirt-engine/api/hosts/123/cpuunits`

您将在 XML 中收到类似以下的响应：

```
<host_cpu_units>
  <host_cpu_unit>
    <core_id>0</core_id>
    <cpu_id>0</cpu_id>
    <socket_id>0</socket_id>
    <vms>
      <vm href="/ovirt-engine/api/vms/def" id="def" />
    </vms>
  </host_cpu_unit>
  <host_cpu_unit>
    <core_id>0</core_id>
    <cpu_id>1</cpu_id>
```

```

<socket_id>1</socket_id>
<runs_vdsm>true</runs_vdsm>
</host_cpu_unit>
<host_cpu_unit>
  <core_id>0</core_id>
  <cpu_id>2</cpu_id>
  <socket_id>2</socket_id>
</host_cpu_unit>
</host_cpu_units>

```

7.124.15. external_network_provider_configurations

主机上调配的外部网络提供程序。

此属性为只读。设置它对主机没有影响。此参数的值反映了集群的 [Default Network Provider](#)。

7.124.16. katello_errata

列出分配到该主机的所有 Katello 勘误表。

```
GET /ovirt-engine/api/hosts/123/katelloerrata
```

您将在 XML 中收到类似以下的响应：

```

<katello_errata>
  <katello_erratum href="/ovirt-engine/api/katelloerrata/456" id="456">
    <name>RHBA-2013:XYZ</name>
    <description>The description of the erratum</description>
    <title>some bug fix update</title>
    <type>bugfix</type>
    <issued>2013-11-20T02:00:00.000+02:00</issued>
    <solution>Few guidelines regarding the solution</solution>
    <summary>Updated packages that fix one bug are now available for XYZ</summary>
    <packages>
      <package>
        <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
      </package>
      ...
    </packages>
  </katello_erratum>
  ...
</katello_errata>

```

7.124.17. statistics

每一主机资源公开一个统计信息子集合，用于特定于主机的统计信息。

XML 表述示例：

```
<statistics>
  <statistic href="/ovirt-engine/api/hosts/123/statistics/456" id="456">
    <name>memory.total</name>
    <description>Total memory</description>
    <kind>gauge</kind>
    <type>integer</type>
    <unit>bytes</unit>
    <values>
      <value>
        <datum>25165824000</datum>
      </value>
    </values>
    <host href="/ovirt-engine/api/hosts/123" id="123"/>
  </statistic>
  ...
</statistics>
```



注意

这个统计子集合是只读的。

以下列表显示了主机的统计类型：

名称	描述
memory.total	主机上的总内存（以字节为单位）。
memory.used	主机上使用的内存（以字节为单位）。
memory.free	主机上空闲的内存（以字节为单位）。
memory.shared	主机上共享的字节内存。
memory.buffers	以字节为单位的 I/O 缓冲区。
memory.cached	以字节为单位的 OS 缓存。
swap.total	主机上的总交换内存（以字节为单位）。

名称	描述
<code>swap.free</code>	以字节为单位在主机上交换内存（以字节为单位）。
<code>swap.used</code>	交换内存，单位为主机上使用的字节数。
<code>swap.cached</code>	以字节为单位的交换内存也缓存在主机的内存中。
<code>ksm.cpu.current</code>	内核同页合并的 CPU 使用量百分比。
<code>cpu.current.user</code>	用户片段的 CPU 使用量百分比。
<code>cpu.current.system</code>	系统的 CPU 使用量百分比。
<code>cpu.current.idle</code>	空闲 CPU 使用率的百分比。
<code>cpu.load.avg.5m</code>	每五分钟的 CPU 负载平均值。
<code>boot.time</code>	计算机的启动时间。

7.125. HOSTCPUUNIT STRUCT

类型，代表主机的物理 CPU，当前具有当前固定状态。

表 7.169. 属性概述

名称	类型	概述
<code>comment</code>	字符串	包含有关此对象的注释的免费文本。
<code>core_id</code>	整数	CPU 所属的内核的 ID。
<code>cpu_id</code>	整数	CPU 的 ID。
<code>description</code>	字符串	以纯文本格式人类可读的描述。
<code>id</code>	字符串	唯一标识符。
<code>name</code>	字符串	纯文本中的人类可读名称。
<code>runs_vdsm</code>	布尔值	指明 CPU 运行 VDSM 的标志
<code>socket_id</code>	整数	CPU 所属的插槽的 id。

表 7.170. 链接摘要

名称	类型	概述
vms	Vm[]	具有其虚拟 CPU 固定到此物理 CPU 的虚拟机列表。

7.126. HOSTDEVICE STRUCT

表 7.171. 属性概述

名称	类型	概述
功能	字符串	
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
driver	字符串	此设备绑定到的驱动程序名称。
id	字符串	唯一标识符。
iommu_group	整数	
m_dev_types	MDevType[]	物理设备上所有支持的 mdev 类型的列表，
name	字符串	纯文本中的人类可读名称。
physical_function	HostDevice	
占位符	布尔值	
product	产品	
vendor	vendor	
virtual_functions	整数	

7.126.1. 驱动程序

此设备绑定到的驱动程序名称。

例如：`pcieport` 或 `wonderci_hcd`。

表 7.172. 链接摘要

名称	类型	概述
<code>host</code>	主机	
<code>parent_device</code>	HostDevice	
<code>vm</code>	VM	

7.127. HOSTDEVICEPASSTHROUGH STRUCT

表 7.173. 属性概述

名称	类型	概述
<code>enabled</code>	布尔值	

7.128. HOSTNIC STRUCT

表示主机 NIC。

例如，主机 NIC 的 XML 表述类似如下：

```
<host_nic href="/ovirt-engine/api/hosts/123/nics/456" id="456">
  <name>eth0</name>
  <boot_protocol>static</boot_protocol>
  <bridged>true</bridged>
  <custom_configuration>true</custom_configuration>
  <ip>
    <address>192.168.122.39</address>
    <gateway>192.168.122.1</gateway>
    <netmask>255.255.255.0</netmask>
    <version>v4</version>
  </ip>
  <ipv6>
    <gateway>::</gateway>
    <version>v6</version>
  </ipv6>
  <ipv6_boot_protocol>none</ipv6_boot_protocol>
  <mac>
    <address>52:54:00:0c:79:1d</address>
  </mac>
</host_nic>
```

```
<mtu>1500</mtu>
<status>up</status>
</host_nic>
```

绑定的接口表示为含有 *绑定* 和 *从属* 属性的 **HostNic** 对象。

例如，绑定主机 NIC 的 XML 表述类似如下：

```
<host_nic href="/ovirt-engine/api/hosts/123/nics/456" id="456">
  <name>bond0</name>
  <mac address="00:00:00:00:00:00"/>
  <ip>
    <address>192.168.122.39</address>
    <gateway>192.168.122.1</gateway>
    <netmask>255.255.255.0</netmask>
    <version>v4</version>
  </ip>
  <boot_protocol>dhcp</boot_protocol>
  <bonding>
    <options>
      <option>
        <name>mode</name>
        <value>4</value>
        <type>Dynamic link aggregation (802.3ad)</type>
      </option>
      <option>
        <name>miimon</name>
        <value>100</value>
      </option>
    </options>
    <slaves>
      <host_nic id="123"/>
      <host_nic id="456"/>
    </slaves>
  </bonding>
  <mtu>1500</mtu>
  <bridged>true</bridged>
  <custom_configuration>false</custom_configuration>
</host_nic>
```

表 7.174. 属性概述

名称	类型	概述
ad_aggregator_id	整数	bond 或 bond slave 的 ad_aggregator_id 属性，用于模式 4 中的绑定。
base_interface	字符串	NIC 的基础接口。
bonding	bonding	NIC 的绑定参数。

名称	类型	概述
boot_protocol	BootProtocol	NIC 的 IPv4 引导协议配置。
bridged	布尔值	定义桥接网络状态。
check_connectivity	布尔值	
comment	字符串	包含有关此对象的注释的免费文本。
custom_configuration	布尔值	
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
ip	ip	NIC 的 IPv4 地址。
ipv6	ip	NIC 的 IPv6 地址。
ipv6_boot_protocol	BootProtocol	NIC 的 IPv6 引导协议配置。
MAC	MAC	NIC 的 MAC 地址。
mtu	整数	接口的最大传输单元。
name	字符串	纯文本中的人类可读名称。
override_configuration	布尔值	
属性	attribute[]	
速度	整数	
status	NicStatus	
virtual_functions_configuration	HostNicVirtualFunctionsConfiguration	描述物理功能 NIC 的虚拟功能配置。
vlan	VLAN	

7.128.1. ad_aggregator_id

bond 或 **bond slave** 的 **ad_aggregator_id** 属性，用于模式 4 中的绑定。绑定模式 4 是 802.3ad 标准，也称为动态链路聚合。（请参阅[维基百科与 演示以了解详细信息](#)）。这只适用于模式 4 中的绑定，或者对作为绑定一部分的 NIC 有效。其他模式中的绑定或模式 4 中不属于绑定的 NIC 不存在。ad_aggregator_id 属性指示哪个绑定从设备处于活动状态。活跃 slave 的 ad_aggregator_id 值与绑定的 ad_aggregator_id 属性的值相同。此参数为只读。设置它不会影响 bond/NIC。它从绑定的 `/sys/class/net/bondX/bonding/ad_aggregator` 文件和一个 NIC 的 `/sys/class/net/ensX/bonding_slave/ad_aggregator_id` 文件中检索。

7.128.2. bridged

定义桥接网络状态。网桥网络设置为 **true**，无网桥网络 设为 **false**。

表 7.175. 链接摘要

名称	类型	概述
host	主机	
network	Network	对接口应连接到的网络的引用。
network_labels	NetworkLabel[]	应用到这个 NIC 的标签。
physical_function	HostNic	对 SR-IOV 虚拟功能 NIC 的物理功能 NIC 的引用。
qos	QoS	到接口服务质量配置的连接。
statistics	Statistic[]	NIC 统计信息的链接。

7.128.3. network

对接口应连接到的网络的引用。允许空白网络 ID。

7.128.4. statistics

NIC 统计信息的链接。

HostNic 统计值的数据类型：

- **data.current.rx** - 收到的数据的每秒字节率。

- `data.current.tx` - 传输数据的每秒字节率。
- `data.current.rx.bps` - 收到的数据的每秒位数（自版本 4.2 起）。
- `data.current.tx.bps` - 传输的数据每秒的位数（自版本 4.2 起）。
- `data.total.rx` - 收到的数据总数。
- `data.total.tx` - 总传输的数据。
- `error.total.rx` - 接收数据的总错误。
- `error.total.tx` - 传输数据的总错误。

7.129. HOSTNICVIRTUALFUNCTIONSCONFIGURATION STRUCT

描述支持 SR-IOV 的物理功能 NIC 的虚拟功能配置。

表 7.176. 属性概述

名称	类型	概述
<code>all_networks_all_owed</code>	布尔值	定义是否允许在相关虚拟功能上定义所有网络，还是仅指定网络。
<code>max_number_of_virtual_functions</code>	整数	NIC 支持的最大虚拟功能数量。
<code>number_of_virtual_functions</code>	整数	当前定义的虚拟功能数量。

7.129.1. `max_number_of_virtual_functions`

NIC 支持的最大虚拟功能数量。此属性为只读。

7.129.2. *number_of_virtual_functions*

当前定义的虚拟功能数量。0 和 *max_number_of_virtual_functions* 之间的用户定义的值。

7.130. *HOSTPROTOCOL* 枚举

引擎用于与主机通信的协议。



警告

自引擎版本 4.1 起，自 *xml* 删除后，协议始终设置为 *stomp*。

表 7.177. 值概述

名称	概述
stomp	位于 STOMP 上的 JSON-RPC 协议。
xml	XML-RPC 协议。

7.131. *HOSTSTATUS ENUM*

键入代表主机状态。

表 7.178. 值概述

名称	概述
connecting	引擎无法针对特定阈值与主机通信，因此现在在通过隔离前尝试连接。
down	主机已停机。
错误	主机处于错误状态。
initializing	主机正在初始化。
install_failed	主机安装失败。

名称	概述
installing	主机正在安装。
installing_os	主机操作系统现已安装。
kdumping	主机内核已崩溃，现在已经经历内存转储。
维护	主机处于维护状态。
non_operational	该主机不正常运行。
non_responsive	主机未响应。
pending_approval	主机正在等待管理员批准。
preparing_for_maintenance	主机正在准备维护。
reboot	主机正在重新启动。
unassigned	主机处于激活过程中。
up	主机已启动。

7.131.1. 错误

主机处于错误状态。如果我们尝试多次运行虚拟机并且将失败，则会出现此情况。

7.131.2. 初始化

主机正在初始化。这是将主机移至"启动"状态之前的一个中间步骤。

7.131.3. install_failed

主机安装失败。在这种情况下，请查看事件日志以了解安装失败并重新安装。

7.131.4. installing_os

主机操作系统现已安装。使用卫星/Foreman 提供程序并发布裸机调配（发现主机调配）时，此状态很重要。

7.131.5. 维护

主机处于维护状态。当主机处于维护状态时，它无法运行虚拟机。

7.131.6. non_operational

该主机不正常运行。由于各种原因（例如，与存储没有连接、不支持专用网络、不支持集群级别等），可能会发生这种情况。

7.131.7. non_responsive

主机未响应。这意味着引擎无法与主机通信。

7.131.8. pending_approval

主机正在等待管理员批准。这只适用于 `vintage ovirt-node / RHV-H`。此属性不再相关，因为 Vintage 节点不再被支持，且已弃用。

7.131.9. preparing_for_maintenance

主机正在准备维护。在这一时间段内，引擎确保将所有虚拟机从此主机实时迁移到其他主机。完成所有迁移后，主机将进入“维护”状态。

7.132. HOSTSTORAGE STRUCT

表 7.179. 属性概述

名称	类型	概述
地址	字符串	
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
driver_options	attribute[]	使用 Cinder 驱动程序创建存储域时要传递的选项。

名称	类型	概述
driver_sensitive_options	attribute[]	包含敏感信息的参数，在使用 Cinder 驱动程序创建存储域时传递。
id	字符串	唯一标识符。
logical_units	LogicalUnit[]	
mount_options	字符串	
name	字符串	纯文本中的人类可读名称。
nfs_retrans	整数	在尝试进一步恢复操作前重试请求的次数。
nfs_timeo	整数	在重试 NFS 请求前等待响应的秒中的时间。
nfs_version	NfsVersion	
override_luns	布尔值	
password	字符串	
路径	字符串	
port	整数	
portal	字符串	
目标	字符串	
type	StorageType	
username	字符串	
vfs_type	字符串	
volume_group	VolumeGroup	

7.132.1. driver_options

使用 Cinder 驱动程序创建存储域时要传递的选项。

例如 (Kaminario 后端) :

POST /ovirt-engine/api/storagedomains/

```
<storage_domain>
  <name>kamniraio-cinder</name>
  <type>managed_block_storage</type>
  <storage>
    <type>managed_block_storage</type>
    <driver_options>
      <property>
        <name>san_ip</name>
        <value>192.168.1.1</value>
      </property>
      <property>
        <name>san_login</name>
        <value>username</value>
      </property>
      <property>
        <name>san_password</name>
        <value>password</value>
      </property>
      <property>
        <name>use_multipath_for_image_xfer</name>
        <value>>true</value>
      </property>
      <property>
        <name>volume_driver</name>
        <value>cinder.volume.drivers.kaminario.kaminario_iscsi.KaminarioISCSIDriver</value>
      </property>
    </driver_options>
  </storage>
  <host>
    <name>host</name>
  </host>
</storage_domain>
```

7.132.2. driver_sensitive_options

包含敏感信息的参数，在使用 **Cinder** 驱动程序创建存储域时传递。这些参数在保存时会加密。

例如：以下 XML 加密并保存用户名、密码和 SAN IP 地址：

POST /ovirt-engine/api/storagedomains/

```
<storage_domain>
  <name>kamniraio-cinder</name>
  <type>managed_block_storage</type>
  <storage>
    <type>managed_block_storage</type>
    <driver_options>
      <property>
```

```

    <name>san_ip</name>
    <value>192.168.1.1</value>
  </property>
</property>
  <name>san_login</name>
  <value>username</value>
</property>
</property>
  <name>san_password</name>
  <value>password</value>
</property>
</property>
  <name>use_multipath_for_image_xfer</name>
  <value>>true</value>
</property>
</property>
  <name>volume_driver</name>
  <value>cinder.volume.drivers.kaminario.kaminario_iscsi.KaminarioISCSIDriver</value>
</property>
</driver_options>
<driver_sensitive_options>
  <property>
    <name>username</name>
    <value>admin</value>
  </property>
  <property>
    <name>password</name>
    <value>123</value>
  </property>
  <property>
    <name>san_ip</name>
    <value>192.168.1.1</value>
  </property>
</driver_sensitive_options>
</storage>
<host>
  <name>host</name>
</host>
</storage_domain>

```

7.132.3. nfs_retrans

在尝试进一步恢复操作前重试请求的次数。该值必须在 0 到 65535 之间。详情请查看 *nfs man page* 中 *retrans* 挂载选项的描述。

7.132.4. nfs_timeo

在重试 NFS 请求前等待响应的秒中的时间。该值必须在 0 到 65535 之间。如需了解更多详细信息，请参阅 *nfs man page* 中 *timeo* 挂载选项的描述。

表 7.180. 链接摘要

名称	类型	概述
host	主机	

7.133. HOSTTYPE ENUM

此枚举类型用于确定主机使用哪种类型的操作系统。

表 7.181. 值概述

名称	概述
ovirt_node	主机包含红帽虚拟化主机(RHVH)：红帽企业虚拟化系统管理程序(RHEV-H)的全新实施，它使用与红帽企业 Linux、CentOS 或 Fedora 相同的安装程序。
rhel	该主机包含完整的 Red Hat Enterprise Linux、CentOS 或 Fedora 安装。
rhev_h	该主机包含红帽企业虚拟化管理程序(RHEV-H)，这是红帽企业 Linux、CentOS 或 Fedora 的小型版本，专门用于托管虚拟机。

7.133.1. ovirt_node

主机包含红帽虚拟化主机(RHVH)：红帽企业虚拟化系统管理程序(RHEV-H)的全新实施，它使用与红帽企业 Linux、CentOS 或 Fedora 相同的安装程序。RHVH 与传统 RHEV-H 之间的主要区别在于，RHVH 具有可写入文件系统，并将自行处理安装，而不是像在传统 RHEV-H 中一样由管理器推送 RPM。

7.133.2. rhev_h

该主机包含红帽企业虚拟化管理程序(RHEV-H)，这是红帽企业 Linux、CentOS 或 Fedora 的小型版本，专门用于托管虚拟机。

此属性不再相关，因为 Vintage 节点不再被支持，且已弃用。

7.134. HOSTEDENGINE STRUCT

表 7.182. 属性概述

名称	类型	概述
Active	布尔值	

名称	类型	概述
已配置	布尔值	
global_maintenance	布尔值	
local_maintenance	布尔值	
分数	整数	

7.135. ICON STRUCT

虚拟机或模板的图标。

表 7.183. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
data	字符串	Base64 编码图标文件的内容。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
media_type	字符串	图标文件格式。
name	字符串	纯文本中的人类可读名称。

7.135.1. media_type

图标文件格式。

其中之一：

- `image/jpeg`

- *image/png*
- *image/gif*

7.136. IDENTIFIED STRUCT

这个接口是所有代表有标识符的对象类型的基本模型。

表 7.184. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

7.137. 镜像 结构

代表镜像实体。

表 7.185. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
Size	整数	镜像文件的大小。
type	ImageFileType	镜像文件的类型。

表 7.186. 链接摘要

名称	类型	概述
storage_domain	StorageDomain	与此映像关联的存储域。

7.138. IMAGEFILETYPE ENUM

表示镜像的文件类型。

表 7.187. 值概述

名称	概述
disk	映像是一种磁盘格式，可用作虚拟机的磁盘。
floppy	映像是可附加到虚拟机的软盘，例如在 Windows 中安装 VirtIO 驱动程序。
iso	图像是'

7.138.1. iso

映像是 a .iso 文件，可用作 CD-ROM 来引导和安装虚拟机。

7.139. IMAGETRANSFER STRUCT

此类型包含有关正在执行的映像传输的信息。

表 7.188. 属性概述

名称	类型	概述
Active	布尔值	指明此传输是否至少有一个活跃的会话，即客户端和守护进程之间至少有一个实时迁移会话。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
方向	ImageTransferDirection	该方向指示传输是发送镜像数据（上传）还是接收镜像数据（下载）。
format	DiskFormat	上传期间上传或收到的数据格式。

名称	类型	概述
id	字符串	唯一标识符。
inactivity_timeout	整数	不活跃客户端的超时时间，之后被 Red Hat Virtualization Manager 中止。
name	字符串	纯文本中的人类可读名称。
phase	ImageTransferPhase	镜像传输的当前阶段正在进行中。
proxy_url	字符串	用户输入或输出的代理服务器的 URL。
shallow	布尔值	仅下载指定的映像，而不是整个镜像链。
timeout_policy	ImageTransferTimeoutPolicy	超时策略决定当客户端闲置超过不活动超时系统如何处理传输。
transfer_url	字符串	用户可以直接输入或输出的守护进程服务器的 URL。
transferred	整数	表示已传输的字节数。

7.139.1. 方向

该方向指示传输是发送镜像数据（上传）还是接收镜像数据（下载）。

如果在添加新传输期间未设置方向，则上传传输的默认方向。

7.139.2. 格式

上传期间上传或收到的数据格式。如果没有指定，则默认为磁盘格式。

7.139.3. inactivity_timeout

不活跃客户端的超时时间，之后被 Red Hat Virtualization Manager 中止。要禁用不活动超时，请指定 '0'。如果没有指定，则该值默认为 engine-config 值：
`TransferImageClientInactivityTimeoutInSeconds`。

7.139.4. phase

镜像传输的当前阶段正在进行中。每次传输都需要一个托管的会话，必须打开该会话供用户输入或输出镜像。有关更多文档，请参阅[图像传送](#)。

7.139.5. proxy_url

用户输入或输出的代理服务器的 URL。只有镜像传输处于传输阶段时，此属性才可用。详情请查看[阶段](#)。

7.139.6. Kow

仅下载指定的映像，而不是整个镜像链。

如果为 `true`，在使用格式=`"raw"` 和方向=`"download"` 时，传输仅包含来自指定磁盘快照的数据，未分配区域则报告为漏洞。默认情况下，传输包含来自所有磁盘快照的数据。

指定磁盘快照时，传输仅包含指定磁盘快照的数据。指定磁盘时，传输仅包含来自活动磁盘 `snaphost` 的数据。

如果没有使用 `format="raw"` 或方向=`"upload"`，这个参数无效。

示例：下载一个快照：

```
<image_transfer>
  <snapshot id="2fb24fa2-a5db-446b-b733-4654661cd56d"/>
  <direction>download</direction>
  <format>raw</format>
  <shallow>true</shallow>
</image_transfer>
```

要下载活跃的快照磁盘镜像（这不能作为磁盘快照访问），请指定磁盘：

```
<image_transfer>
  <disk id="ff6be46d-ef5d-41d6-835c-4a68e8956b00"/>
  <direction>download</direction>
  <format>raw</format>
  <shallow>true</shallow>
</image_transfer>
```

在这两种情况下，您现在可以使用 `imageio` 客户端下载 `qcow2` 镜像：

```
from ovirt_imageio import client
```

```
client.download(
    transfer.transfer_url,
    "51275e7d-42e9-491f-9d65-b9211c897eac",
    backing_file="07c0ccac-0845-4665-9097-d0a3b16cf43b",
    backing_format="qcow2")
```

7.139.7. transfer_url

用户可以直接输入或输出的守护进程服务器的 URL。

这是 `proxy_url` 的替代选择。例如，如果客户端可以访问主机计算机，它可以绕过代理并直接传输至主机，从而可能提高吞吐量性能。只有镜像传输处于传输阶段时，此属性才可用。详情请查看 [阶段](#)。

表 7.189. 链接摘要

名称	类型	概述
<code>backup</code>	Backup	与映像传输关联的备份。
<code>disk</code>	Disk	用于输入或输出的磁盘。
<code>host</code>	主机	用于写入映像的主机，该映像作为输入或输出目标。
<code>image</code>	Image	用于输入或输出的镜像。
<code>snapshot</code>	DiskSnapshot	用于输入或输出的磁盘快照。

7.139.8. backup

与映像传输关联的备份。在为作为备份一部分的磁盘启动镜像传输时指定。

7.139.9. 主机

用于写入映像的主机，该映像作为输入或输出目标。如果未指定，将从数据中心随机选择活动主机。

7.139.10. image

用于输入或输出的镜像。



重要

此属性已弃用，因为引擎的版本 4.2。改为使用 `disk` 或 `snapshot` 属性。

7.140. IMAGETRANSFERDIRECTION ENUM

镜像传输方向。

在添加新传输时，用户可以选择转让是至映像，选择上传，还是从映像中进行传输 - 选择 `download` 作为 `ImageTransferDirection`。

有关更多文档，请参阅[图像传送](#)。

表 7.190. 值概述

名称	概述
下载	当用户想从镜像流传输数据时，用户必须选择 <code>download</code> 。
upload	当用户想要将数据流传输到镜像时，用户可以选择 <code>upload</code> 。

7.141. IMAGETRANSFERPHASE ENUM

映像传输实体可能阶段的列表。这些值各自定义传输流中的特定点。

如需更多信息，请参阅[映像传输](#)。

表 7.191. 值概述

名称	概述
cancelled	此阶段将因用户取消转让而设置。
cancelled_system	此阶段将因系统取消传输而设置。

名称	概述
<code>cancelled_user</code>	此阶段将因用户取消转让而设置。
<code>finalizing_cleanup</code>	此阶段表示用户取消转让，正在进行必要的清理。
<code>文莱_failure</code>	此阶段只能在管理门户中设置，并且表示转移期间出现错误，而且正在最终完成该阶段失败。
<code>finalizing_success</code>	此阶段将在用户调用完成时设置。
<code>finished_cleanup</code>	此阶段表示用户取消转让，并进行了必要的清理。
<code>finished_failure</code>	表示目标镜像失败验证，且无法使用。
<code>finished_success</code>	表示传输会话已成功关闭，目标镜像已验证并就绪可用。
<code>initializing</code>	映像传输的初始阶段。
<code>paused_system</code>	此阶段表示会话超时，或者出现此传输时出现的一些其他错误；例如 <code>ovirt-imageio-daemon</code> 没有在所选主机上运行。
<code>paused_user</code>	此阶段是因为用户使用 <code>pause</code> 进行暂停调用。
恢复	客户端调用恢复了转让的阶段。
传输	打开传输会话的阶段，客户端可以使用首选工具输入或输出所需的镜像。
<code>Unknown</code>	一个未知的阶段。

7.141.1. `cancelled`

此阶段将因用户取消转让而设置。取消只能在管理门户中执行。

7.141.2. `finalizing_success`

此阶段将在用户调用完成时设置。调用完成对完成传输会话非常重要，并使用目标镜像完成。完成之后，该阶段将更改为 `completed_success` 或 `completed_failure`。

如需更多信息，请参阅[镜像传输](#)。

7.141.3. finished_failure

表示目标镜像失败验证，且无法使用。达到此阶段后，镜像传输实体将被删除，目标映像将设置为非法。系统取消传输也将导致。

7.141.4. finished_success

表示传输会话已成功关闭，目标镜像已验证并就绪可用。达到此阶段后，映像传输实体将被删除。

7.141.5. 初始化

映像传输的初始阶段。它在转移会话建立期间设定。会话设定后，阶段将更改为 `传输`

7.141.6. paused_system

此阶段表示会话超时，或者出现此传输时出现的一些其他错误；例如 `ovirt-imageio-daemon` 没有在选择主机上运行。要恢复会话，客户端应调用 `restore`。恢复后，该阶段将变为 `恢复`。

7.141.7. 恢复

客户端调用恢复了转让的阶段。恢复一个新的会话，在调用新会话后，阶段将更改为 `传输` 阶段，如果失败，则会更改为 `paused_system`。

7.141.8. Unknown

一个未知的阶段。只有在出现无法预计的错误时才会设置此设置。

7.142. IMAGETRANSFERTIMEOUTPOLICY ENUM

[镜像传输超时策略](#)。

定义当客户端在不活动超时秒内不活动时系统如何处理传输。

有关更多文档，请参阅[图像传送](#)。

表 7.192. 值概述

名称	概述
取消	取消传输并解锁磁盘。
legacy	强制政策将保留默认设置的传统功能。
pause	暂停转让。

7.142.1. 取消

取消传输并解锁磁盘。对于使用上传方向进行镜像传输，该磁盘将被删除。

7.142.2. legacy

强制政策将保留默认设置的传统功能。如果方向是下载，则默认行为将取消转让，并在上传该方向时暂停传输。

7.142.3. pause

暂停转让。用户可恢复或取消转让。传输暂停时磁盘将保持锁定状态。

7.143. INHERITABLEBOOLEAN ENUM

代表可以设置或从更高级别上继承的布尔值的枚举。继承顺序为虚拟机 → cluster → engine-config。

表 7.193. 值概述

名称	概述
false	在此级别上将值设为 false。
inherit	继承更高级别的值。
true	在此级别上将值设为 true。

7.144. INITIALIZATION STRUCT

表 7.194. 属性概述

名称	类型	概述
<code>active_directory_ou</code>	字符串	
<code>authorized_ssh_keys</code>	字符串	
<code>cloud_init</code>	CloudInit	指定 <code>cloud-init</code> 配置的已弃用属性。
<code>cloud_init_network_protocol</code>	CloudInitNetworkProtocol	属性指定用于格式化 cloud-init 网络参数的 cloud-init 协议。
配置	Configuration	
<code>custom_script</code>	字符串	
<code>dns_search</code>	字符串	
<code>dns_servers</code>	字符串	
<code>domain</code>	字符串	
<code>host_name</code>	字符串	
<code>input_locale</code>	字符串	
<code>nic_configurations</code>	NicConfiguration[]	
<code>org_name</code>	字符串	
<code>regenerate_ids</code>	布尔值	
<code>regenerate_ssh_keys</code>	布尔值	
<code>root_password</code>	字符串	
<code>system_locale</code>	字符串	
<code>timezone</code>	字符串	
<code>ui_language</code>	字符串	

名称	类型	概述
user_locale	字符串	
user_name	字符串	
windows_license_key	字符串	

7.144.1. cloud_init

指定 cloud-init 配置的已弃用属性。

此属性和 [CloudInit](#) 类型已弃用，并将在以后的中删除。要指定 cloud-init 配置，请使用 [Initialization](#) 类型内的属性。这两种类型的属性之间的映射如下：

CloudInit	初始化
authorized_keys	authorized_ssh_keys
dns.search_domains	dns_search
dns.servers	dns_servers
文件	custom_script
host	host_name
network_configuration.nics	nic_configurations
regenerate_ssh_keys	regenerate_ssh_keys
timezone	timezone
用户	user_name & root_password

有关如何使用 cloud-init 的详情，请查看 [Python](#) 和 [Java](#) 中的示例。

7.144.2. cloud_init_network_protocol

属性指定用于格式化 cloud-init 网络参数的 cloud-init 协议。如果省略，则会使用默认值，如

CloudInitNetworkProtocol所述

7.145. INSTANCETYPE STRUCT

描述虚拟机的硬件配置。

例如，Migration 实例类型包括 1 个虚拟 CPU 和 4 GiB 内存。它是顶级实体（例如，不绑定到任何数据中心或群集）。用于实例类型以及虚拟机和模板类型常用的属性有：

- console
- cpu
- custom_cpu_model
- custom_emulated_machine
- display
- high_availability
- io
- 内存
- memory_policy
- 迁移

- ***migration_downtime***
- ***OS***
- ***rng_device***
- ***soundcard_enabled***
- ***usb***
- ***virtio_scsi***

从实例类型和模板创建虚拟机时，虚拟机将从实例类型继承硬件配置



注意

实例类型继承模板实体的属性，尽管大多数模板属性不在实例类型中使用。

表 7.195. 属性概述

名称	类型	概述
auto_pinning_policy	AutoPinningPolicy	指定是否以及如何应用自动 CPU 和 NUMA 配置。
bios	BIOS	参考虚拟机的 BIOS 配置。
comment	字符串	包含有关此对象的注释的免费文本。
console	控制台 (Console)	为此虚拟机配置的控制台。
cpu	cpu	虚拟机 CPU 的配置。
cpu_pinning_policy	CpuPinningPolicy	指定是否以及如何应用 CPU 和 NUMA 配置。

名称	类型	概述
cpu_shares	整数	
creation_time	Date	虚拟机创建日期。
custom_compatibility_version	版本	虚拟机自定义兼容性版本。
custom_cpu_model	字符串	
custom_emulated_machine	字符串	
custom_properties	CustomProperty[]	发送到 VDSM 的属性，以配置各种 hook。
delete_protected	布尔值	如果为 true ，则无法删除虚拟机。
description	字符串	以纯文本格式人类可读的描述。
display	显示	虚拟机显示配置。
domain	域	为此虚拟机配置的域。
high_availability	HighAvailability	虚拟机高可用性配置。
id	字符串	唯一标识符。
初始化	初始化	引用虚拟机的初始化配置。
io	io	用于 IO 线程性能调优。
large_icon	icon	虚拟机的大图标。
lease	StorageDomainLease	虚拟机/模板租用所在存储域的引用。
内存	整数	虚拟机的内存，以字节为单位。
memory_policy	MemoryPolicy	参考虚拟机的内存管理配置。
迁移	MigrationOptions	有关将正在运行的虚拟机迁移到另一台主机的配置。
migration_down_time	整数	虚拟机在实时迁移到 ms 中另一主机期间可以不响应的最大时间。

名称	类型	概述
multi_queues_enabled	布尔值	如果为 true ，则 每个虚拟接口都将获得最佳队列数，具体取决于可用的虚拟 Cpus。
name	字符串	纯文本中的人类可读名称。
Origin	字符串	此虚拟机的起源。
OS	OperatingSystem	虚拟机上安装的操作系统类型。
placement_policy	VmPlacementPolicy	虚拟机放置策略的配置。
rng_device	RngDevice	此虚拟机的随机数字生成器设备配置。
serial_number	SerialNumber	集群中虚拟机的序列号。
small_icon	icon	虚拟机的小图标。
soundcard_enabled	布尔值	如果为 true ，则会在虚拟机中添加声卡。
ssso	SSO	此虚拟机被配置至单点登录配置的引用。
start_paused	布尔值	如果为 true ，则虚拟机将在启动后的初始处于"暂停"状态。
stateless	布尔值	如果为 true ，则虚拟机为无状态 - 它的状态（磁盘）会在关机后回滚。
status	TemplateStatus	模板的状态。
storage_error_resume_behavior	VmStorageErrorResumeBehaviour	确定存储错误后如何恢复虚拟机。
time_zone	timezone	虚拟机的时区由 oVirt 设置。
tpm_enabled	布尔值	如果为 true ，则会在虚拟机中添加 TPM 设备。
tunnel_migration	布尔值	如果为 true ，则网络数据传输将在虚拟机实时迁移期间加密。
type	VmType	确定虚拟机是为桌面还是服务器优化。
usb	usb	配置此虚拟机的 USB 设备（count、类型）。
version	TemplateVersion	指明这是基本版本还是另一个模板的子版本。

名称	类型	概述
<code>virtio_scsi</code>	VirtioScsi	参考 VirtIO SCSI 配置。
<code>virtio_scsi_multi_queues</code>	整数	这个字段的 Virtio-SCSI controller 队列数量需要 <code>virtioScsiMultiQueuesEnabled</code> 。如需更多信息，请参阅 <code>virtioScsiMultiQueuesEnabled</code>
<code>virtio_scsi_multi_queues_enabled</code>	布尔值	如果为 true ，Virtio-SCSI 设备会根据可用的虚拟 Cpus 和磁盘或根据指定的 <code>virtioScsiMultiQueues</code> 来获得多个队列。
<code>vm</code>	VM	与此模板关联的虚拟机配置。

7.145.1. auto_pinning_policy

指定是否以及如何应用自动 CPU 和 NUMA 配置。



重要

由于引擎的 4.5 版已弃用，因此仅保留以向后兼容。以后可能会删除它。请使用 `CpuPinningPolicy`。

7.145.2. cpu

虚拟机 CPU 的配置。

可以在不重新启动虚拟机的情况下更新套接字配置。内核和线程需要重启。

例如：要将套接字数量立即更改为 4，并在重启后将内核和线程数量改为 2，发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```

通过请求正文：

```
<vm>
```

```

<cpu>
  <topology>
    <sockets>4</sockets>
    <cores>2</cores>
    <threads>2</threads>
  </topology>
</cpu>
</vm>

```

7.145.3. cpu_pinning_policy

指定是否以及如何应用 CPU 和 NUMA 配置。如果没有指定 CPU 固定字符串之前的行为，则会决定 CpuPinningPolicy 到 None 或 Manual。

7.145.4. custom_compatibility_version

虚拟机自定义兼容性版本。

使虚拟机能够根据自己的兼容性版本进行自定义。如果设置了 custom_compatibility_version，它将覆盖该特定虚拟机的集群的兼容性版本。

虚拟机的兼容性版本受位于的数据中心的限制，并根据虚拟机计划在其上运行的主机进行检查。

7.145.5. high_availability

虚拟机高可用性配置。如果设置，虚拟机将在意外停机时自动重新启动。

7.145.6. 初始化

引用虚拟机的初始化配置。



注意

自 Red Hat Virtualization 4.1.8 起，可以通过发送空标签来清除此属性。

例如，清除 初始化 属性会发送类似如下的请求：


```
PUT /ovirt-engine/api/vms/123
```

这样的请求正文：

```
<vm>
  <initialization/>
</vm>
```

对此类请求的响应和带有标头 **All-Content: true** 的请求仍将包含此属性。

7.145.7. large_icon

虚拟机的大图标。用户设置或引用根据操作系统设置的镜像集。

7.145.8. lease

虚拟机/模板租用所在存储域的引用。

使用租用运行的虚拟机需要检查运行时不会由另一主机获取租期，从而防止此虚拟机的另一个实例在另一主机上运行。这可防止高可用性虚拟机中的脑裂。模板也可为租用定义存储域，以便从该模板创建虚拟机，预先配置此存储域作为租用位置。

7.145.9. 内存

虚拟机的内存，以字节为单位。

例如，要更新虚拟机使其包含 1 Gibibyte(GiB)内存，请发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```

使用以下请求正文：

```
<vm>
  <memory>1073741824</memory>
</vm>
```

Red Hat Virtualization 3.6 支持内存热插拔。在虚拟机处于 **up** 状态时，您可以使用上述示例来增加内存。增量的大小必须可以被分成 `HotPlugMemoryBlockSizeMb` 配置值（默认为256 MiB）。如果此值无法分隔内存大小递增，则内存大小更改仅存储在下次运行配置中。每个成功的内存热插拔操作都会创建一个或两个新内存设备。

从 Red Hat Virtualization 4.2 开始支持内存热插拔。**内存热拔头只能在虚拟机处于启动状态时执行**。只有之前热插拔内存设备才能通过热拔出操作删除。请求的内存减少向下舍入，以匹配之前热插拔内存设备的组合大小。请求的内存值存储在下次运行配置中，而不舍入。



注意

这个示例中的内存会被转换为字节：
 $1 \text{ GiB} = 2^{30} \text{ 字节} = 1073741824 \text{ 字节}$ 。



注意

红帽虚拟化管理器内部将值向下舍入到整个 MiB ($1 \text{ MiB} = 2^{20} \text{ 字节}$)

7.145.10. migration_downtime

虚拟机在实时迁移到 ms 中另一主机期间可以不响应的最大时间。

为虚拟机明确设置，或者使用 `engine-config -s DefaultMaximumMigrationDowntime=[value]`

7.145.11. Origin

此虚拟机的起源。

可能的值：

- `ovirt`
- `rhev`

- `vmware`
- `Xen`
- `external`
- `hosted_engine`
- `managed_hosted_engine`
- `kvm`
- `physical_machine`
- `hyperv`

7.145.12. `placement_policy`

虚拟机放置策略的配置。

此配置可以更新为将虚拟机固定到一个或多个主机。



注意

固定到多个主机的虚拟机无法实时迁移，但在主机出现故障时，任何配置为高可用性的虚拟机都会自动在虚拟机要固定到的其他主机上重新启动。

例如，要将虚拟机固定到两个主机，请发送以下请求：

```
PUT /api/vms/123
```

这样的请求正文：

```
<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>
```

7.145.13. small_icon

虚拟机的小图标。用户设置或引用根据操作系统设置的镜像集。

7.145.14. sso

此虚拟机被配置至单点登录配置的引用。打开控制台时，可以在虚拟机的操作系统中自动签署该用户。

7.145.15. tpm_enabled

如果为 **true**，则会在虚拟机中添加 TPM 设备。默认值为 **false**。只有在设置了 "All-Content=true" 标头时，此属性才能在获取时可见。

表 7.196. 链接摘要

名称	类型	概述
cdroms	Cdrom[]	引用附加到模板的 CD-ROM 设备。
cluster	Cluster	虚拟机所属群集的引用。
cpu_profile	CpuProfile	引用此虚拟机使用的 CPU 配置集。

名称	类型	概述
disk_attachments	DiskAttachment[]	引用模板所连接的磁盘。
graphics_consoles	GraphicsConsole[]	引用附加到模板的图形控制台。
mediated_devices	VmMediatedDevice[]	介质设备配置。
nics	Nic[]	引用附加到模板的网络接口。
权限	permission[]	引用附加到模板的用户权限。
quota	Quota	引用为此虚拟机设置的配额配置。
storage_domain	StorageDomain	虚拟机所属存储域的引用。
tags	Tag[]	引用附加到模板的标签。
watchdogs	Watchdog[]	引用模板所连接的 watchdog 设备。

7.146. IO STRUCT

表 7.197. 属性概述

名称	类型	概述
线程	整数	

7.147. IP STRUCT

表示网络接口的 IP 配置。

表 7.198. 属性概述

名称	类型	概述
地址	字符串	IP 地址的文本表示。
gateway	字符串	默认网关的地址。
netmask	字符串	网络掩码。

名称	类型	概述
version	IpVersion	IP 协议的版本。

7.147.1. 地址

IP 地址的文本表示。

例如, IPv4 地址将如下所示 :

```
<ip>
  <address>192.168.0.1</address>
  ...
</ip>
```

IPv6 地址将按如下表示 :

```
<ip>
  <address>2620:52:0:20f0:4216:7eff:feaa:1b50</address>
  ...
</ip>
```

7.147.2. 子网掩码

网络掩码。

对于 IPv6 地址, 该值是 0-128 范围内的整数, 代表子网前缀。

7.147.3. version

IP 协议的版本。



注意

从 Manager 的 4.1 版本, 此属性为可选, 未提供值时, 将从 address 属性的值中推断出来。

7.148. IPADDRESSASSIGNMENT STRUCT

表示网络设备的 IP 地址分配。

对于静态启动协议分配，必须在 IP 配置中提供子网掩码和 IP 地址（以及可选的默认网关）。

表 7.199. 属性概述

名称	类型	概述
assignment_method	BootProtocol	设置用于分配网络设备 IP 配置的引导协议。
ip	ip	设置网络设备的 IP 配置。

7.149. IPVERSION ENUM

定义 IP 协议版本的值。

表 7.200. 值概述

名称	概述
v4	IPv4.
v6	IPv6.

7.150. ISCSIBOND STRUCT

表 7.201. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.202. 链接摘要

名称	类型	概述
data_center	DataCenter	
网络	network[]	
storage_connections	StorageConnections[]	

7.151. ISCSIDetails Struct

表 7.203. 属性概述

名称	类型	概述
地址	字符串	
disk_id	字符串	
initiator	字符串	
lun_mapping	整数	
password	字符串	
路径	整数	
port	整数	
portal	字符串	
product_id	字符串	
serial	字符串	
Size	整数	
status	字符串	
storage_domain_id	字符串	
目标	字符串	
username	字符串	

名称	类型	概述
vendor_id	字符串	
volume_group_id	字符串	

7.152. JOB STRUCT

表示作业，用于监控系统中某个流的执行。个作业可以在层次结构结构中包含多个步骤。这些步骤可以并行处理，具体取决于流程的实施。

表 7.204. 属性概述

名称	类型	概述
auto_cleared	布尔值	指明该作业应当在系统完成之后自动清除。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
end_time	Date	作业的结束时间。
external	布尔值	指明作业是否源自外部系统。
id	字符串	唯一标识符。
last_updated	Date	作业的最后更新日期。
name	字符串	纯文本中的人类可读名称。
start_time	Date	作业的开始时间。
status	JobStatus	作业的状态。

7.152.1. external

指明作业是否源自外部系统。外部作业由作业的创建者从外部进行管理。

表 7.205. 链接摘要

名称	类型	概述
所有者	用户	作业所有者的用户。
步骤	第[]步	作业的步骤。

7.153. JOBSTATUS ENUM

表示作业的状态。

表 7.206. 值概述

名称	概述
aborted	中止的作业状态。
Failed	失败的任务状态。
finished	完成的作业状态。
Started	启动的作业状态。
Unknown	未知的作业状态。

7.153.1. 中止

中止的作业状态。此状态适用于强制中止的外部作业。

7.153.2. 完成

完成的作业状态。此状态描述了已完成的作业执行。

7.153.3. Started

启动的作业状态。此状态表示当前正在执行的作业。

7.153.4. Unknown

未知的作业状态。此状态表示其解析未知的作业，即系统意外重启之前执行的作业。

7.154. KATELLOERRATUM STRUCT

键入代表 *Katello 勘误*。

表 7.207. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
issued	Date	签发 Katello 勘误的日期。
name	字符串	纯文本中的人类可读名称。
软件包	package[]	解决 Katello 勘误报告的问题的软件包列表。
严重性	字符串	Katello 勘误的严重性。
解决方案	字符串	Katello 勘误描述的问题的解决方案。
summary	字符串	Katello 勘误摘要。
title	字符串	Katello 勘误的标题。
type	字符串	Katello 勘误的类型。

7.154.1. 严重性

Katello 勘误的严重性。

支持的严重级别为 *中等*、*重要* 或 *关键*。

7.154.2. type

Katello 勘误的类型。

支持的类型有 **漏洞修复**、**增强** 或 **安全性**。

表 7.208. 链接摘要

名称	类型	概述
host	主机	引用 Katello 勘误所分配到的主机。
vm	VM	引用 Katello 勘误所分配到的虚拟机。

7.155. KDUMPSTATUS ENUM

表 7.209. 值概述

名称	概述
disabled	
enabled	
Unknown	

7.156. 内核 结构

表 7.210. 属性概述

名称	类型	概述
version	版本	

7.157. KSM STRUCT

表 7.211. 属性概述

名称	类型	概述
enabled	布尔值	
merge_across_nodes	布尔值	

7.158. LINKLAYERDISCOVERYPROTOCOLELEMENT STRUCT

代表由链路层发现协议(LLDP)接收的信息元素。IEEE 802.1AB 将类型、长度、值(TLV)定义为“短语，信息元素的变量长度编码”。此类型表示此类信息元素。

属性名称是一个人类可读的字符串，用于描述值的用途，可能不唯一。名称具有冗余性，因为它可以从类型、optional oui 和 subtype 创建。名称的目的是简化对信息元素的阅读。属性的名称与 IEEE 802.1AB 第 8 章中使用的字符串完全相同。

特定于组织的信息元素的类型为 127 种，以及属性和子类型。

例如，信息元素的 XML 表述可能类似如下：

```
<link_layer_discovery_protocol_element>
  <name>Port VLAN Id</name>
  <oui>32962</oui>
  <properties>
    <property>
      <name>vlan id</name>
      <value>488</value>
    </property>
    <property>
      <name>vlan name</name>
      <value>v2-0488-03-0505</value>
    </property>
  </properties>
  <subtype>3</subtype>
  <type>127</type>
</link_layer_discovery_protocol_element>
```

表 7.212. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
OUI	整数	组织唯一标识符(OUI)编码为整数。
属性	attribute[]	表示由信息元素传输的结构化数据作为名称/值对的列表。

名称	类型	概述
subtype	整数	组织化定义的子类型，编码为一个整数。
type	整数	LinkLayerDiscoveryProtocolElement 的类型编码为一个整数。

7.158.1. OUI

组织唯一标识符(OUI)编码为整数。仅在类型为 127 时可用。

7.158.2. 子类型

组织化定义的子类型，编码为一个整数。仅在类型为 127 时可用。

7.159. LOGMAXMEMORYUSEDTHRESHOLDTYPE ENUM

描述系统支持的所有最大内存阈值类型。

表 7.213. 值概述

名称	概述
absolute_value_in_mb	绝对值阈值类型。
百分比	百分比阈值类型。

7.159.1. absolute_value_in_mb

绝对值阈值类型。

当指定绝对值时，如果 MB 中的可用内存低于 LogMaxMemoryUsedThreshold 中指定的值，则会记录审计日志事件。

7.159.2. 百分比

百分比阈值类型。

当指定百分比时，如果所使用的内存超过 `LogMaxMemoryUsedThreshold` 中指定的值，则会记录审计日志事件。

7.160. LOGSEVERITY ENUM

代表事件的严重性的枚举。

表 7.214. 值概述

名称	概述
<code>alert</code>	警报严重性.
错误	错误严重性.
<code>Normal</code>	普通严重性.
<code>warning</code>	Warning 严重性.

7.160.1. alert

警报严重性.用于指定需要立即关注的条件。

7.160.2. 错误

错误严重性.用于指定是否存在需要检查的错误。

7.160.3. Normal

普通严重性.用于信息事件。

7.160.4. warning

Warning 严重性.用于警告可能出错的内容。

7.161. LOGICALUNIT STRUCT

表 7.215. 属性概述

名称	类型	概述
地址	字符串	
discard_max_size	整数	单个操作中逻辑单元的底层存储可以丢弃的最大字节数。
discard_zeroes_data	布尔值	正确，如果之前在逻辑单元的底层存储中丢弃的块会写回为零。
disk_id	字符串	
id	字符串	
lun_mapping	整数	
password	字符串	
路径	整数	
port	整数	
portal	字符串	
product_id	字符串	
serial	字符串	
Size	整数	
status	LunStatus	
storage_domain_id	字符串	
目标	字符串	
username	字符串	
vendor_id	字符串	
volume_group_id	字符串	

7.161.1. *discard_max_size*

单个操作中逻辑单元的底层存储可以丢弃的最大字节数。0 表示该设备不支持丢弃功能。



注意

这是软件限制，而非硬件限制，如用于 `discard_max_bytes` 的 `queue-sysfs` 的文档中所述。

7.161.2. `discard_zeroes_data`

正确，如果之前在逻辑单元的底层存储中丢弃的块会写回为零。如需更多信息，请参阅用于 `discard_zeroes_data` 的 `queue-sysfs` 文档。



重要

自系统版本 4.2.1 起，已删除对此属性的支持，因为 `sysfs` 文件 `discard_zeroes_data` 已在内核中弃用。它保留下来以便向后兼容，但值始终为 `false`。

7.162. `LUNSTATUS ENUM`

表 7.216. 值概述

名称	概述
Free	
unusable	
Used	

7.163. `MDEVTYPE STRUCT`

调解设备是一种软件设备，允许分割物理设备的资源。

详情请查看 [Libvirt-MDEV](#)。

表 7.217. 属性概述

名称	类型	概述
available_instances	整数	MDev 类型可用的实例数。
description	字符串	MDev 类型描述。
human_readable_name	字符串	MDev 键入人类可读名称。
name	字符串	MDev 类型名称。

7.164. MAC STRUCT

代表虚拟网络接口的 MAC 地址。

表 7.218. 属性概述

名称	类型	概述
地址	字符串	MAC 地址。

7.165. MACPOOL STRUCT

代表 MAC 地址池。

MAC 地址池的 XML 表示示例：

```
<mac_pool href="/ovirt-engine/api/macpools/123" id="123">
  <name>Default</name>
  <description>Default MAC pool</description>
  <allow_duplicates>false</allow_duplicates>
  <default_pool>true</default_pool>
  <ranges>
    <range>
      <from>00:1A:4A:16:01:51</from>
      <to>00:1A:4A:16:01:E6</to>
    </range>
  </ranges>
</mac_pool>
```

表 7.219. 属性概述

名称	类型	概述
allow_duplicates	布尔值	定义是否允许池中重复的 MAC 地址。
comment	字符串	包含有关此对象的注释的免费文本。
default_pool	布尔值	定义这是否为默认池。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
范围	范围[]	定义池的 MAC 地址范围。

7.165.1. allow_duplicates

定义是否允许池中重复的 MAC 地址。如果没有指定，则默认为 *false*。

7.165.2. default_pool

定义这是否为默认池。如果没有指定，则默认为 *false*。

7.165.3. 范围

定义池的 MAC 地址范围。可以定义多个范围。

表 7.220. 链接摘要

名称	类型	概述
权限	permission[]	返回对与 MacPool 相关联的权限的引用。

7.166. MEMORYOVERCOMMIT STRUCT

表 7.221. 属性概述

名称	类型	概述
%	整数	

7.167. MEMORYPOLICY STRUCT

类似虚拟机实体的内存相关属性的逻辑分组。

表 7.222. 属性概述

名称	类型	概述
ballooning	布尔值	
Guaranteed	整数	以字节为单位的内存量，保证不会排空 Balloon 机制。
max	整数	最大虚拟机内存，以字节为单位。
over_commit	MemoryOverCommit	
transparent_huge_pages	TransparentHugePages	

7.167.1. Guaranteed

以字节为单位的内存量，保证不会排空 Balloon 机制。

红帽虚拟化管理器内部将此值向下舍入到整个 MiB ($1\text{MiB} = 2^{20}$ 字节)。



注意

由于 Red Hat Virtualization 4.2 以后，可以更新它，在同一请求中更新提供的内存，并且虚拟机处于 **up** 状态。

7.167.2. Max

最大虚拟机内存，以字节为单位。

用户以字节为单位提供值，Red Hat Virtualization Manager 会将值向下舍入到最接近的 MiB 值。

例如，如果用户输入值 1073741825 (1 GiB + 1 字节)，则红帽虚拟化管理器会将该值截断到最接近的最小 MiB 边界：本例中为 1073741824(1 GiB)。

7.168. MESSAGEBROKERTYPE ENUM

弃用的消息代理类型。

忽略，因为自 Red Hat Virtualization 4.4.0 起，OpenStack Neutron 代理的部署已被丢弃。

表 7.223. 值概述

名称	概述
qpuid	
rabbit_mq	

7.169. METHOD STRUCT

表 7.224. 属性概述

名称	类型	概述
id	SsoMethod	

7.170. MIGRATEONERROR ENUM

表 7.225. 值概述

名称	概述
do_not_migrate	
migrate	
migrate_highly_available	

7.171. MIGRATIONBANDWIDTH STRUCT

定义迁移所用的带宽。

表 7.226. 属性概述

名称	类型	概述
assignment_method	MigrationBandwidthAssignmentMethod	用于分配带宽的方法。
custom_value	整数	以 Mbps 为单位的自定义带宽。

7.171.1. custom_value

以 Mbps 为单位的自定义带宽。只有在 `assignmentMethod` 属性是 `custom` 时才会应用。

7.172. MIGRATIONBANDWIDTHASSIGNMENTMETHOD ENUM

定义如何分配迁移带宽。

表 7.227. 值概述

名称	概述
auto	如果定义了服务质量，则从服务质量获取带宽。
custom	以 Mbit/s 为单位的自定义定义带宽。
hypervisor_default	取系统管理程序上配置的值。

7.172.1. auto

如果定义了服务质量，则从服务质量获取带宽。如果未定义服务质量，则从检测到使用的链接速度获取带宽。如果未检测到任何内容，带宽将退回到 `hypervisor_default` 值。

7.173. MIGRATIONOPTIONS STRUCT

迁移选项的类型。

表 7.228. 属性概述

名称	类型	概述
auto_converge	InheritableBoolean	

名称	类型	概述
bandwidth	MigrationBandwidth	允许被迁移使用的带宽。
compressed	InheritableBoolean	
已加密	InheritableBoolean	指定是否应加密迁移。

表 7.229. 链接摘要

名称	类型	概述
policy	MigrationPolicy	对迁移策略的引用，如使用 engine-config 定义。

7.174. MIGRATIONPOLICY STRUCT

描述如何处理迁移的策略，如聚合或允许并行迁移数量。

表 7.230. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

7.175. NETWORK STRUCT

逻辑网络的类型。

逻辑网络 JSON 表示形式的示例：

```
{
  "network": [{
    "data_center": {
      "href": "/ovirt-engine/api/datacenters/123",
      "id": "123"
    }
  }]
}
```

```

    },
    "stp" : "false",
    "mtu" : "0",
    "usages" : {
      "usage" : [ "vm" ]
    },
  },
  "name" : "ovirtmgmt",
  "description" : "Management Network",
  "href" : "/ovirt-engine/api/networks/456",
  "id" : "456",
  "link" : [ {
    "href" : "/ovirt-engine/api/networks/456/permissions",
    "rel" : "permissions"
  }, {
    "href" : "/ovirt-engine/api/networks/456/vnicprofiles",
    "rel" : "vnicprofiles"
  }, {
    "href" : "/ovirt-engine/api/networks/456/labels",
    "rel" : "labels"
  } ]
}

```

同一逻辑网络 XML 表示的示例：

```

<network href="/ovirt-engine/api/networks/456" id="456">
  <name>ovirtmgmt</name>
  <description>Management Network</description>
  <link href="/ovirt-engine/api/networks/456/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/networks/456/vnicprofiles" rel="vnicprofiles"/>
  <link href="/ovirt-engine/api/networks/456/labels" rel="labels"/>
  <data_center href="/ovirt-engine/api/datacenters/123" id="123"/>
  <stp>false</stp>
  <mtu>0</mtu>
  <usages>
    <usage>vm</usage>
  </usages>
</network>

```

表 7.231. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
display	布尔值	弃用的"usages"应该用于将网络定义为显示网络。
dns_resolver_configuration	DnsResolverConfiguration	使用 GET 检索网络时，将报告 DNS 解析器配置。

名称	类型	概述
id	字符串	唯一标识符。
ip	ip	已弃用，不在使用中。
mtu	整数	指定网络的最大传输单元。
name	字符串	纯文本中的人类可读名称。
port_isolation	布尔值	定义是否在此网络上阻止了同一主机上运行的虚拟机之间的通信。
profile_required	布尔值	指定在创建网络时是否应自动创建虚拟网络接口配置文件。
必需	布尔值	定义是否对群集中的所有主机强制使用网络。
status	NetworkStatus	网络的状态。
STP	布尔值	指定网络是否启用了范围树协议。
使用方法	NetworkUsage[]	定义网络的一组使用元素。
vdsname	字符串	主机上使用的网络的名称。
vlan	VLAN	VLAN 标签。

7.175.1. dns_resolver_configuration

使用 GET 检索网络时，将报告 DNS 解析器配置。在创建新网络或更新现有网络时，这是可选的。

7.175.2. port_isolation

定义是否在此网络上阻止了同一主机上运行的虚拟机之间的通信。仅适用于虚拟机网络。网络管理员负责确保多个主机之间的通信已被阻止。此属性只能在创建网络时设置，且无法编辑。如果未设置值，则允许同一主机上运行的虚拟机之间的通信。

7.175.3. 必需

定义是否对群集中的所有主机强制使用网络。如果主机中省略了"必需"操作网络，则该主机将被标记为 non_operational。

7.175.4. status

网络的状态。 `non_operational` (如果定义为"必需"的网络并且从任何活跃的集群主机中省略了网络)。运行性, 否则 操作。

7.175.5. 使用方法

定义网络的一组使用元素。

例如, 用户可以指定网络将用于虚拟机流量, 也可使用 `vm` 和 `display` 值来显示 流量。

7.175.6. vdsname

主机上使用的网络的名称。当网络名称不适合充当主机上的网桥名称时, VDSM 会自动生成这个替代名称。不合适的名称包含空格、特殊字符或超过 15 个字符, 并替换为主机上的 UUID。此参数为只读。设置该设置无效。

表 7.232. 链接摘要

名称	类型	概述
<code>cluster</code>	Cluster	附加对这个网络的集群的引用。
<code>data_center</code>	DataCenter	网络所属的数据中心的引用。
<code>external_provider</code>	OpenStackNetworkProvider	可选引用创建该网络的 OpenStack 网络提供程序。
<code>external_provider_physical_network</code>	Network	对网络的可选引用, 应用于物理网络访问。
<code>network_labels</code>	NetworkLabel[]	对分配给网络的标签的引用。
权限	permission[]	对网络权限的引用。
<code>qos</code>	QoS	提及服务质量。
<code>vnic_profiles</code>	VnicProfile[]	网络配置文件的引用。

7.175.7. cluster

附加对这个网络的集群的引用。只有从群集级别访问网络时，才会填充。

7.175.8. external_provider

可选引用创建该网络的 OpenStack 网络提供程序。

如果在创建网络时指定，也会创建匹配的 OpenStack 网络。

7.175.9. external_provider_physical_network

对网络的可选引用，应用于物理网络访问。仅在指定了 external_provider 时有效。

7.176. NETWORKATTACHMENT STRUCT

描述主机如何连接到网络。

主机上网络附加的 XML 表示：

```
<network_attachment href="/ovirt-engine/api/hosts/123/nics/456/networkattachments/789" id="789">
  <network href="/ovirt-engine/api/networks/234" id="234"/>
  <host_nic href="/ovirt-engine/api/hosts/123/nics/123" id="123"/>
  <in_sync>true</in_sync>
  <ip_address_assignments>
    <ip_address_assignment>
      <assignment_method>static</assignment_method>
      <ip>
        <address>192.168.122.39</address>
        <gateway>192.168.122.1</gateway>
        <netmask>255.255.255.0</netmask>
        <version>v4</version>
      </ip>
    </ip_address_assignment>
  </ip_address_assignments>
  <reported_configurations>
    <reported_configuration>
      <name>mtu</name>
      <expected_value>1500</expected_value>
      <actual_value>1500</actual_value>
      <in_sync>true</in_sync>
    </reported_configuration>
    <reported_configuration>
      <name>bridged</name>
      <expected_value>true</expected_value>
```

```

    <actual_value>true</actual_value>
    <in_sync>true</in_sync>
  </reported_configuration>
  ...
</reported_configurations>
</network_attachment>

```

需要具有名称或 id 的网络元素，才能将网络附加到网络接口卡(NIC)。

例如，要将网络附加到主机网络接口卡，请发送类似如下的请求：

```
POST /ovirt-engine/api/hosts/123/nics/456/networkattachments
```

这样的请求正文：

```

<networkattachment>
  <network id="234"/>
</networkattachment>

```

要将网络附加到主机，请发送类似如下的请求：

```
POST /ovirt-engine/api/hosts/123/networkattachments
```

这样的请求正文：

```

<network_attachment>
  <network id="234"/>
  <host_nic id="456"/>
</network_attachment>

```

`ip_address_assignments` 和 `properties` 元素是 `updatable` 后创建。

例如，要更新网络附加，请发送如下请求：

```
PUT /ovirt-engine/api/hosts/123/nics/456/networkattachments/789
```

这样的请求正文：

```

<network_attachment>
  <ip_address_assignments>
    <ip_address_assignment>
      <assignment_method>static</assignment_method>
      <ip>
        <address>7.1.1.1</address>
        <gateway>7.1.1.2</gateway>
        <netmask>255.255.255.0</netmask>
        <version>v4</version>
      </ip>
    </ip_address_assignment>
  </ip_address_assignments>
</network_attachment>

```

要从网络接口卡分离网络，发送类似如下的请求：

```
DELETE /ovirt-engine/api/hosts/123/nics/456/networkattachments/789
```



重要

必须明确提交对网络附加配置的更改。

网络附加的属性子集合的 XML 表示：

```

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

```

表 7.233. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
dns_resolver_configuration	DnsResolverConfiguration	使用 GET 检索网络附加时，将报告 DNS 解析器配置。

名称	类型	概述
id	字符串	唯一标识符。
in_sync	布尔值	
ip_address_assignments	IpAddressAssignment[]	网络的 IP 配置。
name	字符串	纯文本中的人类可读名称。
属性	attribute[]	定义网络配置的自定义属性。
reported_configurations	ReportedConfiguration[]	配置属性的只读列表。

7.176.1. dns_resolver_configuration

使用 **GET** 检索网络附加时，将报告 **DNS 解析器配置**。它是在创建新网络附加或更新现有网络时可选的。

7.176.2. 属性

定义网络配置的自定义属性。

网桥选项的集合名称为 **bridge_opts**。使用空白字符分隔多个条目。以下密钥对 **bridge_opts** 有效：

名称	默认值
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

名称	默认值
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

表 7.234. 链接摘要

名称	类型	概述
host	主机	
host_nic	HostNic	对主机网络接口的引用。
network	Network	对接口所附加的网络的引用。
qos	QoS	

7.177. NETWORKCONFIGURATION STRUCT

表 7.235. 属性概述

名称	类型	概述
dns	DNS	
nics	Nic[]	

7.178. NETWORKFILTER STRUCT

网络过滤器根据定义的规则过滤发送到虚拟机的 NIC 的数据包。

基于 libvirt 支持多种类型的网络过滤器。有关不同网络过滤器的更多详细信息，请参阅[此处](#)。

默认网络过滤器基于网络类型和配置。如果 `EnableMACAntiSpoofingFilterRules` 为 `True`，则 VM 网络的默认过滤器为 `vdsm-no-mac-spoof`，否则不会为未配置过滤器的 OVN 网络配置过滤器。

除了 libvirt 的网络过滤器外，还有两个额外的网络过滤器：第一个过滤器名为 `vdsm-no-mac-spoofing`，它由 `no-mac-spoofing` 和 `no-arp-mac-spoofing` 组成。第二个称为 `ovirt-no-filter`，在为虚拟机的 NIC 定义网络过滤器时使用。`ovirt-no-filter` 网络过滤器仅用于内部实施，且 NIC 中不存在。

这是 XML 表述的示例：

```
<network_filter id="00000019-0019-0019-0019-00000000026c">
  <name>example-filter</name>
  <version>
    <major>4</major>
    <minor>0</minor>
    <build>-1</build>
    <revision>-1</revision>
  </version>
</network_filter>
```

如果任何部分版本不存在，它将由 `-1` 表示。

表 7.236. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。

名称	类型	概述
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
version	版本	特定 NetworkFilter 的最低支持版本。

7.178.1. version

特定 NetworkFilter 的最低支持版本。这是 NetworkFilter 首次引入的版本。

7.179. NETWORKFILTERPARAMETER STRUCT

网络过滤器的参数。

详情请查看 [Libvirt-Filters](#)。这是 XML 表述的示例：

```
<network_filter_parameter id="123">
  <name>IP</name>
  <value>10.0.1.2</value>
</network_filter_parameter>
```

表 7.237. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
value	字符串	表示 参数的值。

表 7.238. 链接摘要

名称	类型	概述
nic	NIC	参与虚拟机 NIC 关联至。

7.180. NETWORKLABELSTRUCT

表示可添加到主机网络接口和网络的标签。标签通过标签 `id` 将网络绑定到主机网络接口。

表 7.239. 属性概述

名称	类型	概述
<code>comment</code>	字符串	包含有关此对象的注释的免费文本。
<code>description</code>	字符串	以纯文本格式人类可读的描述。
<code>id</code>	字符串	唯一标识符。
<code>name</code>	字符串	纯文本中的人类可读名称。

表 7.240. 链接摘要

名称	类型	概述
<code>host_nic</code>	HostNic	对包含此标签的主机网络接口的引用。
<code>network</code>	Network	对包含此标签的网络的引用。

7.181. NETWORKPLUGINTYPE ENUM

网络插件类型。

指定主机上的供应商驱动程序实现。

自 Red Hat Virtualization Manager 版本 4.2 起，此类型已弃用，以代表 `OpenStackNetworkProvider` 类型的 `external_plugin_type` 属性。

表 7.241. 值概述

名称	概述
<code>open_vswitch</code>	Open vSwitch.

7.181.1. open_vswitch

Open vSwitch.

指定应该将基于 **Open vSwitch** 的驱动程序实施用于此提供程序。

自 Red Hat Virtualization Manager 版本 4.2 起, 这个值已被弃用。改为使用 `OpenStackNetworkProvider.external_plugin_type` 属性中的字符串 `open_vswitch`。

7.182. NETWORKSTATUS ENUM

表 7.242. 值概述

名称	概述
<code>non_operational</code>	
操作	

7.183. NETWORKUSAGE 枚举

此类型表示在集群中使用网络的用途。

表 7.243. 值概述

名称	概述
<code>default_route</code>	主机的默认网关和 DNS 解析器配置将从此网络获取。
<code>display</code>	网络将用于 SPICE 和 VNC 通信。
<code>Gluster</code>	网络将用于 Gluster(brick)数据流量。
管理	该网络将用于红帽虚拟化管理器 and 节点之间的通信。
迁移	网络将用于虚拟机迁移。
<code>vm</code>	

7.183.1. default_route

主机的默认网关和 DNS 解析器配置将从此网络获取。

如果此网络附加到主机，则将从网络附加的 `dns_resolver_configuration` 属性获取 DNS 解析器配置。如果网络附加中没有 `dns_resolver_configuration` 属性，则会从网络本身的 `dns_resolver_configuration` 中获取它们。如果 `dns_resolver_configuration` 属性也没有，则不会设置 DNS 解析器配置。

如果在网络中设置此标记，则主机的默认网关将从网络附加的 `ip_address_assignment` 的 `gateway` 属性获取。

7.183.2. 管理

该网络将用于红帽虚拟化管理器和节点之间的通信。这是创建 `ovirtmgmt` 网桥的网络。

7.184. NFSPROFILEDETAIL STRUCT

表 7.244. 属性概述

名称	类型	概述
<code>nfs_server_ip</code>	字符串	
<code>profile_details</code>	<code>ProfileDetail[]</code>	

7.185. NFSVERSION ENUM

表 7.245. 值概述

名称	概述
<code>auto</code>	
<code>v3</code>	
<code>v4</code>	
<code>v4_0</code>	NFS 4.
<code>v4_1</code>	
<code>v4_2</code>	NFS 4.

7.185.1. v4_0

NFS 4.0.**7.185.2. v4_2****NFS 4.2.****7.186. NIC STRUCT**

表示虚拟机 NIC.

例如, NIC 的 XML 表示方式如下:

```
<nic href="/ovirt-engine/api/vms/123/nics/456" id="456">
  <name>nic1</name>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
  <interface>virtio</interface>
  <linked>true</linked>
  <mac>
    <address>02:00:00:00:00:00</address>
  </mac>
  <plugged>true</plugged>
  <vnic_profile href="/ovirt-engine/api/vnicprofiles/789" id="789"/>
</nic>
```

表 7.246. 属性概述

名称	类型	概述
boot_protocol	BootProtocol	定义如何为 NIC 分配 IP 地址。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
Interface	NicInterface	用于 NIC 的驱动程序类型。
已链接	布尔值	定义 NIC 是否链接到虚拟机。
MAC	MAC	接口的 MAC 地址。
name	字符串	纯文本中的人类可读名称。

名称	类型	概述
on_boot	布尔值	定义是否应在操作系统启动时激活网络接口。
inserted	布尔值	定义是否将 NIC 插入到虚拟机。
synced	布尔值	定义虚拟机的 NIC 配置是否与引擎表示的配置同步。

表 7.247. 链接摘要

名称	类型	概述
instance_type	InstanceType	(可选) 引用设备使用的实例类型。
network	Network	对接口应连接到的网络的引用。
network_attachments	NetworkAttachment[]	与主机 NIC 关联的网络附加的集合链接。
network_filter_parameters	NetworkFilterParameter[]	到网络过滤器参数的链接。
network_labels	NetworkLabel[]	与主机 NIC 关联的网络标签集合链接。
reported_devices	ReportedDevice[]	与虚拟网络接口关联的已报告设备的集合链接。
statistics	Statistic[]	NIC 的统计信息链接。
模板	Template	(可选) 引用设备使用的模板。
virtual_function_allowed_labels	NetworkLabel[]	到允许附加到 SR-IOV NIC 的虚拟功能的网络标签集合的链接。
virtual_function_allowed_networks	network[]	到允许附加到 SR-IOV NIC 的虚拟功能的网络集合的链接。
vm	VM	不要使用此元素，而是使用 vms 。
vms	Vm[]	引用使用此设备的虚拟机。
vnic_profile	VnicProfile	到关联的虚拟网络接口配置文件的链接。

7.186.1. network

对接口应连接到的网络的引用。允许空白网络 ID。

此元素用于创建或更新 NIC 已弃用；改为使用 `vnic_profile`。它会被保留，因为它仍然被初始化元素使用，作为 IP 地址和其他网络详细信息的拥有者。

7.186.2. vms

引用使用此设备的虚拟机。一个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.187. NICCONFIGURATION STRUCT

`type` 描述了虚拟网络接口的配置。

表 7.248. 属性概述

名称	类型	概述
<code>boot_protocol</code>	<code>BootProtocol</code>	IPv4 引导协议。
<code>ip</code>	<code>ip</code>	IPv4 地址详细信息。
<code>ipv6</code>	<code>ip</code>	IPv6 地址详细信息。
<code>ipv6_boot_protocol</code>	<code>BootProtocol</code>	IPv6 引导协议。
<code>name</code>	字符串	网络接口名称。
<code>on_boot</code>	布尔值	指定是否应在虚拟机客户机操作系统启动时激活网络接口。

7.188. NICINTERFACE ENUM

定义模拟虚拟网络接口设备模型的选项。

表 7.249. 值概述

名称	概述
<code>e1000</code>	<code>e1000</code> 。

名称	概述
e1000e	e1000e.
pci_passthrough	PCI 直通.
rtl8139	rtl8139.
rtl8139_virtio	双模式 rtl8139, VirtIO.
spapr_vlan	sPAPR VLAN.
VirtIO	VirtIO.

7.189. NICSTATUS ENUM

网络接口卡状态.

表 7.250. 值概述

名称	概述
down	NIC 已停机且不可访问。
up	NIC 已启动，可访问。

7.190. NOTABLEEVENT ENUM

键入代表 Red Hat Virtualization 服务器中的一个事件子集：用户可能订阅的事件，以接收有关。

表 7.251. 值概述

名称	概述
cluster_alert_ha_reservation	HA 保留检查失败
cluster_alert_ha_reservation_down	HA 保留检查已过
dwh_error	ETL Service 错误

名称	概述
dwh_stopped	ETL Service Stopped
engine_backup_completed	引擎备份成功完成
engine_backup_failed	引擎备份失败
engine_backup_started	引擎备份已启动
engine_ca_certification_has_expired	引擎 CA 的认证已过期
engine_ca_certification_is_about_to_expire	引擎 CA 的认证即将过期
engine_certification_has_expired	引擎的认证已过期
engine_certification_is_about_to_expire	引擎的认证即将过期
engine_stop	引擎已停止
faulty_multipaths_on_host	主机上出现故障的多路径路径
gluster_brick_status_changed	检测到 brick 状态的变化
gluster_hook_added_failed	在冲突服务器上添加 Gluster Hook 失败
gluster_hook_added	添加了 Gluster Hook
gluster_hook_conflict_detected	在 Gluster Hook 中检测到冲突
gluster_hook_detected_delete	检测到删除 Gluster Hook

名称	概述
gluster_hook_detected_new	检测到新的 Gluster Hook
gluster_hook_disabled	Gluster Hook Disabled
gluster_hook_disabled_failed	禁用 Gluster Hook 失败
gluster_hook_enabled	启用 Gluster Hook
gluster_hook_enabled_failed	启用 Gluster Hook 失败
gluster_hook_remove_failed	从集群中删除 Gluster Hook 失败
gluster_hook_removed	删除了 Gluster Hook
gluster_server_add_failed	添加 Gluster 服务器失败
gluster_server_remove	Gluster 服务器已删除
gluster_server_remove_failed	删除 Gluster 服务器失败
gluster_service_restart_failed	重新启动 Gluster 服务失败
gluster_service_restarted	重新启动了 Gluster 服务
gluster_service_start_failed	启动 Gluster 服务失败
gluster_service_started	Gluster 服务已启动
gluster_service_stop_failed	停止 Gluster 服务失败

名称	概述
gluster_service_stopped	Gluster 服务已停止
gluster_volume_add_brick	添加了 Gluster 卷 brick
gluster_volume_add_brick_failed	在 Gluster 卷中添加 brick 失败
gluster_volume_all_snapshots_delete_failed	删除卷快照失败
gluster_volume_all_snapshots_deleted	卷上删除的所有快照
gluster_volume_brick_replaced	替换了 Gluster 卷 Brick
gluster_volume_confirmed_space_low	确认卷的空间较低
gluster_volume_create	已创建 Gluster 卷
gluster_volume_create_failed	无法创建 Gluster 卷
gluster_volume_delete	Gluster 卷已删除
gluster_volume_delete_failed	无法删除 Gluster 卷
gluster_volume_migrate_brick_data_finished	用于删除 brick 的 Gluster 卷迁移
gluster_volume_option_added	添加了 Gluster 卷选项
gluster_volume_option_modified	已修改 Gluster 卷选项

名称	概述
gluster_volume_option_set_failed	无法设置 Gluster 卷选项
gluster_volume_options_reset	Gluster 卷选项重置
gluster_volume_options_reset_all	重置所有 Gluster 卷选项
gluster_volume_options_reset_failed	无法重置 Gluster 卷选项
gluster_volume_profile_start	Gluster 卷配置文件已启动
gluster_volume_profile_start_failed	启动 Gluster 卷配置文件失败
gluster_volume_profile_stop	Gluster 卷配置文件已停止
gluster_volume_profile_stop_failed	停止 Gluster 卷配置文件失败
gluster_volume_rebalance_finished	Gluster 卷重新平衡完成
gluster_volume_rebalance_not_found_from_cli	无法从 CLI 查找重新平衡卷的信息。
gluster_volume_rebalance_start	已启动 Gluster 卷重新平衡
gluster_volume_rebalance_start_detected_from_cli	从 CLI 开始重新平衡 gluster 卷

名称	概述
gluster_volume_rebalance_start_failed	无法启动 Gluster 卷重新平衡
gluster_volume_rebalance_stop	Gluster 卷重新平衡已停止
gluster_volume_rebalance_stop_failed	无法停止 Gluster 卷重新平衡
gluster_volume_remove_bricks	Gluster 卷 Bricks Removed
gluster_volume_remove_bricks_failed	无法删除 Gluster 卷 Bricks
gluster_volume_remove_bricks_stop	停止从 Gluster 卷移除 brick
gluster_volume_remove_bricks_stop_failed	从 Gluster 卷停止删除 brick 失败
gluster_volume_replace_brick_failed	Gluster 卷替换 Brick Failed
gluster_volume_replace_brick_start	Gluster 卷替换 Brick Started
gluster_volume_replace_brick_start_failed	无法启动 Gluster 卷替换 Brick
gluster_volume_snapshot_activate_failed	在卷中激活快照失败
gluster_volume_snapshot_activated	卷中激活的快照

名称	概述
gluster_volume_snapshot_create_failed	无法为集群 <code>{clusterName}</code> 上的卷 <code>{glusterVolumeName}</code> 创建快照。
gluster_volume_snapshot_created	为集群 <code>{clusterName}</code> 上的卷 <code>{glusterVolumeName}</code> 创建的快照 <code>{snapname}</code> 。
gluster_volume_snapshot_deactivate_failed	在卷中取消激活快照失败
gluster_volume_snapshot_deactivated	卷中取消激活的快照
gluster_volume_snapshot_delete_failed	删除卷快照失败
gluster_volume_snapshot_deleted	卷上删除的快照
gluster_volume_snapshot_restore_failed	在卷中恢复快照失败
gluster_volume_snapshot_restored	卷中恢复快照
gluster_volume_start	Gluster 卷已启动
gluster_volume_start_failed	无法启动 Gluster 卷
gluster_volume_stop	Gluster 卷已停止
gluster_volume_stop_failed	无法停止 Gluster 卷
ha_vm_failed	高可用 VM 失败

名称	概述
ha_vm_restart_failed	高可用虚拟机重启失败
host_activate_failed	激活主机失败
host_activate_manual_ha	主机已激活，但托管引擎 HA 服务可能仍然处于维护模式
host_approve_failed	批准主机失败
host_bond_slave_state_down	主机的绑定从状态更改为 down
host_certificate_has_invalid_san	主机的证书包含无效的主题备用名称(SAN)
host_certificate_has_expired	主机的证书已过期
host_certificate_is_about_to_expire	主机的认证即将过期
host_failure	主机不响应
host_high_cpu_use	主机 CPU 用量超过定义阈值
host_high_memory_use	主机内存用量超过定义阈值
host_high_swap_use	主机交换内存用量超过定义阈值
host_initiated_run_vm_failed	在不同主机上重启虚拟机失败
host_install_failed	主机安装失败
host_interface_high_network_use	主机网络接口使用量超过定义阈值

名称	概述
host_interface_state_down	主机的接口将状态更改为 down
host_low_mem	主机可用内存低于定义阈值
host_low_swap	主机可用交换内存低于定义阈值
host_recover_failed	主机无法恢复
host_set_nonoperational	主机状态设置为不可运行
host_set_nonoperational_domain	由于无法访问的存储域，主机状态设置为不可运行
host_set_nonoperational_iface_down	因为缺少接口，主机状态被设置为不可运行
host_slow_storage_response_time	存储响应时间较慢
host_time_drift_alert	主机具有时间drift
host_untrusted	主机状态设置为不可运行。
host_updates_are_available	主机有可用的更新
host_updates_are_available_with_packages	主机具有可更新的可用软件包
importexport_import_template_from_trusted_to_untrusted	从可信集群导入到非可信集群的模板
importexport_import_template_from_untrusted_to_trusted	从非可信集群导入到可信集群的模板

名称	概述
<code>importexport_import_vm_from_trusted_to_untrusted</code>	将 VM 从可信集群导入到非可信集群
<code>importexport_import_vm_from_untrusted_to_trusted</code>	将虚拟机从不受信任的集群导入到可信集群
<code>irs_confirmed_disk_space_low</code>	确认低磁盘空间
<code>irs_disk_space_low</code>	低磁盘空间
<code>irs_disk_space_low_error</code>	非常低的磁盘空间
<code>irs_failure</code>	访问存储失败
<code>mac_address_is_external</code>	具有外部 MAC 地址的虚拟机
<code>multipath_devices_without_valid_paths_on_host</code>	主机上没有有效路径的多路径设备
<code>network_update_display_for_cluster_with_active_vm</code>	使用活跃虚拟机在集群中更新了显示网络
<code>network_update_display_for_host_with_active_vm</code>	在带有活跃虚拟机的主机上更新了显示网络
<code>no_faulty_multipaths_on_host</code>	主机上没有出现故障的多路径路径
<code>number_of_lvs_on_storage_domain_exceeded_threshold</code>	存储域的 LV 数量超过阈值

名称	概述
remove_gluster_volume_bricks_not_found_from_cli	无法从 CLI 找到用于删除卷上 brick 的信息。
start_removing_gluster_volume_bricks	开始从卷中删除 brick
start_removing_gluster_volume_bricks_detected_from_cli	从 CLI 中为卷上的 brick 移除检测到开始
start_removing_gluster_volume_bricks_failed	无法删除卷 brick
system_change_storage_pool_status_no_host_for_spm	为 Data-Center 选择 SPM 失败
system_deactivated_storage_domain	Storage Domain state 设置为 inactive
user_add_vm_from_trusted_to_untrusted	非可信虚拟机是从可信模板创建的
user_add_vm_from_untrusted_to_trusted	从不受信任的模板中创建可信虚拟机
user_add_vm_template_from_trusted_to_untrusted	从可信虚拟机创建不可信模板
user_add_vm_template_from_untrusted_to_trusted	从不受信任的虚拟机中创建可信模板
user_host_maintenance	主机已切换到维护模式

名称	概述
user_host_maintenance_manual_ha	主机切换到维护模式，但无法启用托管引擎 HA 维护模式
user_host_maintenance_migration_failed	将 Host 切换到维护模式失败
user_update_vm_from_trusted_to_untrusted	虚拟机从可信集群移到非可信集群
user_update_vm_from_untrusted_to_trusted	虚拟机从不受信任的集群移到可信集群
user_update_vm_template_from_trusted_to_untrusted	模板从可信集群移到非可信集群
user_update_vm_template_from_untrusted_to_trusted	模板从不受信任的集群移到可信集群
vm_console_connected	连接的虚拟机控制台
vm_console_disconnected	虚拟机控制台断开连接
vm_down_error	VM 停机并显示错误
vm_failure	虚拟机无法在主机上找到
vm_migration_failed	迁移失败
vm_migration_start	开始迁移虚拟机
vm_migration_to_server_failed	虚拟机迁移到目标主机失败
vm_not_responding	VM 没有响应

名称	概述
vm_paused	VM 已暂停
vm_paused_eio	虚拟机因为存储 I/O 错误而暂停
vm_paused_enospc	虚拟机因为缺少存储空间而暂停
vm_paused_eperm	由于存储读/写权限问题，虚拟机已被暂停
vm_paused_error	虚拟机已暂停，因为未知存储错误
vm_recovered_from_pause_error	虚拟机已经恢复为已暂停备份至运行
vm_set_ticket	虚拟机控制台会话已启动
vm_status_restored	VM 状态恢复

7.190.1. *gluster_volume_rebalance_not_found_from_cli*

无法从 CLI 查找重新平衡卷的信息。将它标记为 *unknown*。

7.190.2. *host_untrusted*

主机状态设置为不可运行。主机不受 *attestation* 服务信任

7.190.3. *remove_gluster_volume_bricks_not_found_from_cli*

无法从 CLI 找到用于删除卷上 *brick* 的信息。将它标记为 *unknown*。

7.191. NOTIFICATIONMETHOD ENUM

键入代表事件订阅的通知方法。目前，API 只支持 *SMTP*，未来可能会添加对 *SNMP* 通知的支持。

表 7.252. 值概述

名称	概述
smtp	通过电子邮件发送通知.
snmp	SNMP 通知.

7.191.1. smtp

通过电子邮件发送通知.

采用 SMTP 通知方法的 `event-subscriptions` 将在 `address` 字段中包含一个电子邮件地址。

7.191.2. SNMP

SNMP 通知.

使用 SNMP 通知方法的事件订阅将在地址字段中包含 SNMP 地址。

7.192. NUMANO DESTRUCT

代表物理 NUMA 节点。

XML 表述示例：

```
<host_numa_node href="/ovirt-engine/api/hosts/0923f1ea/numanodes/007cf1ab" id="007cf1ab">
  <cpu>
    <cores>
      <core>
        <index>0</index>
      </core>
    </cores>
  </cpu>
  <index>0</index>
  <memory>65536</memory>
  <node_distance>40 20 40 10</node_distance>
  <host href="/ovirt-engine/api/hosts/0923f1ea" id="0923f1ea"/>
</host_numa_node>
```

表 7.253. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
cpu	cpu	
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
index	整数	
内存	整数	NUMA 节点的内存（以 MB 为单位）。
name	字符串	纯文本中的人类可读名称。
node_distance	字符串	

表 7.254. 链接摘要

名称	类型	概述
host	主机	
statistics	Statistic[]	每一主机 NUMA 节点资源公开一个统计子集合，用于主机 NUMA 节点特定统计信息。

7.192.1. statistics

每一主机 NUMA 节点资源公开一个统计子集合，用于主机 NUMA 节点特定统计信息。

XML 表述示例：

```
<statistics>
  <statistic href="/ovirt-engine/api/hosts/123/numanodes/456/statistics/789" id="789">
    <name>memory.total</name>
    <description>Total memory</description>
    <kind>gauge</kind>
    <type>integer</type>
    <unit>bytes</unit>
    <values>
      <value>
        <datum>25165824000</datum>
      </value>
    </values>
  </statistic>
</statistics>
```

```

<host_numa_node href="/ovirt-engine/api/hosts/123/numanodes/456" id="456" />
</statistic>
...
</statistics>

```



注意

这个统计子集合是只读的。

下表显示了主机 NUMA 节点的统计类型：

名称	描述
memory.total	NUMA 节点上的内存总量，以字节为单位。
memory.used	NUMA 节点上使用的内存（以字节为单位）。
memory.free	NUMA 节点上空闲的内存（以字节为单位）。
cpu.current.user	用户片段的 CPU 使用量百分比。
cpu.current.system	系统的 CPU 使用量百分比。
cpu.current.idle	空闲 CPU 使用率的百分比。

7.193. NUMANODEPIN STRUCT

表示虚拟 NUMA 节点固定到物理 NUMA 节点。

表 7.255. 属性概述

名称	类型	概述
host_numa_node	NumaNode	弃用。
index	整数	虚拟 NUMA 节点固定到的物理 NUMA 节点的索引。
pinned	布尔值	弃用。

7.193.1. host_numa_node

弃用.没有功能。

7.193.2. pinned

弃用.应该始终为 `true`。

7.194. NUMATUNEMODE ENUM

表 7.256. 值概述

名称	概述
interleave	
preferred	
strict	

7.195. OPENSTACKIMAGE STRUCT

表 7.257. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.258. 链接摘要

名称	类型	概述
openstack_image_provider	OpenStackImageProvider	

7.196. OPENSTACKIMAGEPROVIDER STRUCT

表 7.259. 属性概述

名称	类型	概述
authentication_url	字符串	定义外部提供程序身份验证 URL 地址。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
password	字符串	定义用户在身份验证过程中的密码。
属性	attribute[]	提供程序名称/值属性数组。
required_authentication	布尔值	定义是否需要供应商身份验证。
tenant_name	字符串	定义 OpenStack Identity API v2 的租户名称。
url	字符串	定义外部提供程序的 URL 地址。
username	字符串	定义要在身份验证过程中使用的用户名。

7.196.1. required_authentication

定义是否需要供应商身份验证。

如果需要身份验证，则身份验证期间将使用 **用户名和密码** 属性。

7.196.2. tenant_name

定义 **OpenStack Identity API v2.0** 的租户名称。

表 7.260. 链接摘要

名称	类型	概述
证书	certificate[]	

名称	类型	概述
images	OpenStackImage[]	

7.197. OPENSTACKNETWORK STRUCT

表 7.261. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.262. 链接摘要

名称	类型	概述
openstack_network_provider	OpenStackNetworkProvider	

7.198. OPENSTACKNETWORKPROVIDER STRUCT

表 7.263. 属性概述

名称	类型	概述
agent_configuration	AgentConfiguration	弃用的代理配置设置。
authentication_url	字符串	定义外部提供程序身份验证 URL 地址。
auto_sync	布尔值	指明此提供程序的网络是否自动同步。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
external_plugin_type	字符串	网络插件类型。

名称	类型	概述
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
password	字符串	定义用户在身份验证过程中的密码。
plugin_type	NetworkPluginType	网络插件类型。
project_domain_name	字符串	定义 OpenStack Identity API v3 的项目域名。
project_name	字符串	定义 OpenStack Identity API v3 的项目名称。
属性	attribute[]	提供程序名称/值属性数组。
read_only	布尔值	指示提供程序是否只读。
required_authentication	布尔值	定义是否需要供应商身份验证。
tenant_name	字符串	定义 OpenStack Identity API v2 的租户名称。
type	OpenStackNetworkProviderType	提供程序的类型。
非受管	布尔值	指明该提供程序是否由红帽虚拟化管理。
url	字符串	定义外部提供程序的 URL 地址。
user_domain_name	字符串	在 <code>ExternalProvider</code> 中为 OpenStack Identity API v3 定义用户名的域名。
username	字符串	定义要在身份验证过程中使用的用户名。

7.198.1. agent_configuration

弃用的代理配置设置。

忽略，因为自 Red Hat Virtualization 4.4.0 起，OpenStack Neutron 代理的部署已被丢弃。

7.198.2. auto_sync

指明此提供程序的网络是否自动同步。

如果为 `true`，则此供应商的网络将自动同步到 Red Hat Virtualization。这意味着此供应商的所有新网络都会被导入，所有丢弃的网络都将从所有将此外部供应商作为默认供应商的集群中删除。如果供应商上更改了网络名称，则会将更改同步到 Red Hat Virtualization 中的网络实体。另外，如果添加了一个具有提供程序作为默认供应商的新集群，则已导入的网络将在同步期间附加到此新集群。

自动启动的导入会触发以下步骤：

- 外部供应商的网络将导入到集群数据中心的拥有该外部供应商作为默认供应商的每个数据中心。
- 将为每个涉及的数据中心和网络创建一个 `vNIC 配置集`。
- 网络将分配给拥有该外部供应商作为默认供应商的每个集群。

所有用户都可以使用新的 `vNIC 配置文件`。

默认值为 `false`，用于向后兼容。

7.198.3. `external_plugin_type`

网络插件类型。

通过此属性，您可以在主机上添加或修改外部 NIC 时选择正确的提供程序驱动程序。如果支持自动安装驱动程序（仅适用于某些预定义实施，如 `ovirt-provider-ovn`），此属性还允许系统决定在新添加的主机上安装哪些驱动程序实施。

7.198.4. `plugin_type`

网络插件类型。

自 Red Hat Virtualization Manager 版本 4.2 起，此属性在支持 `external_plugin_type` 时已弃用。此

属性仅对类型为 `open_vswitch` 的提供商有效，且仅在 `external_plugin_type` 属性值等于 `open_vswitch` 的值时返回。

如果在更新过程中指定了 `plugin_type` 和 `external_plugin_type`，则 `plugin_type` 的值将被忽略。

对于外部供应商，这个值不会显示，并在更新请求期间被忽略。

7.198.5. `read_only`

指示提供程序是否只读。

只读供应商不允许添加、修改或删除网络或子网。允许与端口相关的操作，因为调配虚拟 NIC 需要它们。

7.198.6. `required_authentication`

定义是否需要供应商身份验证。

如果需要身份验证，则身份验证期间将使用 `用户名和密码` 属性。

7.198.7. `tenant_name`

定义 `OpenStack Identity API v2.0` 的租户名称。

7.198.8. 非受管

指明该提供程序是否由红帽虚拟化管理。

如果为 `true`，身份验证和子网控制完全由外部供应商控制，且由 `Red Hat Virtualization` 管理。

默认值为 `false`，用于向后兼容。

表 7.264. 链接摘要

名称	类型	概述
证书	certificate[]	引用证书列表。
网络	OpenStackNetwork[]	参考 OpenStack 网络列表。
子网	OpenStackSubnet[]	引用 OpenStack 网络子网列表。

7.199. OPENSTACKNETWORKPROVIDERTYPE ENUM

OpenStack 网络提供程序可以由 OpenStack Neutron 实施（在这种情况下，Neutron 代理自动安装到主机上），也可以是实施 OpenStack API 的外部提供程序，在这种情况下，虚拟接口驱动程序是手动安装的自定义解决方案。

表 7.265. 值概述

名称	概述
external	表示该提供程序是外部提供商，它实施 OpenStack Neutron API。
Neutron	表示提供程序为 OpenStack Neutron。

7.199.1. external

表示该提供程序是外部提供商，它实施 OpenStack Neutron API。在这种情况下，虚拟接口驱动程序由外部提供程序实施。

7.199.2. Neutron

表示提供程序为 OpenStack Neutron。标准 OpenStack Neutron 代理用作虚拟接口驱动程序。

7.200. OPENSTACKPROVIDER STRUCT

表 7.266. 属性概述

名称	类型	概述
authentication_url	字符串	定义外部提供程序身份验证 URL 地址。

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
password	字符串	定义用户在身份验证过程中的密码。
属性	attribute[]	提供程序名称/值属性数组。
required_authentication	布尔值	定义是否需要供应商身份验证。
tenant_name	字符串	定义 OpenStack Identity API v2 的租户名称。
url	字符串	定义外部提供程序的 URL 地址。
username	字符串	定义要在身份验证过程中使用的用户名。

7.200.1. required_authentication

定义是否需要供应商身份验证。

如果需要身份验证，则身份验证期间将使用 *用户名和密码* 属性。

7.200.2. tenant_name

定义 *OpenStack Identity API v2.0* 的租户名称。

7.201. OPENSTACKSUBNET STRUCT

表 7.267. 属性概述

名称	类型	概述
cidr	字符串	定义网络 CIDR。
comment	字符串	包含有关此对象的注释的免费文本。

名称	类型	概述
description	字符串	以纯文本格式人类可读的描述。
dns_servers	string[]	定义 DNS 服务器列表。
gateway	字符串	定义 IP 网关。
id	字符串	唯一标识符。
ip_version	字符串	定义 IP 版本。
name	字符串	纯文本中的人类可读名称。

7.201.1. ip_version

定义 IP 版本。

IPv4 的值可以是 v4, 对于 IPv6, 可以是 'v 6。

表 7.268. 链接摘要

名称	类型	概述
openstack_network	OpenStackNetwork	引用管理 OpenStack 网络的服务。

7.202. OPENSTACKVOLUMEPROVIDER STRUCT

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 7.269. 属性概述

名称	类型	概述
authentication_url	字符串	定义外部提供程序身份验证 URL 地址。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。

名称	类型	概述
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
password	字符串	定义用户在身份验证过程中的密码。
属性	attribute[]	提供程序名称/值属性数组。
required_authentication	布尔值	定义是否需要供应商身份验证。
tenant_name	字符串	定义 OpenStack Identity API v2 的租户名称。
url	字符串	定义外部提供程序的 URL 地址。
username	字符串	定义要在身份验证过程中使用的用户名。

7.202.1. required_authentication

定义是否需要供应商身份验证。

如果需要身份验证，则身份验证期间将使用 **用户名和密码** 属性。

7.202.2. tenant_name

定义 **OpenStack Identity API v2.0** 的租户名称。

表 7.270. 链接摘要

名称	类型	概述
authentication_keys	OpenstackVolume AuthenticationKey []	
证书	certificate[]	
data_center	DataCenter	
volume_types	OpenStackVolume Type[]	

7.203. OPENSTACKVOLUMETYPE STRUCT

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 7.271. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
属性	attribute[]	

表 7.272. 链接摘要

名称	类型	概述
openstack_volume_provider	OpenStackVolumeProvider	

7.204. OPENSTACKVOLUMEAUTHENTICATIONKEY STRUCT

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 7.273. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
creation_date	Date	
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

名称	类型	概述
usage_type	OpenstackVolume AuthenticationKey UsageType	
uuid	字符串	
value	字符串	

表 7.274. 链接摘要

名称	类型	概述
openstack_volu me_provider	OpenStackVolume Provider	

7.205. OPENSTACKVOLUMEAUTHENTICATIONKEYUSAGETYPE ENUM

OpenStack 卷(Cinder)集成已被 Managed Block Storage 替代。

表 7.275. 值概述

名称	概述
Ceph	

7.206. OPERATINGSYSTEM STRUCT

描述操作系统的信息.这同时用于虚拟机和主机。

表 7.276. 属性概述

名称	类型	概述
boot	引导	配置启动序列。
cmdline	字符串	如果使用 Linux 操作系统，用于启动虚拟机的自定义内核参数。
custom_kernel_ cmdline	字符串	主机内核命令行的自定义部分。

名称	类型	概述
initrd	字符串	如果使用 Linux 操作系统，则在 ISO 存储域中自定义初始 ramdisk 的路径。
kernel	字符串	如果使用 Linux 操作系统，ISO 存储域中自定义内核的路径。
reported_kernel_cmdline	字符串	运行的主机报告的主机内核命令行。
type	字符串	人类可读的操作系统名称。
version	版本	

7.206.1. 引导

配置启动序列。



注意

不用于主机。

7.206.2. cmdline

如果使用 Linux 操作系统，用于启动虚拟机的自定义内核参数。



注意

不用于主机。

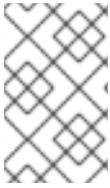
7.206.3. custom_kernel_cmdline

主机内核命令行的自定义部分。这将与现有内核命令行合并。

您必须重新安装，然后重新启动主机，以应用此属性所实施的更改。

在每个主机部署过程中，使用 `grubby --update-kernel DEFAULT --remove-args`

<previous_custom_params> 以及当前的内核命令行自定义使用 `grubby --update-kernel DEFAULT --args <custom_params>` 来应用当前的内核命令行自定义。Manager 在内部跟踪最近应用的内核参数自定义。



注意

此属性目前仅用于主机。

7.206.4. initrd

如果使用 Linux 操作系统，则在 ISO 存储域中自定义初始 ramdisk 的路径。

例如 `iso://initramfs-3.10.0-514.6.1.el7.x86_64.img`。



注意

不用于主机。

7.206.5. kernel

如果使用 Linux 操作系统，ISO 存储域中自定义内核的路径。

例如 `iso://vmlinuz-3.10.0-514.6.1.el7.x86_64`。



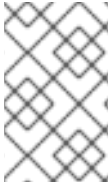
注意

不用于主机。

7.206.6. reported_kernel_cmdline

运行的主机报告的主机内核命令行。

这是一个只读属性。更改此属性的尝试会被静默忽略。



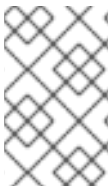
注意

此属性目前仅用于主机。

7.206.7. type

人类可读的操作系统名称。

例如 *Fedora* 或 *RHEL*。通常，操作系统服务返回的其中一个名称。



注意

主机只读。

7.207. OPERATINGSYSTEMINFO STRUCT

表示客户机操作系统。

表 7.277. 属性概述

名称	类型	概述
架构	构架	操作系统架构。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
large_icon	icon	客户机操作系统的大图标。
name	字符串	纯文本中的人类可读名称。
small_icon	icon	客户机操作系统的小图标。

7.207.1. large_icon

客户机操作系统的大图标。最大尺寸：宽 150px，高度 120px。

7.207.2. small_icon

客户机操作系统的小图标。最大尺寸：宽度 43px，高度 43px。

7.208. OPTION STRUCT

表 7.278. 属性概述

名称	类型	概述
name	字符串	
type	字符串	
value	字符串	

7.209. OSTYPE ENUM

键入代表操作系统的类型。



警告

随着 `OperatingSystemInfo` 类型的引入，此类型已弃用。操作系统作为 API 中的顶层集合提供：`oper_systems`

最终用户通过选择这些值之一来声明虚拟机（guest 操作系统）中安装的操作系统的类型。此声明使系统能够调优虚拟机配置，以获得更好的用户体验。例如，系统选择最适合操作系统的设备。请注意，系统依赖于用户的选择，且不会通过检查安装的实际 guest 操作系统进行验证。

表 7.279. 值概述

名称	概述
其他	其他类型的操作系统，并未由其他值指定。
other_linux	除其他值指定的 Linux 之外的 Linux 发行版。

名称	概述
rhel_3	红帽企业 Linux 3 32 位.
rhel_3x64	红帽企业 Linux 3 64 位.
rhel_4	红帽企业 Linux 4 32 位.
rhel_4x64	红帽企业 Linux 4 64 位.
rhel_5	红帽企业 Linux 5 32 位.
rhel_5x64	红帽企业 Linux 5 64 位.
rhel_6	红帽企业 Linux 6 32 位.
rhel_6x64	红帽企业 Linux 6 64 位.
unassigned	此值映射到 其他 值。
windows_2003	Windows 2003 32-bit.
windows_2003x64	Windows 2003 64-bit.
windows_2008	Windows 2008 32-bit.
windows_2008r2x64	Windows 2008 R2 64-bit.
windows_2008x64	Windows 2008 64-bit.
windows_2012x64	Windows 2012 64-bit.
windows_7	Windows 7 32-bit.
windows_7x64	Windows 7 64-bit.
windows_8	Windows 8 32-bit.
windows_8x64	Windows 8 64-bit.
windows_xp	Windows XP.

7.210. 软件包 结构

键入代表包。

这是软件包元素的一个示例：

```
<package>
  <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
</package>
```

表 7.280. 属性概述

名称	类型	概述
name	字符串	软件包的名称。

7.211. PAYLOAD STRUCT

表 7.281. 属性概述

名称	类型	概述
文件	file[]	
type	VmDeviceType	
volume_id	字符串	

7.212. PAYLOADENCODING 枚举

表 7.282. 值概述

名称	概述
base64	
plaintext	

7.213. PERMISSION STRUCT

类型表示权限。

表 7.283. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.284. 链接摘要

名称	类型	概述
cluster	Cluster	对群集的引用。
data_center	DataCenter	数据中心参考。
disk	Disk	引用磁盘。
group	组	对组的引用。
host	主机	主机引用。
role	角色	角色参考。
storage_domain	StorageDomain	对存储域的引用。
模板	Template	引用模板。
user	用户	用户引用。
vm	VM	引用虚拟机。
vm_pool	VmPool	虚拟机池的引用。

7.214. PERMIT STRUCT

类型表示允许。

表 7.285. 属性概述

名称	类型	概述
----	----	----

名称	类型	概述
管理	布尔值	指定许可是否是管理权限。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.286. 链接摘要

名称	类型	概述
role	角色	对允许所属角色的引用。

7.215. PMPROXY STRUCT

表 7.287. 属性概述

名称	类型	概述
type	PmProxyType	

7.216. PMPROXYTYPE ENUM

表 7.288. 值概述

名称	概述
cluster	隔离代理从与隔离的主机相同的集群中选择。
dc	隔离代理从与隔离的主机相同的数据中心选择。
other_dc	隔离代理是从与隔离的主机不同的数据中心选择的。

7.217. POLICYUNITTYPE ENUM

包含所有内部策略单元类型的类型。

表 7.289. 值概述

名称	概述
filter	
load_balancing	
weight	

7.218. PORTMIRRORING STRUCT

7.219. POWERMANAGEMENT STRUCT

表 7.290. 属性概述

名称	类型	概述
地址	字符串	主机的主机名或 IP 地址。
代理	agent[]	当使用多个隔离时，指定隔离代理选项。
automatic_pm_enabled	布尔值	切换主机的自动电源控制以节省能源。
enabled	布尔值	指明电源管理配置是启用或禁用的。
kdump_detection	布尔值	在关闭 kdump 前，切换是否在主机上运行 kdump。
选项	选项[]	使用选项 name="" 和 value="" 字符串指定的 type= 的隔离选项。
password	字符串	有效、稳定的电源管理密码。
pm_proxies	PmProxy[]	确定电源管理代理。
status	PowerManagementStatus	确定主机的电源状态。
type	字符串	隔离设备代码。
username	字符串	用于电源管理的有效用户名。

7.219.1. 代理

当使用多个隔离时，指定隔离代理选项。

使用排序子元素来优先选择隔离代理。代理会按照其顺序按顺序运行，直到隔离操作成功为止。当两个或多个隔离代理具有相同的顺序时，它们会被同时运行。其他子元素包括 `type`、`ip`、`user`、`password` 和 `options`。

7.219.2. `automatic_pm_enabled`

切换主机的自动电源控制以节省能源。当设置为 `true` 时，如果集群的负载较低，则主机将自动关闭，并在需要时再次打开。当主机创建时，这设置为 `true`，除非被用户禁用。

7.219.3. `kdump_detection`

在关闭 `kdump` 前，切换是否在主机上运行 `kdump`。当设置为 `true` 时，主机不会在 `kdump` 过程中关闭。当主机启用了电源管理时，这设置为 `true`，除非用户禁用。

7.219.4. `type`

隔离设备代码。

功能集中提供了有效隔离设备代码的列表。

7.220. `POWERMANAGEMENTSTATUS ENUM`

表 7.291. 值概述

名称	概述
<code>off</code>	主机为 OFF。
<code>on</code>	主机是 ON。
<code>Unknown</code>	未知状态。

7.221. 产品结构

表 7.292. 属性概述

名称	类型	概述
<code>comment</code>	字符串	包含有关此对象的注释的免费文本。

名称	类型	概述
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

7.222. PRODUCTINFO STRUCT

产品信息。

入口点包含一个 `product_info` 元素，可帮助 API 用户确定 Red Hat Virtualization 环境的真实性。这包括产品的名称、供应商和版本。

验证真正的红帽虚拟化环境

以下元素确定真正的 Red Hat Virtualization 环境：

```
<api>
...
<product_info>
  <name>oVirt Engine</name>
  <vendor>ovirt.org</vendor>
  <version>
    <build>0</build>
    <full_version>4.1.0_master</full_version>
    <major>4</major>
    <minor>1</minor>
    <revision>0</revision>
  </version>
</product_info>
...
</api>
```

表 7.293. 属性概述

名称	类型	概述
instance_id	字符串	产品此特定安装的 ID。
name	字符串	产品的名称，如 oVirt Engine 。

名称	类型	概述
vendor	字符串	供应商的名称，如 'ovirt。
version	版本	产品的版本号。

7.222.1. vendor

供应商的名称，如 *ovirt.org*。

7.223. PROFILEDETAIL STRUCT

表 7.294. 属性概述

名称	类型	概述
block_statistics	BlockStatistic[]	
duration	整数	
fop_statistics	FopStatistic[]	
profile_type	字符串	
statistics	Statistic[]	

7.224. 属性 结构

表 7.295. 属性概述

名称	类型	概述
name	字符串	
value	字符串	

7.225. PROXYTICKET STRUCT

表 7.296. 属性概述

名称	类型	概述
value	字符串	

7.226. QCOWVERSION ENUM

QCOW 版本将卷支持的 qemu 版本指定为 qemu 版本。

该字段可以使用更新 API 进行更新，并且仅报告 QCOW 卷，由创建磁盘的存储域版本决定。版本低于 V4 的存储域支持 QCOW2 版本 2 卷，而 V4 存储域也支持 QCOW2 版本 3。有关不同 QCOW 版本功能的更多信息，请参阅[这里](#)。

表 7.297. 值概述

名称	概述
qcow2_v2	Copy On Write 默认兼容性版本意味着每个 QEMU 都可以使用它。
qcow2_v3	QEMU 1 中引入的 Copy On Write 兼容性版本。

7.226.1. qcow2_v3

QEMU 1.1 中引入的 Copy On Write 兼容性版本意味着新的格式正在使用中。

7.227. QOS STRUCT

此类型表示用于定义服务质量(QoS)的属性。

对于存储，类型为 **storage** 的属性

`max_throughput`、`max_read_throughput`、`max_write_throughput`、`max_iops`、`max_read_iops` 和 `max_write_iops` 相关。

对于具有计算能力的资源，类型为 **cpu**，属性 `cpu_limit` 相关。

对于虚拟机网络，类型为 **network**，`in bound_average`、`in bound_peak`、`in bound_burst`、`outbound_average`、`outbound_peak` 和 `outbound_burst` 相关。

对于主机网络，类型为 **hostnetwork**，属性 `outbound_average_linkshare`、`outbound_average_upperlimit` 和 `outbound_average_realtime` 相关。

表 7.298. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
cpu_limit	整数	% 的最大处理能力。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
inbound_average	整数	所需的平均入站率以 Mbps 为单位（每秒多位数）。
inbound_burst	整数	单个突发中可以传输的数据量（以 MB 为单位）。
inbound_peak	整数	Mbps 的最大入站率（每秒的兆位数）。
max_iops	整数	每秒允许的输入和输出操作数上限。
max_read_iops	整数	每秒允许的输入操作数量上限。
max_read_throughput	整数	读取操作允许的最大吞吐量。
max_throughput	整数	允许的最大总吞吐量数。
max_write_iops	整数	每秒允许的输入操作数量上限。
max_write_throughput	整数	写入操作允许的最大吞吐量。
name	字符串	纯文本中的人类可读名称。
outbound_average	整数	以 Mbps 为单位所需的平均出站率（每秒 Megabits）。
outbound_average_linkshare	整数	加权份额。
outbound_average_realtime	整数	提交率（以 Mbps 为单位）。
outbound_average_upperlimit	整数	网络以 Mbps 为单位使用的最大带宽（每 sec 的 Megabits）。
outbound_burst	整数	单个突发中可以发送的数据量（以 MB 为单位）。

名称	类型	概述
outbound_peak	整数	Mbps 中最大出站率（每 sec 兆位数）。
type	QoSType	可为此条目分配的资源类型。

7.227.1. *cpu_limit*

% 的最大处理能力。

用于配置计算资源。

7.227.2. *inbound_average*

所需的平均入站率以 Mbps 为单位（每秒多位数）。

*用于配置虚拟机网络。如果定义，还必须设置 *inbound_peak* 和 *inbound_burst*。*

详情请查看 [Libvirt-QOS](#)。

7.227.3. *inbound_burst*

单个突发中可以传输的数据量（以 MB 为单位）。

*用于配置虚拟机网络。如果定义，还必须设置 *inbound_average* 和 *inbound_peak*。*

详情请查看 [Libvirt-QOS](#)。

7.227.4. *inbound_peak*

Mbps 的最大入站率（每秒的兆位数）。

用于配置虚拟机网络。如果定义，还必须设置 `inbound_average` 和 `inbound_burst`。

详情请查看 [Libvirt-QOS](#)。

7.227.5. `max_iops`

每秒允许的输入和输出操作数上限。

用于配置存储。如果设置了 `max_read_iops` 或 `max_write_iops`，则不得设置 `max_iops`。

7.227.6. `max_read_iops`

每秒允许的输入操作数量上限。

用于配置存储。如果设置了 `max_iops`，则不能设置 `max_read_iops`。

7.227.7. `max_read_throughput`

读取操作允许的最大吞吐量。

用于配置存储。如果设置了 `max_throughput`，则不能设置 `max_read_throughput`。

7.227.8. `max_throughput`

允许的最大总吞吐量数。

用于配置存储。如果设置了 `max_read_throughput` 或 `max_write_throughput`，则不能设置 `max_throughput`。

7.227.9. `max_write_iops`

每秒允许的输入操作数量上限。

用于配置存储。如果设置了 `max_iops`，则不能设置 `max_iops`。

7.227.10. `max_write_throughput`

写入操作允许的最大吞吐量。

用于配置存储。如果设置了 `max_throughput`，则不能设置 `max_throughput`。

7.227.11. `outbound_average`

以 `Mbps` 为单位所需的平均出站位率（每秒Megabits）。

用于配置虚拟机网络。如果定义，还必须设置 `outbound_peak` 和 `outbound_burst`。

详情请查看 [Libvirt-QOS](#)。

7.227.12. `outbound_average_linkshare`

加权份额。

用于配置主机网络。相对于附加到同一逻辑链接的其他网络，表示应分配特定网络的逻辑网络的容量。确切的共享取决于该链接上所有网络的共享总和。默认情况下，这是 1 到 100 范围内的一个数字。

7.227.13. `outbound_average_realtime`

提交率（以 `Mbps` 为单位）。

用于配置主机网络。网络需要的最小带宽。请求的提交率不受保证，并且将根据网络基础架构和其他网络在同一逻辑链路上请求的提交费率而有所不同。

7.227.14. `outbound_average_upperlimit`

网络以 Mbps 为单位使用的最大带宽（每 sec 的 Megabits）。

用于配置主机网络。如果提供了 `outboundAverageUpperlimit` 和 `outbound_average_realtime`，则 `出站_averageUpperlimit` 不能低于 `outbound_average_realtime`。

详情请查看 [Libvirt-QoS](#)。

7.227.15. `outbound_burst`

单个突发中可以发送的数据量（以 MB 为单位）。

用于配置虚拟机网络。如果定义，还必须设置 `outbound_average` 和 `outbound_peak`。

详情请查看 [Libvirt-QoS](#)。

7.227.16. `outbound_peak`

Mbps 中最大出站率（每 sec 兆位数）。

用于配置虚拟机网络。如果定义，还必须设置 `outbound_average` 和 `outbound_burst`。

详情请查看 [Libvirt-QoS](#)。

表 7.299. 链接摘要

名称	类型	概述
<code>data_center</code>	<code>DataCenter</code>	QoS 分配到的数据中心。

7.228. `QOSTYPE ENUM`

此类型表示可分配给服务质量(QoS)的资源类型。

表 7.300. 值概述

名称	概述
cpu	服务质量(QoS) 可分配给具有计算能力的资源。
hostnetwork	服务质量(QoS) 可以分配给主机网络。
network	服务质量(QoS) 可以分配给虚拟机网络。
storage	服务质量(QoS) 可分配到存储。

7.229. QUOTA STRUCT

代表配额对象。

配额的 XML 示例：

```
<quota href="/ovirt-engine/api/datacenters/7044934e/quotas/dcad5ddc" id="dcad5ddc">
  <name>My Quota</name>
  <description>A quota for my oVirt environment</description>
  <cluster_hard_limit_pct>0</cluster_hard_limit_pct>
  <cluster_soft_limit_pct>0</cluster_soft_limit_pct>
  <data_center href="/ovirt-engine/api/datacenters/7044934e" id="7044934e"/>
  <storage_hard_limit_pct>0</storage_hard_limit_pct>
  <storage_soft_limit_pct>0</storage_soft_limit_pct>
</quota>
```

表 7.301. 属性概述

名称	类型	概述
cluster_hard_limit_pct	整数	
cluster_soft_limit_pct	整数	
comment	字符串	包含有关此对象的注释的免费文本。
data_center	DataCenter	
description	字符串	以纯文本格式人类可读的描述。
磁盘	disk[]	
id	字符串	唯一标识符。

名称	类型	概述
name	字符串	纯文本中的人类可读名称。
storage_hard_limit_pct	整数	
storage_soft_limit_pct	整数	
用户	user[]	
vms	Vm[]	

表 7.302. 链接摘要

名称	类型	概述
权限	permission[]	
quota_cluster_limits	QuotaClusterLimit[]	
quota_storage_limits	QuotaStorageLimit[]	

7.230. QUOTA_CLUSTER_LIMIT_STRUCT

表 7.303. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
memory_limit	十进制	
memory_usage	十进制	
name	字符串	纯文本中的人类可读名称。
vcpu_limit	整数	

名称	类型	概述
vcpu_usage	整数	

表 7.304. 链接摘要

名称	类型	概述
cluster	Cluster	
quota	Quota	

7.231. QUOTAMODETYPE ENUM

表 7.305. 值概述

名称	概述
Audit	
disabled	
enabled	

7.232. QUOTASTORAGELIMIT STRUCT

表 7.306. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
limit	整数	
name	字符串	纯文本中的人类可读名称。
usage	十进制	

表 7.307. 链接摘要

名称	类型	概述
quota	Quota	
storage_domain	StorageDomain	

7.233. RANGE STRUCT

表 7.308. 属性概述

名称	类型	概述
来自	字符串	
为	字符串	

7.234. RATE STRUCT

确定随机数字生成器设备的最大字节消耗速度。

表 7.309. 属性概述

名称	类型	概述
字节	整数	每个期间内允许消耗的字节数。
周期	整数	持续一个时段（毫秒）。

7.235. REGISTRATIONAFFINITYGROUPMAPPING STRUCT

此类型描述了如何在对象注册过程中映射关联性组。对象可以是虚拟机、模板等。

使用这个映射的 XML 表述示例：

```
<action>
  <registration_configuration>
    <affinity_group_mappings>
      <registration_affinity_group_mapping>
        <from>
          <name>affinity</name>
        </from>
        <to>
```

```

    <name>affinity2</name>
  </to>
</registration_affinity_group_mapping>
</affinity_group_mappings>
</registration_configuration>
</action>

```

表 7.310. 链接摘要

名称	类型	概述
来自	AffinityGroup	引用原始关联性组。
为	AffinityGroup	引用目标关联性组。

7.235.1. 来自

引用原始关联性组。它可以通过 *名称* 来指定。

7.236. REGISTRATIONAFFINITYLABELMAPPING STRUCT

此类型描述了如何在对象注册过程中映射关联性标签。对象可以是虚拟机、模板等。

使用映射的 XML 表示示例：

```

<action>
  <registration_configuration>
    <affinity_label_mappings>
      <registration_affinity_label_mapping>
        <from>
          <name>affinity_label</name>
        </from>
        <to>
          <name>affinity_label2</name>
        </to>
      </registration_affinity_label_mapping>
    </affinity_label_mappings>
  </registration_configuration>
</action>

```

表 7.311. 链接摘要

名称	类型	概述
来自	AffinityLabel	引用原始关联性标签。

名称	类型	概述
为	AffinityLabel	引用目的地关联性标签。

7.236.1. 来自

引用原始关联性标签。它可以通过 `名称` 来指定。

7.237. REGISTRATIONCLUSTERMAPPING STRUCT

此类型描述了如何在对象注册过程中映射集群。对象可以是虚拟机、模板等。

使用这个映射的 XML 表述示例：

```
<action>
  <registration_configuration>
    <cluster_mappings>
      <registration_cluster_mapping>
        <from>
          <name>myoriginalcluster</name>
        </from>
        <to>
          <name>mynewcluster</name>
        </to>
      </registration_cluster_mapping>
    </cluster_mappings>
  </registration_configuration>
</action>
```

表 7.312. 链接摘要

名称	类型	概述
来自	Cluster	引用原始集群。
为	Cluster	引用目标集群。

7.237.1. 来自

引用原始集群。它可使用 `id` 或 `名称` 指定。

7.237.2. 为

引用目标集群。它可使用 `id` 或 `名称` 指定。

7.238. REGISTRATIONCONFIGURATION STRUCT

此类型描述了对对象（虚拟机、模板等）如何注册，用于灾难恢复解决方案的实施。

此类型中包含的每个映射都可用于将原始系统中的对象映射到注册虚拟机或模板的系统中的对象。例如，可能会有一个主设置，其中包含在群集 A 上配置的虚拟机，以及一个带有群集 B 的活动次要设置。群集 B 与该虚拟机兼容，如果发生灾难恢复方案，存储域可以导入到次要设置，用户可以将虚拟机注册到群集 B。

在这种情况下，我们可以通过定义集群映射来自动执行恢复过程。注册实体后，其 OVF 将指示它属于群集 A，但映射将指示群集 A 将替换为群集 B。红帽虚拟化管理器应执行该交换机，并将虚拟机注册到次要站点中的群集 B。

集群映射只是一个示例，有不同的映射类型：

- 群集映射.
- LUN 映射.
- 角色映射.
- 域映射.
- 权限映射.
- 关联性组映射.
- 关联性标签映射.

• **虚拟 NIC 配置文件映射.**

在 *Red Hat Virtualization Manager* 中执行寄存器操作后，每个映射都将用于其特定的 OVF 数据。

使用映射的 XML 表述示例：

```
<action>
  <registration_configuration>
    <cluster_mappings>
      <registration_cluster_mapping>
        <from>
          <name>myoriginalcluster</name>
        </from>
        <to>
          <name>mynewcluster</name>
        </to>
      </registration_cluster_mapping>
    </cluster_mappings>
    <role_mappings>
      <registration_role_mapping>
        <from>
          <name>SuperUser</name>
        </from>
        <to>
          <name>UserVmRunTimeManager</name>
        </to>
      </registration_role_mapping>
    </role_mappings>
    <domain_mappings>
      <registration_domain_mapping>
        <from>
          <name>redhat</name>
        </from>
        <to>
          <name>internal</name>
        </to>
      </registration_domain_mapping>
    </domain_mappings>
    <lun_mappings>
      <registration_lun_mapping>
        <from id="111">
        </from>
        <to id="222">
          <alias>weTestLun</alias>
          <lun_storage>
            <type>iscsi</type>
            <logical_units>
              <logical_unit id="36001405fb1ddb4b91e44078f1ffcfef">
                <address>44.33.11.22</address>
                <port>3260</port>
              </logical_unit>
            </logical_units>
          </lun_storage>
        </to>
      </registration_lun_mapping>
    </lun_mappings>
  </registration_configuration>
</action>
```

```

        <portal>1</portal>
        <target>iqn.2017-11.com.name.redhat:444</target>
    </logical_unit>
</logical_units>
</lun_storage>
</to>
</registration_lun_mapping>
</lun_mappings>
<affinity_group_mappings>
<registration_affinity_group_mapping>
  <from>
    <name>affinity</name>
  </from>
  <to>
    <name>affinity2</name>
  </to>
</registration_affinity_group_mapping>
</affinity_group_mappings>
<affinity_label_mappings>
<registration_affinity_label_mapping>
  <from>
    <name>affinity_label</name>
  </from>
  <to>
    <name>affinity_label2</name>
  </to>
</registration_affinity_label_mapping>
</affinity_label_mappings>
<vnic_profile_mappings>
<registration_vnic_profile_mapping>
  <from>
    <name>gold</name>
    <network>
      <name>red</name>
    </network>
  </from>
  <to id="738dd914-8ec8-4a8b-8628-34672a5d449b"/>
</registration_vnic_profile_mapping>
<registration_vnic_profile_mapping>
  <from>
    <name>silver</name>
    <network>
      <name>blue</name>
    </network>
  </from>
  <to>
    <name>copper</name>
    <network>
      <name>orange</name>
    </network>
  </to>
</registration_vnic_profile_mapping>
</vnic_profile_mappings>
</registration_configuration>
</action>

```

表 7.313. 属性概述

名称	类型	概述
affinity_group_mappings	RegistrationAffinityGroupMapping[]	描述如何映射关联性组。
affinity_label_mappings	RegistrationAffinityLabelMapping[]	描述如何映射关联性标签。
cluster_mappings	RegistrationClusterMapping[]	描述如何映射对象引用的集群。
domain_mappings	RegistrationDomainMapping[]	描述如何映射用户的域。
lun_mappings	RegistrationLunMapping[]	描述如何映射 LUN。
role_mappings	RegistrationRoleMapping[]	描述如何映射角色。
vnic_profile_mappings	RegistrationVnicProfileMapping[]	映射将在注册过程中应用的虚拟 NIC 配置集的规则。

7.239. REGISTRATIONDOMAINMAPPING STRUCT

此类型描述了如何在对象注册过程中映射用户的域。对象可以是虚拟机、模板等。注意：这是基于用户名相同且仅更改域名这一假设。

使用这个映射的 XML 表述示例：

```
<action>
  <registration_configuration>
    <domain_mappings>
      <registration_domain_mapping>
        <from>
          <name>redhat</name>
        </from>
        <to>
          <name>internal</name>
        </to>
      </registration_domain_mapping>
    </domain_mappings>
  </registration_configuration>
</action>
```

表 7.314. 链接摘要

名称	类型	概述
来自	域	引用原始域.
为	域	引用目标域.

7.239.1. 来自

引用原始域.它可以通过 名称 来指定.

7.240. REGISTRATIONLUNMAPPING STRUCT

此类型描述了如何在对象注册过程中映射 LUN。对象可以是虚拟机、模板等。

外部 LUN 磁盘是不驻留于存储域中的实体。必须指定它，因为它不需要存在于对象注册的环境中。使用这个映射的 XML 表述示例：

```
<action>
  <registration_configuration>
    <lun_mappings>
      <registration_lun_mapping>
    <lun_mappings>
      <registration_lun_mapping>
        <from id="111">
          </from>
        <to id="222">
          <alias>weTestLun</alias>
          <lun_storage>
            <type>iscsi</type>
            <logical_units>
              <logical_unit id="36001405fb1ddb4b91e44078f1ffcfef">
                <address>44.33.11.22</address>
                <port>3260</port>
                <portal>1</portal>
                <target>iqn.2017-11.com.name.redhat:444</target>
              </logical_unit>
            </logical_units>
          </lun_storage>
        </to>
      </registration_lun_mapping>
    </lun_mappings>
  </registration_configuration>
</action>
```

表 7.315. 链接摘要

名称	类型	概述
来自	Disk	引用原始 LUN。
为	Disk	引用要添加到虚拟机的 LUN。

7.240.1. 来自

引用原始 LUN。这必须使用 `id` 属性指定。

7.241. REGISTRATIONROLEMAPPING STRUCT

此类型描述了如何在对象注册过程中映射角色。对象可以是虚拟机、模板等。

角色映射旨在映射主站点和次要站点之间的角色关联。例如，注册的虚拟机可能有权限，角色 `UserVmRunTimeManager` 角色。因此，我们可以发送一个映射，它将使用 `SuperUser` 角色而不是 `UserVmRunTimeManager` 在第二个设置中注册虚拟机。使用这个映射的 XML 表示示例：

```
<action>
  <registration_configuration>
    <role_mappings>
      <registration_eole_mapping>
        <from>
          <name>SuperUser</name>
        </from>
        <to>
          <name>UserVmRunTimeManager</name>
        </to>
      </registration_role_mapping>
    </role_mappings>
  </registration_configuration>
</action>
```

表 7.316. 链接摘要

名称	类型	概述
来自	角色	对原始角色的引用。
为	角色	对目标角色的引用。

7.241.1. 来自

对原始角色的引用.它可以通过 `名称` 来指定。

7.242. REGISTRATIONVNICPROFILEMAPPING STRUCT

将外部虚拟 NIC 配置集映射到 Red Hat Virtualization Manager 中存在的 NIC 配置集。目标可指定为配置文件 ID 或一对配置文件名称和网络名称。

例如，如果所需的虚拟 NIC 配置集映射包括以下行：

源网络名称	源网络配置集名称	目标虚拟 NIC 配置集 ID\names
红色	gold	738dd914-8ec8-4a8b-8628-34672a5d449b
<empty> (无网络名称)	<empty> (无网络配置集名称)	892a12ec-2028-4451-80aa-ff3bf55d6bac
blue	Silver	orange\copper
黄色	Platinum	<empty> (无配置集)
绿色	bronze	

然后，应将以下代码片段添加到 `RegistrationConfiguration`

```
<vnic_profile_mappings>
  <registration_vnic_profile_mapping>
    <from>
      <name>gold</name>
      <network>
        <name>red</name>
      </network>
    </from>
    <to id="738dd914-8ec8-4a8b-8628-34672a5d449b"/>
  </registration_vnic_profile_mapping>
  <registration_vnic_profile_mapping>
    <from>
      <name></name>
      <network>
        <name></name>
      </network>
    </from>
    <to id="892a12ec-2028-4451-80aa-ff3bf55d6bac"/>
  </registration_vnic_profile_mapping>
```

```

<registration_vnic_profile_mapping>
  <from>
    <name>silver</name>
    <network>
      <name>blue</name>
    </network>
  </from>
  <to>
    <name>copper</name>
    <network>
      <name>orange</name>
    </network>
  </to>
</registration_vnic_profile_mapping>
<registration_vnic_profile_mapping>
  <from>
    <name>platinum</name>
    <network>
      <name>yellow</name>
    </network>
  </from>
  <to>
    <name></name>
    <network>
      <name></name>
    </network>
  </to>
</registration_vnic_profile_mapping>
<registration_vnic_profile_mapping>
  <from>
    <name>bronze</name>
    <network>
      <name>green</name>
    </network>
  </from>
</registration_vnic_profile_mapping>
</vnic_profile_mappings>

```

表 7.317. 链接摘要

名称	类型	概述
来自	VnicProfile	对外部网络和外部网络配置文件的引用。
为	VnicProfile	引用现有虚拟 NIC 配置文件。

7.242.1. 来自

对外部网络和外部网络配置文件的引用。这两个都应通过其名称来指定。

7.242.2. 为

引用现有虚拟 NIC 配置文件.它应使用其名称或 ID 指定。应指定 name 或 id, 但不指定两者。

7.243. REPORTEDCONFIGURATION STRUCT

表 7.318. 属性概述

名称	类型	概述
actual_value	字符串	
expected_value	字符串	
in_sync	布尔值	当网络附加包含未分配的网络配置时为 false 。
name	字符串	

7.244. REPORTEDDEVICE STRUCT

表 7.319. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
ips	Ip[]	
MAC	MAC	
name	字符串	纯文本中的人类可读名称。
type	ReportedDeviceType	

表 7.320. 链接摘要

名称	类型	概述
vm	VM	

7.245. REPORTEDDEVICETYPE ENUM

表 7.321. 值概述

名称	概述
network	

7.246. RESOLUTIONTYPE 枚举

表 7.322. 值概述

名称	概述
add	
copy	

7.247. RNGDEVICE STRUCT

随机数字生成器(RNG)设备模型。

表 7.323. 属性概述

名称	类型	概述
速率	速率	确定随机数字生成器设备的最大字节消耗速度。
source	RngSource	随机数字生成器设备的后端。

7.248. RINGSOURCE ENUM

代表随机生成器后端类型。

表 7.324. 值概述

名称	概述
hwrng	从 <code>/dev/hwrng</code> (通常是专用的 HW 生成器) 设备获取随机数据。
random	从 <code>/dev/random</code> 设备获取随机数据。
urandom	从 <code>/dev/urandom</code> 设备获取随机数据。

7.248.1. urandom

从 `/dev/urandom` 设备获取随机数据。

此 RNG 源旨在为非集群识别实体（如 Blank 模板和实例类型）和与具有兼容性版本 4.1 或更高版本的集群关联的实体替换随机 RNG 源。

7.249. ROLE STRUCT

代表系统角色。

表 7.325. 属性概述

名称	类型	概述
管理	布尔值	该角色定义为仅限管理员或非管理员角色。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
可变	布尔值	定义更新或删除角色的功能。
name	字符串	纯文本中的人类可读名称。

7.249.1. 可变

定义更新或删除角色的功能。

将 `mutable` 设置为 `false` 的角色是预定义角色。

表 7.326. 链接摘要

名称	类型	概述
Allow	Permit[]	角色允许使用子集合的链接。
user	用户	

7.250. ROLETYPE ENUM

键入 表示角色是否为管理角色。授予至少一个管理角色的用户被视为管理员。

表 7.327. 值概述

名称	概述
admin	管理角色.
user	用户角色.

7.251. SCHEDULINGPOLICY STRUCT

表 7.328. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
default_policy	布尔值	
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
Locked	布尔值	
name	字符串	纯文本中的人类可读名称。
属性	attribute[]	

表 7.329. 链接摘要

名称	类型	概述
平衡	balance[]	
过滤器	filter[]	
weight	Weight[]	

7.252. SCHEDULINGPOLICYUNIT STRUCT

表 7.330. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
enabled	布尔值	
id	字符串	唯一标识符。
internal	布尔值	
name	字符串	纯文本中的人类可读名称。
属性	attribute[]	
type	PolicyUnitType	

7.253. SCSIGENERICIO ENUM

当直接 LUN 磁盘使用 SCSI 直通时，特权 I/O 策略由这个枚举决定。

表 7.331. 值概述

名称	概述
disabled	禁用 SCSI 直通。
filtered	不允许特权 SCSI I/O。
unfiltered	允许特权 SCSI I/O。

7.254. SELINUX STRUCT

在系统中表示 SELinux。

表 7.332. 属性概述

名称	类型	概述
mode	SeLinuxMode	SELinux 当前模式。

7.255. SELINUXMODE ENUM

表示 SELinux 强制模式。

表 7.333. 值概述

名称	概述
disabled	SELinux 在内核中被禁用。
enforcing	SELinux 正在运行并强制实施权限。
Permissive	SELinux 运行和记录，但不强制实施权限。

7.256. SERIALNUMBER STRUCT

表 7.334. 属性概述

名称	类型	概述
policy	SerialNumberPolicy	
value	字符串	

7.257. SERIALNUMBERPOLICY ENUM

键入 代表序列号策略。

表 7.335. 值概述

名称	概述
custom	此策略允许用户提供任意字符串作为 Serial Number。
host	此策略是传统策略。
none	此策略用于删除 Serial Number 策略，并将其移到 default: null。
vm	此策略将使用虚拟机 ID 作为串行号。

7.257.1. 主机

此策略是传统策略。它将使用主机 ID 作为串行号。

7.258. SESSION STRUCT

描述虚拟机的用户会话。

表 7.336. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
console_user	布尔值	指明这是否为控制台会话。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
ip	ip	用户从中连接的 IP 地址。
name	字符串	纯文本中的人类可读名称。
protocol	字符串	会话使用的协议。

7.258.1. console_user

指明这是否为控制台会话。

控制台用户 (SPICE 或 VNC) 的值为 `true`，其他人 (如 RDP 或 SSH) 的值为 `false`。

7.258.2. ip

用户从中连接的 IP 地址。

目前仅适用于控制台用户。

7.258.3. 协议

会话使用的协议。

当前未使用.用于关于如何连接用户的信息：通过 **SPICE**、**VNC**、**SSH** 或 **RDP**。

表 7.337. 链接摘要

名称	类型	概述
user	用户	与这个会话相关的用户。
vm	VM	与本次会话相关的虚拟机的链接。

7.258.4. user

与这个会话相关的用户。

如果用户是控制台用户，则此为实际 **Red Hat Virtualization** 用户的链接。否则，仅提供用户名。

7.259. SKIPIFCONNECTIVITYBROKEN STRUCT

表 7.338. 属性概述

名称	类型	概述
enabled	布尔值	如果启用，我们将不会在群集中主机超过连接丢失百分比时隔离主机。
threshold	整数	连接测试的阈值。

7.259.1. enabled

如果启用，我们将不会在群集中主机超过连接丢失百分比时隔离主机。这可用于防止在群集中存在全局网络问题时出现隔离风暴。

7.259.2. threshold

连接测试的阈值.如果群集中的至少阈值百分比的主机丢失了连接，则不会发生隔离。

7.260. SKIPFSDACTIVE STRUCT

此类型代表隔离策略中与存储相关的配置。

表 7.339. 属性概述

名称	类型	概述
enabled	布尔值	如果启用，我们将在主机在存储中维持其租用时跳过隔离。

7.260.1. enabled

如果启用，我们将在主机在存储中维持其租用时跳过隔离。这意味着，如果主机仍具有存储访问权限，则它不会被隔离。

7.261. SNAPSHOT STRUCT

表示快照对象。

XML 表述示例：

```
<snapshot id="456" href="/ovirt-engine/api/vms/123/snapshots/456">
  <actions>
    <link rel="restore" href="/ovirt-engine/api/vms/123/snapshots/456/restore"/>
  </actions>
  <vm id="123" href="/ovirt-engine/api/vms/123"/>
  <description>Virtual Machine 1 - Snapshot A</description>
  <type>active</type>
  <date>2010-08-16T14:24:29</date>
  <persist_memorystate>false</persist_memorystate>
</snapshot>
```

表 7.340. 属性概述

名称	类型	概述
auto_pinning_policy	AutoPinningPolicy	指定是否以及如何应用自动 CPU 和 NUMA 配置。
bios	BIOS	参考虚拟机的 BIOS 配置。
comment	字符串	包含有关此对象的注释的免费文本。

名称	类型	概述
console	控制台 (Console)	为此虚拟机配置的控制台。
cpu	cpu	虚拟机 CPU 的配置。
cpu_pinning_policy	CpuPinningPolicy	指定是否以及如何应用 CPU 和 NUMA 配置。
cpu_shares	整数	
creation_time	Date	虚拟机创建日期。
custom_compatibility_version	版本	虚拟机自定义兼容性版本。
custom_cpu_model	字符串	
custom_emulated_machine	字符串	
custom_properties	CustomProperty[]	发送到 VDSM 的属性，以配置各种 hook。
date	Date	创建此快照的日期。
delete_protected	布尔值	如果为 true ，则无法删除虚拟机。
description	字符串	以纯文本格式人类可读的描述。
display	显示	虚拟机显示配置。
domain	域	为此虚拟机配置的域。
fqdn	字符串	虚拟机的完全限定域名。
guest_operating_system	GuestOperatingSystem	虚拟机上安装什么操作系统。
guest_time_zone	timezone	虚拟机使用什么时区（由虚拟客户机代理返回）。
has_illegal_images	布尔值	指明虚拟机是否有带有 ILLEGAL 状态的磁盘的快照。

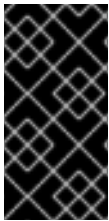
名称	类型	概述
high_availability	HighAvailability	虚拟机高可用性配置。
id	字符串	唯一标识符。
初始化	初始化	引用虚拟机的初始化配置。
io	io	用于 IO 线程性能调优。
large_icon	icon	虚拟机的大图标。
lease	StorageDomainLe ase	虚拟机/模板租用所在存储域的引用。
内存	整数	虚拟机的内存，以字节为单位。
memory_policy	MemoryPolicy	参考虚拟机的内存管理配置。
迁移	MigrationOptions	有关将正在运行的虚拟机迁移到另一台主机的配置。
migration_down time	整数	虚拟机在实时迁移到 ms 中另一主机期间可以不响应的最大时间。
multi_queues_e nabled	布尔值	如果为 true ，则每个虚拟接口都将获得最佳队列数，具体取决于可用的虚拟 Cpus。
name	字符串	纯文本中的人类可读名称。
next_run_config uration_exists	布尔值	虚拟机配置已更改，需要重启虚拟机。
numa_tune_mo de	NumaTuneMode	如何应用 NUMA 拓扑。
Origin	字符串	此虚拟机的起源。
OS	OperatingSystem	虚拟机上安装的操作系统类型。
有效负载	payload[]	虚拟机的可选有效负载，用于 ISO 进行配置。
persist_memory state	布尔值	指明虚拟机的内存内容是否包含在快照中。
placement_poli cy	VmPlacementPolic y	虚拟机放置策略的配置。

名称	类型	概述
rng_device	RngDevice	此虚拟机的随机数字生成器设备配置。
run_once	布尔值	如果为 true ，则使用 <code>run</code> 命令启动虚拟机一次，这意味着它的配置可能与存储的配置不同，用于本次运行。
serial_number	SerialNumber	集群中虚拟机的序列号。
small_icon	icon	虚拟机的小图标。
snapshot_statuses	SnapshotStatus	快照的状态。
snapshot_type	SnapshotType	快照类型。
soundcard_enabled	布尔值	如果为 true ，则会在虚拟机中添加声卡。
ssso	SSO	此虚拟机被配置至单点登录配置的引用。
start_paused	布尔值	如果为 true ，则虚拟机将在启动后的初始处于"暂停"状态。
start_time	Date	虚拟机启动的日期。
stateless	布尔值	如果为 true ，则虚拟机为无状态 - 它的状态（磁盘）会在关机后回滚。
status	VmStatus	虚拟机的当前状态。
status_detail	字符串	当前状态人类可读的详细信息。
stop_reason	字符串	虚拟机被停止的原因。
stop_time	Date	虚拟机停止的日期。
storage_error_resume_behaviour	VmStorageErrorResumeBehaviour	确定存储错误后如何恢复虚拟机。
time_zone	timezone	虚拟机的时区由 oVirt 设置。
tpm_enabled	布尔值	如果为 true ，则会在虚拟机中添加 TPM 设备。
tunnel_migration	布尔值	如果为 true ，则网络数据传输将在虚拟机实时迁移期间加密。
type	VmType	确定虚拟机是为桌面还是服务器优化。

名称	类型	概述
<code>usb</code>	<code>usb</code>	配置此虚拟机的 USB 设备（count、类型）。
<code>use_latest_template_version</code>	布尔值	如果为 true ，则虚拟机在启动时将其重新配置为模板的最新版本。
<code>virtio_scsi</code>	<code>VirtioScsi</code>	参考 VirtIO SCSI 配置。
<code>virtio_scsi_multi_queues</code>	整数	这个字段的 Virtio-SCSI controller 队列数量需要 <code>virtioScsiMultiQueuesEnabled</code> 。如需更多信息，请参阅 <code>virtioScsiMultiQueuesEnabled</code>
<code>virtio_scsi_multi_queues_enabled</code>	布尔值	如果为 true ，Virtio-SCSI 设备会根据可用的虚拟 Cpus 和磁盘或根据指定的 <code>virtioScsiMultiQueues</code> 来获得多个队列。

7.261.1. `auto_pinning_policy`

指定是否以及如何应用自动 CPU 和 NUMA 配置。



重要

由于引擎的 4.5 版已弃用，因此仅保留以向后兼容。以后可能会删除它。请使用 `CpuPinningPolicy`。

7.261.2. `cpu`

虚拟机 CPU 的配置。

可以在不重新启动虚拟机的情况下更新套接字配置。内核和线程需要重启。

例如：要将套接字数量立即更改为 4，并在重启后将内核和线程数量改为 2，发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```


通过请求正文：

```
<vm>
  <cpu>
    <topology>
      <sockets>4</sockets>
      <cores>2</cores>
      <threads>2</threads>
    </topology>
  </cpu>
</vm>
```

7.261.3. cpu_pinning_policy

指定是否以及如何应用 CPU 和 NUMA 配置。如果没有指定 CPU 固定字符串之前的行为，则会决定 CpuPinningPolicy 到 None 或 Manual。

7.261.4. custom_compatibility_version

虚拟机自定义兼容性版本。

使虚拟机能够根据自己的兼容性版本进行自定义。如果设置了 custom_compatibility_version，它将覆盖该特定虚拟机的集群的兼容性版本。

虚拟机的兼容性版本受位于的数据中心的限制，并根据虚拟机计划在其上运行的主机进行检查。

7.261.5. high_availability

虚拟机高可用性配置。如果设置，虚拟机将在意外停机时自动重新启动。

7.261.6. 初始化

引用虚拟机的初始化配置。



注意

自 Red Hat Virtualization 4.1.8 起，可以通过发送空标签来清除此属性。

例如，清除初始化属性会发送类似如下的请求：

```
PUT /ovirt-engine/api/vms/123
```

这样的请求正文：

```
<vm>  
  <initialization/>  
</vm>
```

对此类请求的响应和带有标头 `All-Content: true` 的请求仍将包含此属性。

7.261.7. large_icon

虚拟机的大图标。用户设置或引用根据操作系统设置的镜像集。

7.261.8. lease

虚拟机/模板租用所在存储域的引用。

使用租用运行的虚拟机需要检查运行时不会由另一主机获取租期，从而防止此虚拟机的另一个实例在另一主机上运行。这可防止高可用性虚拟机中的脑裂。模板也可为租用定义存储域，以便从该模板创建虚拟机，预先配置此存储域作为租用位置。

7.261.9. 内存

虚拟机的内存，以字节为单位。

例如，要更新虚拟机使其包含 1 Gibibyte(GiB)内存，请发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```

使用以下请求正文：

```
<vm>
  <memory>1073741824</memory>
</vm>
```

Red Hat Virtualization 3.6 支持内存热插拔。在虚拟机处于 **up** 状态时，您可以使用上述示例来增加内存。增量的大小必须可以被分成 `HotPlugMemoryBlockSizeMb` 配置值（默认为 256 MiB）。如果此值无法分隔内存大小递增，则内存大小更改仅存储在下次运行配置中。每个成功的内存热插拔操作都会创建一个或两个新内存设备。

从 Red Hat Virtualization 4.2 开始支持内存热插拔。**内存热拔头只能在虚拟机处于启动状态时执行**。只有之前热插拔内存设备才能通过热拔出操作删除。请求的内存减少向下舍入，以匹配之前热插拔内存设备的组合大小。请求的内存值存储在下次运行配置中，而不舍入。



注意

这个示例中的内存会被转换为字节：
 $1 \text{ GiB} = 2^{30} \text{ 字节} = 1073741824 \text{ 字节}$ 。



注意

红帽虚拟化管理器内部将值向下舍入到整个 MiB ($1 \text{ MiB} = 2^{20} \text{ 字节}$)

7.261.10. migration_downtime

虚拟机在实时迁移到 `ms` 中另一主机期间可以不响应的最大时间。

为虚拟机明确设置，或者使用 `engine-config -s DefaultMaximumMigrationDowntime=[value]`

7.261.11. next_run_configuration_exists

虚拟机配置已更改，需要重启虚拟机。更改的配置在处理虚拟机的关机时应用。

7.261.12. numa_tune_mode

如何应用 NUMA 拓扑。弃用了每个 vNUMA 节点的 NUMA 调优。

7.261.13. Origin

此虚拟机的起源。

可能的值：

- *ovirt*
- *rhev*
- *vmware*
- *Xen*
- *external*
- *hosted_engine*
- *managed_hosted_engine*
- *kvm*
- *physical_machine*
- *hyperv*

7.261.14. persist_memorystate

指明虚拟机的内存内容是否包含在快照中。

创建快照时，默认值为 `true`。

7.261.15. placement_policy

虚拟机放置策略的配置。

此配置可以更新为将虚拟机固定到一个或多个主机。



注意

固定到多个主机的虚拟机无法实时迁移，但在主机出现故障时，任何配置为高可用性的虚拟机都会自动在虚拟机要固定到的其他主机上重新启动。

例如，要将虚拟机固定到两个主机，请发送以下请求：

```
PUT /api/vms/123
```

这样的请求正文：

```
<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>
```

7.261.16. small_icon

虚拟机的小图标。用户设置或引用根据操作系统设置的镜像集。

7.261.17. sso

此虚拟机被配置至单点登录配置的引用。打开控制台时，可以在虚拟机的操作系统中自动签署该用户。

7.261.18. stop_reason

虚拟机被停止的原因。关闭虚拟机时由用户设置（可选）。

7.261.19. tpm_enabled

如果为 `true`，则会在虚拟机中添加 TPM 设备。默认值为 `false`。只有在设置了 "All-Content=true" 标头时，此属性才能在获取时可见。

表 7.341. 链接摘要

名称	类型	概述
<code>affinity_labels</code>	AffinityLabel[]	可选。
应用程序	application[]	在虚拟机上安装的应用程序列表。
<code>cdroms</code>	Cdrom[]	引用挂载到 CDROM 的 ISO。
<code>cluster</code>	Cluster	虚拟机所属群集的引用。
<code>cpu_profile</code>	CpuProfile	引用此虚拟机使用的 CPU 配置集。
<code>disk_attachments</code>	DiskAttachment[]	引用附加到虚拟机的磁盘。
磁盘	disk[]	链接到快照的磁盘列表。
<code>dynamic_cpu</code>	DynamicCpu	虚拟机 CPU 的动态配置。
<code>external_host_provider</code>	ExternalHostProvider	
<code>floppies</code>	Floppy[]	引用挂载到软盘的 ISO。

名称	类型	概述
graphics_consoles	GraphicsConsole[]	为此虚拟机配置的图形控制台列表。
host	主机	引用正在运行的虚拟机的主机。
host_devices	HostDevice[]	参考与此虚拟机关联的设备。
instance_type	InstanceType	虚拟机配置（可选）可通过其中一个实例类型预定义。
katello_errata	KatelloErratum[]	列出分配到虚拟机的所有 Katello 勘误表。
mediated_devices	VmMediatedDevice[]	介质设备配置。
nics	Nic[]	参考虚拟机上的网络接口设备列表。
numa_nodes	NumaNode[]	指此虚拟机使用的 NUMA 节点配置。
original_template	Template	引用用于创建虚拟机的原始模板。
权限	permission[]	为此虚拟机设置权限。
quota	Quota	引用为此虚拟机设置的配额配置。
reported_devices	ReportedDevice[]	
会话	session[]	为此虚拟机打开的用户会话列表。
snapshots	Snapshot[]	指的是从虚拟机获取的所有快照。
statistics	Statistic[]	从此虚拟机收集的统计信息。
storage_domain	StorageDomain	虚拟机所属存储域的引用。
tags	Tag[]	
模板	Template	对虚拟机所基于的模板的引用。
vm	VM	为此快照所生成的虚拟机。
vm_pool	VmPool	虚拟机所属池的引用（可选）

名称	类型	概述
watchdogs	Watchdog[]	引用 Watchdog 配置。

7.261.20. affinity_labels

可选。用于标记子集群。

7.261.21. katello_errata

列出分配到虚拟机的所有 Katello 勘误表。

GET /ovirt-engine/api/vms/123/katelloerrata

您将在 XML 中收到类似以下的响应：

```
<katello_errata>
  <katello_erratum href="/ovirt-engine/api/katelloerrata/456" id="456">
    <name>RHBA-2013:XYZ</name>
    <description>The description of the erratum</description>
    <title>some bug fix update</title>
    <type>bugfix</type>
    <issued>2013-11-20T02:00:00.000+02:00</issued>
    <solution>Few guidelines regarding the solution</solution>
    <summary>Updated packages that fix one bug are now available for XYZ</summary>
    <packages>
      <package>
        <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
      </package>
      ...
    </packages>
  </katello_erratum>
  ...
</katello_errata>
```

7.261.22. original_template

引用用于创建虚拟机的原始模板。

如果虚拟机是从模板或其他虚拟机克隆的，则 `模板` 链接到空白模板，而 `original_template` 则用于跟踪历史记录。

否则，模板和 `original_template` 相同。

7.261.23. statistics

从此虚拟机收集的统计信息。

请注意，只有在虚拟机上安装 [Red Hat Virtualization 客户机代理](#) 时才提供一些统计，特别是 `memory.buffered` 和 `memory.cached`。

7.262. SNAPSHOTSTATUS ENUM

表示快照的当前状态。

表 7.342. 值概述

名称	概述
<code>in_preview</code>	快照正在预览。
<code>Locked</code>	快照已被锁定。
确定	快照是确定的。

7.262.1. Locked

快照已被锁定。

快照的创建、删除、恢复或预览过程中将锁定。

7.263. SNAPSHOTTYPE ENUM

表示快照的类型。

表 7.343. 值概述

名称	概述
Active	引用虚拟机的当前配置。
预览	如果 正在预览 某些快照，活动快照将变为预览。
Normal	用户创建的快照。
stateless	为无状态虚拟机内部创建的快照。

7.263.1. 预览

如果 **正在预览** 某些快照，活动快照将变为预览。

换句话说，这是预览前的活动快照。

7.263.2. 无状态

为无状态虚拟机内部创建的快照。

此快照在虚拟机启动时创建，并在虚拟机关闭时恢复。

7.264. SPECIALOBJECTS STRUCT

此类型包含对特殊对象的引用，如空白模板和标签层次结构的根。

表 7.344. 链接摘要

名称	类型	概述
blank_template	Template	对空白模板的引用。
root_tag	Tag	对标签层次结构的根目录的引用。

7.265. SPM STRUCT

表 7.345. 属性概述

名称	类型	概述
priority	整数	
status	SpmStatus	

7.266. SPMSTATUS ENUM

表 7.346. 值概述

名称	概述
contending	
none	
spm	

7.267. SSH STRUCT

表 7.347. 属性概述

名称	类型	概述
authentication_method	SshAuthentication Method	
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
fingerprint	字符串	主机 SSH 公钥的指纹。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
port	整数	
public_key	字符串	使用 RFC4253 中定义的 SSH 公钥格式的 SSH 公钥。
user	用户	

7.267.1. 指纹

主机 SSH 公钥的指纹.此字段自 4.4.5 起已弃用, 并将在以后的中删除。

请改为使用公钥。

7.267.2. public_key

使用 RFC4253 中定义的 SSH 公钥格式的 SSH 公钥。 <https://tools.ietf.org/html/rfc4253>

7.268. SSHAUTHENTICATIONMETHOD ENUM

表 7.348. 值概述

名称	概述
password	
publickey	

7.269. SHHPUBLICKEY STRUCT

表 7.349. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
content	字符串	包含保存的 SSH 密钥。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.350. 链接摘要

名称	类型	概述
user	用户	

7.270. SSO STRUCT

表 7.351. 属性概述

名称	类型	概述
方法	Method[]	

7.271. SSOMETHOD ENUM

表 7.352. 值概述

名称	概述
guest_agent	

7.272. STATISTICSTRUCT

用于所有统计数据的通用类型。

统计包含各种实体的统计值。以下对象包含统计信息：

- *Disk*
- *主机*
- *HostNic*
- *NumaNode*
- *NIC*
- *VM*
- *GlusterBrick*

- 步骤
- **GlusterVolume**

XML 表述示例：

```
<statistics>
  <statistic id="1234" href="/ovirt-engine/api/hosts/1234/nics/1234/statistics/1234">
    <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="1234" href="/ovirt-engine/api/hosts/1234/nics/1234"/>
  </statistic>
  ...
</statistics>
```



注意

这个统计子集合是只读的。

表 7.353. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
kind	StatisticKind	统计测量结果的类型。
name	字符串	纯文本中的人类可读名称。
type	ValueType	后续统计值的数据类型。
unit	StatisticUnit	测量统计值的单位或速率。

名称	类型	概述
值	<code>value[]</code>	包含 datum 的数据集。

表 7.354. 链接摘要

名称	类型	概述
brick	<code>GlusterBrick</code>	
disk	<code>Disk</code>	与包含的 磁盘 资源的关系。
gluster_volume	<code>GlusterVolume</code>	
host	主机	
host_nic	<code>HostNic</code>	对主机 NIC 的引用。
host_numa_node	<code>NumaNode</code>	
nic	<code>NIC</code>	
步骤	步骤	
vm	<code>VM</code>	

7.273. STATISTICKIND ENUM

表 7.355. 值概述

名称	概述
计数	
量表	

7.274. STATISTICUNIT ENUM

表 7.356. 值概述

名称	概述
bits_per_second	

名称	概述
字节	
bytes_per_second	
count_per_second	
none	
%	
秒	

7.275. STEP STRUCT

表示步骤，这是作业执行的一部分。步骤用于描述和跟踪作为更广泛的序列一部分的特定执行单元。某些步骤支持报告其进度。

表 7.357. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
end_time	Date	步骤的结束时间。
external	布尔值	指明该步骤是否源自外部系统。
external_type	ExternalSystemType	步骤中引用的外部系统。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
数字	整数	当前层次结构级别中步骤的顺序。
progress	整数	步骤进度（如已报告）以百分比表示。
start_time	Date	步骤的开始时间。

名称	类型	概述
status	StepStatus	步骤的状态。
type	StepEnum	步骤的类型。

7.275.1. external

指明该步骤是否源自外部系统。外部步骤由步骤的创建者从外部进行管理。

表 7.358. 链接摘要

名称	类型	概述
execution_host	主机	用于步骤执行的主机（可选）。
job	Job	引用当前步骤层次结构顶部的作业。
parent_step	步骤	引用层次结构中当前步骤的父步骤。
statistics	Statistic[]	

7.276. STEPENUM ENUM

键入代表步骤类型。

表 7.359. 值概述

名称	概述
执行	执行步骤类型。
挪威	强制步骤类型。
rebalancing_volume	重新平衡卷 步骤类型。
removing_bricks	删除 brick 步骤类型。
Unknown	未知步骤类型。
validating	验证步骤类型。

7.276.1. 执行

执行步骤类型。用于跟踪作业的主要执行块。它通常是多个子步骤的父步骤，用于描述执行步骤的部分内容。

7.276.2. 挪威

强制步骤类型。描述完成该作业所需的执行后步骤。

7.276.3. rebalancing_volume

重新平衡卷步骤类型。描述作为 Gluster 流一部分的步骤类型。

7.276.4. removing_bricks

删除 brick 步骤类型。描述作为 Gluster 流一部分的步骤类型。

7.276.5. Unknown

未知步骤类型。描述其原始卷未知的步骤类型。

7.276.6. 验证

验证步骤类型。用于验证参数的正确性和参数执行前的有效性。

7.277. STEPSTATUS ENUM

表示步骤的状态。

表 7.360. 值概述

名称	概述
aborted	中止的步骤状态。
Failed	失败的步骤状态。

名称	概述
finished	完成的步骤状态。
Started	已启动步骤状态。
Unknown	未知步骤状态。

7.277.1. 中止

中止的步骤状态。此状态适用于强制中止的外部步骤。

7.277.2. 完成

完成的步骤状态。此状态描述了完成的步骤执行。

7.277.3. Started

已启动步骤状态。此状态代表当前正在执行的一个步骤。

7.277.4. Unknown

未知步骤状态。此状态表示其解析未知的步骤，即系统意外重启之前执行的步骤。

7.278. STORAGECONNECTION STRUCT

表示存储服务器连接。

XML 表述示例：

```
<storage_connection id="123">
  <address>my nfs.example.com</address>
  <type>nfs</type>
  <path>/exports/mydata</path>
</storage_connection>
```

表 7.361. 属性概述

名称	类型	概述
地址	字符串	存储服务器连接的地址。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
mount_options	字符串	NFS 存储服务器连接的挂载选项。
name	字符串	纯文本中的人类可读名称。
nfs_retrans	整数	NFS 存储服务器连接的 NFS 重新传输值。
nfs_timeo	整数	NFS 存储服务器连接的 NFS 时间值。
nfs_version	NfsVersion	NFS 存储服务器连接的 NFS 版本。
password	字符串	iSCSI 存储服务器连接的密码。
路径	字符串	NFS 存储服务器连接的路径。
port	整数	iSCSI 存储服务器连接的端口。
portal	字符串	iSCSI 存储服务器连接的门户。
目标	字符串	iSCSI 存储服务器连接的目标。
type	StorageType	存储服务器连接的类型。
username	字符串	iSCSI 存储服务器连接的用户名。
vfs_type	字符串	NFS 存储服务器连接的 VFS 类型。

表 7.362. 链接摘要

名称	类型	概述
gluster_volume	GlusterVolume	链接供该存储域使用的 gluster 卷。
host	主机	

7.279. STORAGECONNECTIONEXTENSION STRUCT

表 7.363. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
password	字符串	
目标	字符串	
username	字符串	

表 7.364. 链接摘要

名称	类型	概述
host	主机	

7.280. STORAGEDOMAIN STRUCT

存储域。

NFS 存储域的 XML 表示，其标识符为 123：

```
<storage_domain href="/ovirt-engine/api/storagedomains/123" id="123">
  <name>mydata</name>
  <description>My data</description>
  <available>38654705664</available>
  <committed>1073741824</committed>
  <critical_space_action_blocker>5</critical_space_action_blocker>
  <external_status>ok</external_status>
  <master>true</master>
  <storage>
    <address>my nfs.example.com</address>
    <nfs_version>v3</nfs_version>
    <path>/exports/mydata</path>
    <type>nfs</type>
  </storage>
  <storage_format>v3</storage_format>
  <type>data</type>
```

```

<used>13958643712</used>
<warning_low_space_indicator>10</warning_low_space_indicator>
<wipe_after_delete>false</wipe_after_delete>
<data_centers>
  <data_center href="/ovirt-engine/api/datacenters/456" id="456"/>
</data_centers>
</storage_domain>

```

表 7.365. 属性概述

名称	类型	概述
可用	整数	
backup	布尔值	此属性指示数据存储域是否用作备份域。
block_size	整数	指定存储域的块大小，以字节为单位。
comment	字符串	包含有关此对象的注释的免费文本。
committed	整数	
critical_space_action_blocker	整数	
description	字符串	以纯文本格式人类可读的描述。
discard_after_delete	布尔值	指明在删除块存储域之前是否要丢弃块存储域上的磁盘块。
external_status	ExternalStatus	
id	字符串	唯一标识符。
import	布尔值	
master	布尔值	
name	字符串	纯文本中的人类可读名称。
status	StorageDomainStatus	
storage	HostStorage	
storage_format	StorageFormat	
supports_discard	布尔值	指明块存储域是否支持丢弃操作。

名称	类型	概述
supports_discard zeroes_data	布尔值	指明块存储域是否支持丢弃零数据的属性。
type	StorageDomainType	
Used	整数	
warning_low_space_indicator	整数	
wipe_after_delete	布尔值	作为此 存储域 上 磁盘 的 wipe_after_delete 的默认值。

7.280.1. backup

此属性指示数据存储域是否用作备份域。如果将域设置为 **backup**，它将用于存储虚拟机和模板用于灾难恢复目的，方式与使用导出存储域相同。此属性仅可用于数据存储域，不适用于 ISO 域或导出存储域。用户可以在创建数据存储域或导入数据存储域时使用此功能。

7.280.2. block_size

指定存储域的块大小，以字节为单位。可以省略，并且在那种情况下，将默认为 512 字节。并非所有存储域都支持所有可能的大小。

7.280.3. discard_after_delete

指明在删除块存储域之前是否要丢弃块存储域上的磁盘块。

如果为 **true**，且此存储域中的磁盘启用了 **wipe_after_delete** 值，则当磁盘被删除时：

1. 首先被擦除。
2. 然后丢弃其块。
3. 最后它将被删除。

请注意：

- 对于非块存储类型，删除后丢弃始终为 `false`。
- 只有存储域支持丢弃时，才可在删除后丢弃。

7.280.4. `supports_discard`

指明块存储域是否支持丢弃操作。仅当从支持丢弃构建的所有逻辑单元都是逻辑单元时，存储域才支持丢弃；也就是说，如果每个逻辑单元的 `discard_max_size` 值大于 0。这是此存储域中虚拟磁盘启用其 `pass_discard` 属性的先决条件之一。

7.280.5. `supports_discard_zeroes_data`

指明块存储域是否支持丢弃零数据的属性。存储域仅支持从其构建的所有逻辑单元都支持丢弃零数据的属性；也就是说，如果每个逻辑单元的 `discard_zeroes_data` 值为 `true`。



重要

自系统版本 4.2.1 起，已删除对此属性的支持，因为 `sysfs` 文件 `discard_zeroes_data` 已在内核中弃用。它保留下来以便向后兼容，但值始终为 `false`。

7.280.6. `wipe_after_delete`

作为此存储域上磁盘的 `wipe_after_delete` 的默认值。

也就是说，默认情况下，新创建的磁盘将从其存储域中获得其 `wipe_after_delete` 值。请注意，配置值 `SANWipeAfterDelete` 作为块存储域的 `wipe_after_delete` 值的默认值。

表 7.366. 链接摘要

名称	类型	概述
<code>data_center</code>	<code>DataCenter</code>	存储域所连接的数据中心的链接。
<code>data_centers</code>	<code>DataCenter[]</code>	到存储域所连接的数据中心的一组链接。

名称	类型	概述
disk_profiles	DiskProfile[]	
disk_snapshots	DiskSnapshot[]	
磁盘	disk[]	
文件	file[]	
host	主机	主机仅在创建时相关。
images	image[]	
权限	permission[]	
storage_connections	StorageConnection[]	
templates	template[]	
vms	Vm[]	

7.280.7. data_center

存储域所连接的数据中心的链接。这仅用于向后兼容，因为存储域可以连接到多个数据中心（如果是 ISO 域）。改为使用 `dataCenters` 元素。

7.281. STORAGEDOMAINLEASE STRUCT

表示驻留于存储域上的租用。

租期是一个位于存储域上特殊卷的 `Sanlock` 资源，这个 `Sanlock` 资源用于提供存储基础锁定。

表 7.367. 链接摘要

名称	类型	概述
storage_domain	StorageDomain	引用锁定所在的存储域。

7.282. STORAGEDOMAINSTATUS ENUM

表 7.368. 值概述

名称	概述
激活	
Active	
分离	
Inactive	
Locked	
维护	
混合	
preparing_for_maintenance	
未附加	
Unknown	

7.283. STORAGEDOMAINTYPE ENUM

表示由存储域管理的数据种类。

表 7.369. 值概述

名称	概述
data	数据域用于存储系统中虚拟机和模板的磁盘和快照。
export	导出域是用于在数据中心和红帽虚拟化环境之间复制和移动虚拟机和模板的临时存储库。
image	映像域存储可以从外部系统导入的映像。
iso	ISO 域存储用于为虚拟机安装和启动操作系统及应用的 ISO 文件（或逻辑 CD）。
managed_block_storage	在块存储设备上创建托管块存储域。
卷	卷域存储逻辑卷，它们可以用作虚拟机的磁盘。

7.283.1. data

数据域用于存储系统中虚拟机和模板的磁盘和快照。此外，磁盘快照也存储在数据域中。数据域无法在数据中心间共享。

7.283.2. export

导出域是用于在数据中心和红帽虚拟化环境之间复制和移动虚拟机和模板的临时存储库。导出域也可用于备份虚拟机。导出域可以在数据中心之间移动，但一次只能在一个数据中心内处于活动状态。

7.283.3. image

映像域存储可以从外部系统导入的映像。例如，来自 OpenStack Glance 镜像存储库的镜像。

7.283.4. iso

ISO 域存储用于为虚拟机安装和启动操作系统及应用的 ISO 文件（或逻辑 CD）。ISO 域消除了数据中心对物理介质的需求。ISO 域可以在不同的数据中心之间共享。

7.283.5. managed_block_storage

在块存储设备上创建托管块存储域。这些域由 Cinder 访问和管理。

7.283.6. 卷

卷域存储逻辑卷，它们可以用作虚拟机的磁盘。例如，来自 OpenStack Cinder 块存储服务的卷。

7.284. STORAGEFORMAT ENUM

类型表示存储域的格式。

表 7.370. 值概述

名称	概述
v1	存储域格式的版本 1 适用于 NFS、iSCSI 和 FC 存储域。

名称	概述
v2	存储域格式的版本 2 适用于 iSCSI 和 FC 存储域。
v3	存储域格式的版本 3 适用于 NFS、POSIX、iSCSI 和 FC 存储域。
v4	存储域格式的版本 4。
v5	存储域格式的版本 5 适用于 NFS、POSIX 和 Gluster 存储域。

7.284.1. v1

存储域格式的版本 1 适用于 NFS、iSCSI 和 FC 存储域。

每个存储域包含描述其自身结构的元数据，以及用于支持虚拟机磁盘映像的所有物理卷名称。主域还包含存储池中所有域和物理卷名称的元数据。此元数据的总大小限制为 2 KiB，限制池中的存储域数量。模板和虚拟机基础映像是只读的。

7.284.2. v2

存储域格式的版本 2 适用于 iSCSI 和 FC 存储域。

所有存储域和池元数据都存储为逻辑卷标签，而不是写入逻辑卷。有关虚拟机磁盘卷的元数据仍然存储在域中的逻辑卷中。元数据中不再包含物理卷名称。模板和虚拟机基础映像是只读的。

7.284.3. v3

存储域格式的版本 3 适用于 NFS、POSIX、iSCSI 和 FC 存储域。

所有存储域和池元数据都存储为逻辑卷标签，而不是写入逻辑卷。有关虚拟机磁盘卷的元数据仍然存储在域中的逻辑卷中。虚拟机和模板基础映像不再只读。此更改启用实时快照、实时迁移和快照克隆。添加了对 Unicode 元数据的支持，用于非英语卷名称。

7.284.4. v5

存储域格式的版本 5 适用于 NFS、POSIX 和 Gluster 存储域。

添加了对 4096 字节块大小和变量符号链接的支持。

7.285. STORAGETYPE ENUM

键入代表存储域类型。

表 7.371. 值概述

名称	概述
cinder	Cinder 存储域.
fcp	Fibre-Channel 存储域.
Glance	Glance 存储域.
glusterfs	Gluster-FS 存储域.
iscsi	iSCSI 存储域.
localfs	本地存储上的存储域.
managed_block_storage	托管块存储域.
nfs	NFS 存储域.
posixfs	POSIX-FS 存储域.

7.285.1. cinder

Cinder 存储域.有关 **Cinder** 的更多详细信息, 请转至 [Cinder](#)。

7.285.2. Glance

Glance 存储域.有关 **Glance** 的更多详细信息, 请转至 [Glance](#)。

7.285.3. glusterfs

Gluster-FS 存储域.如需关于 **Gluster** 的更多详细信息, 请参阅 [Gluster](#)。

7.285.4. managed_block_storage

托管块存储域。使用 `cinderlib` 管理的存储域。有关支持的存储驱动程序，请参阅[可用驱动程序](#)。

7.286. SWITCHTYPE ENUM

描述 Manager 支持的所有交换机类型。

表 7.372. 值概述

名称	概述
legacy	原生交换机类型。
OVS	Open vSwitch 类型。

7.287. SYSTEMOPTIONSTRUCT

键入，表示系统的配置选项。

表 7.373. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
值	SystemOptionValue[]	各种系统版本的 选项值。

7.288. SYSTEMOPTIONVALUE STRUCT

键入 表示配置选项的一对值和版本。

表 7.374. 属性概述

名称	类型	概述
value	字符串	特定版本的配置选项值。
version	字符串	配置选项的版本。

7.289. TAG STRUCT

表示系统中的标签。

表 7.375. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.376. 链接摘要

名称	类型	概述
group	组	引用分配了此标签的组。
host	主机	引用分配了此标签的主机。
父级	Tag	引用此标签的父标签。
模板	Template	引用分配了此标签的模板。
user	用户	引用分配了此标签的用户。
vm	VM	引用分配了此标签的虚拟机。

7.290. TEMPLATE STRUCT

代表虚拟机模板的类型。模板支持快速实例化具有常见配置和磁盘状态的虚拟机。

表 7.377. 属性概述

名称	类型	概述
auto_pinning_policy	AutoPinningPolicy	指定是否以及如何应用自动 CPU 和 NUMA 配置。
bios	BIOS	参考虚拟机的 BIOS 配置。
comment	字符串	包含有关此对象的注释的免费文本。
console	控制台 (Console)	为此虚拟机配置的控制台。
cpu	cpu	虚拟机 CPU 的配置。
cpu_pinning_policy	CpuPinningPolicy	指定是否以及如何应用 CPU 和 NUMA 配置。
cpu_shares	整数	
creation_time	Date	虚拟机创建日期。
custom_compatibility_version	版本	虚拟机自定义兼容性版本。
custom_cpu_model	字符串	
custom_emulated_machine	字符串	
custom_properties	CustomProperty[]	发送到 VDSM 的属性，以配置各种 hook。
delete_protected	布尔值	如果为 true ，则无法删除虚拟机。
description	字符串	以纯文本格式人类可读的描述。
display	显示	虚拟机显示配置。
domain	域	为此虚拟机配置的域。
high_availability	HighAvailability	虚拟机高可用性配置。
id	字符串	唯一标识符。
初始化	初始化	引用虚拟机的初始化配置。

名称	类型	概述
io	io	用于 IO 线程性能调优。
large_icon	icon	虚拟机的大图标。
lease	StorageDomainLe ase	虚拟机/模板租用所在存储域的引用。
内存	整数	虚拟机的内存，以字节为单位。
memory_policy	MemoryPolicy	参考虚拟机的内存管理配置。
迁移	MigrationOptions	有关将正在运行的虚拟机迁移到另一台主机的配置。
migration_down time	整数	虚拟机在实时迁移到 ms 中另一主机期间可以不响应的最大时间。
multi_queues_e nabled	布尔值	如果为 true ，则 每个虚拟接口都将获得最佳队列数，具体取决于可用的虚拟 Cpus。
name	字符串	纯文本中的人类可读名称。
Origin	字符串	此虚拟机的起源。
OS	OperatingSystem	虚拟机上安装的操作系统类型。
placement_poli cy	VmPlacementPolic y	虚拟机放置策略的配置。
rng_device	RngDevice	此虚拟机的随机数字生成器设备配置。
serial_number	SerialNumber	集群中虚拟机的序列号。
small_icon	icon	虚拟机的小图标。
soundcard_ena bled	布尔值	如果为 true ，则会在虚拟机中添加声卡。
sso	SSO	此虚拟机被配置至单点登录配置的引用。
start_paused	布尔值	如果为 true ，则虚拟机将在启动后的初始处于"暂停"状态。
stateless	布尔值	如果为 true ，则虚拟机为无状态 - 它的状态（磁盘）会在关机后回滚。
status	TemplateStatus	模板的状态。

名称	类型	概述
storage_error_resume_behavior	VmStorageErrorResumeBehaviour	确定存储错误后如何恢复虚拟机。
time_zone	timezone	虚拟机的时区由 oVirt 设置。
tpm_enabled	布尔值	如果为 true ，则会在虚拟机中添加 TPM 设备。
tunnel_migration	布尔值	如果为 true ，则网络数据传输将在虚拟机实时迁移期间加密。
type	VmType	确定虚拟机是为桌面还是服务器优化。
usb	usb	配置此虚拟机的 USB 设备（count、类型）。
version	TemplateVersion	指明这是基本版本还是另一个模板的子版本。
virtio_scsi	VirtioScsi	参考 VirtIO SCSI 配置。
virtio_scsi_multi_queues	整数	这个字段的 Virtio-SCSI controller 队列数量需要 virtioScsiMultiQueuesEnabled。如需更多信息，请参阅 virtioScsiMultiQueuesEnabled
virtio_scsi_multi_queues_enabled	布尔值	如果为 true ，Virtio-SCSI 设备会根据可用的虚拟 Cpus 和磁盘或根据指定的 virtioScsiMultiQueues 来获得多个队列。
vm	VM	与此模板关联的虚拟机配置。

7.290.1. auto_pinning_policy

指定是否以及如何应用自动 CPU 和 NUMA 配置。



重要

由于引擎的 4.5 版已弃用，因此仅保留以向后兼容。以后可能会删除它。请使用 CpuPinningPolicy。

7.290.2. cpu

虚拟机 CPU 的配置。

可以在不重新启动虚拟机的情况下更新套接字配置。内核和线程需要重启。

例如：要将套接字数量立即更改为 4，并在重启后将内核和线程数量改为 2，发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```

通过请求正文：

```
<vm>
  <cpu>
    <topology>
      <sockets>4</sockets>
      <cores>2</cores>
      <threads>2</threads>
    </topology>
  </cpu>
</vm>
```

7.290.3. cpu_pinning_policy

指定是否以及如何应用 CPU 和 NUMA 配置。如果没有指定 CPU 固定字符串之前的行为，则会决定 CpuPinningPolicy 到 None 或 Manual。

7.290.4. custom_compatibility_version

虚拟机自定义兼容性版本。

使虚拟机能够根据自己的兼容性版本进行自定义。如果设置了 custom_compatibility_version，它将覆盖该特定虚拟机的集群的兼容性版本。

虚拟机的兼容性版本受位于的数据中心的限制，并根据虚拟机计划在其上运行的主机进行检查。

7.290.5. high_availability

虚拟机高可用性配置。如果设置，虚拟机将在意外停机时自动重新启动。

7.290.6. 初始化

引用虚拟机的初始化配置。



注意

自 Red Hat Virtualization 4.1.8 起，可以通过发送空标签来清除此属性。

例如，清除初始化属性会发送类似如下的请求：

```
PUT /ovirt-engine/api/vms/123
```

这样的请求正文：

```
<vm>
  <initialization/>
</vm>
```

对此类请求的响应和带有标头 `All-Content: true` 的请求仍将包含此属性。

7.290.7. large_icon

虚拟机的大图标。用户设置或引用根据操作系统设置的镜像集。

7.290.8. lease

虚拟机/模板租用所在存储域的引用。

使用租用运行的虚拟机需要检查运行时不会由另一主机获取租期，从而防止此虚拟机的另一个实例在另一主机上运行。这可防止高可用性虚拟机中的脑裂。模板也可为租用定义存储域，以便从该模板创建虚拟机，预先配置此存储域作为租用位置。

7.290.9. 内存

虚拟机的内存，以字节为单位。

例如，要更新虚拟机使其包含 1 Gibibyte(GiB)内存，请发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```

使用以下请求正文：

```
<vm>
  <memory>1073741824</memory>
</vm>
```

Red Hat Virtualization 3.6 支持内存热插拔。在虚拟机处于 **up** 状态时，您可以使用上述示例来增加内存。增量的大小必须可以被分成 `HotPlugMemoryBlockSizeMb` 配置值（默认为 256 MiB）。如果此值无法分隔内存大小递增，则内存大小更改仅存储在下一次运行配置中。每个成功的内存热插拔操作都会创建一个或两个新内存设备。

从 Red Hat Virtualization 4.2 开始支持内存热插拔。内存热拔头只能在虚拟机处于启动状态时执行。只有之前热插拔内存设备才能通过热拔出操作删除。请求的内存减少向下舍入，以匹配之前热插拔内存设备的组合大小。请求的内存值存储在下次运行配置中，而不舍入。



注意

这个示例中的内存会被转换为字节：
1 GiB = 2^{30} 字节 = 1073741824 字节。



注意

红帽虚拟化管理器内部将值向下舍入到整个 MiB (1MiB = 2^{20} 字节)

7.290.10. migration_downtime

虚拟机在实时迁移到 ms 中另一主机期间可以不响应的最大时间。

为虚拟机明确设置，或者使用 `engine-config -s DefaultMaximumMigrationDowntime=[value]`

7.290.11. Origin

此虚拟机的起源。

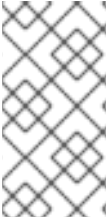
可能的值：

- *ovirt*
- *rhev*
- *vmware*
- *Xen*
- *external*
- *hosted_engine*
- *managed_hosted_engine*
- *kvm*
- *physical_machine*
- *hyperv*

7.290.12. *placement_policy*

虚拟机放置策略的配置。

此配置可以更新为将虚拟机固定到一个或多个主机。

**注意**

固定到多个主机的虚拟机无法实时迁移，但在主机出现故障时，任何配置为高可用性的虚拟机都会自动在虚拟机要固定到的其他主机上重新启动。

例如，要将虚拟机固定到两个主机，请发送以下请求：

```
PUT /api/vms/123
```

这样的请求正文：

```
<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>
```

7.290.13. small_icon

虚拟机的小图标。用户设置或引用根据操作系统设置的镜像集。

7.290.14. sso

此虚拟机被配置至单点登录配置的引用。打开控制台时，可以在虚拟机的操作系统中自动签署该用户。

7.290.15. tpm_enabled

如果为 **true**，则会在虚拟机中添加 TPM 设备。默认值为 **false**。只有在设置了 "All-Content=true" 标头时，此属性才能在获取时可见。

表 7.378. 链接摘要

名称	类型	概述
cdroms	Cdrom[]	引用附加到模板的 CD-ROM 设备。
cluster	Cluster	虚拟机所属群集的引用。
cpu_profile	CpuProfile	引用此虚拟机使用的 CPU 配置集。
disk_attachments	DiskAttachment[]	引用模板所连接的磁盘。
graphics_consoles	GraphicsConsole[]	引用附加到模板的图形控制台。
mediated_devices	VmMediatedDevice[]	介质设备配置。
nics	Nic[]	引用附加到模板的网络接口。
权限	permission[]	引用附加到模板的用户权限。
quota	Quota	引用为此虚拟机设置的配额配置。
storage_domain	StorageDomain	虚拟机所属存储域的引用。
tags	Tag[]	引用附加到模板的标签。
watchdogs	Watchdog[]	引用模板所连接的 watchdog 设备。

7.291. TEMPLATESTATUS ENUM

键入代表虚拟机模板状态的状态。

表 7.379. 值概述

名称	概述
illegal	此状态表示模板中至少有一个磁盘是非法的。
Locked	此状态表示一些操作会阻止通过模板执行其他操作。
确定	此状态表示模板有效且可供使用。

7.292. TEMPLATEVERSION STRUCT

键入表示虚拟机模板版本。

表 7.380. 属性概述

名称	类型	概述
version_name	字符串	此版本的名称。
version_number	整数	此版本在模板的版本层次结构中的索引。

7.292.1. version_number

此版本在模板的版本层次结构中的索引。索引 1 代表模板的原始版本， 它也称为基础版本。

表 7.381. 链接摘要

名称	类型	概述
base_template	Template	引用此版本关联的模板。

7.293. TICKETSTRUCT

键入代表允许虚拟机访问的票据。

表 7.382. 属性概述

名称	类型	概述
到期	整数	票据的生存时间（以秒为单位）。
value	字符串	虚拟机访问票据。

7.294. TIMEZONE STRUCT

时区表示。

表 7.383. 属性概述

名称	类型	概述
name	字符串	时区的名称。
utc_offset	字符串	UTC 偏移。

7.294.1. utc_offset

UTC 偏移。

与 [UTC 偏移](#)。

7.295. TRANSPARENTHUGEPAGES STRUCT

输入代表透明大内存页(THP)支持的内容。

表 7.384. 属性概述

名称	类型	概述
enabled	布尔值	启用 THP 支持。

7.296. TRANSPORTTYPE ENUM

用于访问 Gluster 卷的协议。

表 7.385. 值概述

名称	概述
rdma	远程直接内存访问。
tcp	TCP :

7.297. UNMANAGEDNETWORK STRUCT

表 7.386. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.387. 链接摘要

名称	类型	概述
host	主机	
host_nic	HostNic	

7.298. USB STRUCT

配置虚拟机的 USB 设备。

表 7.388. 属性概述

名称	类型	概述
enabled	布尔值	确定是否应包含 USB 设备。
type	UsbType	USB 类型，当前仅支持 native 。

7.299. USBTYPE ENUM

USB 设备重定向的类型。

表 7.389. 值概述

名称	概述
legacy	传统 USB 重定向。
native	本地 USB 重定向。

7.299.1. legacy

传统 USB 重定向。

从引擎版本 3.6 起，此 USB 类型已弃用，并在版本 4.1 中完全删除。它只会保留，以避免现有脚本中的语法错误。如果使用它，系统将自动替换为 `native`。

7.299.2. native

本地 USB 重定向。

本地 USB 重定向允许 Linux 和 Windows 虚拟机的 KVM/SPICE USB 重定向。虚拟机（客户机）计算机不需要针对本地 USB 的客户机安装代理或驱动程序。在 Linux 客户端上，USB 重定向所需的所有软件包都由 `virt-viewer` 软件包提供。在 Windows 客户端上，还必须安装 `usbdk` 软件包。

7.300. 用户 结构

代表系统中的用户。

表 7.390. 属性概述

名称	类型	概述
<code>comment</code>	字符串	包含有关此对象的注释的免费文本。
<code>Department</code>	字符串	
<code>description</code>	字符串	以纯文本格式人类可读的描述。
<code>domain_entry_id</code>	字符串	
<code>email</code>	字符串	
<code>id</code>	字符串	唯一标识符。
<code>last_name</code>	字符串	
<code>logged_in</code>	布尔值	
<code>name</code>	字符串	纯文本中的人类可读名称。

名称	类型	概述
namespace	字符串	用户所在的命名空间。
password	字符串	
principal	字符串	与 <code>user_name</code> 类似。
user_name	字符串	用户的用户名。
user_options	attribute[]	用户选项允许您保存用于为每个用户自定义设置的键/值属性。

7.300.1. namespace

用户所在的命名空间。在使用在 LDAP 服务器中存储用户的授权提供商时，此属性等同于 LDAP 服务器的命名上下文。如需更多信息，请参阅 <https://github.com/oVirt/ovirt-engine-extension-aaa-ldap>。使用在数据库中存储用户的内置授权提供程序时，此属性将被忽略。如需更多信息，请参阅 <https://github.com/oVirt/ovirt-engine-extension-aaa-jdbc>。

7.300.2. principal

与 `user_name` 类似。格式取决于 LDAP 提供程序。对于大多数 LDAP 提供程序，这是 uid LDAP 属性的值。对于 Active Directory，这是用户主体名称(UPN)。

7.300.3. user_name

用户的用户名。格式取决于授权提供程序类型。在大多数 LDAP 提供程序中，这是 uid LDAP 属性的值。在 Active Directory 中，它是用户主体名称(UPN)。UPN 或 uid 后必须跟着授权提供程序名称。例如，如果是 LDAP 的 uid 属性，则为：`my user@myextension-authz`。对于使用 UPN 的 Active Directory，它为：`myuser@mysubdomain.mydomain.com@myextension-authz`。此属性是添加新用户时必需的参数。

7.300.4. user_options

用户选项允许您保存用于为每个用户自定义设置的键/值属性。请注意，由于版本 4.4.5 已弃用，并保留此属性仅用于向后兼容。以后会将其移除。请改为使用选项端点。

表 7.391. 链接摘要

名称	类型	概述
domain	域	

名称	类型	概述
groups	group[]	
选项	UserOption[]	
权限	permission[]	
角色	role[]	到用户资源的角色子集合的链接。
ssh_public_key s	SshPublicKey[]	
tags	Tag[]	到用户资源的标签子集合的链接。

7.301. USEROPTIONSTRUCT

用户选项允许您保存用于为每个用户自定义设置的键/值属性。

表 7.392. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
content	字符串	JSON 内容编码为字符串。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

7.301.1. 内容

JSON 内容编码为字符串。支持任何有效的 JSON。

表 7.393. 链接摘要

名称	类型	概述
user	用户	

7.302. VALUE STRUCT

表 7.394. 属性概述

名称	类型	概述
datum	十进制	
详情	字符串	

7.303. VALUETYPE ENUM

表 7.395. 值概述

名称	概述
十进制	
整数	
字符串	

7.304. VCPUPIN STRUCT

表 7.396. 属性概述

名称	类型	概述
cpu_set	字符串	
vcpu	整数	

7.305. VENDOR STRUCT

表 7.397. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

7.306. 版本 结构

表 7.398. 属性概述

名称	类型	概述
build	整数	
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
full_version	字符串	
id	字符串	唯一标识符。
major	整数	
次	整数	
name	字符串	纯文本中的人类可读名称。
修订	整数	

7.307. VGPUPLACEMENT ENUM

vGPU 放置策略。

它可将 vGPU 放置到第一个可用的物理卡上，或者将它们分布到多个物理卡上。

表 7.399. 值概述

名称	概述
consolidated	使用整合放置。
separated	使用分隔放置。

7.307.1. 合并

使用整合放置.每个 vGPU 放置在第一个带有可用空间的物理卡上。

这是默认放置，利用物理卡上的所有可用空间。

7.307.2. separated

使用分隔放置。每个 vGPU 都放置在单独的物理卡上（如果可能）。

这对提高 vGPU 性能非常有用。

7.308. VIRTIO SCSI STRUCT

键入代表 virtio-SCSI 支持的内容。如果支持，我们将 virtio 驱动程序用于 SCSI 客户机设备。

表 7.400. 属性概述

名称	类型	概述
enabled	布尔值	启用 Virtio SCSI 支持。

7.309. VIRTUALNUMANODE STRUCT

代表虚拟 NUMA 节点。

XML 表述示例：

```
<vm_numa_node href="/ovirt-engine/api/vms/123/numanodes/456" id="456">
  <cpu>
    <cores>
      <core>
        <index>0</index>
      </core>
    </cores>
  </cpu>
  <index>0</index>
  <memory>1024</memory>
  <numa_node_pins>
    <numa_node_pin>
      <index>0</index>
    </numa_node_pin>
  </numa_node_pins>
  <vm href="/ovirt-engine/api/vms/123" id="123" />
</vm_numa_node>
```

表 7.401. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
cpu	cpu	
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
index	整数	
内存	整数	NUMA 节点的内存（以 MB 为单位）。
name	字符串	纯文本中的人类可读名称。
node_distance	字符串	
numa_node_pins	NumaNodePin[]	
numa_tune_mode	NumaTuneMode	如何应用 NUMA 拓扑。

表 7.402. 链接摘要

名称	类型	概述
host	主机	
statistics	Statistic[]	每一主机 NUMA 节点资源公开一个统计子集合，用于主机 NUMA 节点特定统计信息。
vm	VM	

7.309.1. statistics

每一主机 NUMA 节点资源公开一个统计子集合，用于主机 NUMA 节点特定统计信息。

XML 表述示例：

```
<statistics>
  <statistic href="/ovirt-engine/api/hosts/123/numanodes/456/statistics/789" id="789">
```

```

<name>memory.total</name>
<description>Total memory</description>
<kind>gauge</kind>
<type>integer</type>
<unit>bytes</unit>
<values>
  <value>
    <datum>25165824000</datum>
  </value>
</values>
<host_numa_node href="/ovirt-engine/api/hosts/123/numanodes/456" id="456" />
</statistic>
...
</statistics>

```



注意

这个统计子集合是只读的。

下表显示了主机 NUMA 节点的统计类型：

名称	描述
memory.total	NUMA 节点上的内存总量，以字节为单位。
memory.used	NUMA 节点上使用的内存（以字节为单位）。
memory.free	NUMA 节点上空闲的内存（以字节为单位）。
cpu.current.user	用户片段的 CPU 使用量百分比。
cpu.current.system	系统的 CPU 使用量百分比。
cpu.current.idle	空闲 CPU 使用率的百分比。

7.310. VLAN STRUCT

键入代表虚拟 LAN(VLAN)类型。

表 7.403. 属性概述

名称	类型	概述
id	整数	虚拟局域网 ID.

7.311. VM STRUCT

表示虚拟机。

表 7.404. 属性概述

名称	类型	概述
auto_pinning_policy	AutoPinningPolicy	指定是否以及如何应用自动 CPU 和 NUMA 配置。
bios	BIOS	参考虚拟机的 BIOS 配置。
comment	字符串	包含有关此对象的注释的免费文本。
console	控制台 (Console)	为此虚拟机配置的控制台。
cpu	cpu	虚拟机 CPU 的配置。
cpu_pinning_policy	CpuPinningPolicy	指定是否以及如何应用 CPU 和 NUMA 配置。
cpu_shares	整数	
creation_time	Date	虚拟机创建日期。
custom_compatibility_version	版本	虚拟机自定义兼容性版本。
custom_cpu_model	字符串	
custom_emulated_machine	字符串	
custom_properties	CustomProperty[]	发送到 VDSM 的属性，以配置各种 hook。
delete_protected	布尔值	如果为 true ，则无法删除虚拟机。
description	字符串	以纯文本格式人类可读的描述。
display	显示	虚拟机显示配置。
domain	域	为此虚拟机配置的域。

名称	类型	概述
fqdn	字符串	虚拟机的完全限定域名。
guest_operating_system	GuestOperatingSystem	虚拟机上安装什么操作系统。
guest_time_zone	timezone	虚拟机使用什么时区（由虚拟客户机代理返回）。
has_illegal_images	布尔值	指明虚拟机是否有带有 ILLEGAL 状态的磁盘的快照。
high_availability	HighAvailability	虚拟机高可用性配置。
id	字符串	唯一标识符。
初始化	初始化	引用虚拟机的初始化配置。
io	io	用于 IO 线程性能调优。
large_icon	icon	虚拟机的大图标。
lease	StorageDomainLease	虚拟机/模板租用所在存储域的引用。
内存	整数	虚拟机的内存，以字节为单位。
memory_policy	MemoryPolicy	参考虚拟机的内存管理配置。
迁移	MigrationOptions	有关将正在运行的虚拟机迁移到另一台主机的配置。
migration_down_time	整数	虚拟机在实时迁移到 ms 中另一主机期间可以不响应的最大时间。
multi_queues_enabled	布尔值	如果为 true ，则每个虚拟接口都将获得最佳队列数，具体取决于可用的虚拟 Cpus。
name	字符串	纯文本中的人类可读名称。
next_run_configuration_exists	布尔值	虚拟机配置已更改，需要重启虚拟机。
numa_tune_mode	NumaTuneMode	如何应用 NUMA 拓扑。
Origin	字符串	此虚拟机的起源。

名称	类型	概述
OS	OperatingSystem	虚拟机上安装的操作系统类型。
有效负载	payload[]	虚拟机的可选有效负载，用于 ISO 进行配置。
placement_policy	VmPlacementPolicy	虚拟机放置策略的配置。
rng_device	RngDevice	此虚拟机的随机数字生成器设备配置。
run_once	布尔值	如果为 true ，则使用 <code>run</code> 命令启动虚拟机一次，这意味着它的配置可能与存储的配置不同，用于本次运行。
serial_number	SerialNumber	集群中虚拟机的序列号。
small_icon	icon	虚拟机的小图标。
soundcard_enabled	布尔值	如果为 true ，则会在虚拟机中添加声卡。
ssso	SSO	此虚拟机被配置至单点登录配置的引用。
start_paused	布尔值	如果为 true ，则虚拟机将在启动后的初始处于"暂停"状态。
start_time	Date	虚拟机启动的日期。
stateless	布尔值	如果为 true ，则虚拟机为无状态 - 它的状态（磁盘）会在关机后回滚。
status	VmStatus	虚拟机的当前状态。
status_detail	字符串	当前状态人类可读的详细信息。
stop_reason	字符串	虚拟机被停止的原因。
stop_time	Date	虚拟机停止的日期。
storage_error_resume_behaviour	VmStorageErrorResumeBehaviour	确定存储错误后如何恢复虚拟机。
time_zone	timezone	虚拟机的时区由 oVirt 设置。
tpm_enabled	布尔值	如果为 true ，则会在虚拟机中添加 TPM 设备。
tunnel_migration	布尔值	如果为 true ，则网络数据传输将在虚拟机实时迁移期间加密。

名称	类型	概述
<code>type</code>	<code>VmType</code>	确定虚拟机是为桌面还是服务器优化。
<code>usb</code>	<code>usb</code>	配置此虚拟机的 USB 设备 (count、类型)。
<code>use_latest_template_version</code>	布尔值	如果为 true ，则虚拟机在启动时将其重新配置为模板的最新版本。
<code>virtio_scsi</code>	<code>VirtioScsi</code>	参考 VirtIO SCSI 配置。
<code>virtio_scsi_multi_queues</code>	整数	这个字段的 Virtio-SCSI controller 队列数量需要 <code>virtioScsiMultiQueuesEnabled</code> 。如需更多信息，请参阅 <code>virtioScsiMultiQueuesEnabled</code>
<code>virtio_scsi_multi_queues_enabled</code>	布尔值	如果为 true ，Virtio-SCSI 设备会根据可用的虚拟 Cpus 和磁盘或根据指定的 <code>virtioScsiMultiQueues</code> 来获得多个队列。

7.311.1. `auto_pinning_policy`

指定是否以及如何应用自动 CPU 和 NUMA 配置。



重要

由于引擎的 4.5 版已弃用，因此仅保留以向后兼容。以后可能会删除它。请使用 `CpuPinningPolicy`。

7.311.2. `cpu`

虚拟机 CPU 的配置。

可以在不重新启动虚拟机的情况下更新套接字配置。内核和线程需要重启。

例如：要将套接字数量立即更改为 4，并在重启后将内核和线程数量改为 2，发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```

通过请求正文：

```
<vm>
  <cpu>
    <topology>
      <sockets>4</sockets>
      <cores>2</cores>
      <threads>2</threads>
    </topology>
  </cpu>
</vm>
```

7.311.3. cpu_pinning_policy

指定是否以及如何应用 CPU 和 NUMA 配置。如果没有指定 CPU 固定字符串之前的行为，则会决定 CpuPinningPolicy 到 None 或 Manual。

7.311.4. custom_compatibility_version

虚拟机自定义兼容性版本。

使虚拟机能够根据自己的兼容性版本进行自定义。如果设置了 custom_compatibility_version，它将覆盖该特定虚拟机的集群的兼容性版本。

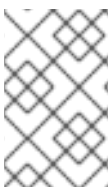
虚拟机的兼容性版本受位于的数据中心的限制，并根据虚拟机计划在其上运行的主机进行检查。

7.311.5. high_availability

虚拟机高可用性配置。如果设置，虚拟机将在意外停机时自动重新启动。

7.311.6. 初始化

引用虚拟机的初始化配置。



注意

自 Red Hat Virtualization 4.1.8 起，可以通过发送空标签来清除此属性。

例如，清除初始化属性会发送类似如下的请求：

```
PUT /ovirt-engine/api/vms/123
```

这样的请求正文：

```
<vm>  
  <initialization/>  
</vm>
```

对此类请求的响应和带有标头 `All-Content: true` 的请求仍将包含此属性。

7.311.7. large_icon

虚拟机的大图标。用户设置或引用根据操作系统设置的镜像集。

7.311.8. lease

虚拟机/模板租用所在存储域的引用。

使用租用运行的虚拟机需要检查运行时不会由另一主机获取租期，从而防止此虚拟机的另一个实例在另一主机上运行。这可防止高可用性虚拟机中的脑裂。模板也可对租用定义存储域，以便从该模板创建虚拟机，预先配置此存储域作为租用位置。

7.311.9. 内存

虚拟机的内存，以字节为单位。

例如，要更新虚拟机使其包含 1 Gibibyte(GiB)内存，请发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```

使用以下请求正文：

```
<vm>
  <memory>1073741824</memory>
</vm>
```

Red Hat Virtualization 3.6 支持内存热插拔。在虚拟机处于 **up** 状态时，您可以使用上述示例来增加内存。增量的大小必须可以被分成 `HotPlugMemoryBlockSizeMb` 配置值（默认为 256 MiB）。如果此值无法分隔内存大小递增，则内存大小更改仅存储在下一次运行配置中。每个成功的内存热插拔操作都会创建一个或两个新内存设备。

从 Red Hat Virtualization 4.2 开始支持内存热插拔。**内存热拔头只能在虚拟机处于启动状态时执行**。只有之前热插拔内存设备才能通过热拔出操作删除。请求的内存减少向下舍入，以匹配之前热插拔内存设备的组合大小。请求的内存值存储在下次运行配置中，而不舍入。



注意

这个示例中的内存会被转换为字节：
 $1 \text{ GiB} = 2^{30} \text{ 字节} = 1073741824 \text{ 字节}$ 。



注意

红帽虚拟化管理器内部将值向下舍入到整个 MiB ($1 \text{ MiB} = 2^{20} \text{ 字节}$)

7.311.10. migration_downtime

虚拟机在实时迁移到 `ms` 中另一主机期间可以不响应的最大时间。

为虚拟机明确设置，或者使用 `engine-config -s DefaultMaximumMigrationDowntime=[value]`

7.311.11. next_run_configuration_exists

虚拟机配置已更改，需要重启虚拟机。更改的配置在处理虚拟机的关机时应用。

7.311.12. numa_tune_mode

如何应用 NUMA 拓扑。弃用了每个 vNUMA 节点的 NUMA 调优。

7.311.13. Origin

此虚拟机的起源。

可能的值：

- *ovirt*
- *rhev*
- *vmware*
- *Xen*
- *external*
- *hosted_engine*
- *managed_hosted_engine*
- *kvm*
- *physical_machine*
- *hyperv*

7.311.14. placement_policy

虚拟机放置策略的配置。

此配置可以更新为将虚拟机固定到一个或多个主机。



注意

固定到多个主机的虚拟机无法实时迁移，但在主机出现故障时，任何配置为高可用性的虚拟机都会自动在虚拟机要固定到的其他主机上重新启动。

例如，要将虚拟机固定到两个主机，请发送以下请求：

```
PUT /api/vms/123
```

这样的请求正文：

```
<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>
```

7.311.15. small_icon

虚拟机的小图标。用户设置或引用根据操作系统设置的镜像集。

7.311.16. sso

此虚拟机被配置至单点登录配置的引用。打开控制台时，可以在虚拟机的操作系统中自动签署该用户。

7.311.17. stop_reason

虚拟机被停止的原因。关闭虚拟机时由用户设置（可选）。

7.311.18. tpm_enabled

如果为 `true`，则会在虚拟机中添加 TPM 设备。默认值为 `false`。只有在设置了 `"All-Content=true"` 标头时，此属性才能在获取时可见。

表 7.405. 链接摘要

名称	类型	概述
<code>affinity_labels</code>	<code>AffinityLabel[]</code>	可选。
应用程序	<code>application[]</code>	在虚拟机上安装的应用程序列表。
<code>cdroms</code>	<code>Cdrom[]</code>	引用挂载到 CDROM 的 ISO。
<code>cluster</code>	<code>Cluster</code>	虚拟机所属群集的引用。
<code>cpu_profile</code>	<code>CpuProfile</code>	引用此虚拟机使用的 CPU 配置集。
<code>disk_attachments</code>	<code>DiskAttachment[]</code>	引用附加到虚拟机的磁盘。
<code>dynamic_cpu</code>	<code>DynamicCpu</code>	虚拟机 CPU 的动态配置。
<code>external_host_provider</code>	<code>ExternalHostProvider</code>	
<code>floppies</code>	<code>Floppy[]</code>	引用挂载到软盘的 ISO。
<code>graphics_consoles</code>	<code>GraphicsConsole[]</code>	为此虚拟机配置的图形控制台列表。
<code>host</code>	主机	引用正在运行的虚拟机的主机。
<code>host_devices</code>	<code>HostDevice[]</code>	参考与此虚拟机关联的设备。
<code>instance_type</code>	<code>InstanceType</code>	虚拟机配置（可选）可通过其中一个实例类型预定义。
<code>katello_errata</code>	<code>KatelloErratum[]</code>	列出分配到虚拟机的所有 Katello 勘误表。
<code>mediated_devices</code>	<code>VmMediatedDevice[]</code>	介质设备配置。

名称	类型	概述
nics	Nic[]	参考虚拟机上的网络接口设备列表。
numa_nodes	NumaNode[]	指此虚拟机使用的 NUMA 节点配置。
original_template	Template	引用用于创建虚拟机的原始模板。
权限	permission[]	为此虚拟机设置权限。
quota	Quota	引用为此虚拟机设置的配额配置。
reported_devices	ReportedDevice[]	
会话	session[]	为此虚拟机打开的用户会话列表。
snapshots	Snapshot[]	指的是从虚拟机获取的所有快照。
statistics	Statistic[]	从此虚拟机收集的统计信息。
storage_domain	StorageDomain	虚拟机所属存储域的引用。
tags	Tag[]	
模板	Template	对虚拟机所基于的模板的引用。
vm_pool	VmPool	虚拟机所属池的引用（可选）
watchdogs	Watchdog[]	引用 Watchdog 配置。

7.311.19. affinity_labels

可选。用于标记子集群。

7.311.20. katello_errata

列出分配到虚拟机的所有 Katello 勘误表。

GET /ovirt-engine/api/vms/123/katelloerrata

您将在 XML 中收到类似以下的响应：

```
<katello_errata>
  <katello_erratum href="/ovirt-engine/api/katelloerrata/456" id="456">
    <name>RHBA-2013:XYZ</name>
    <description>The description of the erratum</description>
    <title>some bug fix update</title>
    <type>bugfix</type>
    <issued>2013-11-20T02:00:00.000+02:00</issued>
    <solution>Few guidelines regarding the solution</solution>
    <summary>Updated packages that fix one bug are now available for XYZ</summary>
    <packages>
      <package>
        <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
      </package>
      ...
    </packages>
  </katello_erratum>
  ...
</katello_errata>
```

7.311.21. original_template

引用用于创建虚拟机的原始模板。

如果虚拟机是从模板或其他虚拟机克隆的，则 `模板` 链接到空白模板，而 `original_template` 则用于跟踪历史记录。

否则，模板和 `original_template` 相同。

7.311.22. statistics

从此虚拟机收集的统计信息。

请注意，只有在虚拟机上安装 [Red Hat Virtualization 客户机代理](#) 时才提供一些统计，特别是 `memory.buffered` 和 `memory.cached`。

7.312. VMAFFINITY ENUM

表 7.406. 值概述

名称	概述
migratable	
pinned	
user_migratable	

7.313. VMBASE STRUCT

表示基本虚拟机配置。这供虚拟机、模板和实例类型使用。

表 7.407. 属性概述

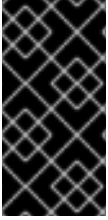
名称	类型	概述
auto_pinning_policy	AutoPinningPolicy	指定是否以及如何应用自动 CPU 和 NUMA 配置。
bios	BIOS	参考虚拟机的 BIOS 配置。
comment	字符串	包含有关此对象的注释的免费文本。
console	控制台 (Console)	为此虚拟机配置的控制台。
cpu	cpu	虚拟机 CPU 的配置。
cpu_pinning_policy	CpuPinningPolicy	指定是否以及如何应用 CPU 和 NUMA 配置。
cpu_shares	整数	
creation_time	Date	虚拟机创建日期。
custom_compatibility_version	版本	虚拟机自定义兼容性版本。
custom_cpu_model	字符串	
custom_emulated_machine	字符串	
custom_properties	CustomProperty[]	发送到 VDSM 的属性，以配置各种 hook。

名称	类型	概述
delete_protected	布尔值	如果为 true ，则无法删除虚拟机。
description	字符串	以纯文本格式人类可读的描述。
display	显示	虚拟机显示配置。
domain	域	为此虚拟机配置的域。
high_availability	HighAvailability	虚拟机高可用性配置。
id	字符串	唯一标识符。
初始化	初始化	引用虚拟机的初始化配置。
io	io	用于 IO 线程性能调优。
large_icon	icon	虚拟机的大图标。
lease	StorageDomainLease	虚拟机/模板租用所在存储域的引用。
内存	整数	虚拟机的内存，以字节为单位。
memory_policy	MemoryPolicy	参考虚拟机的内存管理配置。
迁移	MigrationOptions	有关将正在运行的虚拟机迁移到另一台主机的配置。
migration_downtime	整数	虚拟机在实时迁移到 ms 中另一主机期间可以不响应的最大时间。
multi_queues_enabled	布尔值	如果为 true ，则每个虚拟接口都将获得最佳队列数，具体取决于可用的虚拟 Cpus。
name	字符串	纯文本中的人类可读名称。
Origin	字符串	此虚拟机的起源。
OS	OperatingSystem	虚拟机上安装的操作系统类型。
placement_policy	VmPlacementPolicy	虚拟机放置策略的配置。
rng_device	RngDevice	此虚拟机的随机数字生成器设备配置。

名称	类型	概述
serial_number	SerialNumber	集群中虚拟机的序列号。
small_icon	icon	虚拟机的小图标。
soundcard_enabled	布尔值	如果为 true ，则会在虚拟机中添加声卡。
ssso	SSO	此虚拟机被配置至单点登录配置的引用。
start_paused	布尔值	如果为 true ，则虚拟机将在启动后的初始处于"暂停"状态。
stateless	布尔值	如果为 true ，则虚拟机为无状态 - 它的状态（磁盘）会在关机后回滚。
storage_error_resume_behavior	VmStorageErrorResumeBehaviour	确定存储错误后如何恢复虚拟机。
time_zone	timezone	虚拟机的时区由 oVirt 设置。
tpm_enabled	布尔值	如果为 true ，则会在虚拟机中添加 TPM 设备。
tunnel_migration	布尔值	如果为 true ，则网络数据传输将在虚拟机实时迁移期间加密。
type	VmType	确定虚拟机是为桌面还是服务器优化。
usb	usb	配置此虚拟机的 USB 设备（count、类型）。
virtio_scsi	VirtioScsi	参考 VirtIO SCSI 配置。
virtio_scsi_multi_queues	整数	这个字段的 Virtio-SCSI controller 队列数量需要 virtioScsiMultiQueuesEnabled。如需更多信息，请参阅 virtioScsiMultiQueuesEnabled
virtio_scsi_multi_queues_enabled	布尔值	如果为 true ，Virtio-SCSI 设备会根据可用的虚拟 Cpus 和磁盘或根据指定的 virtioScsiMultiQueues 来获得多个队列。

7.313.1. auto_pinning_policy

指定是否以及如何应用自动 CPU 和 NUMA 配置。

**重要**

由于引擎的 4.5 版已弃用，因此仅保留以向后兼容。以后可能会删除它。请使用 `CpuPinningPolicy`。

7.313.2. cpu

虚拟机 CPU 的配置。

可以在不重新启动虚拟机的情况下更新套接字配置。内核和线程需要重启。

例如：要将套接字数量立即更改为 4，并在重启后将内核和线程数量改为 2，发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```

通过请求正文：

```
<vm>
  <cpu>
    <topology>
      <sockets>4</sockets>
      <cores>2</cores>
      <threads>2</threads>
    </topology>
  </cpu>
</vm>
```

7.313.3. cpu_pinning_policy

指定是否以及如何应用 CPU 和 NUMA 配置。如果没有指定 CPU 固定字符串之前的行为，则会决定 `CpuPinningPolicy` 到 `None` 或 `Manual`。

7.313.4. custom_compatibility_version

虚拟机自定义兼容性版本。

使虚拟机能够根据自己的兼容性版本进行自定义。如果设置了 `custom_compatibility_version`，它将覆盖该特定虚拟机的集群的兼容性版本。

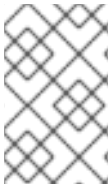
虚拟机的兼容性版本受位于的数据中心的限制，并根据虚拟机计划在其上运行的主机进行检查。

7.313.5. high_availability

虚拟机高可用性配置。如果设置，虚拟机将在意外停机时自动重新启动。

7.313.6. 初始化

引用虚拟机的初始化配置。



注意

自 Red Hat Virtualization 4.1.8 起，可以通过发送空标签来清除此属性。

例如，清除初始化属性会发送类似如下的请求：

```
PUT /ovirt-engine/api/vms/123
```

这样的请求正文：

```
<vm>  
  <initialization/>  
</vm>
```

对此类请求的响应和带有标头 **All-Content: true** 的请求仍将包含此属性。

7.313.7. large_icon

虚拟机的大图标。用户设置或引用根据操作系统设置的镜像集。

7.313.8. lease

虚拟机/模板租用所在存储域的引用。

使用租用运行的虚拟机需要检查运行时不会由另一主机获取租期，从而防止此虚拟机的另一个实例在另一主机上运行。这可防止高可用性虚拟机中的脑裂。模板也可为租用定义存储域，以便从该模板创建虚拟机，预先配置此存储域作为租用位置。

7.313.9. 内存

虚拟机的内存，以字节为单位。

例如，要更新虚拟机使其包含 1 Gibibyte(GiB)内存，请发送以下请求：

```
PUT /ovirt-engine/api/vms/123
```

使用以下请求正文：

```
<vm>
  <memory>1073741824</memory>
</vm>
```

Red Hat Virtualization 3.6 支持内存热插拔。在虚拟机处于 **up** 状态时，您可以使用上述示例来增加内存。增量的大小必须可以被分成 `HotPlugMemoryBlockSizeMb` 配置值（默认为 256 MiB）。如果此值无法分隔内存大小递增，则内存大小更改仅存储在下次运行配置中。每个成功的内存热插拔操作都会创建一个或两个新内存设备。

从 Red Hat Virtualization 4.2 开始支持内存热插拔。**内存热拔头只能在虚拟机处于启动状态时执行**。只有之前热插拔内存设备才能通过热拔出操作删除。请求的内存减少向下舍入，以匹配之前热插拔内存设备的组合大小。请求的内存值存储在下次运行配置中，而不舍入。



注意

这个示例中的内存会被转换为字节：
 $1 \text{ GiB} = 2^{30} \text{ 字节} = 1073741824 \text{ 字节}$ 。



注意

红帽虚拟化管理器内部将值向下舍入到整个 MiB ($1 \text{ MiB} = 2^{20} \text{ 字节}$)

7.313.10. migration_downtime

虚拟机在实时迁移到 `ms` 中另一主机期间可以不响应的最大时间。

为虚拟机明确设置，或者使用 `engine-config -s DefaultMaximumMigrationDowntime=[value]`

7.313.11. Origin

此虚拟机的起源。

可能的值：

- `ovirt`
- `rhev`
- `vmware`
- `Xen`
- `external`
- `hosted_engine`
- `managed_hosted_engine`
- `kvm`
- `physical_machine`

- **hyperv**

7.313.12. placement_policy

虚拟机放置策略的配置。

此配置可以更新为将虚拟机固定到一个或多个主机。

**注意**

固定到多个主机的虚拟机无法实时迁移，但在主机出现故障时，任何配置为高可用性的虚拟机都会自动在虚拟机要固定到的其他主机上重新启动。

例如，要将虚拟机固定到两个主机，请发送以下请求：

```
PUT /api/vms/123
```

这样的请求正文：

```
<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>
```

7.313.13. small_icon

虚拟机的小图标。用户设置或引用根据操作系统设置的镜像集。

7.313.14. sso

此虚拟机被配置至单点登录配置的引用。打开控制台时，可以在虚拟机的操作系统中自动签署该用户。

7.313.15. tpm_enabled

如果为 `true`，则会在虚拟机中添加 TPM 设备。默认值为 `false`。只有在设置了 "All-Content=true" 标头时，此属性才能在获取时可见。

表 7.408. 链接摘要

名称	类型	概述
<code>cluster</code>	Cluster	虚拟机所属群集的引用。
<code>cpu_profile</code>	CpuProfile	引用此虚拟机使用的 CPU 配置集。
<code>quota</code>	Quota	引用为此虚拟机设置的配额配置。
<code>storage_domain</code>	StorageDomain	虚拟机所属存储域的引用。

7.314. VMDEVICETYPE ENUM

表 7.409. 值概述

名称	概述
<code>cdrom</code>	
<code>floppy</code>	

7.315. VMMEDIEDDEVICE STRUCT

VM mediated device 是一个虚拟设备，用于指定 vGPU 介质设备的属性。它并不是实际设备，它正作为指定如何配置主机设备的一部分的方法。

表 7.410. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。
spec_params	attribute[]	设备的属性。

表 7.411. 链接摘要

名称	类型	概述
instance_type	InstanceType	(可选) 引用设备使用的实例类型。
模板	Template	(可选) 引用设备使用的模板。
vm	VM	不要使用此元素，而是使用 vms 。
vms	Vm[]	引用使用此设备的虚拟机。

7.315.1. vms

引用使用此设备的虚拟机。一个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.316. VMPLACEMENTPOLICY STRUCT

表 7.412. 属性概述

名称	类型	概述
关联性	VmAffinity	

表 7.413. 链接摘要

名称	类型	概述
主机	Host[]	

7.317. VMPOOL STRUCT

类型，代表虚拟机池。

表 7.414. 属性概述

名称	类型	概述
auto_storage_select	布尔值	指明池是否应该在复制模板的多个存储域中自动分发虚拟机的磁盘。
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
display	显示	为池中的虚拟机配置的显示设置。
id	字符串	唯一标识符。
max_user_vms	整数	池中可以分配给特定用户的最大虚拟机数量。
name	字符串	纯文本中的人类可读名称。
prestarted_vms	整数	系统尝试从池中预启动指定数量的虚拟机。
rng_device	RngDevice	为池中的虚拟机配置的随机数生成器设备。
Size	整数	池中虚拟机的数量。
soundcard_enabled	布尔值	指明是否应为池中的虚拟机配置声卡。
有状态	布尔值	虚拟机池的有状态标志。
tpm_enabled	布尔值	如果为 true ，则会在虚拟机中添加 TPM 设备。
type	VmPoolType	池中虚拟机的取消分配策略。
use_latest_template_version	布尔值	指明池中的虚拟机是否已更新为池所基于的较新版本的模板。

7.317.1. auto_storage_select

指明池是否应该在复制模板的多个存储域中自动分发虚拟机的磁盘。

当池使用的模板存在于多个存储域中时，池中的虚拟机磁盘将创建在其中一个存储域中。默认情况

下，或者此属性的值为 `false` 时，该存储域会在池创建时选择，并且所有虚拟机都将使用相同的存储域。如果此属性为 `true`，那么当虚拟机添加到池中时，系统会选择具有更多可用空间的存储域。

7.317.2. `prestarted_vms`

系统尝试从池中预启动指定数量的虚拟机。

这些虚拟机已启动，无需附加到任何用户。这样，用户可以更快地从池中获取虚拟机。

7.317.3. 有状态

虚拟机池的有状态标志。

有状态虚拟机池的虚拟机始终以有状态模式启动（不会创建无状态快照）。即使虚拟机传递给其他用户，虚拟机的状态也会保留。

7.317.4. `tpm_enabled`

如果为 `true`，则会在虚拟机中添加 TPM 设备。默认值为 `false`。只有在设置了 "All-Content=true" 标头时，此属性才能在获取时可见。

表 7.415. 链接摘要

名称	类型	概述
<code>cluster</code>	<code>Cluster</code>	对池驻留的集群的引用。
<code>instance_type</code>	<code>InstanceType</code>	引用此池所基于的实例类型。
权限	<code>permission[]</code>	为此虚拟机池设置权限。
模板	<code>Template</code>	对池基于的模板的引用。
<code>vm</code>	<code>VM</code>	对属于池的任意虚拟机的引用。

7.317.5. `instance_type`

引用此池所基于的实例类型。它只能在创建池时设置，且不可编辑。

7.317.6. vm

对属于池的任意虚拟机的引用。

请注意，此虚拟机可能不以池模板的最新版本为基础。

7.318. VMPOOLTYPE ENUM

在虚拟机池中表示虚拟机取消分配策略的类型。

表 7.416. 值概述

名称	概述
automatic	此策略表示池中的虚拟机自动由系统取消分配。
Manual	此策略表示管理员手动取消池中的虚拟机分配。

7.318.1. 自动

此策略表示池中的虚拟机自动由系统取消分配。

通过此策略，当虚拟机属于池并分配给用户时，虚拟机会从用户分离，其状态将恢复到池的默认状态，虚拟机将返回到池（即，可以将虚拟机分配给另一个用户）。

7.318.2. Manual

此策略表示管理员手动取消池中的虚拟机分配。

采用此策略时，属于池的虚拟机仍会分配给用户，并在关机时保持其状态。要将虚拟机返回到池，管理员需要通过删除用户在该虚拟机上的权限来显式取消分配。

7.319. VMSTATUS ENUM

类型，代表虚拟机的状态。

表 7.417. 值概述

名称	概述
down	此状态表示虚拟机进程未在运行。
image_locked	此状态表示虚拟机进程没有运行，并且虚拟机磁盘上有一些操作会阻止它启动。
迁移	此状态表示虚拟机进程正在运行，并且虚拟机正在从一个主机迁移到另一个主机。
not_responding	此状态表示系统管理程序检测到虚拟机没有响应。
paused	此状态表示虚拟机进程正在运行，并且虚拟机已暂停。
powering_down	此状态表示虚拟机进程正在运行，并即将停止运行。
powering_up	此状态表示虚拟机进程正在运行，并且正在加载客户机操作系统。
reboot_in_progress	此状态表示虚拟机进程正在运行，并且客户机操作系统正在重启。
restoring_state	此状态表示虚拟机进程即将运行，并且虚拟机也将从休眠中中止。
saving_state	此状态表示虚拟机进程正在运行，并且虚拟机正在休眠。
暂停	此状态表示虚拟机进程未在运行，并且已保存虚拟机的运行状态。
unassigned	当收到无效状态时，会设置此状态。
Unknown	此状态表示系统无法确定虚拟机的状态。
up	此状态表示虚拟机进程正在运行，并且已加载客户机操作系统。
wait_for_launch	此状态表示虚拟机进程即将运行。

7.319.1. paused

此状态表示虚拟机进程正在运行，并且虚拟机已暂停。这在两种情形中可能会发生：运行虚拟机时为暂停模式，以及虚拟机因错误而自动暂停时。

7.319.2. powering_up

此状态表示虚拟机进程正在运行，并且正在加载客户机操作系统。请注意，如果没有安装 guest-agent，则在运行虚拟机时，会为预定义的时间（默认为 60 秒）设置此状态。

7.319.3. restoring_state

此状态表示虚拟机进程即将运行，并且虚拟机也将从休眠中中止。在此状态下，正在恢复虚拟机的运行状态。

7.319.4. saving_state

此状态表示虚拟机进程正在运行，并且虚拟机正在休眠。在此状态下，将保存虚拟机的运行状态。请注意，此状态并不意味着客户机操作系统正在休眠。

7.319.5. 暂停

此状态表示虚拟机进程未在运行，并且已保存虚拟机的运行状态。这个状态与 Down 类似，但在此状态下启动虚拟机时，其保存的运行状态将被恢复，而不是使用普通的 `procedue` 启动。

7.319.6. Unknown

此状态表示系统无法确定虚拟机的状态。虚拟机进程可能正在运行或未在此状态中运行。例如，当主机变得不响应时，其上运行的虚拟机将设置为此状态。

7.319.7. up

此状态表示虚拟机进程正在运行，并且已加载客户机操作系统。请注意，如果没有安装 `guest-agent`，则在预定义的时间后（默认为 60 秒）在运行虚拟机时设置此状态。

7.319.8. wait_for_launch

此状态表示虚拟机进程即将运行。当运行虚拟机的请求到达主机时，会设置此状态。虚拟机进程可能无法运行。

7.320. VMSTORAGEERRORRESUMEBEHAVIOUR ENUM

如果存储此虚拟机具有部分磁盘无响应，则虚拟机将暂停。

这是可能的选项，在存储再次可用时虚拟机应发生什么情况。

表 7.418. 值概述

名称	概述
<code>auto_resume</code>	当存储再次可用时，虚拟机会自动恢复。
<code>kill</code>	虚拟机将在超时时终止（可在系统管理程序上配置）。
<code>leave_paused</code>	切勿与虚拟机执行任何操作。

7.320.1. `auto_resume`

当存储再次可用时，虚拟机会自动恢复。

这是 4.2 之前唯一可用的行为。

7.320.2. `kill`

虚拟机将在超时时终止（可在系统管理程序上配置）。

这是对具有租用的高可用性虚拟机唯一支持的选项。原因在于，高可用性虚拟机使用基础架构重新启动，并且任何类型的恢复风险都存在脑裂的风险。

7.320.3. `leave_paused`

切勿与虚拟机执行任何操作。

如果实施了自定义故障转移，并且用户不希望虚拟机恢复，则非常有用。

7.321. VMSUMMARY STRUCT

类型，包含与特定主机上虚拟机相关的信息。

表 7.419. 属性概述

名称	类型	概述
<code>Active</code>	整数	主机上活跃的虚拟机数量。

名称	类型	概述
迁移	整数	迁移至主机或从主机迁移的虚拟机数量。
总计	整数	主机上存在的虚拟机数量。

7.322. VMTYPE ENUM

类型表示要优化的虚拟机。

表 7.420. 值概述

名称	概述
desktop	虚拟机旨在用作桌面。
high_performance	虚拟机旨在用作高性能虚拟机。
server	虚拟机旨在用作服务器。

7.322.1. desktop

虚拟机旨在用作桌面。

目前，这意味着声调设备将自动添加到虚拟机中。

7.322.2. high_performance

虚拟机旨在用作高性能虚拟机。

目前，其含义在于，虚拟机配置将自动设置为以最高性能运行，并且具有尽可能接近裸机的性能指标。

无法自动设置适用于最高性能的部分推荐配置设置；建议在运行虚拟机前手动设置它们。

以下配置更改会自动设置：

- 启用无外设模式。
- 启用串行控制台。
- 启用传递主机 CPU。
- 启用 I/O 线程。
- 启用 I/O 线程固定并设置固定拓扑。
- 启用半虚拟化随机数字生成器 PCI(virtio-rng)设备。
- 禁用所有 USB 设备。
- 禁用声卡设备。
- 禁用智能卡设备。
- 禁用内存气球设备。
- 禁用 watchdog 设备。
- 禁用迁移。
- 禁用高可用性。

以下推荐的配置更改必须由用户手动设置：

- 启用 CPU 固定拓扑。
- 启用非统一内存访问(NUMA)固定拓扑。
- 启用和设置巨页配置。
- 禁用内核同页合并(KSM)。

7.322.3. server

虚拟机旨在用作服务器。

目前，其含义在于音响设备不会自动添加到虚拟机中。

7.323. VNICPASSTHROUGH STRUCT

表 7.421. 属性概述

名称	类型	概述
mode	VnicPassThrough Mode	定义 vNIC 将实施为虚拟设备，还是作为传递到主机设备的直通实现。

7.324. VNICPASSTHROUGHMODE ENUM

描述 vNIC 作为直通设备或虚拟设备实施。

表 7.422. 值概述

名称	概述
disabled	作为虚拟设备实施。
enabled	作为直通设备实施。

7.325. VNICPROFILE STRUCT

vNIC 配置集是一组可应用于单个 NIC 的设置。

表 7.423. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
custom_properties	CustomProperty[]	应用到 vNIC 配置集的自定义属性。
description	字符串	以纯文本格式人类可读的描述。
id	字符串	唯一标识符。
migratable	布尔值	标记 pass_through NIC 是否是 migratable。
name	字符串	纯文本中的人类可读名称。
pass_through	VnicPassThrough	启用传递给启用了 SR-IOV 的主机 NIC。
port_mirroring	布尔值	启用端口镜像。

7.325.1. migratable

标记 **pass_through** NIC 是否是 migratable。

如果将 **pass_through.mode** 设置为 **disabled** 这个选项没有含义，它将被视为 **true**。如果您从请求中省略这个选项，则默认为 **true**。

迁移虚拟机时，只有在所有 **pass_through** NIC 都标记为 **migratable** 时，才会迁移此虚拟机。

7.325.2. pass_through

启用传递给启用了 SR-IOV 的主机 NIC。

vNIC 配置集允许 NIC 直接连接到支持 SR-IOV 的主机 NIC 的虚拟功能(VF) (如果启用了 **passthrough**)。然后 NIC 将绕过软件网络虚拟化，并直接连接到 VF 进行直接设备分配。

如果 vNIC 配置集已附加到 NIC，则无法启用 `passthrough`。如果启用了 `passthrough` 配置集，则 vNIC 配置集会禁用 `qos` 和 `port_mirroring`。

7.325.3. `port_mirroring`

启用端口镜像。

端口镜像会将给定逻辑网络和主机上的第 3 层网络流量复制到虚拟机上的 NIC。此虚拟机可用于网络调试和调优、入侵检测，以及监控同一主机和逻辑网络上其他虚拟机的行为。复制的唯一流量是一台主机上一个逻辑网络的内部流量。主机外部的网络上的流量不会增加；但是，与其他虚拟机相比，启用了端口镜像的虚拟机会使用更多的主机 CPU 和 RAM。

端口镜像有以下限制：

- 不支持使用启用了端口镜像的 vNIC 配置文件热插拔 NIC。
- 当 vNIC 配置集附加到虚拟机时，无法更改端口镜像。

鉴于上述限制，建议您在额外的专用 vNIC 配置集上启用端口镜像。



重要

启用端口镜像可以降低其他网络用户的隐私。

表 7.424. 链接摘要

名称	类型	概述
故障切换	VnicProfile	SR-IOV 迁移的故障切换 vNIC 配置集，无需停机
network	Network	引用 vNIC 配置集应用到的网络。
network_filter	NetworkFilter	引用应用到使用此配置集的 NIC 的顶级网络过滤器。
权限	permission[]	允许使用 vNIC 配置集的权限。

名称	类型	概述
qos	QoS	引用应用到 vNIC 配置集的服务属性质量。

7.325.4. network_filter

引用应用到使用此配置集的 NIC 的顶级网络过滤器。

网络过滤器增强了管理进出虚拟机的网络数据包流量的能力。网络过滤器可以包含对其他过滤器的引用、流量过滤规则或两者的组合。

7.325.5. qos

引用应用到 vNIC 配置集的服务属性质量。

Service 属性的质量控制 NIC 的进站和出站网络流量。

7.326. VNICPROFILEMAPPING STRUCT

将外部虚拟 NIC 配置集映射到 Red Hat Virtualization Manager 中存在的已弃用类型。

例如，如果所需的虚拟 NIC 配置集映射包括以下两行：

源网络名称	源网络配置集名称	目标虚拟 NIC 配置集 ID
红色	gold	738dd914-8ec8-4a8b-8628-34672a5d449b
blue	Silver	892a12ec-2028-4451-80aa-ff3bf55d6bac

以下表单自 4.2.1 起已弃用，并将在以后的中删除：

```
<vnic_profile_mappings>
  <vnic_profile_mapping>
    <source_network_name>red</source_network_name>
```

```

<source_network_profile_name>gold</source_network_profile_name>
<target_vnic_profile id="738dd914-8ec8-4a8b-8628-34672a5d449b"/>
</vnic_profile_mapping>
<vnic_profile_mapping>
  <source_network_name>blue</source_network_name>
  <source_network_profile_name>silver</source_network_profile_name>
  <target_vnic_profile id="892a12ec-2028-4451-80aa-ff3bf55d6bac"/>
</vnic_profile_mapping>
</vnic_profile_mappings>

```

表 7.425. 属性概述

名称	类型	概述
<code>source_network_name</code>	字符串	描述外部网络名称的属性。
<code>source_network_profile_name</code>	字符串	描述外部网络配置文件名称的属性。

7.326.1. `source_network_name`

描述外部网络名称的属性。



警告

请注意，此属性自引擎版本 4.2.1 起已弃用，仅用于向后兼容。以后会将其移除。

7.326.2. `source_network_profile_name`

描述外部网络配置文件名称的属性。

**警告**

请注意，此属性自引擎版本 4.2.1 起已弃用，仅用于向后兼容。以后会将其移除。

表 7.426. 链接摘要

名称	类型	概述
target_vnic_profile	VnicProfile	描述现有虚拟 NIC 配置文件的已弃用属性。

7.326.3. target_vnic_profile

描述现有虚拟 NIC 配置文件的已弃用属性。

**警告**

请注意，此属性自引擎版本 4.2.1 起已弃用，仅用于向后兼容。以后会将其移除。

7.327. VOLUMEGROUP STRUCT

表 7.427. 属性概述

名称	类型	概述
id	字符串	
logical_units	LogicalUnit[]	
name	字符串	

7.328. WATCHDOG STRUCT

此类型代表 `watchdog` 配置。

表 7.428. 属性概述

名称	类型	概述
<code>action</code>	WatchdogAction	触发 <code>watchdog</code> 时执行的 <code>watchdog</code> 操作。
<code>comment</code>	字符串	包含有关此对象的注释的免费文本。
<code>description</code>	字符串	以纯文本格式人类可读的描述。
<code>id</code>	字符串	唯一标识符。
<code>model</code>	WatchdogModel	<code>watchdog</code> 设备模型。
<code>name</code>	字符串	纯文本中的人类可读名称。

7.328.1. `model`

`watchdog` 设备模型。目前仅支持 `I6300ESB`。

表 7.429. 链接摘要

名称	类型	概述
<code>instance_type</code>	InstanceType	(可选) 引用设备使用的实例类型。
模板	Template	(可选) 引用设备使用的模板。
<code>vm</code>	VM	不要使用此元素，而是使用 <code>vms</code> 。
<code>vms</code>	Vm[]	引用使用此设备的虚拟机。

7.328.2. `vms`

引用使用此设备的虚拟机。一个设备可以由多个虚拟机使用；例如，我的共享磁盘由两个或多个虚拟机同时使用。

7.329. WATCHDOGACTION ENUM

此类型描述了可用的 `watchdog` 操作。

表 7.430. 值概述

名称	概述
<code>dump</code>	虚拟机进程将核心转储到主机上的默认路径。
<code>none</code>	触发 <code>watchdog</code> 操作时不会执行任何操作。
<code>pause</code>	触发 <code>watchdog</code> 操作时，虚拟机将暂停。
<code>poweroff</code>	触发 <code>watchdog</code> 操作时，虚拟机将被关闭。
<code>reset</code>	触发 <code>watchdog</code> 操作时，虚拟机将重新启动。

7.329.1. `none`

触发 `watchdog` 操作时不会执行任何操作。但是，仍会生成日志消息。

7.330. `WATCHDOGMODEL ENUM`

这个类型代表 `watchdog` 模型。

表 7.431. 值概述

名称	概述
<code>diag288</code>	S390X 计算机的 <code>watchdog</code> 模型。
<code>i6300esb</code>	基于 PCI 的监控模型。

7.330.1. `diag288`

S390X 计算机的 `watchdog` 模型。

S390X 具有集成的 `watchdog` 设备，可通过 `DIAG288` 指令控制。对 S390X 虚拟机使用此模型。

7.330.2. `i6300esb`

基于 PCI 的监控模型。

将 I6300ESB watchdog 用于 x86_64 和 PPC64 虚拟机。

7.331. WEIGHT STRUCT

表 7.432. 属性概述

名称	类型	概述
comment	字符串	包含有关此对象的注释的免费文本。
description	字符串	以纯文本格式人类可读的描述。
factor	整数	
id	字符串	唯一标识符。
name	字符串	纯文本中的人类可读名称。

表 7.433. 链接摘要

名称	类型	概述
scheduling_policy	SchedulingPolicy	
scheduling_policy_unit	SchedulingPolicyUnit	

附录 A. 原语类型

本节论述了 API 支持的原始数据类型。

A.1. 字符串 原语

无限的 **Unicode** 字符序列。

A.2. 布尔值 原语

代表数学逻辑中使用的假和 真实概念。

有效值是字符串 `false` 和 `true`。

引擎将忽略大小写，例如 `False` 和 `FALSE` 也有效值。但是，服务器将始终返回小写值。

为了与较旧版本的引擎向后兼容，也接受值 `0` 和 `1`。值 `0` 的含义与 `false` 相同，`1` 的含义与 `true` 相同。尽量避免使用这些值，因为将来可能会删除对这些值的支持。

A.3. 整数 原语

代表整数的数学概念。

有效值是十进制数字的有限序列。

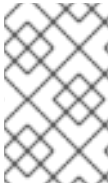
目前引擎使用签名的 32 位整数实施此类型，因此最小值为 -2^{31} (-2147483648)，最大值为 $2^{31}-1$ (2147483647)。

但是，系统中有一些属性，其中可能具有 32 位的值范围不够。在这些情况下，引擎使用 64 位整数，特别是以下属性：

- `disk.actual_size`

- *Disk.provisioned_size*
- *GlusterClient.bytes_read*
- *GlusterClient.bytes_written*
- *Host.max_scheduling_memory*
- *Host.memory*
- *HostNic.speed*
- *LogicalUnit.size*
- *MemoryPolicy.guaranteed*
- *NumaNode.memory*
- *QuotaStorageLimit.limit*
- *StorageDomain.available*
- *StorageDomain.used*
- *StorageDomain.committed*
- *VmBase.memory*

在这些例外情况下，最小值为 -2^{63} (-9223372036854775808)，最大值为 $2^{63}-1$ (9223372036854775807)。



注意

将来，整数类型将使用无限制的前序整数实施，因此上述限制和例外最终会消失。

A.4. 十进制 原语

代表真实数字的数学概念。

目前，引擎使用 32 位 [IEEE 754](#) 单精确浮动点数实施此类型。

对于某些属性而言，这不够精确。在这种特殊情况下，引擎使用 64 位双精确浮动点数，特别是以下属性：

- `QuotaStorageLimit.usage`
- `QuotaStorageLimit.memory_limit`
- `QuotaStorageLimit.memory_usage`



注意

将来，十进制类型将使用无限制的十进制数字实施，因此上述限制和例外最终会消失。

A.5. 日期 原语

表示日期和时间。

引擎返回的格式是请求 [XML Schema](#) 规格中描述的格式。例如，如果您发送类似此请求来检索虚拟机的 XML 表述：

```
GET /ovirt-engine/api/vms/123
Accept: application/xml
```

响应正文将包含以下 XML 文档：

```
<vm id="123" href="/ovirt-engine/api/vms/123">
...
<creation_time>2016-09-08T09:53:35.138+02:00</creation_time>
...
</vm>
```

在请求 JSON 表示时，引擎使用不同的格式：整数，其包含自1970年1月1日起的毫秒数，也称为epoch 时间。例如，如果您发送类似此请求，以检索虚拟机的 JSON 表示形式：

```
GET /ovirt-engine/api/vms/123
Accept: application/json
```

响应正文将包含以下 JSON 文档：

```
{
  "id": "123",
  "href": "/ovirt-engine/api/vms/123",
  ...
  "creation_time": 1472564909990,
  ...
}
```



注意

在这两种情况下，引擎返回的日期使用运行它的服务器中配置的时区，在上面的示例中是 UTC+2。

附录 B. API 版本 4 的更改

本节说明了从 API 版本 4 开始的后向兼容性破坏所引入的更改。

B.1. API 版本 4.0 的更改 (显示版本 3.6)

B.1.1. 删除了 YAML 支持

对 YAML 的支持已完全删除。

B.1.2. 重新命名复杂的类型

以下 XML 模式复杂类型已被重命名：

第 3 版	第 4 版
API	api
CPU	cpu
CPU	CPU
CdRom	cdrom
CdRoms	CDROM
DNS	DNS
GuestNicConfiguration	NicConfiguration
GuestNicsConfiguration	NicConfigurations
HostNICStates	HostNicStates
HostNIC	HostNic
HostStorage	HostStorages
IO	io
IP	ip

第 3 版	第 4 版
ips	ips
KSM	km
MAC	MAC
NIC	NIC
PreviewVMs	PreviewVms
QoS	QoS
QoSs	Qoss
RSDL	RsdI
SELinux	SeLinux
SPM	SPM
SSHPublicKey	SshPublicKey
SSHPublicKeys	SshPublicKeys
SSH	ssh
SkipIfSDActive	SkipIfSdActive
slaves	HostNics
存储	HostStorage
SupportedVersions	版本
VCpuPin	VcpuPin
VLAN	VLAN
VM	VM
VM	VM
VirtIO_SCSI	VirtioScsi
WatchDog	Watchdog

第 3 版	第 4 版
WatchDogs	watchdogs

B.1.3. 使用 enum 类型替换 Status 类型

目前，不同对象的状态是使用 **Status** 类型报告，它包含一个 **状态** 字符串和另一个 **详细信息** 字符串以了解更多详细信息。例如，当前报告因为 IO 错误而暂停的虚拟机的状态，如下所示：

```
<vm>
...
<status>
  <state>paused</state>
  <detail>eio</detail>
</status>
...
</vm>
```

在 API 的版本 4 中，此 **Status** 类型已删除，并被 **enum type** 替换。需要额外的 **详细信息** 字符串时，它已被额外的 **status_detail** 属性替代。例如，同一虚拟机的状态现在将报告如下：

```
<vm>
...
<status>paused</status>
<status_detail>eio</status_detail>
...
</vm>
```

B.1.4. 删除 NIC 网络和 port_mirroring 属性

NIC 网络和 **port_mirroring** 元素已被 **vnic_profile** 元素替代，因此在创建或更新 **NIC** 时，不会指定网络和端口镜像配置，这些会预先指定创建 **vNIC 配置集**：

POST /ovirt-engine/api/vnicprofiles

```
<vnic_profile>
  <name>myprofile</name>
  <network id="..." />
  <port_mirroring>true</port_mirroring>
</vnic_profile>
```

然后，创建 **NIC** 或引用现有的 **vNIC 配置集**：

PUT /ovirt-engine/api/vms/123/nics/456

```
<nic>  
  <vnic_profile id="/vnicprofiles/...">  
</nic>
```

旧元素及其含义保留下来，以便向后兼容，但现已完全删除。

请注意，网络元素尚未从 XML 架构中删除，因为它仍由初始化元素使用，但是如果在创建或更新 NIC 时提供，则完全被忽略。

B.1.5. 删除 NIC active 属性

NIC active 属性在一段时间前被插入。现在它已被完全删除。

B.1.6. 删除磁盘类型属性

磁盘的类型属性已被删除，但保留在 XML 架构中并忽略。现在它已被完全删除。

B.1.7. 删除磁盘大小属性

磁盘大小属性在之前已被 provisioned_size 替代。现在它已被完全删除。

B.1.8. 删除了对将虚拟机固定到单一主机的支持

在版本 3.6 之前，API 可以使用 VM 实体的 placement_policy 元素将虚拟机固定到单一主机上：

PUT /ovirt-engine/api/vms/123

```
<vm>  
  <placement_policy>  
    <host id="456"/>  
  </placement_policy>  
</vm>
```

在版本 3.6 中，这个功能被改进来支持多个主机，并添加了一个新的主机元素：

PUT /ovirt-engine/api/vms/123

```

<vm>
  <placement_policy>
    <hosts>
      <host id="456"/>
      <host id="789"/>
      ...
    </hosts>
  </placement_policy>
</vm>

```

为了保持向后兼容性，单一 `主机` 元素已被保留。在 4.0 中，这已被删除，因此即使固定到单个主机，应用也需要使用 `hosts` 元素。

B.1.9. 删除了 `capabilities.permits` 元素

对于每个集群级别，其允许列表可能有所不同，而且几年前已添加到 `version` 元素中，但它也保留在 `capabilities` 元素中，只是为了向后兼容。

在 4.0 中，功能 `服务` 已被完全删除，并被新的集群级别 `服务` 替代。要查找集群级别 4.0 支持的允许，应使用如下请求：

```
GET /ovirt-engine/api/clusterlevels/4.0
```

结果将是包含特定于该集群级别的信息的文档，特别是所支持的允许集合：

```

<cluster_level id="4.0" href="/clusterlevels/4.0">
  ...
  <permits>
    <permit id="1">
      <name>create_vm</name>
      <administrative>false</administrative>
    </permit>
    ...
  </permits>
</cluster_level>

```

B.1.10. 删除了 `storage_manager` 元素

`storage_manager` 元素在一段时间前已被 `spm` 元素取代。旧版本保留为向后兼容性，但现已完全删除。

B.1.11. 删除了数据中心 `storage_type` 元素

数据中心过去与特定的存储类型（NFS、光纤通道、iSCSI 等）关联，但是它们已经进行了更改，使得只有两种类型：本地存储以及共享存储。引入了一个新的本地元素来指示这一点，并且保留了旧的 `storage_type` 元素，以便向后兼容。这个旧元素现已完全删除。

B.1.12. 删除 `timezone` 元素

用于包含用于表示时区的 `timezone` 元素的虚拟机资源。这个元素只允许字符串：

```
<vm>
  <timezone>Europe/Madrid</timezone>
</vm>
```

这不允许扩展，因为需要添加 UTC 偏移，它被替换为一个新的结构化 `time_zone` 元素：

```
<vm>
  <time_zone>
    <name>Europe/Madrid</name>
    <utc_offset>GMT+1</utc_offset>
  </time_zone>
</vm>
```

旧时区元素已被保留，但现已完全删除。

B.1.13. 删除了 `guest_info` 元素

`guest_info` 元素用于存放客户机代理收集的信息，如 IP 地址和完全限定主机名。此信息在其他地点也可获得。例如，IP 地址在 VM 资源中可用：

GET /ovirt-engine/api/vms/123

```
<vm>
  <guest_info>
    <ips>
      <ip address="192.168.122.30"/>
    </ips>
    <fqdn>myvm.example.com</fqdn>
  </guest_info>
</vm>
```

另外，在 NIC 资源中使用较新的 `reporting_devices` 元素：

GET /ovirt-engine/api/vms/{vm:id}/nics/{nic:id}

```

<nic>
  <reported_devices>
    <reported_device>
      <name>eth0</name>
      <mac address="00:1a:4a:b5:4c:94"/>
      <ips>
        <ip address="192.168.1.115" version="v4"/>
        <ip address="fe80::21a:4aff:feb5:4c94" version="v6"/>
        <ip address="::1:21a:4aff:feb5:4c94" version="v6"/>
      </ips>
    </reported_device>
  </reported_devices>
</nic>

```

此外，这个较新的 `report_devices` 元素还提供更完整的信息，如多个 IP 地址、MAC 地址等。

为删除此重复，`guest_info` 元素已被删除。

为了支持完全限定域名，在 VM 资源中添加了新的 `fqdn` 元素：

GET /ovirt-engine/api/vms/123

```

<vm>
  <fqdn>myvm.example.com</fqdn>
</vms>

```

此命令将包含 `guest_info.fqdn` 用于包含的相同信息。

B.1.14. 使用 `type` 元素替换 CPU id 属性

用于具有代表 CPU 类型的 `id` 属性的 `cpu` 元素：

```

<cpu id="Intel Nehalem Family">
  <architecture>X86_64</architecture>
  ...
</cpu>

```

这与 API 模型的其余元素有关，其中 `id` 属性用于不透明标识符。这个 `id` 属性已被一个新的 `type` 元素替代：

```
<cpu>
  <type>Intel Nehalem Family</type>
  <architecture>X86_64</architecture>
</cpu>
```

B.1.15. 在 CPU 拓扑中使用元素而不是属性

在过去，CPU 拓扑元素将属性用作其属性：

```
<cpu>
  <topology sockets="1" cores="1" threads="1"/>
  ...
</cpu>
```

这与 API 中的常见做法相反。它们已被内联元素取代：

```
<cpu>
  <topology>
    <sockets>1</sockets>
    <cores>1</cores>
    <threads>1</threads>
  </topology>
  ...
</cpu>
```

B.1.16. 在 VCPU pin 中使用元素而不是属性

在过去，VCPU pin 元素将属性用作其属性：

```
<cpu_tune>
  <vcpu_pin vcpu="0" cpu_set="0"/>
</cpu_tune>
```

这与 API 中的常见做法相反。它们已被内联元素取代：

```
<cpu_tune>
  <vcpu_pin>
    <vcpu>0</vcpu>
    <cpu_set>0</cpu_set>
  </vcpu_pin>
</cpu_tune>
```

B.1.17. 在 VCPU pin 中使用元素而不是属性

在过去，`version` 元素的属性是它的属性：

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

这与 API 中的常见做法相反。它们已被内联元素取代：

```
<version>
  <major>3</minor>
  <minor>5</minor>
  ...
</version>
```

B.1.18. 在内存过量使用中元素而不是属性

过去，`过量使用` 元素的属性是它的属性：

```
<memory_policy>
  <overcommit percent="100"/>
  ...
</memory_policy>
```

这与 API 中的常见做法相反。它们已被内联元素取代：

```
<memory_policy>
  <overcommit>
    <percent>100</percent>
  </overcommit>
  ...
</memory_policy>
```

B.1.19. 在控制台中使用元素而不是属性

在过去，`console` 元素 将属性用作其属性：

```
<console enabled="true"/>
```

这与 API 中的常见做法相反。它们已被内联元素取代：

```
<console>
  <enabled>true</enabled>
</console>
```

B.1.20. 在 VIRTIO SCSI 中使用元素而不是属性

在过去，VIRTIO ISCSI 元素在其属性中使用属性：

```
<virtio_scsi enabled="true"/>
```

这与 API 中的常见做法相反。它们已被内联元素取代：

```
<virtio_scsi>
  <enabled>true</enabled>
</virtio_scsi>
```

B.1.21. 对于电源管理代理类型，使用 element 而不是属性

电源管理类型属性表示为属性：

```
<agent type="apc">
  <username>myuser</username>
  ...
</agent>
```

这与 API 中的常见做法相反。它已被一个内部元素替代：

```
<agent>
  <type>apc</type>
  <username>myuser</username>
  ...
</agent>
```

B.1.22. 在电源管理代理选项中使用元素而不是属性

在过去，电源管理代理选项元素在其属性中使用属性：

```
<options>
  <option name="port" value="22"/>
  <option name="slot" value="5"/>
  ...
</options>
```


这与 API 中的常见做法相反。它们已被内联元素取代：

```
<options>
  <option>
    <name>port</name>
    <value>22</value>
  </option>
  <option>
    <name>slot</name>
    <value>5</value>
  </option>
  ...
</options>
```

B.1.23. 在 IP 地址中使用元素而不是属性：

在过去，IP 地址元素将属性用作其属性：

```
<ip address="192.168.122.1" netmask="255.255.255.0"/>
```

这与 API 中的常见做法相反。它们已被内联元素取代：

```
<ip>
  <address>192.168.122.1</address>
  <netmask>255.255.255.0</netmask>
</ip>
```

B.1.24. 在 MAC 地址中使用元素而不是属性：

在过去，MAC 地址元素将属性用作其属性：

```
<mac address="66:f2:c5:5f:bb:8d"/>
```

这与 API 中的常见做法相反。它们已被内联元素取代：

```
<mac>
  <address>66:f2:c5:5f:bb:8d</address>
</mac>
```

B.1.25. 在引导设备中使用元素而不是属性：

在过去，引导设备元素将属性用作其属性：

```
<boot dev="cdrom"/>
```

这与 API 中的常见做法相反。它们已被内联元素取代：

```
<boot>
  <dev>cdrom</dev>
</boot>
```

B.1.26. 对于操作系统类型，使用 element 而不是 属性

操作系统类型属性表示为属性：

```
<os type="other">
  ...
</os>
```

这与 API 中的常见做法相反。它已被一个内部元素替代：

```
<os>
  <type>other</type>
  ...
</os>
```

B.1.27. 从请求中删除了 force 参数，以检索主机

在从数据库检索之前，要检索用于支持强制矩阵参数的主机的请求，以指示应该刷新主机数据（扩展 VDSM 以重新加载主机功能和设备）：

```
GET /ovirt-engine/api/hosts/123;force
```

此 force 参数已被主机刷新操作取代，但保留为向后兼容性。现在它已被完全删除。需要此功能的应用程序应执行两个请求，第一个用于刷新主机：

```
POST /ovirt-engine/api/hosts/123/refresh
```

```
<action/>
```

然后，在没有 `force` 参数的情况下检索它：

```
GET /ovirt-engine/api/hosts/123
```

B.1.28. 删除了已弃用的主机电源管理配置

主机电源管理配置过去是主机资源的一部分，使用嵌入式配置元素：

```
<power_management type="apc">
  <enabled>true</enabled>
  <address>myaddress</address>
  <username>myaddress</username>
  <options>
    <option name="port" value="22"/>
    </option name="slot" value="5"/>
  </options>
  ...
</power_management>
```

这在几年前已经改变，为了支持多个电源管理代理，引入一个新的 `/hosts/123/fenceagents` 集合。

旧类型属性、旧地址、用户名和密码元素以及直接在 `power_management` 内的内代理元素保留下来，以便向后兼容性。所有这些元素已完全删除，因此查询或修改电源管理代理的唯一方法是 `/hosts/123/fenceagents` 子集合。

B.1.29. 使用多个 `boot.devices.device` 而不是多次引导

过去，在启动虚拟机时指定启动序列的方法是使用多个引导元素，每个元素包含一个 `dev` 元素。例如，指定虚拟机应首先尝试从 `CDROM` 引导，然后从硬盘引导，并使用以下请求：

```
POST /ovirt-engine/api/vms/123/start
```

```
<action>
  <vm>
    ...
    <boot>
      <dev>cdrom</dev>
    </boot>
    <boot>
      <dev>hd</dev>
    </boot>
  </vm>
</action>
```

API 其它部分的常见做法是使用 `wrapper` 元素表示数组。在这种情况下，`wrapper` 元素可以命名为 `boots`，但这并不合理，因为此处的有多个值是引导设备，而不是启动序列。要修复这个不一致性的问题，这个错误已被一个可以包含多个设备的 `引导` 元素替代：

POST /ovirt-engine/api/vms/123/start

```
<action>
  <vm>
    ...
    <boot>
      <devices>
        <device>cdrom</device>
        <device>hd</device>
      </devices>
    </boot>
  </vm>
</action>
```

B.1.30. 删除了 `disk.clone` 和 `disk.detach_only` 元素

这些元素不是磁盘表示的一部分，而是添加和删除虚拟机的操作参数。

`disks.clone` 元素用于指示必须克隆新虚拟机的磁盘：

POST /ovirt-engine/api/vms

```
<vm>
  ...
  <disks>
    <clone>true</clone>
  </disks>
</vm>
```

现在，这已被删除，并使用一个新的 `克隆` 查询参数替换：

POST /ovirt-engine/api/vms?clone=true

```
<vm>
  ...
</vm>
```

`disks.detach_only` 元素用于指示在删除虚拟机时磁盘不必删除，而是从虚拟机分离：

DELETE /ovirt-engine/api/vms/123

```

<action>
  <vm>
    <disks>
      <detach_only>true</detach_only>
    </disks>
  </vm>
</action>

```

现在，这已被删除，并被一个新的 `detach_only` 查询参数替代：

DELETE /ovirt-engine/api/vms/123?detach_only=true**B.1.31. 将元素 `vmpool` 重命名为 `vm_pool`**

代表虚拟机池的元素的名称，如 `vmpool` 和 `vm pools`。它们已被重命名为 `vm_pool` 和 `vm_pools`，以便在复杂类型的名称（本例中为 `VmPool` 和 `VmPools`）和元素之间保持一致。

B.1.32. 使用 `logical_units` 而不是多个 `logical_unit`

作为卷组的一部分的逻辑单元，报告为未绑定的 `logical_unit` 元素。例如，当报告存储域详情时：

GET /ovirt-engine/api/storagedomains/123

```

<storage_domain>
  ...
  <storage>
    ...
    <volume_group>
      <logical_unit>
        <!-- First LU -->
      </logical_unit>
      <logical_unit>
        <!-- Second LU -->
      </logical_unit>
    ...
  </volume_group>
</storage>
</storage_domain>

```

这与 API 中的常见做法相反，因为元素列表始终包含在元素中。这个问题现已解决，因此逻辑单元列表将与 `logical_units` 元素一起换行：

GET /ovirt-engine/api/storagedomains/123

```

<storage_domain>
  ...
  <storage>
    ...
    <volume_group>
      <logical_units>
        <logical_unit>
          <!-- First LU -->
        </logical_unit>
        <logical_unit>
          <!-- Second LU -->
        </logical_unit>
      ...
    </logical_units>
  </volume_group>
</storage>
</storage_domain>

```

B.1.33. 删除了 `snapshot.collapse_snapshots` 元素

此元素实际上不是快照表示的一部分，而是从导出存储域导入虚拟机的操作参数：

```
POST /ovirt-engine/api/storagedomains/123/vms/456/import
```

```

<action>
  <vm>
    <snapshots>
      <collapse_snapshots>true</collapse_snapshots>
    </snapshots>
  </vm>
</action>

```

现在，这已被删除，并被一个新的 `折叠_snapshots` 查询参数替代：

```
POST /ovirt-engine/api/storagedomains/123/vms/456/import?collapse_snapshots=true
```

```
<action/>
```

B.1.34. 重命名 `存储` 和 `host_storage` 元素

主机存储集合使用 `存储` 和 `host_storage` 元素以及 `Storage` 和 `HostStorage` 复杂类型来报告与主机关联的存储：

```
GET /ovirt-engine/api/hosts/123/storage
```

```

<host_storage>
  <storage>
    ...
  </storage>
  <storage>
    ...
  </storage>
  ...
</host_storage>

```

这不遵循 API 的其余部分中使用的模式，其中外部元素是复数名称，内元素是同名，但单数表示。现在，它已被修改为使用 `host_storages` 作为外部元素，`host_storage` 作为内部元素：

GET /ovirt-engine/api/hosts/123/storage

```

<host_storages>
  <host_storage>
    ...
  </host_storage>
  <host_storage>
    ...
  </host_storage>
  ...
</host_storages>

```

B.1.35. 删除了 `permissions.clone` 元素

此元素不是权限表示的一部分，而是创建虚拟机或模板的操作参数：

POST /ovirt-engine/api/vms

```

<vm>
  <template id="...">
    <permissions>
      <clone>true</clone>
    </permissions>
  </template>
</action>

```

POST /ovirt-engine/api/templates

```

<template>
  <vm id="...">
    <permissions>
      <clone>true</clone>
    </permissions>
  </vm>
</template>

```

现在，这已被删除，并被新的 `clone_permissions` 查询参数替代：

POST /ovirt-engine/api/vms?clone_permissions=true

```
<vm>
  <template id="..."/>
</vm>
```

POST /ovirt-engine/api/templates?clone_permissions=true

```
<template>
  <vm id="..."/>
</template>
```

B.1.36. 重新命名随机数生成器 源 元素

用来使用一组源元素来报告随机数生成器 源，以一个元素括起，名称应反映其用途。例如，以前报告的集群的随机数字生成器源如下：

GET /ovirt-engine/api/clusters/123

```
<cluster>
  ...
  <required_rng_sources>
    <source>random</source>
  </required_rng_sources>
  ...
</cluster>
```

以及一个用于报告的主机的随机数字生成器源，如下所示：

GET /ovirt-engine/api/hosts/123

```
<host>
  ...
  <hardware_information>
    <supported_rng_sources>
      <source>random</source>
    </supported_rng_sources>
  </hardware_information>
  ...
</host>
```

这与 **API** 的其余部分不一致，其中集合用一个名称括起，以单数形式按相同名称括起元素。这个问题现已解决。现在会报告所需的随机数生成器源，如下所示：

GET /ovirt-engine/api/clusters/123

```

<cluster>
  <required_rng_sources>
    <required_rng_source>random</required_rng_source>
  </required_rng_sources>
  ...
</cluster>

```

主机支持的随机数生成器源将报告如下：

GET /ovirt-engine/api/hosts/123

```

<host>
  ...
  <hardware_information>
    <supported_rng_sources>
      <supported_rng_source>random</supported_rng_source>
    </supported_rng_sources>
  </hardware_information>
  ...
</host>

```

注意 `required_rng_source` 和 `supported_rng_source` 的使用，而不是只使用 `source`。

B.1.37. 删除了中间 标签.parent 元素

关系建立标签，过去使用非终端父标签表示的父标签，后者包含另一个标签元素：

```

<tag>
  <name>mytag</name>
  <parent>
    <tag id="..." href="..." />
  </parent>
</tag>

```

这个结构已被简化，现在只使用一个父元素：

```

<tag>
  <name>mytag</name>
  <parent id="..." href="..." />
</tag>

```

B.1.38. 删除调度内置名称和阈值

在过去，集群的调度策略的规格基于内置名称和阈值。例如，使用均匀分布式调度策略的集群被表示如下：

```
<cluster>
  <name>mycluster</name>
  <scheduling_policy>
    <policy>evenly_distributed</policy>
    <thresholds high="80" duration="120"/>
  </scheduling_policy>
  ...
</cluster>
```

这种机制被一个顶层 `/schedulingpolicies` 集合替代，在这里可以使用任意名称和属性定义调度策略。例如，相同的调度策略在顶级集合中被表示如下：

```
<scheduling_policy>
  <name>evenly_distributed</name>
  <properties>
    <property>
      <name>CpuOverCommitDurationMinutes</name>
      <value>2</value>
    </property>
    <property>
      <name>HighUtilization</name>
      <value>80</value>
    </property>
  </properties>
</scheduling_policy>
```

集群的表示法引用调度策略及其标识符：

```
<cluster>
  <name>mycluster</name>
  <scheduling_policy id="...">
  ...
</cluster>
```

为保持向后兼容性，旧策略和阈值元素得以保留。嵌入在集群中的调度策略表示也被保留。所有这些操作现已被完全删除，因此在检索、创建或更新集群时引用调度策略的唯一方法是使用其标识符引用现有的集群。例如，当检索集群时，只会填充 `id`（和 `href`）：

```
GET /ovirt-engine/api/clusters/123
```

```
<cluster>
```

```
...
<scheduling_policy id="..." href="..."/>
...
</cluster>
```

在创建或更新集群时，仅接受 `id`。

B.1.39. 删除了 `bricks.replica_count` 和 `bricks.stripe_count` 元素

这些元素实际上不是 `brick` 集合表示的一部分，而是用于添加和删除 `brick` 的操作参数。现在删除了它们，并被新的 `replica_count` 和 `stripe_count` 参数替代：

```
POST .../bricks?replica_count=3&stripe_count=2
```

```
DELETE .../bricks?replica_count=3
```

B.1.40. 将统计类型属性重命名为 `kind`

用于使用表示统计类型的类型元素（计量、计数器等）以及表示值类型（整数、字符串等）的 `type` 属性来表示统计：

```
<statistic>
  <type>GAUGE</type>
  <values type="INTEGER">
    <value>...</value>
    <value>...</value>
  ...
</values>
</statistic>
```

为避免对第二种事物都使用类型概念，现在 `kind` 和 `type` 都是元素：

```
<statistic>
  <kind>gauge</kind>
  <type>integer</type>
  <values>
    <value>...</value>
    <value>...</value>
  ...
</values>
</statistic>
```

B.1.41. 使用多个 `vcpu_pins.vcpu_pin`，而不是多个 `vcpu_pin`

过去，将虚拟机的虚拟 CPU 固定指定为物理 CPU 固定的方法是使用多个 `vcpu_pin` 元素：

```
<vm>
  <cpu>
    <cpu_tune>
      <vcpu_pin>...</vcpu_pin>
      <vcpu_pin>...</vcpu_pin>
      ...
    </cpu_tune>
  </cpu>
</vm>
```

为了遵循 API 其它部分中的常见做法，它已改为使用 `wrapper` 元素，本例中为 `vcpu_pins`：

```
<vm>
  <cpu>
    <cpu_tune>
      <vcpu_pins>
        <vcpu_pin>...</vcpu_pin>
        <vcpu_pin>...</vcpu_pin>
        ...
      </vcpu_pins>
    </cpu_tune>
  </cpu>
</vm>
```

B.1.42. 使用 `force` 参数强制删除数据中心

删除数据中心的操作支持 `force` 参数。要使用它来支持可选的 `action` 参数，请执行 `DELETE` 操作：

```
DELETE /ovirt-engine/api/datacenters/123
```

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

此可选的 `action` 参数已被一个可选参数替代：

```
DELETE /ovirt-engine/api/datacenters/123?force=true
```

B.1.43. 使用 `force` 参数强制删除主机

删除主机的操作支持 `force` 参数。要使用它来支持可选的 `action` 参数，请执行 `DELETE` 操作：

```
DELETE /ovirt-engine/api/host/123
```

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

此可选的 `action` 参数已被一个可选参数替代：

```
DELETE /ovirt-engine/api/host/123?force=true
```

B.1.44. 使用参数强制删除存储域

删除存储域的操作支持 `强制`、`销毁`和`主机` 参数。这些参数使用作为正文的存储域表示给 `DELETE` 方法：

```
DELETE /ovirt-engine/api/storagedomains/123
```

```
<storage_domain>
  <force>...</force>
  <destroy>...</destroy>
  <host id="...">
    <name>...</name>
  </host>
</storage_domain>
```

这个问题存在问题，因为 `HTTP DELETE` 参数不应具有正文，并且存储域的代表形式不应包含存储域属性之外的内容，而是包含操作的参数。

`force`、`delete` 和 `host` 属性已被等同的参数替代，操作现在不接受正文。例如，现在使用 `force` 参数删除存储域的正确方法是：

```
DELETE /ovirt-engine/api/storagedomain/123?host=myhost&force=true
```

使用 `destroy` 参数删除：

```
DELETE /ovirt-engine/api/storagedomain/123?host=myhost&destroy=true
```

B.1.45. 使用主机参数删除存储服务器连接

删除存储服务器连接的操作支持 `主机` 参数。要使用它来支持可选的 `action` 参数，请使用 `DELETE` 方法：

DELETE /ovirt-engine/api/storageconnections/123

```
<action>
  <host id="...">
    <name>...</name>
  </host>
</action>
```

此可选的 `action` 参数已被一个可选参数替代：

DELETE /ovirt-engine/api/storageconnections/123?host=myhost**B.1.46. 使用 `force` 和 `storage_domain` 参数来删除模板磁盘**

删除模板磁盘的操作支持 `force` 和 `storage_domain` 参数。要使用它，可使用 `DELETE` 方法来支持可选的 `action` 参数：

DELETE /ovirt-engine/api/templates/123/disks/456

```
<action>
  <force>...</force>
  <storage_domain id="..."/>
</action>
```

在 API 版本 4 中，此操作被移到新的 磁盘附加集中，请求正文已被查询参数 `force` 和 `storage_domain` 替代：

DELETE /ovirt-engine/api/templates/123/disksattachments/456?force=true**DELETE /ovirt-engine/api/templates/123/disksattachments/456?storage_domain=123****B.1.47. 不要通过 VM 磁盘 API 删除磁盘**

通过删除 `/vms/123/disks/456` 来删除实体意味着删除虚拟机与磁盘之间的关系 - 例如，此操作应该仅将磁盘从虚拟机分离。此操作无法从系统完全删除磁盘，这容易出现用户错误，并产生无法更正的后果。要删除磁盘，请使用 `/disk/456` API：

DELETE /ovirt-engine/api/disks/456**B.1.48. 使用 强制 查询参数强制删除虚拟机**

删除虚拟机的操作支持 `force` 参数。要使用它来支持可选的 `action` 参数，请使用 `DELETE` 方法：

```
DELETE /ovirt-engine/api/vms/123
```

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

此可选的 `action` 参数已被一个可选的查询参数替代：

```
DELETE /ovirt-engine/api/vms/123?force=true
```

B.1.49. 使用 `POST` 而不是 `DELETE` 来删除多个 `brick`

删除多个 `Gluster brick` 的操作是使用 `DELETE` 方法实施的，并将 `brick` 列表作为请求的正文传递：

```
DELETE /ovirt-engine/api/clusters/123/glustervolumes/456/bricks
```

```
<bricks>
  <bricks id="...">
  <bricks id="...">
  ...
</bricks>
```

这有问题，因为 `DELETE` 方法不应该有正文，因此它已被使用 `POST` 方法的新删除操作替代：

```
POST /ovirt-engine/api/clusters/123/glustervolumes/456/bricks/remove
```

```
<bricks>
  <bricks id="...">
  <bricks id="...">
  ...
</bricks>
```

B.1.50. 删除 `scheduling_policy.policy` 元素

该元素保留下来以便向后兼容。改为使用 `scheduling_policy.name`。

```
POST /ovirt-engine/api/schedulingpolicies
```

```
<scheduling_policy>
```

```
...
<name>policy_name</name>
...
</scheduling_policy>
```

PUT /ovirt-engine/api/schedulingpolicies/123

```
<scheduling_policy>
...
<name>policy_name</name>
...
</scheduling_policy>
```

B.1.51. 添加了 snapshot.snapshot_type

API 正在逐渐引入枚举。直到现在为止，一些将成为字符串的字段被替换为适当的枚举。其中一个字段是 `vm.type`。但是，此字段由快照继承，快照类型与 `vm` 类型不同。因此，快照实体添加了一个新字段：`snapshot.snapshot_type`。

```
<snapshot>
...
<snapshot_type>regular|active|stateless|preview</snapshot_type>
...
</snapshot>
```

B.1.52. 从虚拟机中删除了移动操作

虚拟机实体已弃用的移动操作已被删除。相反，您可以移动内置磁盘。

B.1.53. 将 report_configurations.in_sync 移到 network_attachment

在 API 的版本 3 中，XML 模式类型 `ReportedConfigurations` 具有 `in_sync` 属性：

```
<network_attachment>
  <reported_configurations>
    <in_sync>true</in_sync>
    <reported_configuration>
      ...
    </reported_configuration>
    ...
  </reported_configurations>
</network_attachment>
```

在 API 版本 4 使用的规范机制中，这无法表达，因为列表类型（报告配置列表）不能有属性。为了代表它，属性已被移到括起的 `network_attachment` 中：


```

<network_attachment>
  <in_sync>true</in_sync>
  <reported_configurations>
    <reported_configuration>
      ...
    </reported_configuration>
    ...
  </reported_configurations>
</network_attachment>

```

B.1.54. 使用集群级别替换 的功能

顶级 功能 集合已被新的 集群级别集合 取代。这个新集合将包含模型中不可用的信息，如每个集群级别的 CPU 类型列表：

GET /ovirt-engine/api/clusterlevels

这将返回一个 **ClusterLevel** 对象列表，其中包含系统支持的所有集群级别的详情：

```

<cluster_levels>
  <cluster_level id="3.6" href="/clusterlevels/3.6">
    <cpu_types>
      <cpu_type>
        <name>Intel Nehalem Family</name>
        <level>2</level>
        <architecture>x86_64</architecture>
      </cpu_type>
      ...
    </cpu_types>
    ...
  </cluster_level>
</cluster_levels>

```

每个特定集群级别都有自己的子资源，由版本本身标识：

GET /ovirt-engine/api/clusterlevels/3.6

这将返回该版本的详情：

```

<cluster_level id="3.6" href="/clusterlevels/3.6">
  <cpu_types>
    <cpu_type>
      <name>Intel Nehalem Family</name>

```

```

<level>2</level>
<architecture>x86_64</architecture>
</cpu_type>
...
</cpu_types>
...
</cluster_level>

```

B.1.55. 使用 磁盘 附加替换的磁盘

在 API 虚拟机的版本 3 中，模板有一个 磁盘 集合，其中包含附加的磁盘的所有信息。在 API 版本 4 中，这些 磁盘 集合已删除，并替换为一个新的 磁盘附加集合，该集合将仅包含对磁盘的引用以及磁盘与其附加的虚拟机或模板之间的关系特定属性：接口和可启动。

要查找附加到虚拟机的磁盘，例如，发送如下请求：

```
GET /ovirt-engine/api/vms/123/diskattachments
```

这会返回类似如下的响应：

```

<disk_attachments>
  <disk_attachment href="/vms/123/diskattachments/456" id="456">
    <bootable>false</bootable>
    <interface>virtio</interface>
    <disk href="/disks/456" id="456"/>
    <vm href="/vms/123" id="123"/>
  </disk_attachment>
  ...
</disk_attachments>

```

要查找磁盘的其余详细信息，请使用提供的链接。

在虚拟机或模板中添加磁盘也使用新的 `disk_attachment` 元素：如下请求：

```
POST /ovirt-engine/api/vms/123/diskattachments
```

如果磁盘不存在且您要创建它，请使用以下正文：

```

<disk_attachment>
  <bootable>false</bootable>
  <interface>virtio</interface>
  <disk>

```

```

<description>My disk</description>
<format>cow</format>
<name>mydisk</name>
<provisioned_size>1048576</provisioned_size>
<storage_domains>
  <storage_domain>
    <name>mydata</name>
  </storage_domain>
</storage_domains>
</disk>
</disk_attachment>

```

或者，如果磁盘已存在，则使用以下正文，且您只想将其附加到虚拟机：

```

<disk_attachment>
  <bootable>false</bootable>
  <interface>virtio</interface>
  <disk id="456"/>
</disk_attachment>

```

考虑 `vm.disks` 和 `template.disks attributes` 具有所有使用的 `disk_attachments`。例如，在创建模板时，`vm.disks` 元素用于指示在其中创建模板磁盘的存储域。这个用法已被 `vm.disk_attachments` 替代，因此，在特定存储域中创建带有磁盘的模板的请求现在类似如下：

```

<template>
  <name>mytemplate</name>
  <vm id="123">
    <disk_attachments>
      <disk_attachment>
        <disk id="456">
          <storage_domains>
            <storage_domain id="789"/>
          </storage_domains>
        </disk>
      </disk_attachment>
      ...
    </disk_attachments>
  </vm>
</template>

```

B.1.56. 使用 `iscsi_targets` 元素发现未注册的存储

在 API 版本 3 中，使用多个 `iscsi_target` 元素来发现用于接收 iSCSI 目标列表的未注册存储域的操作：

```
POST /ovirt-engine/api/hosts/123/unregisteredstoragedomaindiscover
```

```
<action>
```

```
<iscsi>
  <address>myiscsiserver</address>
</iscsi>
<iscsi_target>iqn.2016-07.com.example:mytarget1</iscsi_target>
<iscsi_target>iqn.2016-07.com.example:mytarget2</iscsi_target>
</action>
```

在 API 版本 4 中，所有重复元素（本例中为 `iscsi_target`）都使用另一个元素 `iscsi_targets` 进行打包，以防出现 `iscsi_targets`。因此，同一请求现在应类似如下：

POST /ovirt-engine/api/hosts/123/unregisteredstoragedomaindiscover

```
<action>
  <iscsi>
    <address>myiscsiserver</address>
  </iscsi>
  <iscsi_targets>
    <iscsi_target>iqn.2016-07.com.example:mytarget1</iscsi_target>
    <iscsi_target>iqn.2016-07.com.example:mytarget2</iscsi_target>
  </iscsi_targets>
</action>
```

B.2. ENGINE 版本 4.5 中的更改

B.2.1. 由 Managed Block Storage 取代的 OpenStack 卷(Cinder)集成。

附录 C. 法律通知

Copyright © 2022 Red Hat, Inc.

Licensed under the ([Creative Commons Attribution–ShareAlike 4.0 International License](#)).从 ([oVirt Project](#))的文档衍生而来。如果您发布本文档或对其进行改编，您必须提供原始版本的 URL。

修改后的版本必须删除所有红帽商标。

Red Hat、Red Hat Enterprise Linux、Red Hat 商标、Shadowman 商标、JBoss、OpenShift、Fedora、Infinity 商标以及 RHCE 都是在美国及其他国家的注册商标。

Linux® 是 Linus Torvalds 在美国和其他国家/地区的注册商标。

Java® 是 Oracle 和/或其附属公司的注册商标。

XFS® 是 Silicon Graphics International Corp. 或其子公司在美国和/或其他国家的商标。

MySQL® 是 MySQL AB 在美国、欧盟和其他国家/地区的注册商标。

Node.js® 是 Joyent 的官方商标。Red Hat Software Collections 与官方 Joyent Node.js 开源或商业项目没有正式关联或被正式认可。

The OpenStack® Word Mark 和 OpenStack 标识是 OpenStack Foundation 在美国及其他国家的注册商标/服务标记或商标/服务标记，可根据 OpenStack Foundation 授权使用。我们不附属于 OpenStack Foundation 或 OpenStack 社区。

所有其他商标均由其各自所有者所有。