文書番号: **DSP2027**

日付: **2012-09-12**

バージョン: **1.0.1**

Cloud Infrastructure Management Interface (CIMI) 入門

文書のタイプ: ホワイト・ペーパー

文書の位置付け: **DMTF** 参考情報

文書の使用言語: **ja-JP**

Copyright Notice
Copyright © 2012 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit http://www.dmtf.org/about/policies/disclosures.php.

目次

目次 3
概要 6
Scenario 1：マシンの新規作成 7
Step 1：CEP の取得 7
Step 2：マシン・イメージのリストの取得 7
Step 3：マシン構成のリストの取得 8
Step 4：認証情報リソースの新規作成 10
Step 5：マシンの新規作成 11
Step 6：新しいマシンの照会 11
Step 7：マシンの起動 12
Step 8：マシンの停止 13
Step 9：マシンの属性の更新 13
Scenario 2：マシンへの新しいボリュームの追加 14
Step 1：マシンの URL の取得 14
Step 2：CEP の取得 14
Step 3：使用するボリューム構成を判断するためのリストの取得 14
Step 4：ボリュームの新規作成 15
Step 5：ボリューム情報の取得 16
Step 6：マシンのボリューム・コレクションの取得 16
Step 7：マシンへの新しいボリュームの接続 17
Step 8：更新を確認するためのマシンのボリューム・コレクションの照会 18
Scenario 3：マシン・テンプレートの定義および使用 19
Step 1：CEP の取得 19
Step 2：マシン・テンプレートの新規作成 19
Step 3：マシン・テンプレートによるマシンの新規作成 20
Scenario 4：既存のボリュームからのマシンの新規作成 21
Step 1：CEP を取得する 21
Step 2：ボリュームのリストを取得し、使用するボリュームを決定する 21
Step 3：新しいマシンを作成する 22
Step 4：新しいマシンを照会する 22
Scenario 5：システム・テンプレートの定義および使用 24
Step 1：CEP の取得 24
Step 2：システム・テンプレートの新規作成 24
Step 3：システム・テンプレートによるシステムの新規作成 25
Step 4：新しいシステムの照会 26
Scenario 6：システム・テンプレートの編集 28
Step 1：既存のシステム・テンプレートの編集 28
Step 2：システム・テンプレートによるシステムの新規作成 30
Step 3：新しいシステムの照会 30
変更履歴： 31

序文

本書では、CIMI プロトコルの一般的な使用法をシナリオの形式で説明する。

謝辞
本書の作成にあたり、以下の方々に感謝の意を表す。

Editors:

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

Contributors:

- 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 - Telefónica
- 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.
- Galán, Fermín - Telefónica
- 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.
- Peñalvo, Miguel - Telefónica
- Pilz, Gilbert - Oracle
- Polo, Alvaro - Telefónica
- 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

概要

便宜上、ここで説明する各シナリオでは、クラウド・プロバイダーは、そのシナリオで着目する機能のデモンストレーションに必要な最小限の機能のみをサポートする。したがって、サポートされている機能のリストを取得するためにクラウド・エントリー・ポイント（CEP）を照会した結果はシナリオごとにカスタマイズされている。また、クラウド・エントリー・ポイントの URI を http://example.com/CEP と仮定している。

Scenario 1 : マシンの新規作成

このシナリオでは新しいマシンを作成する。この新しいマシンの構成は、プロバイダーから提供された既存の構成とイメージに基づいたものとする。一方、認証情報のリソース（ユーザーID およびパスワード）は新たに作成する。

Step 1 : CloudEntryPoint(CEP)の取得

CloudEntryPoint（CEP）は、このクラウドで使用できる一連のリソースへのリンクを提供する。CEP を取得して、各タイプごとのリソースの集合（コレクション）への URL を読み出す。

注意：CIMI 仕様では、タイプ別のリソース（マシン、マシン構成、マシンイメージ、ボリューム、ネットワーク・・）の集合をコレクションとよび、コレクションに対する操作を規定している。詳細は CIMI1.0 仕様書（DSP0263）5.5.12 節を参照のこと。

```
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" }
}
```

Step 2 : マシン・イメージのリストの取得

新しいマシンを作成する前に、事前にインストールしておくオペレーティング・システムとソフトウェアを決定する。Machine Images コレクションは、このクラウドが提供するマシン・イメージの集合である。これには、クラウドによりあらかじめ定義されているものもあれば、ユーザーが作成したものもある。

```
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" }
    }

  ]
}
```

Step 2.1 : マシン・イメージの選択

次に、それぞれのマシン・イメージを調べて、自分のニーズに適したものを探し出す。ここでは最初のマシン・イメージがニーズに適しているので、これを後で使用する。

リストの先頭にあるイメージを使用することはすでにわかっていて、単に直前に検索したリソースの 1 つを確認するだけである場合は、代わりに次の処理を実行すればよいことに注意されたい。

```
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" }
    }
  ]
}
```

この場合は$first=1 を指定する必要はない。"1"がデフォルト値となっているからである。

注意：CIMI 仕様では、リソース集合（コレクション）の検索において、集合中で検索する最初の要素の位置（$first）と最後の要素の位置（$last)を指定することで、検索範囲を絞り込むことができる。本文のケースでは$first を省略し、$last=1 としているが、仕様によれば $ first=1, $last=1,すなわち集合の最初の要素のみ検索、と解釈される。詳細は CIMI1.0 仕様書（DSP0263）4.1.6.2 節を参照のこと。

Step 3 : マシン構成のリストの取得

次に、マシン・イメージのインストール先とする仮想ハードウェアの種類を決定する。使用するマシン・イメージの種類を決定した場合と同様に、使用可能なマシン構成のリストを要求する。

```
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 },
      ]
    }
  ]
}
```

Step 3.1 : マシン構成の選択

次に、返されたリストを調べ、ニーズに適したマシン構成を選択する。ここでは最初のマシン構成がニーズに適しているので、これを後で使用する。

Step 4 : 認証情報リソースの新規作成

新たに作成したこのマシンでは自分専用のユーザーID とパスワードを使用するので、認証情報リソースを新規に作成する必要がある。この処理は POST 操作で実行するが、その前に認証情報のコレクションを取得する必要がある。これは、POST を使用して新しい認証情報リソースを配置する場所（URI）を把握しておくためである。

```
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" }]
}
```

現時点では、この環境に認証情報リソースは存在していない。認証情報リソースを新たに作成するには、その認証情報リソースに対するこのクラウド・プロバイダーの拡張属性（たとえばパスワード）を検出しておく必要がある。デフォルトでは、CIMI 仕様には新しいマシンの最初のユーザーを指定する方法は定義されていない。この情報を提供する方法の決定は、各クラウド・プロバイダーに委ねられている。クライアントは、認証情報リソースのメタデータ・リソースを検索することにより、この情報を検出できる。このメタデータ・リソースを調べるには、まず、ResourceMetadata コレクションに目を通して、目的の認証情報リソースに対するこのプロバイダーの説明を確認する。クラウドのエントリー・ポイントで参照されている URI から ResourceMetadata コレクションを取得することから着手する。

```
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" }
      ]
    }
  ]
}
```

このコレクションの resourceMetadata エントリーのリストで各エントリーを確認し、"typeURI"が"http://schemas.dmtf.org/cimi/1/Credential"であるものを探し出す。そのエントリーが見つかれば、このプロバイダーがこの認証情報リソースに追加した拡張を確認できる。上記のコードは、Credential リソースが拡張されていて、"userID"および"password"という 2 つの属性を指定する必要があることを示している。どちらの属性も"string"型である。

ここで、POST 操作を使用して、新しい認証情報リソースである Credential を作成する。

```
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"属性および"password"属性は、認証情報の ResourceMetadata を使用して検出したが、"name"属性および"description"属性はすべてのリソースで使用できる共通属性セットに属している。この後のシナリオで、"userID"と"password"がこのイメージ・タイプとクラウド・プロバイダーの固有な属性名であることをクライアントがどのように知ったか、をみることにする。

Step 5 : マシンの新規作成

Machines コレクションを取得する。これにより、新しいマシンの POST 先がわかる。

```
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" } ]
}
```

ここで新しいマシンを作成する。

```
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
```

Step 6 : 新しいマシンの照会

このマシンを取得して、新しいマシンの完全な表現を得る。

```
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" }
  ]
}
```

このマシンの"state"属性は、新しいマシンの初期状態である"STOPPED"になっている。

Step 7 : マシンの起動

このマシンの表現にある"operations"配列に"start"操作が存在することは、"start"操作の POST 先 URI を示しているだけではなく、この時点で"start"操作が可能であることも示している。

```
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
```

Step 7.1 : マシンの起動を確認するためのマシンへの照会

マシンを再度検索し、マシンが起動していることを確認する。

```
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" }
  ]
}
```

現在、このマシンの"state"属性は"STARTED"になっており、"operations"配列では"start"操作ではなく、"stop"操作が可能であることが示されている。

Step 8 : マシンの停止

"stop"操作が指定されている URL を使用して、このマシンの停止を要求できる。

```
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
```

Step 9 : マシンの属性の更新

"edit"操作が指定されている URL に対して PUT 操作を使用し、"name"や"description"など、マシンの属性をいくつか更新できる。

```
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" }
```

"edit"操作の URL は、更新対象となる属性が追加されていることに注意されたい。これらの属性のみが更新の対象となる。この URL には"description"属性が記述されているが HTTP 要求の本文には記述されていないので、この属性は消去される。

注意：PUT /machines/843752?$select=name,description という指定により name, description 属性が変更されるはずだが、メッセージ本文には description の記載がない。これは、description　属性は不要、という意味である。

Scenario 2 : マシンへの新しいボリュームの追加

このシナリオでは、新しいボリュームを作成し、それを既存のマシンに接続する。

Step 1 : マシンの URL の取得

目的のマシン：

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

Step 2 : CEP の取得

CEP は、このクラウドで使用できる一連のリソースへのリンクを提供する。CEP を取得し、各コレクションへの URL を読み出す。

```
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" }
}
```

Step 3 : 使用するボリューム構成を決定するためのリストの取得

新しいボリュームを作成するときは、そのボリュームの種類（サイズ、フォーマットなど）を決定する必要がある。Volume Configurations コレクションは、このクラウドが提供する事前定義されたボリューム構成の集合である。

```
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
    }
  ]
}
```

Step 3.1 : ボリューム構成の選択

次に、それぞれのボリューム構成を調べて、ニーズに適したものを探し出す。ここでは最初のボリューム構成がニーズに適しているので、これを後で使用する。

Step 4 : ボリュームの新規作成

Volumes コレクションを取得する。これは、POST を使用して新しいボリュームを配置する場所を把握しておくためである。

```
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" } ]
}
```

ここで新しいボリュームを作成する。

```
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
```

Step 5 : ボリューム情報の取得

新たに作成してマシンに接続するボリュームが目的どおりのものであることを確認するには、前の手順で返された参照に従う。

```
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" }
  ]
}
```

Step 6 : マシンのボリューム・コレクションの取得

この新しいボリュームを目的のマシンに接続できるようにするには、まず、そのマシンのボリュームのコレクションを取得し、要求の送り先を確認する必要がある。まず目的のマシンを取得し、ボリュームのコレクションへの参照を入手する。

```
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" }
  ]
}
```

前のシナリオでは焦点を絞っていたことから"volumes"属性は存在していなかったが、ここでは"volumes"属性が表示されている。これは、このシナリオ（およびここで例として使用しているプロバイダーの機能）が拡張され、ボリュームのサポートが追加されているからである。

ここでボリュームのコレクションを取得する。

```
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" }
  ]
}
```

このマシンには、今のところボリュームは接続されていない。

あるいは、最適な方法として、$expand クエリー・パラメータを使用し、元のマシンと同時にこのコレクションを取得することもできる。

注意：$ expand クエリー・パラメータにより、参照（URI)だけでなく、その属性を展開した形式で検索できる。詳細は CIMI1.0 仕様書（DSP0263）4.1.6.4 節を参照のこと。

```
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" }
  ]
}
```

Step 7 : マシンへの新しいボリュームの接続

ボリュームのコレクションに対して"add"操作を使用し、このボリュームを目的のマシンに接続して、新しい MachineVolume リソースができる。

注意：ここでは、ID 番号 843752 のマシンの volumes 属性（マシンに接続するボリュームのコレクション。MachineVolume コレクションと呼ばれる）に ID 番号 35782 番のボリューム情報を追加する操作を行っている。結果として、volumes 属性にボリュームの参照（URI）が追加される。詳細は CIMI 仕様書（DSP0263）5.5.12 節を参照のこと。

```
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"}
  ]
}
```

Step 8 : 更新を確認するためのマシンのボリューム・コレクションの照会

マシンのボリュームのコレクションを取得して、すべてのボリュームのリストを入手し、正常に更新されたことを確認する。

```
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" }
  ]
}
```

Scenario 3 : マシン・テンプレートの定義および使用

このシナリオでは新しいマシン・テンプレートを作成する。このテンプレートは、この後、新しいマシンの作成に使用する。マシン・テンプレートは、マシンの構成、イメージなどについて明確な規定ができる便利なリソースであり、後で再利用できるような継続性を備えている。この機能は、新しいマシンのユーザーが技術的には深い知識を持たず、マシンの作成に必要な詳細をすべては理解できない可能性がある場合に効果的である。マシン・テンプレートは、デモを実施する場合や、特定のマシン構成で特定のマシン・イメージを使用することが必要となる複雑な構成向けに作成することが普通である。マシン・テンプレートを使用すると、このような情報が恒常的に保持されるので、容易に再利用できる。

便宜上、ここでは前のシナリオで用いられた構成情報を再利用する。

Step 1 : CEP の取得

CEP は、このクラウドで使用できる一連のリソースへのリンクを提供する。CEP を取得し、各コレクションへの URL を読み出す。

```
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" }
}
```

Step 2 : マシン・テンプレートの新規作成

このマシン・テンプレートで再利用するマシン構成、マシン・イメージ、および認証情報の各リソースは前のシナリオで取得済みである。

取得済みのマシン・イメージ：

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

取得済みのマシン構成：

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

取得済みの認証情報：

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

新しいマシン・テンプレートを作成できるようにするには、POST の送信先 URL を決定しておく必要がある。この場所は、CEP の一部として返されたマシン・テンプレートのコレクションにある URL から取得する。

注意：CIMI 仕様では、テンプレートの情報も CEP が参照している。 GET /machineTemplates HTTP/1.1 とは、CEP 直下のマシン・テンプレートのコレクションを読み出す操作である。詳細は CIMI 仕様書（DSP0263) 5.12 節を参照のこと。

```
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" }
  ]
}
```

現在のところ、この環境にはマシン・テンプレートが存在していない。

ここで、新しいマシン・テンプレートのリソースを作成する。

```
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
```

Step 3 : マシン・テンプレートによるマシンの新規作成

次に、このマシン・テンプレートを使用して新しいマシンを作成する。

```
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
```

Scenario 4 : 既存のボリュームからのマシンの新規作成

このシナリオでは、既存のボリュームから起動する新しいマシンを作成する。この簡単な例では、起動プロパティーを有効に設定したボリュームがあることをユーザーが認識していると仮定している。

Step 1 : CEP を取得する

CEP は、このクラウドで使用できる一連のリソースへのリンクを提供する。CEP を取得し、各コレクションへの URL を読み出す。

```
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" }
}
```

Step 2 : ボリュームのリストを取得し、使用するボリュームを決定する

新しいマシンをボリュームから作成するには、どのボリュームを使うかを決定する必要がある。ボリュームのコレクションは、このクラウドが提供する既存のボリュームの集合である。

```
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
    }
  ]
}
```

Step 2.1：ボリュームを選択する

次に、それぞれのボリュームを調べて、ニーズに適したものを探し出す。ここでは最初のボリュームがニーズに適しているので、これを後で使用する。

Step 3：新しいマシンを作成する

Machines コレクションを取得する。これにより、新しいマシンの POST 先がわかる。

```
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" }
  ]
}
```

新しいマシンを作成し、起動可能なボリュームに接続する。

```
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" }
    ]
  }
}
```

ただし、このマシン・テンプレートでは、使用するマシン・イメージもマシン構成も指定されていない。便宜上、この例では、プロバイダーがこれらのデフォルト値を持っていると見なすことにする。

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

Step 4：新しいマシンを照会する

作成したマシンを取得して、新しいマシンの完全な表現を得る。

```
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" }
  ]
}
```

このマシンの"state"属性は、新しいマシンの初期状態である"STOPPED"になっている。

Scenario 5 : システム・テンプレートの定義および使用

このシナリオでは新しいシステム・テンプレートを作成する。このテンプレートは、この後で新しいシステムの作成に使用する。システム・テンプレートは、システムの構成、イメージなどについて明確な規定ができる便利なリソースであり、後で再利用できるような継続性を備えている。

Step 1 : CEP の取得

CEP は、このクラウドで使用できる一連のリソースへのリンクを提供する。CEP を取得し、各コレクションへの URL を読み出す。

```
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" }
}
```

Step 2 : システム・テンプレートの新規作成

システム・テンプレートをインスタンス化した時点で、マシン、そのマシンに接続したボリューム、および認証情報の各リソースが作成されるように、そのシステム・テンプレートを定義する。このような構成を実現するには、システム・テンプレートの定義にマシン・テンプレートを値で指定し、ボリューム・テンプレートと認証情報テンプレートを参照で指定する。ボリューム・テンプレートと認証情報テンプレートの各リソースはすでに存在している。

使用可能なボリューム・テンプレート：

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

使用可能な認証情報テンプレート：

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

注意：CIMI 仕様では、リソースのテンプレートを指定する方法として、外部参照のほか、値を直接設定することもできる。詳細は CIMI 仕様書（DSP0263 1.0.0） 3.10 節を参照のこと。

注意：別の方法として、以下のマシン・テンプレート定義にボリューム・テンプレートと認証情報テンプレートを値で指定することもできる。しかし、どのリソースが関連しているのかをシステム・テンプレートですぐに確認できることは有益であり、また一般的に認証情報を自動作成することで高いセキュリティーを実現できる。

新しいマシン・テンプレートを作成するには、POST の送信先 URL を決定しておく必要がある。この場所は、CEP の一部として返されたシステム・テンプレートのコレクションにある URL から取得する。

```
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" }
  ]
}
```

新しいシステム・テンプレートのリソースを作成する。

```
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
```

Step 3 : システム・テンプレートによるシステムの新規作成

作成したシステム・テンプレートを使用して新しいシステムを作成する。

```
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
```

別の方法として、プロバイダーは、システム新規作成の要求に対して、このシステムの作成完了を待たずに、ジョブのリソースへの参照を返すこともできる。その場合は、上記の 201 応答の代わりに次の応答が得られる。

```
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" }
  ]
}
```

この応答によると、このプロバイダーはネストした 2 つのジョブを作成していて、"affectedResources" 属性には新たに作成されたシステムへの参照が記述されている。このジョブの表現を定期的に取得することにより、利用者はこのジョブが完了する時期、つまり"progress"属性の値が 100 になる時期を判断できる。

注意：ジョブは、プロバイダーの実行するプロセスを表現する。詳細は CIMI 仕様書（DSP0263）5.17.1 節を参照のこと。

Step 4 : 新しいシステムの照会

作成したシステムを取得して、新しいシステムの完全な表現を得る。

```
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" }
  ]
}
```

Scenario 6 : システム・テンプレートの編集

このシナリオでは、既存のシステム・テンプレートにマシンをもう 1 つ追加する。

Step 1 : 既存のシステム・テンプレートの編集

ここでは、前のシナリオで作成したシステム・テンプレートを編集し、このテンプレートの認証情報とボリュームのリソースを共有する別のマシンを 1 つ追加する。

作成済みのシステム・テンプレート：

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

既存のシステム・テンプレートの定義を取得する。

```
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" }
  ]
}
```

このシステム・テンプレートのリソースを更新する。

```
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" }
  ]
}
```

Step 2 : システム・テンプレートによるシステムの新規作成

作成したシステム・テンプレートを使用して新しいシステムを作成する。そのためには、前のシナリオで新しいシステムを作成したときに使用した URL に POST を送信する。

```
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
```

Step 3 : 新しいシステムの照会

作成したシステムを取得して、新しいシステムの完全な表現を得る。

```
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" }
  ]
}
```

変更履歴：

バージョン	日付	説明
1.0.0	2012-08-28	バージョン 1.0
1.0.1	2012-09-12	誤記修正