

オープンクラウドコンピューティングインタフェース – クラウドAPIのユースケースおよび要件

1. はじめに	1
2. OCCIのユースケース	2
2.1. SLA@SOI を用いたサービスレベルアグリーメント (SLA) を意識したクラウドインフラストラクチャー.....	2
2.2. サービスのライフサイクルを管理するサービスマネージャー	2
2.3. OpenNebulaを用いるクラウドインフラストラクチャー間の相互運用性	4
2.4. APIを直接コールするAJAXウェブフロントエンド	5
2.5.複数のサービスプロバイダーをサポートする単一の技術統合	5
2.6. OCCIにEC2を含むこと	6
2.7. 自動ビジネス継続性およびディザスタリカバリー	6
2.8. Unix shellからのクラウドの簡潔なスクリプト	6
2.9. 代表的なウェブホスティングクラスター	6
2.10. 集中ダッシュボードからクラウドリソースを管理する.....	7
2.11. コンピュートクラウド	7
2.12. 複数の割り当て	8
2.13. クラウドプロバイダーのVMインプットおよびアウトプットフォーマットのサポートをクラウドコンシューマーが知ることに	8
2.14. クラウドプロバイダーのデータセットインプットおよびアウトプットフォーマットのサポートをクラウドコンシューマーが知ることに	9
3. OCCIの要件	9
3.1. 機能要件	9
3.2. 非機能要件	12
4. クラウドAPI 特徴マトリックス	12
5. 結論	14
6. 寄稿者	14
7. 知的財産権について	14
8. 免責条項	15
9. 著作権情報	15
10. 参考文献	15

本文書の位置づけ

本文書はクラウドおよびグリッドコミュニティに情報を提供するものである。本書の配布に制限はない。

著作権情報

Copyright (C) Open Grid Forum (2009, 2010). All Rights Reserved.

要約

本文書は、オープンクラウドコンピューティングインタフェース (OCCI)クラウドAPIのユースケースおよび要件に関する非公式な記述である。本文書はオープンクラウドコンピューティングインタフェース・ワーキンググループにより作成された。本文書はIaaSクラウドコンピューティングマネージャおよびアドミニストレータのニーズをユースケースの形式で記録する。ユースケースはAPI要件開発の一義的なガイドとして機能する。本文書はオープンクラウドコンピューティングインタフェースの特徴を実証および検証する初めての文書である。

1. はじめに

本文書にはワーキンググループが収集したすべてのユースケースと要件が含まれる。

本文書は、ユースケース、要件、および現在のクラウドAPIの比較マトリックスの3章で構成されている。本文書では、ユースケースを名前と、機能・非機能要件のセットで定義し、すべての要件を分類し、表に記載する。各要件は (可能な限り) ユースケースにマッピングされている。すべての要件には優先度が付けられ、クラウドAPIの比較マトリックスは各APIの一般的小さいおよび詳細な特徴を明細に記し、各特徴をどのAPIがサポートするかを示している。

2. OCCIのユースケース

次章ではOCCIワーキンググループの要件分析中に収集したユースケースを説明する。これらのユースケースを用いて要件を設定し、その後OCCIの仕様を検証する。

2.1. SLA@SOI を用いたサービスレベルアグリーメント (SLA) を意識したクラウドインフラストラクチャ

動的インフラストラクチャプロビジョニングに対する標準インタフェースが必要である。標準インタフェースを準備している間に、インフラストラクチャプロビジョニングが「機械可読」のSLAを使っていることを保証、検証しなければならない。¹

機能要件

- 仮想マシン (VM) の記述: リクエストフォーマットの重要性 - プロバイダ間の一貫性が最も低い分野である。
- VMの記述: 機能属性に非機能制約を加える手段。

¹ SLA@SOIプロジェクトのウェブサイト - <http://sla-at-soi.eu>

- **VMマネジメント:** リクエストのすべてのパラメータは「モニター可能」であり検証できなくてはならない。割り当てたリソース (VM) の完全な管理が必要であり、最低でも、スタート、ストップ、サスペンド、レジュームが必要である。
- **VMモニタリング:** プロビジョニングリクエストで宣言された非機能制約のモニタリング。
- **ネットワークマネジメント:** ネットワークタグで割り当て可能なリソース。パブリックとプライベートの初期設定をさらに分類できる。たとえばウェブのタグをパブリックネットワークグループに割り当てできる。
- **ストレージマネジメント:** 簡潔なマウントポイント、SaaSが提供するストレージ再利用。

非機能要件

- **セキュリティ:** トランスポートおよびユーザレベル (ACLs? OAuth?) でのセキュリティ。
- **サービスの品質:** 多岐にわたることもある。インフラストラクチャプロバイダが提供するサービス、たとえば、セキュリティ、サービス品質 (QoS)、位置情報通知機能 (geo-location)、アイソレーションレベルの一部である。NFPは、IaaSプロバイダを差別化する基本的なビルディングブロックである。
- **スケジューリング情報:** いつ特定のリソースを作動させるか。リソースがお互いに依存している場合にどの順序でリソース群を作動させるか。

2.2. サービスのライフサイクルを管理するサービスマネージャ

本ユースケースはRESERVOIRプロジェクトアーキテクチャの「サービスマネージャ」(SM) 層を基盤とする。クラウドでのサービス展開を希望する「サービスプロバイダ」(SP) はこの層を使ってサービスのライフサイクルを管理する。SMはサービスのリクエストに従いクラウドインフラストラクチャ上で自動的に動作する。ある意味では、SMはサービスのコンフィギュレーションやニーズをクラウドインフラストラクチャへのコールにマッピングするため、SMが課す要件の多くはSMがSPに提供することを目指している柔軟性によるものである。²

機能要件

- **ネットワークマネジメント:** VMを接続できるようなプライベートネットワークの割り当てに関する方法がなければならない。特別なネットワーク (「パブリックネットワーク」等) が利用できなければならない。ネットワークインタフェースが接続している場合には、インフラストラクチャはパブリックIPアドレスを割り当てなければならない。
- **イメージマネジメント:** ディスクイメージをレジスタ、アップロード、アップデートおよびダウンロードする方法がなければならない。

² RESERVOIR プロジェクトウェブサイト- <http://www.reservoir-fp7.eu>

- **VMの記述:** すべてのVMハードウェアコンポーネントおよびその属性を、VMの場所に関する制約がある場合はその制約とともに記述できなければならない。
 - **メモリ:** サイズ。
 - **CPU:** アーキテクチャ、CPUの量およびスピード。
 - **ディスク:** サイズ、インタフェース (SCSI、IDE、SATA等)、RAID (有無およびRAIDレベル)、マウントするディスクイメージ、自動バックアップ (有無、バックアップの頻度)。
 - **ネットワーク:** インタフェース、各インタフェースの帯域幅、インタフェースが接続しているネットワーク。
 - **地理的制約:** (法的な目的等で) VMをデプロイできる場所またはデプロイできない場所。
 - **移行が可能か (はい/いいえ):** データの移行がインフラストラクチャによってサポートされている場合、VMに許可されていればこのフラグを立てる。
- **VMマネジメント:** インフラストラクチャが遷移を許可するのであれば (つまり、OCCIのステートマシンに定義されているならば)、SMがVMの状況を(たとえばACTIVEからSUSPENDEDに) 変更できる方法がなければならない。VMの記述はマシンが動作中 (ACTIVE、SUSPENDED等)に変更できる。しかし、その変更が地理的または移行の制約に関するものでない限り、マシンが停止し、再起動するまでは考慮されない。各ディスクのバックアップはSMが定義するイメージとしてIDを持つ。バックアップをダウンロードする方法を提供しなくてはならない。つまり、各バックアップはディスクイメージであるため、どのVMにもマウント可能でなくてはならない。たとえば、あるVMを停止し、そのVMのコンフィギュレーションをそのディスクがバックアップイメージをマウントするように変更し、その後VMを再開することができなくてはならない。
- **モニタリング:** あるエレメントのステータスの表現をキーとその値の表として提示する (モニタリングについては「ステータス」という用語を使い、OCCIのステートマシンのステートと混同しないように「ステート」という用語を使わない)。たとえば、メモリコンポーネントのステータスは使用されたメモリおよびキャッシュメモリの量で表すことができる。従って、キーは「使用された」および「キャッシュ」となり値は「142MB」および「430MB」となる。リクエストとリプライの両方とも、対応するエレメント識別子を用いる。2種類のモニタリングをサポートしなくてはならない。
 - **プル型:** SMはレジスタしたエレメント、つまりVM、ネットワーク等のステータスをリクエストできる。また、SMはコンポーネントのステータス、たとえばあるVMのあるディスクのステータスをリクエストできる。
 - **パブリッシュ/サブスクライブ型:** SMはVMおよび/またはネットワーク上のイベントについて通知を受けるようにサブスクライブできる。通知が受けられるイベントには次のようなものがある。

- VMのコンポーネントのエラー。
- VMの状況 (たとえばACTIVE からSUSPENDEDへ) の変更。
- あるエレメントの状況に関する定期的な通知。本通知の頻度はサブスクリプションメッセージで設定できる。
- エラーメッセージ: VMが作成できなかったときやイメージをアップロードできなかったとき等、プラットフォームは理由の詳細な説明と共にエラーメッセージを返す。
- 識別表示: ネットワーク、VMおよびイメージは固有のID (UUID、URI等) を持たなければならない。VMのコンポーネント (ディスク、メモリ等) も固有のIDを持つべきか決めなければならない。IDは対応するエレメントが作成されるときに、クラウドインフラストラクチャから割り当てられる。

非機能要件

- ハードウェアコンフィギュレーションおよびモニタリング値のどちらも、その値でどの単位で表すかの明確な基準を持たなくてはならない。たとえば、メモリサイズを「2」に設定するとき、MBではなくGBを意味することを明確にしなければならない。別の方法として、値を「2GB」に設定することもできる。また別の方法としては、値と単位の両方を設定できるようにすること、つまり、値が「2」で単位を「GB」とする方法が挙げられる。
- プロトコル: トランスポート、メッセージフォーマット、および状況の表現はオープンで標準的なプロトコルを用いなければならない、それぞれソフトウェアにより強力にサポートされていない (たとえば、複数のプログラミング言語に適用できるライブラリおよびフレームワークなど)。

2.3. OpenNebulaを用いるクラウドインフラストラクチャ間の相互運用性

OpenNebulaはRESERVOIRプロジェクトで強化されている、仮想インフラストラクチャエンジンであるが、これにより物理的リソースのプール上で仮想マシンを管理できる。OpenNebulaは主に3機能を提供している。パブリッククラウドのバックエンド、データセンターまたはクラスター (プライベートクラウド) のバーチャルインフラストラクチャの管理、クラウドの相互運用の達成 (ハイブリッドクラウド) である。最後の機能は本ユースケースに関連する。

本ユースケースの目的は仮想マシングループの管理を可能にするインタフェースを公開するために、クラウドプロバイダAPIが考慮すべき要件を述べることである。これらの要件は様々なクラウドプロバイダが仮想マシンを管理するためにOpenNebulaを使用した経験から収集したものである。現時点では、Amazon EC2およびElasticHostsクラウドプロバイダにアクセスするためにOpenNebulaには2セットのプラグインがあり、エンドユーザが気付かない方法で両方のク

クラウドプロバイダを活用する。³

機能要件

- VMの記述: 仮想マシンはクラウドプロバイダ間で常に、次のような少量の不可欠な属性を使って記述されるべきである。
 - メモリ: 仮想マシンが必要とするRAMの量。
 - CPU: 仮想マシンが必要とするCPU数 (標準化するべき)。
 - ディスク: 仮想マシンの基本的なファイルシステムおよび使用する可能性のあるその他に整合するディスク。
 - ネットワーク: ある仮想マシンがいくつのネットワークインタフェースを持つべきか、どこに接続すべきか。
- VMマネジメント: APIは仮想マシン上の動作を強化するために次のような機能を提供すべきである。
 - DEPLOY: 仮想マシンを起動する。
 - SHUTDOWN: 仮想マシンをシャットダウンする。
 - CANCEL: 故障時に仮想マシンをキャンセルする、または仮想マシンが作動中の場合は破壊する。
 - CHECKPOINT: 仮想マシンのスナップショットを作成する。
 - SAVE: 仮想マシンのスナップショットを作成し、一時中断する。
 - RESTORE: 従前のスナップショットから仮想マシンを再開する。
 - POLL: 仮想マシンの状況および消費属性に関する情報を取り出す (メモリのパーセント、使用したCPU、移行したバイト等)。
- さらに、仮想マシンは次のいずれかの状態であるべきである。
 - PENDING: VMは物理的リソーススロットを待っている。
 - BOOTING: VM は起動中である。
 - RUNNING: VMは作動中であり、サービスの提供を開始できるべきである。
 - SUSPENDED: VM は一時停止中であり、再開を待っている。
 - SHUTDOWN: VMはシャットダウン中である。
 - CANCEL: VM はユーザまたはスケジューラーによってキャンセルされた。
 - FAILED: VM はクラッシュしたか、適切に開始しなかった。
- ネットワークマネジメント: APIは以下を行うために機能を公表すべきである。
 - プライベート仮想ネットワークの構築
 - パブリックIPを仮想マシンに加える。
- イメージマネジメント: 各クラウドプロバイダがソフトウェアを再インストールしなくて

³ OpenNebula ウェブサイト - <http://www.opennebula.org>

すむようにディスクイメージをアップロードする機能は、仮想マシンマネジメントの基礎である。アップロードプロセスは仮想マシン記述で使うように識別子を返すべきである。

非機能要件

- セキュリティ: 認証用にX509証明書を用いてセキュリティに対処しなければならない。また、承認は前記証明書およびACLリストに基づいてもよい。
- サービスの品質: Haizeaと一緒に用いた場合、OpenNebulaは高度な予約機能を提供する。クラウドプロバイダAPIは適切なサービスの品質 (QoS) を提供できるように同様の能力を提供すべきである。

2.4. APIを直接コールするAJAXウェブフロントエンド

本ユースケースはクラウドのウェブフロントエンドを作成する能力を説明する。クラウドプロバイダはカスタマーのウェブフロントエンドをOCCIのAPIを直接コールする完全なクライアントサイドのAJAXアプリケーションとして実装する。

機能要件

- 完全性: API はクラウドを完全に特定し管理するための完全なコール一式を含まなければならない (しかし、3、4個の名詞に対する15から20個の動詞のみと思われる)。
- 応答性: コールを速やかに返さなければならない。特に、存在するサーバ、ドライブ等のリストをそのプロパティすべてをリストせずにポーリングするために、簡潔で迅速なコールを提供すべきである。この方法はクラウドが返答するには計算的にはるかに簡単だからである。さらに、インタフェース外に作成されたサーバ等の情報を得るために定期的に調査する必要がある。

非機能要件

- シンタックス: APIの簡潔なJSONシンタックスによってAJAX インタフェースの実装がはるかに簡単になる。

2.5. 複数のサービスプロバイダをサポートする単一の技術統合

今日では、各クラウドプロバイダ (ElasticHosts、GoGrid、Amazon等) が独自にクラウドエコシステムの他のすべてのプレイヤー (CohesiveFT、RightScale等) と統合し、 $O(n^2)$ の別々の技術統合を行っている。将来的に、すべてのクラウドプロバイダおよびクラウドエコシステムのパートナーがい1つの標準APIを使うようになれば、 $O(n)$ の技術統合をすることができ、パートナーになる可能性を持つ者はすぐに相互運用性を持つことができる。

非機能要件

- アップテイク: 標準化IaaS APIはクラウドプロバイダとクラウドエコシステム双方からの強力なアップテイクを必要とする。

2.6. OCCIにEC2を含むこと

本文書作成時には、Amazon EC2がIaaSの有名なクラウドAPIである。EC2またはその他独自のAPIやオープンクラウドAPIを実装するクラウドプロバイダはOCCIを実装しないこともありうる。OCCI APIがゲートウェイとしてでもEC2にインタフェースできるようにするために、プロバイダ操作に対する影響を最小限にする。

機能要件

- セマンティック: 中心となるEC2のオブジェクトおよび操作を完全に記述する能力を含まなければならない。

非機能要件

- OCCIとEC2の統合をサポートするゲートウェイ。

2.7. 自動ビジネス継続性およびディザスタリカバリ

最新の物理マシンおよび/または仮想マシンのリモートシャドウを維持し、障害発生時にはリモートマシンを起動し、リモートマシンに切り替えることができる。

機能要件

- VMの記述: レガシーシステムにマッピングされているメタデータ。
- VMマネジメント: 障害発生時の自動マネジメント (たとえば、スタートアップ、IP変更等)。
- ネットワークマネジメント: IPのランタイム変更。
- イメージマネジメント: 物理的に同等なマシンと同期化するための高度なrsyncスタイルの更新 (たとえば、rsyncブロックデバイスからリモートローディスクファイルへ)。

非機能要件

- サービスの品質: フェイルオーバーに対する十分な能力を備えていること

2.8. Unix shellからのクラウドの簡潔なスクリプト

エンドユーザは簡単なタスクのスクリプトを望んでいる。簡単なタスクとは、たとえば、サーバを每晚深夜に立ち上げ、その1時間後にシャットダウンすることや、自動フェイルオーバー、レポーティング等である。ユーザは一般的なUnix/Linuxセットアップを使っているので、上記タス

クを実行するために簡単なcron ジョブを書くことを望んでいる。

非機能要件

- シンタックス: ユーザが抵抗無く使えるようにできるだけ簡潔でなければならない。ユーザが別の発展ツールやライブラリを使う必要があってはならない。ユーザが1、2行のshellスクリプトを書き、`curl`や`wget`等を使って5行以下の簡潔なコマンドデータをポストイングできなければならない。

2.9. 代表的なウェブホスティングクラスター

エンドユーザは代表的なホスティングクラスターをクラウド上で動作させる。n個のデータベースサーバ、m個のフロントエンドウェブサーバ (ロード下でx個に増える) および (専門の仮想マシンまたはGoGrid等のクラウドが提供する) ロードバランサーを用いる。

機能要件

- 完全性: API はこのクラスターを完全に表現できなくてはならない。このクラスターは、少なくとも (n+m+x台) の仮想マシン、各仮想マシンのストレージ、2つのネットワーク (プライベートネットワークがマシンにつながり、パブリックインターネットもロードバランサーに接続する)、パブリックインターネット上のウェブサイト用の固定静的IP、ロードバランサー自体の考えられる仕様、を必要とする。

2.10. 集中ダッシュボードからクラウドリソースを管理する

エンドユーザは軽量 (AJAXベースのものが多い) コンソールで自分のクラウドベースのリソースすべてを見ること、そして管理することを望んでいる。そのコンソールとは本ユースケースで言及したウェブフロントエンドと同じウェブと思われる。つまり、APIに直接コールするAJAXウェブフロントエンドである。

機能要件

- 完全性: クラウドが提供するすべてのリソースはAPIによって発見可能であり、これらのリソース上で実行できるすべてのアクションは、これらのアクションを実際に実行するアクチュエータとともにAPI経由でも利用可能であり、すべてのリソースの属性はAPI経由で利用可能である。
- 応答性: コールを速やかに返さなければならない。特に、存在するサーバ、ドライブ等のリストをそのプロパティすべてをリストせずには返すために、簡潔で迅速なコールを提供すべきである。この方法は、クラウドが返答するのに計算上、はるかにコストが安いからである。さらに、インタフェース外に作成されたサーバ等の情報を得るために定期的に調査する必要がある。(本文はAPIを直接コールするAJAXウェブフロントエンドからコピーした)。

- 分類可能性: (もっと適切な言葉があるはずである) クライアントは類似している種類のリソースを一緒に提示するために、そしてある種のリソースの種類に特化した個別のUIビューを提供するために、各リソースがどの種類かを識別できなくてはならない。たとえば、クライアントは実際のCPUを表示しないコンピュータリソース (おそらくこれがコンピュータテンプレートである) と作動中のCPUを実際に表示するコンピュータリソースとを識別できなければならない。実際に作動中のCPUのインタフェースはインスタンスの現在のIPアドレスを表示することがあり、ユーザはインスタンスにSSH接続できるが、インタフェースの別のタブはすべてのコンピュータテンプレートを表示し、そこからインスタンスを作成できる。
- タグ付け可能性: APIで見つけることができる各リソースはユーザによってタグ付けできなければならない。これにより、リソースは実装特有の識別子によって特定されるものの、ユーザがユーザ固有のコンテキストで定義する用語を使って容易にリソースを特定できる良く起こる状況をサポートする。たとえば、リソース「/compute/instanceABCDEFGH」に「データベースサーバ」というラベルをタグ付けし、リソース「/storage/disk12345678」に「superSecretCorporateData」というラベルをタグ付けすることができる。
- 検索性: リソースのリストをリクエストする能力によって、結果をフィルターされるカテゴリまたはタグを特定できるオプションフィルターが使えなければならない。これにより、さらに、「productionEnvironment」とタグ付けしたリソース、または「ストレージ」カテゴリのリソースに対するビューをさらに制限できる。

非機能要件

- ユーザビリティ: ユーザがそれぞれのリソースでどのアクションが行われているか容易に分かるようなコンテキストメニューおよびコンテキストアウェアリンクを持つユーザインタフェースでなくてはならない。

2.11. コンピュートクラウド

クラウドプロバイダはタスクのプロビジョニング、実行およびモニタリングをするRESTful APIを実装する。

機能要件

- 安全性: 承認された者のみがAPIの使用を許可されるようにし、APIを安全に保たなければならない。
- リソース: エンドポイントをタスクの外部モニタリング、ステータス、および監査のために作成しなくてはならない。このエンドポイントはAJAXおよびその他クライアントをサポートするRESTfulコールに反応する。
- スクリプト: ターゲットシステムはタスクとともに提供される指令を理解し処理する必要

がある。これらの指令はバイナリーまたはデータをシステム、**run executables**、およびシステムリソースステータスにプルすることができる。

非機能要件

- ・ シングルコンピュートメソッド: 得られたサービスはその他多くの目的のために使うことができるサービスと同じでなくてはならない。このサービスはシステムの健康度、システムのライフサイクルマネジメント、システムパッチング、およびコンフィギュレーションの変更をモニタリングするために使うことができる。もしこのサービスが当初、システムの唯一のサービスであったなら、プラグインマナーで他のサービスを積み上げていくために使うことができる。

2.12. 複数の割り当て

全体のクラスターを1回のコールで割り当てる。

機能要件

- ・ グループの定義: コンピュータのグループを定義する方法がなくてはならない。クラスターの例では、2 グループとなる。ヘッドノードと一組のワーカーノードである。
- ・ 情報: 定義されたグループメンバーのコンフィギュレーションを行うために、グループ全体に関しておよび基本的なコンフィギュレーションを見つけるための方法 (**URLが有力**) があるべきである。たとえば、ヘッドノードはすべてのワーカーノードのIPまたはホスト名前を知ることを求める。ワーカーノードも同様に知ることを望むであろうが、ヘッドノードが異なるグループにあることを知る必要がある。

2.13. クラウドプロバイダのVMインプットおよびアウトプットフォーマットのサポートを、クラウドコンシューマが知ることに

クラウドコンシューマは、クラウドプロバイダが認証し提供するVMのインプットおよびアウトプットフォーマットを知りたいことを望んでいる

機能要件

- ・ プロバイダは、安全ではないネットワーク接続上で利用できるAPIを提供する。
- ・ プロバイダは、安全なネットワーク接続上で利用できるAPIを提供する。
- ・ プロバイダが提供するAPIは、すべてのコンシューマ認証と承認レベルに使用できる。
- ・ プロバイダが提供するAPIは、サポートするVMのインプットフォーマットAPIを、すべてのプロバイダに対して独自にそして一般的に識別する。
- ・ プロバイダが提供するAPIは、サポートするVMのアウトプットフォーマットAPIを、すべてのプロバイダに対して独自にそして一般的に識別する。

- プロバイダが提供するAPIは、サポートするVMのフォーマットを、すべてのプロバイダに対して独自にそして一般的に識別する。
- プロバイダのAPIは、サポートする複数のVMのインプットフォーマットをリストとして、すべてのプロバイダに対して独自にそして一般的に識別する。
- プロバイダAPI識別子は独特であり、すべてのAPI表現と整合性を持つ。
- プロバイダAPIのVMインプットおよびアウトプットフォーマット識別子は独特であり、すべてのプロバイダに対して整合性を持つ。
- 報告されたVMインプットおよびアウトプットフォーマットは、対称的、同等および一致した順番でなくてもよい。

2.14. クラウドプロバイダのデータセットインプットおよびアウトプットフォーマットのサポートを、クラウドコンシューマが知る事

クラウドコンシューマは、クラウドプロバイダが認証し提供するデータセットのインプットおよびアウトプットフォーマットを知りたいことを望んでいる。

機能要件

- プロバイダは、安全ではないネットワーク接続上で利用できるAPIを提供する。
- プロバイダは、安全なネットワーク接続上で利用できるAPIを提供する。
- プロバイダが提供するAPIは、すべてのコンシューマ認証と承認レベルに使用できる。
- プロバイダが提供するAPIは、サポートするデータセットのインプットフォーマットAPIを、すべてのプロバイダに対して独自にそして一般的に識別する。
- プロバイダが提供するAPIは、サポートするデータセットのアウトプットフォーマットAPIを、すべてのプロバイダに対して独自にそして一般的に識別する。
- プロバイダが提供するAPIは、サポートするデータセットのフォーマットを、すべてのプロバイダに対して独自にそして一般的に識別する。
- プロバイダAPIは、サポートする複数のデータセットのインプットフォーマットをリストとして、すべてのプロバイダに対して独自にそして一般的に識別する。
- プロバイダAPI識別子は独特であり、すべてのAPI表現と整合性を持つ。
- プロバイダAPIのデータセットインプットおよびアウトプットフォーマット識別子は独特であり、すべてのプロバイダに対して整合性を持つ。
- 報告されたデータセットインプットおよびアウトプットフォーマットは、対称的、同等および一致した順番でなくてもよい。

3. OCCIの要件

3.1. 機能要件

本章では機能要件を説明する。要件を次の表に分類し、優先度を付けた。

表1. VM記述の機能要件

ID	説明	ユースケース	優先度
A.1.1	メモリ、CPU、ディスク、ネットワークを定義するための属性が利用できなくてはならない。	2.2、2.3、2.6	高
A.1.2	地理的場所等の配置の制約を定義する属性がサポートされなければならない。	2.2	中
A.1.3	移行がインフラストラクチャによってサポートされるかを属性は実証すべきである。	2.2	中
A.1.4	APIは完全にクラスターを表現できるべきである (たとえばVM、各VMのストレージ、2つのネットワーク (マシンに接続したプライベートネットワークおよびロードバランサーにも接続するパブリックインターネット)、パブリックインターネット上のウェブサイト用固定静的IP等)。	2.9	高
A.1.5.	プロビジョニングリクエストで宣言される属性に (非機能、機能) 制約を加える方法。	2.1	高
A.1.6.	リソース実行のスケジューリングをサポートする。プロビジョンされたリソースがオリジナルのリクエストによって今後いつか実行されるようにする。	2.1	中
A.1.7.	一般的なオペレーティングシステムがサポートされるべきである。	-	高
A.1.8.	プロバイダポリシーに従ってリソースが分類されるべきである。	-	高
A.1.9.	その次に、新しいリソースをリクエストし、リクエストは完全に実現/また記述されなければならない。	-	高

表2. VMマネジメントの機能要件

ID	説明	ユースケース	優先度
----	----	--------	-----

A.2.1.	VMを開始、停止、一時停止、再開する方法が利用できなくてはならない。	2.1, 2.2, 2.3, 2.5, 2.11, 2.10	高
A.2.2.	障害時の自動管理がサポートされなければならない。	2.1, 2.7	低
A.2.3.	各バックアップディスクおよびイメージにIDを提供する。	2.2	高
A.2.4.	バックアップをダウンロードする方法を提供する。	2.2	中
A.2.5.	APIは次の動作を強化する機能を提供すべきである。デプロイ、シャットダウン、キャンセル、チェックポイント、セーブ、リストア、ポール (をモニタリングと統合しうる)。	2.3	高
A.2.6.	状態モデルには次を含むべきである。ペンディング、ブーティング、ランニング、サスペンディッド、シャットダウン、キャンセル、フェイルド。	2.3	中
A.2.7.	エンタリーごとにすべてのプロパティをリストしなくても、リスティングの収集が可能でなければならない。	2.4	中
A.2.8.	リソース表現が更新されるようにし、その変化がVM上のイベント/変更を起こすようにする。	-	低
A.2.9.	ターミナル、ウェブ、デスクトップおよび自動マネジメントインタフェースをサポートする。	2.10	低
A.2.10.	物理的リソースからクラウドへ、クラウドから別のクラウドへ、仮想リソースからクラウドへのリソースの移行をサポートする (相互運用性に関する議題である)。	-	中
A.2.11.	現在のIaaSベースのクラウド (Amazon EC2等) の、すべての機能のサブセットをサポートする。	2.6	中
A.2.12.	多くのクラウドサービスプロバイダがサポートできる一般的なインタフェースを (インフラストラクチャおよびデータイン	2.13, 2.14	中

	タフェースに関して) 使用すべきである。		
--	----------------------	--	--

表3. ネットワークマネジメントに関する機能要件

ID	説明	ユースケース	優先度
A.3.1	VPNの作成をサポートする。	2.3	低
A.3.2.	複数のネットワーク接続をサポートする (パブリックおよびプライベート)	2.1, 2.2, 2.3	高
A.3.3.	ランタイム時にIPのアタッチや変更がで きななければならない。	2.3, 2.7	中
A.3.4.	複数のネットワーク接続に対するタギン グメカニズムをサポートする。	2.1, 2.2, 2.3	低
A.3.5.	「インタークラウド」セットアップができ るネットワークセットアップをサポート する (インテグレーションに関連する)。	-	中

表4. ストレージマネジメントの機能要件

ID	説明	ユースケース	優先度
A.4.1	マウントポイントでのURIの使用を許可 する。ストレージサービス製品の再使用を 許可する。	2.1	高
A.4.2	ランタイム時に追加のストレージリソー スのアタッチメントを許可する。	-	中

表5. イメージマネジメントの機能要件

ID	説明	ユースケース	優先度
A.5.1.	ディスクイメージをレジスタ、アップロー ド、アップデート、ダウンロードできる方 法が利用できなければならない。	2.2	中
A.5.2.	Rynscコマンドに基づく物理的に同等な マシンを同期化するアップデートがサポ ートされなければならない。	2.7	中
A.5.3.	アップロードが問題なく終了したときに、 識別子を返さなければならない。	2.2	低

表6. 識別/参照

ID	説明	ユースケース	優先度
A.6.1.	VMイメージおよびそのコンポーネントに対する独自のIDが使用できない。	2.2, 2.13, 2.14	高
A.6.2.	リソースとそのコンポーネントにタグできない。	2.10, 2.12	中
A.6.3.	タグ等に基づきリソースを検索できない。	2.10, 2.12	中

表7. モニタリング

ID	説明	ユースケース	優先度
A.7.1.	ネットワーク、VM等のエレメントのステータスをリクエストするプル型のモニタリングをサポートする。	2.1, 2.2, 2.3	中
A.7.2.	VMまたはネットワークで起こるイベントをリクエストするパブリッシュ/サブスクライブ・パターンをサポートする (たとえば、あるコンポーネントのエラー、VM状況の変更、その他定期的な通知など)。	2.2	中
A.7.3.	サーバ、ドライブ等のリストをポーリングするための簡潔で迅速なコールを定義する属性をモニターできない。	2.4	低
A.7.4.	ハイパーバイザーからVMのリソース消費に関する属性をモニターできない(CPU、メモリ等)。	2.1, 2.2	中
A.7.5.	マネジメントレポートをXML、PDFいずれかのフォーマットで作成しない。	-	低

3.2. 非機能要件

本章はすべての非機能要件を説明する。

表8. セキュリティ要件

ID	説明	ユースケース	優先度
B.1.1.	X509証明書の使用をサポートする。	2.3, 2.13, 2.14	高
B.1.2.	ACLの使用をサポートする。	B.1, 2.1	高
B.1.3.	セキュリティレベルを定義する属性が記	2.1	高

	述内で利用できなくてはならない。		
B.1.4.	転送とユーザレベルセキュリティが備わっていないといけない。	2.1, 2.13, 2.14	高
B.1.5.	地理的領域の特定を許可する。	B.4	高

表9. サービスの品質

ID	説明	ユースケース	優先度
B.2.1.	リカバリ/フェイルオーバーの場合のキャパシティ要件をサポートする。	2.7	低
B.2.2.	サービスの品質 (QoS) レベルを定義する VM記述の属性をサポートする (レスポンスタイムも含む)。	2.1	高
B.2.3.	アイソレーションレベルを記述するVMの属性をサポートする。	2.1	中
B.2.4.	高度な予約機能の属性をサポートする。	2.3	低
B.2.5.	VMのレスポンスタイムの特定を許可する。	B.4	高

表10. シンタックス

ID	説明	ユースケース	優先度
B.3.1.	エンドユーザが開発ツールやライブラリを必要としてはならない。	2.8	中
B.3.2.	Ajaxインタフェースをサポートする簡潔なJSONシンタックスをサポートする。	2.4, 2.10	中
B.3.3.	単位 (MB、GB等) の明確な定義を (IEC60027-2A.2が規定する単位のように) リクエストで使わなくてはならない。	A.2, 2.4	中

表11. バックアップ/ディザスタリカバリ

ID	説明	ユースケース	優先度
B.4.1.	クラウドリソースのバックアップ機能をサポートする。	-	低
B.4.2.	インタフェースはフェイルオーバー、ディザスタリカバリおよびビジネス継続プランを再考すべきである。	-	中

4. クラウドAPI 特徴マトリックス

IaaSベースのクラウドの既存API はすでにいくつかのユースケースを実現し、対処している要件もある。次のAPIのリストはOCCIワーキンググループプロセスの作成時に評価されたものである。

- OCCI – オープンクラウドコンピューティングインタフェース
- EC2 – Amazon Elastic Compute Cloud
- EH - ElasticHosts
- FS - Flexiscale
- GG - GoGrid
- SC – Sun Cloud API
- CS – Rackspace Cloud Servers
- VM - VMware vSphere

次のマトリックスはAPIの完全な評価でも詳細な評価でもないが、API比較の際に基本的な基準点として使用することができる。

表12. クラウドAPI 特徴マトリックス

特徴	要件	OCCI	EC2 ^a	EH ^b	FS ^c	GG ^d	SC ^e	CS ^f	VM ^g
HTTP経由の認証	クライアント互換性	OK	N/A	OK	N/A	N/A	OK	N/A	N/A
リクエストサイ ン経由の認証	信用できない第三 者のリクエスト	OK	OK	N/A	N/A	OK	N/A	N/A	?
Ephemeralスト レージリソース	軽いサーバ	OK	OK	OK	N/A	N/A	N/A	N/A	?
パーシステント コンピュートリ ソース	停止したサーバ	OK	N/A	N/A	OK	OK	OK	OK	OK
Ephemeralスト レージリソース	一時的ストレージ	OK	OK	N/A	N/A	N/A	N/A	N/A	N/A
パーシステント ストレージリソ ース	永久ストレージ	OK	OK	OK	OK	OK	OK	OK	OK
複数のストレ ージリソース	複雑なアーキテク チャ	OK	OK	OK	N/A	N/A	N/A	N/A	OK
複数のネットワ ークリソース	複雑なアーキテク チャ	OK	N/A	OK	N/A	N/A	OK	N/A	OK

静的IP	インターネットフ ェーシングアプリ ケーション	OK	OK	OK	OK	OK	OK	OK	OK
ファイアウォー ル構築	ベーシックネット ワークセキュリテ ィ	?	OK	N/A	OK	OK	N/A	N/A	OK
ロードバランシ ング	水平拡張性	?	N/A	N/A	N/A	OK	N/A	OK	N/A
ビルディング	ビジネス	?	N/A	OK	OK	N/A	N/A	N/A	?
リソース分類	リソースの組織	OK	N/A	N/A	N/A	N/A	N/A	N/A	?
リソースサーチ	リソース発見	OK	N/A	N/A	N/A	N/A	N/A	N/A	?
リソースタギン グ	リソースの組織	OK	N/A	N/A	N/A	N/A	OK	N/A	?
コレクション (参照渡し)	列挙内容	OK	N/A	OK	OK	N/A	N/A	N/A	?
コレクション (値渡し)	シリアライゼーシ ョン	OK	OK	OK	OK	OK	OK	OK	?

a Amazon Elastic Compute Cloud - <http://aws.amazon.com/ec2>

b ElasticHosts API - <http://www.elastichosts.com/products/api>

c Flexiscale API - <http://api.flexiscale.com/current/doc>

d GoGrid API - <http://wiki.gogrid.com/wiki/index.php/API>

e The Sun Cloud API - <http://www.kenai.com/projects/suncloudapis/pages/Home>

f Cloud Server API - http://www.rackspacecloud.com/cloud_hosting_products/servers/api

gVMware vSphere - <http://www.VMware.com/products/vsphere>

5. 結論

前章までに、APIに対してどの程度、クラウド要件が存在するかを記載した。(意味がわかりません。) 現時点で不足しているのは、多くのニーズを満たし、実装が簡単で、高度にレスポンス(スループット)であり、世界中で適用可能(動的接続だけでなく周知の規格)であり、安全で、広く認識されている(一般に知られていて実装されている)、そして最後に大事なことであるが、標準化された一般的なソリューションである。

本ワーキンググループはこれらのユースケースおよび要件をクラウドAPIの作成に使用する予定である。

同様な作業がクラウドコンピューティングユースケースグループによって行われてきた。本文書は「クラウド・コンピューティング・ユース・ケース ホワイト・ペーパー」(CCUCW)とし

て発行される。

6. 寄稿者

次表に記す方々から要件の収集およびユースケースの文書化に協力を得た。

表13. 寄稿者一覧

氏名	所属
Michael Behrens	R2AD
Richard Davies	ElasticHosts
Andy Edmonds	Intel - SLA@SOI project
Sam Johnston	Australian Online Solutions
Ignacio M. Llorente	DSA-Research at UCM – RESERVOIR project
ThijSMetsch (Editor)	Sun Microsystems - RESERVOIR project
Gary Mazzaferro	OCCI Counselour - Exxia, Inc.
George Reese	O'Reilly
Luis Rodero-Merino	Telefónica I+D
Shlomo Swidler	MyDrifts
Tino Vázquez	DSA-Research at UCM – RESERVOIR project

7. 知的財産権について

本文書に記載された技術の実装や使用に関連すると考えられるいかなる知的財産権またはその他の権利について、OGFは、その有効性や範囲に関して特定の立場をとるものではない。また、こうした権利下にあるいかなるライセンスについても、それが使用できるあるいは使用できない範囲について特定の立場をとるものではない。さらに、これらのいかなる権利についても特定する努力をOGFが行うものではない。出版やライセンス保証のために用意した権利請求書、および、開発者や本仕様書の使用者が所有権を使用する上での一般的なライセンスや許可を取得する作業の結果については、OGF事務局から入手可能である。

関係者の方々は、本文書で薦められていることを実践する上で必要なあらゆる著作権、特許、特許利用、その他の所有権に対して注意を向けるようOGFは希望する。情報はOGF委員長までお寄せいただければ幸いである。（本当にこういう意味ですか？）

8. 免責条項

本文書と本文書に含まれる情報は現時点のものであり、OGFは、すべての明示あるいは暗示の保証（本文書の情報の使用がいかなる権利をも侵害しないこと、または、市販性、特定目的との適合性に関するいかなる暗示の保証をも侵害していないことを含むが、必ずしもこれらに限定されない）を拒否するものである。

9. 著作権情報

Copyright (C) Open Grid Forum (2009, 2010). All Rights Reserved.

本文書およびその翻訳物は、複製し、他者に提供することができる。本文書に関してコメントや別の説明を与えている派生著作物、あるいは本文書の普及を助ける派生著作物についても、そのままあるいは部分的に、制約なしに作成、複製、発行、頒布が可能である。ただしそのような複製物や派生著作物については、上記の著作権情報と本段落に書かれていることを記載しなければならない。ただし、OGFや他の組織に対する著作権情報や参考文献を削除するなどといった本文書自体の変更は、いかなる形であっても禁止する。なお、グリッド提言を開発する目的で変更が必要ならば、その限りではない。この場合、OGFドキュメントプロセスで決められた著作権に対する手続きを遵守する必要がある。また、英語以外の言語に翻訳する際に変更が必要な場合も、その限りではない。

上に与えた制限付き許諾は永続的なものであり、OGFあるいはその後続組織または委嘱組織が無効にすることはない。

10. 参考文献

[CCUCW] *Cloud Computing Use Cases Whitepaper..*

<http://www.scribd.com/doc/17929394/Cloud-Computing-Use-Cases-Whitepaper>
r. Cloud Computing Use Case Discussion Group 2009-08.