文件编号：DSP2027
日期：2012 年 9 月 12 日
版本：1.0.1

云基础设施管理接口 (CIMI) 入门

文件类型：白皮书

文件状态：DMTF 信息

文件语言：简体中文

DMTF 是一家由行业会员组成的非盈利性协会，致力于促进企业和系统的管理和可交互操作性。会员和非会员可以在注明出处的前提下复制 DMTF 规范和文件。由于 DMTF 规范可能随时修订，应始终注意具体的版本和发布日期。

本标准或提议标准某些要素的实施会受制于第三方专利权，包括临时专利权（本规范中统称为“专利权”）。DMTF 不向用户做任何有关存在该等权利的陈述，不负责认可、披露或识别任何或所有该等第三方专利权、所有者或申索人，对于该等权利、所有者或申索人的不完整或不准确识别或披露不承担任何责任。不论该等实施是否可以预见，DMTF 不以任何方式也不在任何情况下对任何实施该标准的一方承担任何责任，也不对任何专利所有者或申索人承担任何责任，如果一个标准在发布后被撤销或修改，DMTF 也不承担由此产生的费用或损失，实施本标准的任何一方应使 DMTF 免于专利所有者对于该等实施提出的任何和所有侵权主张。

登录 http://www.dmtf.org/about/policies/disclosures.php 了解第三方持有的已通知 DMTF 他们认为可能与 DMTF 标准的实施有关或有影响的专利方面的信息。

目录

前言

本文件描述了 CIMI 协议的各种常见使用场景。

鸣谢：

作者感谢下列人士对本文件所作的贡献。

编辑人：

- Davis, Doug - IBM
- Pilz, Gilbert – Oracle
- Zhang, Aaron - Huawei

贡献人：

- Ali, Ghazanfar - ZTE Corporation
- Andreou, Marios - Red Hat
- Bankston, Keith - Microsoft Corporation
- Bumpus, Winston - VMware Inc.
- Burkhart, Nathan - Microsoft Corporation
- Carlson, Mark - Oracle
- Carter, Steve - Novell
- Chu, Junsheng - ZTE Corporation
- Cohen, Josh - Microsoft Corporation
- Coleman, Derek - Hewlett-Packard Company
- Crandall, John - Brocade Communications Systems
- Davis, Doug - IBM
- Davis, Jim - WBEM Solutions
- de la Iglesia, Fernando - Telefonica
- Dempo, Hiroshi - NEC Corporation
- Durand, Jacques - Fujitsu
- Edery, Yigal - Microsoft Corporation
- Ericson, George - EMC
- Evans, Colleen - Microsoft Corporation
- Floeren, Norbert - Ericsson AB
- Freund, Robert - Hitachi, Ltd.
- Galan, Fermin - Telefonica
- Gopalan, Krishnan - Microsoft Corporation
- Iwasa, Kazunori - Fujitsu

- Johnson, Mark - IBM
- Khasnabish, Bhumip - ZTE Corporation
- Kowalski, Vincent - BMC Software
- Krishnaswamy, Ruby - France Telecom Group
- Lamers, Lawrence - VMware Inc.
- Lipton, Paul - CA Technologies
- Livingston, James - NEC Corporation
- Lubsey, Vince - Virtustream Inc.
- Lutterkort, David - Red Hat
- Maciel, Fred - Hitachi, Ltd.
- Maier, Andreas - IBM
- Malhotra, Ashok - Oracle
- Mischkinsky, Jeff - Oracle
- Molina, Jesus – Fujitsu
- Moscovich, Efraim - CA Technologies
- Murray, Bryan - Hewlett-Packard Company
- Neely, Steven - Cisco
- Ogawa, Ryuichi - NEC Corporation
- Parchem, John - Microsoft Corporation
- Pardikar, Shishir - Citrix Systems Inc.
- Penalvo, Miguel - Telefonica
- Pilz, Gilbert - Oracle
- Polo, Alvaro - Telefonica
- Ronco, Enrico - Telecom Italia
- Rossini, Federico - Telecom Italia
- Rutkowski, Matthew - IBM
- Rutt, Tom - Fujitsu
- Shah, Hemal - Broadcom
- Shah, Nihar - Microsoft Corporation
- Sill, Alan - Texas Tech University
- Song, Zhexuan - Huawei
- Waschke, Marvin - CA Technologies
- Wells, Eric - Hitachi, Ltd.
- Wheeler, Jeff - Huawei
- Wiggers, Maarten - Fujitsu
- Winkler, Steve - SAP AG
- Yu, Jack - Oracle
- Zhang, Aaron – Huawei
- Zhang, HengLiang - Huawei

概述

为简化问题，在以下各种场景中，云提供商仅支持展示场景重点特性所需的最低特性。因此，在查询支持的特性列表的云入口点 (CEP) 时，查询结果将按场景进行自定义。此外，以下假设云入口点的 URI 为 http://example.com/CEP。

场景 1：创建新机器

本场景将创建一台新机器。系统根据提供商提供的现有配置和镜像来配置新机器，并创建新的凭据资源（用户名和密码）。

第 1 步：检索 CEP

CEP 提供该云中可用资源的链接。检索 CEP，找到各资源的 URL：

```
GET / HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
    "id": "http://example.com/CEP",
    "baseURI": "http://example.com/",
    "resourceMetadata": { "href": "http://example.com/resourceMetadata" },
    "machines": { "href": "http://example.com/machines" },
    "machineConfigs": { "href": "http://example.com/machineConfigs" },
    "machineImages": { "href": "http://example.com/machineImages" },
    "credentials": { "href": "http://example.com/credentials" }
}
```

第 2 步：检索机器镜像列表

在创建新机器之前，应首先确定要预装的操作系统和/或软件。机器镜像集包含了该云提供的所有机器镜像。注意，有些镜像是云预定义的，有些可能是由用户创建的：

```
GET /machineImages HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImageCollection",
    "id": "http://example.com/machineImages",
    "machineImages": [
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
            "id": "http://example.com/images/WinXP-SP2",
            "name": "WinXP SP2",
            "description": "Windows XP with Service Pack 2",
            "created": "2012-01-01T12:00:00Z",
            "updated": "2012-01-01T12:00:00Z",
            "imageLocation": { "href": "http://example.com/data/8934322" }
    },
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
        "id": "http://example.com/images/Win7",
        "name": "Windows 7",
        "description": "Windows 7",
        "created": "2012-01-01T12:00:00Z",
        "updated": "2012-01-01T12:00:00Z",
        "imageLocation": { "href": "http://example.com/data/8934344" }
    },
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",

        "id": "http://example.com/images/Linux-SUSE",
                "name": "Linux SUSE",
                "description": "Linux SUSE",
                "created": "2012-01-01T12:00:00Z",
                "updated": "2012-01-01T12:00:00Z",
                "imageLocation": { "href": "http://example.com/data/8934311" }
            }
    ]
}
```

第 2.1 步：选择机器镜像

查看各机器镜像，找到符合要求的镜像。第一个镜像符合要求，随后会使用该镜像。

需要注意的是，如果你事先知道自己要使用列表中的第一个镜像，只想在前面的查询结果中看到该资源，则可以执行以下步骤：

```
GET /machineImages?$last=1 HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImageCollection",
    "id": "http://example.com/machineImages",
    "machineImages": [
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
            "id": "http://example.com/images/WinXP-SP2",
            "name": "WinXP SP2",
            "description": "Windows XP with Service Pack 2",
            "created": "2012-01-01T12:00:00Z",
            "updated": "2012-01-01T12:00:00Z",
            "imageLocation": { "href": "http://example.com/data/8934322" }
        }
    ]
}
```

注意，在以上示例中，“1”为是默认值，因此无需指定 $first=1。

第 3 步：检索机器配置列表

确定将机器镜像安装在哪种虚拟硬件上。与确定机器镜像类型一样，首先查询可用镜像配置列表：

```
GET /machineConfigs HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json
{ "resourceURI":
        "http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection",
    "id": "http://example.com/machineConfigs",
    "machineConfigurations": [
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
            "id": "http://example.com/configs/tiny",
            "name": "tiny",
            "description": "a teenie tiny one",
            "created": "2012-01-01T12:00:00Z",
            "updated": "2012-01-01T12:00:00Z",
            "cpu": 1,
            "memory": 4000000,
            "disks" : [
                { "capacity": 50000000 }
            ]
        },

        { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
            "id": "http://example.com/configs/small",
            "name": "small",
            "description": "a small sized one",
            "created": "2012-01-01T12:00:00Z",
            "updated": "2012-01-01T12:00:00Z",
            "cpu": 1,
            "memory": 8000000,
            "disks" : [
                { "capacity": 500000000 }
            ]
        },

        { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
            "id": "http://example.com/configs/medium",
            "name": "medium",
            "description": "a medium one",
            "created": "2012-01-01T12:00:00Z",
            "updated": "2012-01-01T12:00:00Z",
            "cpu": 1,
            "memory": 16000000,
            "disks" : [
                { "capacity": 1000000000 },
                { "capacity": 1000000000 },
            ]
          }
        ]
}
```

第 3.1 步：选择机器配置

查看返回的列表，选择满足要求的机器配置。第一个配置符合要求，随后会使用该配置。

第 4 步：创建新凭据资源

如想在新机器上使用自己的用户名和密码，就要创建新的凭据资源。该过程通过 POST 操作来完成，但首先要检索凭据集，以确定将新的凭据资源 POST 到什么位置：

```
GET /credentials HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialCollection",
        "id": "http://example.com/credentials",
        "operations": [ { "rel": "add", "href": "http://example.com/credentials" }]
}
```

注意，此时的环境中还没有凭据资源。在创建新的凭据资源之前，必须先找到该云提供商的凭据资源的扩展属性。默认情况下，CIMI 规范并没有定义如何对新机器的初始用户进行规定；各云提供商可以自由决定如何提供该信息。用户可以通过查询凭证资源元数据资源来获取该

信息。要查看该资源，应首先查看 ResourceMetadata 集中该提供商对凭据资源的描述。首先，从云入口点引用的 URI 检索 ResourceMetadata 集：

```
GET /resourceMetadata HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadataCollection",
    "id": "http://example.com/resourceMetadata",
    "resourceMetadatas": [
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
            "id": "http://example.com/resources/Credential",
            "typeURI": "http://schemas.dmtf.org/cimi/1/Credential",
            "name": "Credential",
            "attributes": [
                { "name": "userID", "namespace": "http://example.com",
                    "type": "string", "required": "true" },
                { "name": "password", "namespace": "http://example.com",
                    "type": "string", "required": "true" }
            ]
          }
        ]
}
```

在“typeURI”为“http://schemas.dmtf.org/cimi/1/Credential”的集合中对 resourceMetadata 输入值列表进行迭代。找到之后，可以查看提供商已经添加到凭据资源中的扩展，这表明已经对凭证资源进行了扩展，并且凭证资源必须包含“userID”和“password”这两个属性。这两个属性均为“string”类型。

现在用 POST 操作创建新的凭据资源：

```
POST /credentials HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialCreate",
    "name": "Default",
    "description": "My Default User",
    "credentialTemplate": {
        "userID": "JoeSmith",
        "password": "letmein"
    }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/creds/12345
```

注意：“userID”和“password”属性是通过 Credential ResourceMetadata 得到的，而“名称”和“描述”属性是所有资源的共同属性。在下一场景中，我们将介绍客户如何确定“userID”和“password”属性是否匹配该镜像类型和云提供商。

第 5 步：创建新机器

检索机器集，确定将新机器 POST 到什么位置：

```
GET /machines HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCollection",
        "id": "http://example.com/machines",
        "operations": [ { "rel": "add", "href": "http://example.com/machines" } ]
}
```

现在创建新机器：

```
POST /machines HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate",
        "name": "myMachine1",
        "description": "My very first machine",
        "machineTemplate": {
                "machineConfig": { "href": " http://example.com/configs/tiny" },
                "machineImage": { "href": " http://example.com/images/WinXP-SP2" },
                "credential": { "href": "http://example.com/creds/12345" }
        }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/machines/843752
```

第 6 步：查询新机器

检索新机器，获取完整描述：

```
GET /machines/843752 HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
        "id": "http://example.com/machines/843752",
        "name": "myMachine1",
        "description": "My very first machine",
        "created": "2012-08-15T12:15:00Z",
        "updated": "2012-08-15T12:15:00Z",
        "state": "STOPPED",
        "cpu": 1,
        "memory": 4000000,
        "disks" : { "href": "http://example.com/machines/843752/disks",
        "networkInterfaces": { "href": "http://example.com/machines/843752/NIs",
        "operations": [
                { "rel": "edit", "href": "http://example.com/machines/843752" },
                { "rel": "delete", "href": "http://example.com/machines/843752" },
                { "rel": "http://schemas.dmtf.org/cimi/1/action/start",
                "href": "http://example.com/machines/843752" }
        ]
}
```

注意，机器的“状态”属性是“STOPPED”，这是新机器的初始状态。

第 7 步：启动新机器

机器描述的“操作”阵列中出现“启动”操作，这不仅指定了 POST“启动”操作的目的 URI，还说明可以立即执行该操作。

```
POST /machines/843752 HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
    "action": "http://schemas.dmtf.org/cimi/1/action/start"
}
```

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

第 7.1 步：查询机器，验证其是否已经启动

再次查询机器，验证其是否已启动：

```
GET /machines/843752 HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
    "id": "http://example.com/machines/843752",
    "name": "myMachine1",
    "description": "My very first machine",
    "created": "2012-08-15T12:15:00Z",
    "updated": "2012-08-15T12:15:00Z",
    "state": "STARTED",
    "cpu": 1,
    "memory": 4000000,
    "disks" : { "href": "http://example.com/machines/843752/disks",
    "networkInterfaces": { "href": "http://example.com/machines/843752/NIs",
    "operations": [
        { "rel": "edit", "href": "http://example.com/machines/843752" },
        { "rel": "delete", "href": "http://example.com/machines/843752" },
        { "rel": "http://schemas.dmtf.org/cimi/1/action/stop",
            "href": "http://example.com/machines/843752" }
    ]
}
```

注意，机器的“状态”属性是“STARTED”，“操作”阵列不再指示“启动”操作可用；而是“关闭”操作可用。

第 8 步：关闭机器

可通过“关闭”操作的 URL 来关闭机器：

```
POST /machines/843752 HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
    "action": "http://schemas.dmtf.org/cimi/1/action/stop"
}
```

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

第9步：更新机器属性

使用“编辑”操作URL中的PUT操作，可以更新机器的部分属性，如“名称”和“描述”：

```
PUT /machines/843752?$select=name,description HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
    "name" : "Cool Demo #1"
}
```

```
HTTP/1.1 200 OK

{ "name" : "Cool Demo #1" }
```

注意，“编辑”操作的URL已修改，指示要更新的属性；只有这些属性会被更新。因为URL包含“描述”属性，而HTTP请求正文中不包含该属性，因此该属性被删除。

场景 2：在机器中添加新存储卷

该场景将创建新存储卷，并连接到一台现有机器。

第 1 步：获取机器的 URL

机器：

```
http://example.com/machines/843752
```

第 2 步：检索 CEP

CEP 提供该云中可用资源的链接。检索 CEP，找到各资源的 URL：

```
GET / HTTP/1.1 HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
    "id": "http://example.com/CEP",
    "baseURI": "http://example.com/",
    "machines": { "href": "http://example.com/machines" },
    "machineConfigs": { "href": "http://example.com/machineConfigs" },
    "machineImages": { "href": "http://example.com/machineImages" },
    "credentials": { "href": "http://example.com/credentials" },
    "volumes": { "href": "http://example.com/volumes" },
    "volumeConfigs": { "href": "http://example.com/vConfigs" }
}
```

第 3 步：获取 VolumeConfigurations 列表，以确定使用哪个 VolumeConfigurations

在创建新存储卷时，需要确定要创建的卷的类型，如大小、格式等。存储卷配置集包含了该云提供的所有预定义卷配置：

```
GET /vConfigs HTTP/1.1
```

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

{ "resourceURI":
        "http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection",
    "id": "http://example.com/volumeConfigs",
    "volumeConfigurations": [
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
            "id": "http://example.com/vConfigs/small",
            "name": "Small",
            "description": "A pretty small one",
            "created": "2012-08-15T12:15:00Z",
            "updated": "2012-08-15T12:15:00Z",
            "type": "http://schemas.dmtf.org/cimi/1/mapped",
            "format": "NTFS",
            "capacity": 60000000
        },
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
            "id": "http://example.com/vConfigs/medium",
            "name": "Small",
            "description": "A medium sized one",
            "created": "2012-08-15T12:15:00Z",
            "updated": "2012-08-15T12:15:00Z",
            "type": "http://schemas.dmtf.org/cimi/1/mapped",
            "format": "NTFS",
            "capacity": 500000000
        },
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
            "id": "http://example.com/vConfigs/large",
            "name": "Large",
            "description": "A large one",
            "created": "2012-08-15T12:15:00Z",
            "updated": "2012-08-15T12:15:00Z",
            "type": "http://schemas.dmtf.org/cimi/1/mapped",
            "format": "NTFS",
            "capacity": 1000000000
        }
    ]
}
```

第 3.1 步：选择卷配置

查看各卷配置，找到符合要求的卷配置。第一个卷配置符合要求，随后将使用该卷配置。

第 4 步：创建新卷

检索卷集，确定将新卷 POST 到什么位置：

```
GET /volumes HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeCollection",
        "id": "http://example.com/volumes",
        "operations": [ { "rel": "add", "href": "http://example.com/volumes" } ]
}
```

现在创建新卷：

```
POST /volumes HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeCreate",
        "name": "myVolume1",
        "description": "My first new volume",
        "volumeTemplate": {
                "volumeConfig": { "href": "http://example.com/vConfigs/small" }
        }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/volumes/35782
```

第 5 步：检索卷信息

要验证你创建并连接到机器的卷是否正确，请遵循上述步骤返回的参考：

```
GET /volumes/35782 HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
      "id": "http://example.com/volumes/35782",
      "name": "myVolume1",
      "created": "2012-09-15T12:15:00Z",
      "updated": "2012-09-15T12:15:00Z",
      "description": "My first new volume",
      "type": "http://schemas.dmtf.org/cimi/1/mapped",
      "capacity": 60000000,
      "operations": [
            { "rel": "edit", "href": "http://example.com/volumes/35782" },
            { "rel": "delete", "href": "http://example.com/volumes/35782" }
      ]
}
```

第 6 步：检索机器的卷集

在将新卷连接到机器之前，首先要检索机器的卷集，以确定请求发送的目的地。首先检索机器，获取该集的引用：

```
GET /machines/843752 HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
      "id": "http://example.com/machines/843752",
      "name": "myMachine1",
      "description": "My very first machine",
      "created": "2012-08-15T12:15:00Z",
      "updated": "2012-08-15T12:15:00Z",
      "state": "STARTED",
      "cpu": 1,
      "memory": 4000000,
      "disks" : { "href": "http://example.com/machines/843752/disks",
      "volumes": { "href": "http://example.com/machines/843752/volumes" },
      "networkInterfaces": { "href": "http://example.com/machines/843752/NIs" },
      "operations": [
            { "rel": "edit", "href": "http://example.com/machines/843752" },
            { "rel": "delete", "href": "http://example.com/machines/843752" },
            { "rel": "http://schemas.dmtf.org/cimi/1/action/stop",
                  "href": "http://example.com/machines/843752" }
      ]
}
```

注意，由于上一场景的范围有限，因此不存在“卷”属性；但本场景（及样本提供商的特性）的范围已扩展，可支持卷，因此存在“卷”属性。

现在检索卷集：

```
GET /machines/843752/volumes HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",
    "id": "http://example.com/machines/843752/volumes",
    "operations": [
        { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
    ]
}
```

注意，目前没有与该机器连接的卷。

另一优化方案是，通过$expand 查询参数同时检索原始机器与该卷集：

```
GET /machines/843752?$expand=volumes HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
    "id": "http://example.com/machines/843752",
    "name": "myMachine1",
    "description": "My very first machine",
    "created": "2012-08-15T12:15:00Z",
    "updated": "2012-08-15T12:15:00Z",
    "state": "STARTED",
    "cpu": 1,
    "memory": 4000000,
    "disks" : { "href": "http://example.com/machines/843752/disks",
    "volumes": {
        "href": "http://example.com/machines/843752/volumes",
        "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",
        "id": "http://example.com/machines/843752/volumes",
        "operations": [
            { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
        ]
    },
    "networkInterfaces": { "href": "http://example.com/machines/843752/NIs" },
    "operations": [
        { "rel": "edit", "href": "http://example.com/machines/843752" },
        { "rel": "delete", "href": "http://example.com/machines/843752" },
        { "rel": "http://schemas.dmtf.org/cimi/1/action/stop",
            "href": "http://example.com/machines/843752" }
    ]
}
```

第 7 步：将新卷连接到机器

使用卷集的“添加”操作将卷连接到机器，并进入新的 MachineVolume 资源：

```
POST /machines/843752/volumes HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolume",
    "initialLocation": "V",
    "volume": { "href": "http://example.com/volumes/35782" }
}
```

```
HTTP/1.1 201 Created
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolume",
"id": "http://example.com/machines/843752/volumes/1",
"initialLocation": "V",
"volume": { "href": "http://example.com/volumes/35782" },
"operations": [
        { "rel": "edit", "href": "http://example.com/machines/843752/volumes/1"},
        { "rel": "delete", "href":"http://example.com/machines/843752/volumes/1"}
    ]
}
```

第 8 步：查询机器卷集以验证更新结果

检索机器的卷集，获取卷的完整列表，并用该列表验证更新是否成功：

```
GET /machines/843752/volumes HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",
    "id": "http://example.com/machines/843752/volumes",
    "machineVolumes": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumes",
        "id": "http://example.com/machines/843752/volumes/1",
        "initialLocation": "V",
        "volume": { "href": "http://example.com/volumes/35782" },
        "operations": [
            { "rel":"edit","href":"http://example.com/machines/843752/volumes/1"},
            { "rel":"delete","href":"http://example.com/machines/843752/volumes/1"}
        ]
    },
    "operations": [
        { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
    ]
}
```

场景 3：定义和使用机器模板

该场景将创建用于创建新机器的新机器模板。机器模板可以方便地保存对机器的准确描述（配置、镜像等），便于以后重复使用。当新机器的用户在技术上不是非常熟练，不了解创建机器的具体方法时，该特性尤为有用。创建机器模板通常是用于演示，或用于需使用特定机器镜像的复杂机器配置。机器模板可以保存配置信息，并易于重复使用。

为方便起见，以下示例重复使用在之前的场景中已经获取的配置信息。

第 1 步：检索 CEP

CEP 提供云中可用资源的链接。检索 CEP，找到各资源的 URL：

```
GET / HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
        "id": "http://example.com/CEP",
        "baseURI": "http://example.com/",
        "machines": { "href": "http://example.com/machines" },
        "machineTemplates": { "href": "http://example.com/machineTemplates" },
        "machineConfigs": { "href": "http://example.com/machineConfigs" },
        "machineImages": { "href": "http://example.com/machineImages" },
        "credentials": { "href": "http://example.com/credentials" }
}
```

第 2 步：创建新的机器模板

可重复使用在之前的场景中获得的 MachineConfiguration、MachineImage 和凭据资源来创建该 MachineTemplate：

MachineImage：

```
http://example.com/images/WinXP-SP2
```

MachineConfiguration：

```
http://example.com/configs/tiny
```

凭据：

```
http://example.com/creds/12345
```

在创建新的 MachineTemplate 之前，首先确定将 POST 发送到哪个 URL。可通过返回的 CEP 中的 MachineTemplate 集的 URL 来确定该位置：

```
GET /machineTemplates HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplateCollection",
        "id": "http://example.com/machineTemplates",
        "operations": [
                { "rel": "add", "href": "http://example.com/machineTemplates" }
        ]
}
```

注意，目前环境中没有 MachineTemplates。

现在创建新的 MachineTemplate 资源：

```
POST /machineTemplates HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplate",
        "name": "Demo1",
        "description": "My first demo",
        "machineConfig": { "href": "http://example.com/configs/tiny" },
        "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
        "credential": { "href": "http://example.com/creds/12345" }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/machineTemplates/82754
```

第 3 步：使用机器模板创建新机器

使用该机器模板创建新机器：

```
POST /machines HTTP/1.1
Content-Type: application/json


{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate",
        "name": "myMachine2",
        "description": "My second machine",
        "machineTemplate": { "href": "http://example.com/machineTemplates/82754" }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/machines/843799
```

场景 4：从现有卷创建新机器

该场景将创建从现有卷启动的新机器。在以下简单示例中，假设用户知道存在启动属性为“是”的卷。

第 1 步：检索 CEP

CEP 提供云中可用资源的链接。检索 CEP，找到各资源的 URL：

```
GET / HTTP/1.1
```

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

{ "resourceType": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
    "id": "http://example.com/CEP",
    "baseURI": "http://example.com/",
    "resourceMetadata": { "href": "http://example.com/resourceMetadata" },
    "machines": { "href": "http://example.com/machines" },
    "machineConfigs": { "href": "http://example.com/machineConfigs" },
    "machineImages": { "href": "http://example.com/machineImages" },
    "credentials": { "href": "http://example.com/credentials" },
    "volumes": { "href": "http://example.com/volumes" },
    "volumeConfigs": { "href": "http://example.com/vConfigs" }
}
```

第 2 步：获取卷列表，确定要使用的卷

由卷创建新机器时，需要确定使用哪个卷。卷集包含了该云提供的所有卷：

```
GET /volumes HTTP/1.1
```

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

{ "resourceType": "http://schemas.dmtf.org/cimi/1/VolumeCollection",
    "id": "http://example.com/volumes",
    "volumes": [
        { "resourceType": "http://schemas.dmtf.org/cimi/1/Volume",
            "id": "http://example.com/volumes/vol1",
            "name": "Win7-Bootable",
            "created": "2012-08-15T12:15:00Z",
            "updated": "2012-08-15T12:15:00Z",
            "description": "A bootable volume running Windows 7",
            "state": "AVAILABLE",
            "capacity": 60000000,
            "bootable": true
    },

    { "resourceType": "http://schemas.dmtf.org/cimi/1/Volume",
        "id": "http://example.com/volumes/vol2",
        "name": "Generic Volume",
        "created": "2012-08-15T12:15:00Z",
        "updated": "2012-08-15T12:15:00Z",
        "description": "A generic volume for Windows",
        "state": "AVAILABLE",
        "capacity": 60000000,
            "bootable": true
```

```
            }
      ]
}
```

第 2.1 步：选择卷

查看各卷，找到符合要求的卷。第一个卷符合要求，随后将使用该卷。

第 3 步：创建新机器

检索机器集，确定将新机器 POST 到什么位置：

```
GET /machines HTTP/1.1
```

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

{ "resourceType": "http://schemas.dmtf.org/cimi/1/MachineCollection",
      "id": "http://example.com/machines",
      "operations": [
            { "rel": "add", "href": "http://example.com/machines" }
      ]
}
```

创建新机器，并连接到可启动的卷：

```
POST /machines HTTP/1.1
Content-Type: application/json

{ "resourceType": "http://schemas.dmtf.org/cimi/1/MachineCreate",
      "name": "myMachine2",
      "description": "My second machine",
      "machineTemplate": {
            "volumes": [
                  { "initialLocation": "V",
                        "href": "http://example.com/volumes/vol1" }
            ]
      }
}
```

注意，本例中的 MachineTemplate 未指定要使用的 MachineImage 或 MachineConfiguration。为简化问题，可假设提供商已给出了 MachineImage 和 MachineConfiguration 的默认值。

```
HTTP/1.1 201 Created
Location: http://example.com/machines/852108
```

第 4 步：查询新机器

检索机器，获取新机器的完整描述：

```
GET /machines/852108 HTTP/1.1
```

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

{ "resourceType": "http://schemas.dmtf.org/cimi/1/Machine",
    "id": "http://example.com/machines/852108",
    "name": "myMachine2",
    "description": "My second machine",
    "created": "2012-03-26T10:04:00Z",
    "updated": "2012-03-26T10:04:00Z",
    "state": "STOPPED",
    "cpu": "1",
    "memory": 4000000,
    "disks" : { "href": "http://example.com/machines/852108/disks",
    "volumes": { "href": "http://example.com/machines/852108/volumes" },
    "networkInterfaces": { "href": "http://example.com/machines/852108/NIs",
    "operations": [
        { "rel": "edit", "href": "http://example.com/machines/852108" },
        { "rel": "delete", "href": "http://example.com/machines/852108" },
        { "rel": "http://schemas.dmtf.org/cimi/1/action/start",
            "href": "http://example.com/machines/852108" }
    ]
}
```

注意，机器的“状态”属性是“STOPPED”，这是新机器的初始状态。

场景 5：定义和使用系统模板

该场景将创建用于创建新机器的新系统模板。系统模板可以方便地保存对机器的准确描述（配置、镜像等），便于以后重复使用。

第 1 步：检索 CEP

CEP 提供云中可用资源的链接。检索 CEP，找到各资源的 URL：

```
GET / HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
        "id": "http://example.com/CEP",
        "baseURI": "http://example.com/",
        "systemTemplates": { "href": "http://example.com/systemTemplates" },
        "machineTemplates": { "href": "http://example.com/machineTemplates" },
        "credentialTemplates": { "href": "http://example.com/credentialTemplates"},
        "volumeTemplates": { "href": "http://example.com/volumeTemplates" }
}
```

第 2 步：创建新的系统模板。

定义系统模板，用于创建机器、与机器连接的卷和凭据资源。在配置系统模板时，需要通过数值将 MachineTemplate 引入 SystemTemplate 中，并通过引用将 VolumeTemplate 和 CredentialTemplate 引入 SystemTemplate。VolumeTemplate 和 CredentialTemplate 资源已经存在：

VolumeTemplate：

```
http://example.com/volumeTemplates/95839
```

CredentialTemplate：

```
http://example.com/credentialTemplates/72000
```

注意：也可通过数值将 VolumeTemplate 和 CredentialTemplate 引入以下 MachineTemplate 定义中。但更好的做法是，立即查看 SystemTemplate 中已引用的资源，自动创建凭据往往更加安全。

在创建新的 SystemTemplate 之前，首选要确定需要将 POST 发送到哪个 URL。可通过返回的 CEP 中的 SystemTemplate 集的 URL 来确定该位置：

```
GET /systemTemplates HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplateCollection",
        "id": "http://example.com/systemTemplates",
        "operations": [
                { "rel": "add", "href": "http://example.com/systemTemplates" }
        ]
}
```

创建新的系统模板资源：

```
POST /systemTemplates HTTP/1.1
Content-Type: application/json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
    "name": "System Demo1",
    "description": "My first system template demo",
    "componentDescriptors": [
      { "name": "MyMachine",
        "type": "http://schemas.dmtf.org/cimi/1/Machine",
        "machineTemplate":
          { "name" : "Machine in system demo",
            "description" : "Machine in system",
            "machineConfig": { "href": "http://example.com/configs/tiny" },
            "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
            "credential": { "href": "#MyCredential" },
            "volumes": [
              { "initialLocation": "/vol",
                "href": "#MyVolume"
              }
            ]
        }
    },
    { "name": "MyCredential",
        "type": "http://schemas.dmtf.org/cimi/1/Credential",
        "credentialTemplate":
            { "href": "http://example.com/credentialTemplates/72000" }
    },
    { "name": "MyVolume",
        "type": "http://schemas.dmtf.org/cimi/1/Volume",
        "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839"}
    }
  ]
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/systemTemplates/48920
```

第 3 步：使用系统模板创建新系统

使用该系统模板创建新系统：

```
POST /systems HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCreate",
    "name": "MySystem1",
    "description": "My first system",
    "systemTemplate": { "href": "http://example.com/systemTemplates/48920" }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/systems/78342
```

注意，还有一种可能，就是提供商已经决定返回任务资源的引用，而不是等待系统创建完成。在这种情况下，你可能收到以下响应，而不是上文的 201 响应：

```
HTTP/1.1 202 Accepted
```

```
CIMI-Job-URI: http://example.com/Jobs/90001

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Job",
      "id": "http://example.com/Jobs/90001",
      "name": "SystemCreationJob",
      "created": "2012-03-15T12:15:00Z",
      "updated": "2012-03-15T12:15:00Z",
      "targetResource": { "href": "http://example.com/systems"},
      "affectedResources": {
            "href": " http://example.com/systems/110001",
      },
      "action": "add",
      "status": "RUNNING",
      "progress": 30,
      "timeOfStatusChange": "2012-03-15T12:15:00Z",
      "isCancellable": "true",
      "nestedJobs": [
            { "href": "http://example.com/Jobs/90002" },
            { "href": "http://example.com/Jobs/90003" }
      ]
}
```

该响应表明，提供商选择创建两个嵌套任务，而且“affectedResources”属性包含新建系统的引用。通过定期检索任务描述，用户可以确定任务何时完成，例如，如果“进度”属性的值为 100，则表示任务已完成。

第 4 步：查询新系统

检索系统，获取新系统的完整描述：

```
GET /systems/87342
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
      "id": "http://example.com/systems/78342",
      "name": "MySystem1",
      "description": "My first system",
      "created": "2012-08-15T12:15:00Z",
      "updated": "2012-08-15T12:15:00Z",
      "state": "STOPPED",
      "machines": { "href": "http://example.com/systems/87432/machines" },
      "credentials": { "href": "http://example.com/systems/87342/creds" },
      "volumes": { "href": "http://example.com/systems/87342/vols" },
      "operations" : [
            {"rel": "edit", "href": "http://example.com/systems/78342" }
      ]
}
```

场景 6：编辑系统模板

该场景将第二台机器添加到现有系统模板中。

第 1 步：编辑现有系统模板

编辑在之前的场景中创建的系统模板，并使用该模板中的凭据和卷资源添加另一台机器：

SystemTemplate：

```
http://example.com/systemTemplates/48920
```

检索现有 SystemTemplate 定义：

```
GET /systemTemplates/48920 HTTP/1.1
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
    "name": "System Demo1",
    "description": "My first system template demo",
    "created": "2012-08-15T12:15:00Z",
    "updated": "2012-08-15T12:15:00Z",
    "componentDescriptors": [
        { "name": "MyMachine",
            "type": "http://schemas.dmtf.org/cimi/1/Machine",
            "machineTemplate":
            { "name" : "Machine in system demo",
            "description" : "Machine in system",
            "machineConfig": { "href": "http://example.com/configs/tiny" },
            "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
            "credential": { "href": "#MyCredential" },
            "volumes": [
                { "initialLocation": "/vol",
                    "href": "#MyVolume"
                }
            ]
        }
    },
    { "name": "MyCredential",
    "type": "http://schemas.dmtf.org/cimi/1/Credential",
    "credentialTemplate":
        { "href": "http://example.com/credentialTemplates/72000" }
    },
    { "name": "MyVolume",
        "type": "http://schemas.dmtf.org/cimi/1/Volume",
        "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839"}
    }
  ],
    "operations" : [
        { "rel":"edit", "href": "http://example.com/systemTemplates/48920" }
  ]
}
```

更新 SystemTemplate 资源：

```
PUT /systemTemplates/48920 HTTP/1.1
Content-Type: application/json
```

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
    "name": "System Demo1",
    "description": "My first system template demo",
    "componentDescriptors": [
        { "name": "MyMachine",
          "type": "http://schemas.dmtf.org/cimi/1/Machine",
          "machineTemplate":
            { "name" : "Machine in system demo",
              "description" : "Machine in system",
              "machineConfig": { "href": "http://example.com/configs/tiny" },
              "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
              "credential": { "href": "#MyCredential" },
              "volumes": [
                { "initialLocation": "/vol",
                  "href": "#MyVolume",
                }
              ]
          },
      "quantity": 2
    },
    { "name": "MyCredential",
      "type": "http://schemas.dmtf.org/cimi/1/Credential",
      "credentialTemplate":
        { "href": "http://example.com/credentialTemplates/72000" }
    },
    { "name": "MyVolume",
      "type": "http://schemas.dmtf.org/cimi/1/Volume",
      "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839"}
    }
  ]
}
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
    "name": "System Demo1",
    "description": "My first system template demo",
    "created": "2012-09-15T12:15:00Z",
    "updated": "2012-09-15T12:15:00Z",
    "componentDescriptors": [
      { "name": "MyMachine",
        "type": "http://schemas.dmtf.org/cimi/1/Machine",
        "machineTemplate":
          { "name" : "Machine in system demo",
            "description" : "Machine in system",
            "machineConfig": { "href": "http://example.com/configs/tiny" },
            "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
            "credential": { "href": "#MyCredential" },
            "volumes": [
              { "initialLocation": "/vol",
                "href": "#MyVolume"
              }
          ]
        },
      "quantity": 2
      },
      { "name": "MyCredential",
            "type": "http://schemas.dmtf.org/cimi/1/Credential",
            "credentialTemplate":
                  { "href": "http://example.com/credentialTemplates/72000" }
      },
      { "name": "MyVolume",
            "type": "http://schemas.dmtf.org/cimi/1/Volume",
            "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839"}
      }
   ],
   "operations": [
      { "rel": "edit", "href": "http://example.com/systemTemplates/48920" }
      ]
}
```

第 2 步：用系统模板创建新系统

将 POST 发送到之前创建新系统场景所使用的 URL，用该系统模板创建新系统：

```
POST /systems HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCreate",
      "name": "MySystem2",
      "description": "My second system",
      "systemTemplate": { "href": "http://example.com/systemTemplates/48920" }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/systems/78343
```

第 3 步：查询新系统

检索系统，获取新系统的完整描述：

```
GET /systems/87343
```

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

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
      "id": "http://example.com/systems/78342",
      "name": "MySystem2",
      "description": "My second system",
      "created": "2012-10-15T12:15:00Z",
      "updated": "2012-10-15T12:15:00Z",
      "state": "STOPPED",
      "machines": { "href": "http://example.com/systems/87432/machines" },
      "credentials": { "href": "http://example.com/systems/87342/creds" },
      "volumes": { "href": "http://example.com/systems/87342/vols" },
      "operations" : [
            { "rel": "edit","href": "http://example.com/systems/78342" }
      ]
}
```

修订记录：

版本	**日期**	**描述**
1.0.0	2012-08-28	版本1.0
1.0.1	2012-09-12	勘误表