VCP (Virtual Cloud Provider) の mdx サポートに向けた実装と活用例

ASCADE, Inc.
那須野 淳

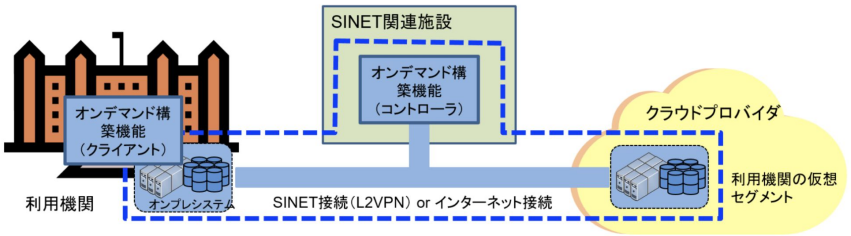
2022/6/1

オンデマンド構築サービスとVCP (Virtual Cloud Provider)

- 構築・管理ソフトウェアVCP
  クラウドプロバイダごとのAPIを抽象化し、制御・管理を容易にするソフトウェア

- オンデマンド構築機能
  VCPと関連ソフトウェアを組み合わせた「Virtual Cloud (VC) コントローラ」を提供

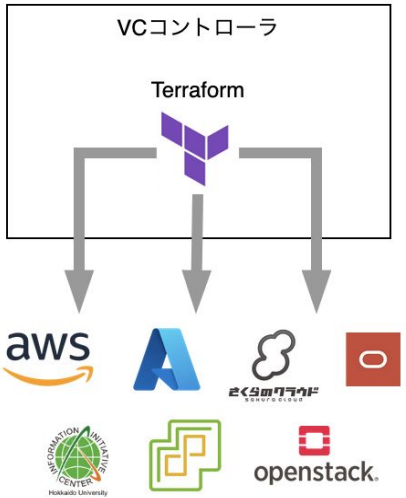
- VCコントローラの主な役割
  - クラウドAPI操作
  - クラウド(間)ネットワーク接続（仮想ルータ）
  - Dockerコンテナでのアプリケーション配備
  - アプリケーション実行環境のモニタリング



VCP から利用可能なクラウド環境

商用、学術機関クラウドなど複数のプロバイダに対応

- Amazon Web Services
- Microsoft Azure
- Oracle Cloud Infrastructure
- さくらのクラウド
- 北海道大学インタークラウドシステム
- VMware vSphere（オンプレミスシステム）
- OpenStack（オンプレミスシステム）



クラウド API に対応した Terraform Provider Plugin が提供されていることが必要

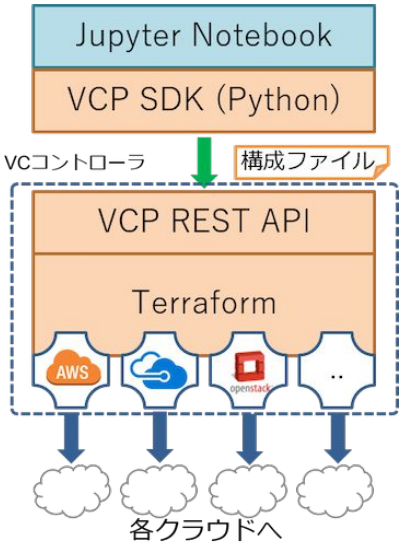
VCP のサービスインターフェース

2種類のインターフェースを提供

- VCP REST API
  - 詳細な構築・管理が可能な REST インタフェース
  - YAML形式でクラウド構成情報を記述して POST
- VCP SDK
  - Python3 ベースの開発キット
  - Jupyter Notebook 環境からの利用も可能

例 (AWSに計算ノードを起動)

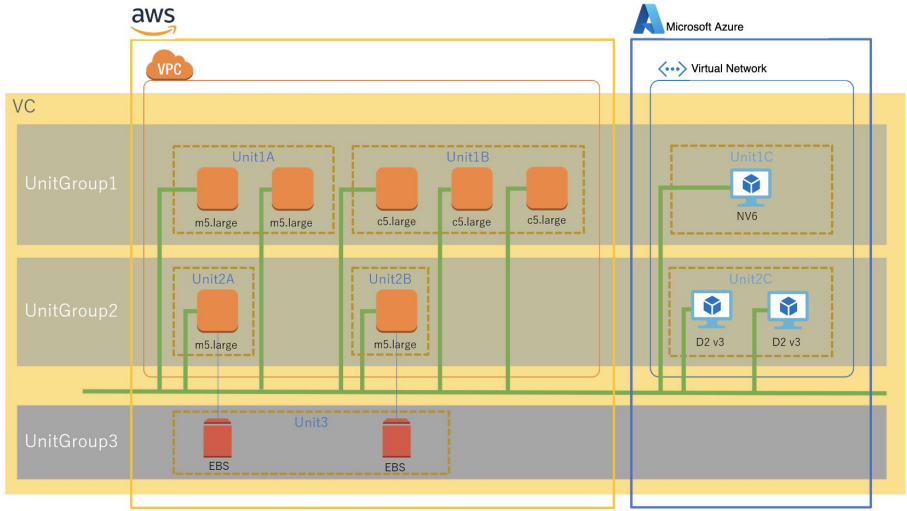
```
vc = VcpSDK("access_token", "my_vc_name")    # SDK初期化
spec = vc.spec.find("aws", "small")          # 性能スペック指定
spec.volume_size = 512                       # スペック追加・修正
nodes = vc.unit.create("test_server", spec) # ノード起動
```



Virtual Cloud (VC) の構成要素

VCP SDK で対応可能なインタークラウド環境でのシステム構築を支援するための構成要素

- VC (Virtual Cloud)
  - 複数クラウドにまたがるひとつの仮想システム環境
- Unit Group
  - 異なる性質の Node群をまとめて扱う（計算資源、ストレージなど）
- Unit
  - 同じスペック (cpu, memory, ...) のNode群をまとめて扱う
- Node
  - 個々のクラウドインスタンス



VCP の mdx 向け機能拡張

目的

mdx 仮想マシン(VM) を VCP で扱う計算資源として利用できるようにする。

- 他のクラウド上で起動したサーバと同様に、 VCP SDK で Unit Group / Unit / Node 操作

提供形態

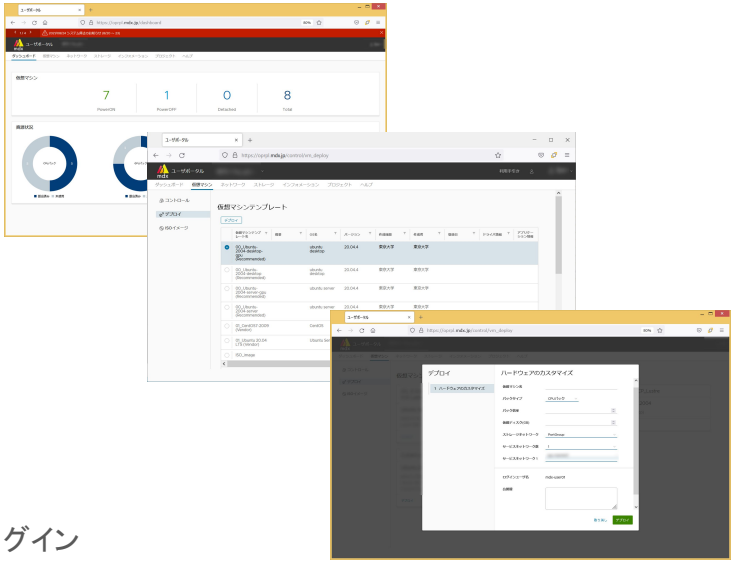
mdx REST API に対応した Python ライブラリを実装 ⇒ “mdx Extension”

- mdx REST API を VCP SDK とともに利用しやすいインターフェース
- mdx REST API でサポートされている機能はひととおりカバー

mdx の基本的な使い方

- ユーザポータルのWeb UI による各種操作
  - 仮想マシン作成・削除
  - 仮想マシン起動・停止
  - ネットワーク管理（e.g. 外部からのアクセス制御）
  - ストレージ管理（内部、大容量、オブジェクト）

- 仮想マシン利用の流れ
  - ダッシュボードで利用可能な資源量を確認
  - 仮想マシンテンプレート（ OSイメージ）一覧から選択
  - ハードウェア・スペック、接続先ネットワークなどを指定
  - デプロイ実行
  - 仮想マシンの起動状態を確認
    - IPアドレスが付与（起動から約 5分）されたら、SSHでログイン



mdx REST API

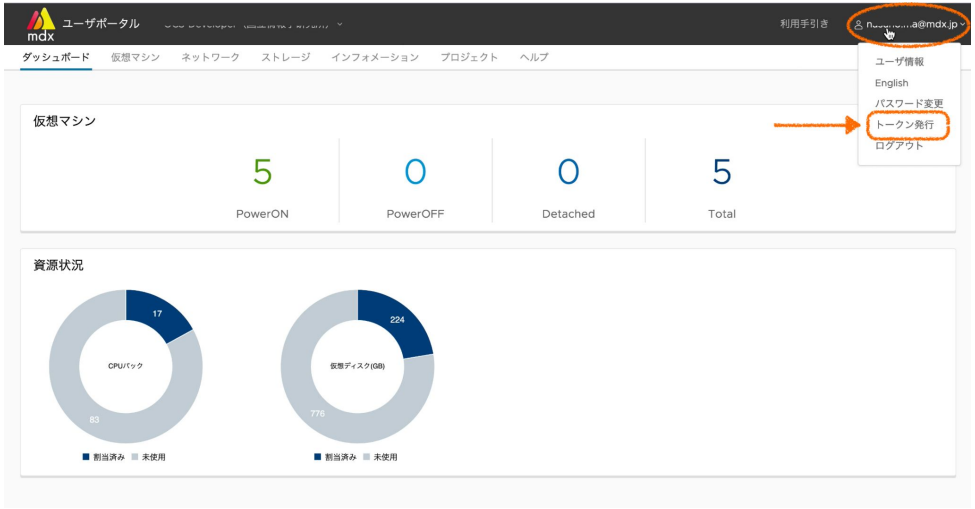
プログラムからのmdx 操作に対応するため、ユーザポータル機能として新たに開発された。

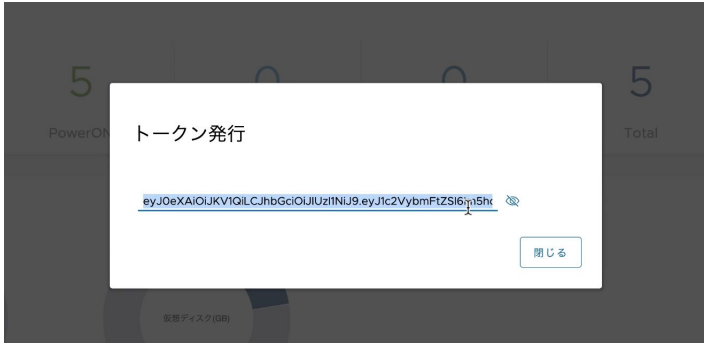
- 主な mdx REST API 機能
  - トークンベース認証
  - プロジェクト、ネットワークセグメント情報取得
  - カタログ（仮想マシンテンプレート）情報取得
  - 仮想マシン デプロイ
  - 仮想マシン 情報取得
  - 仮想マシン OSシャットダウン、再起動
  - 操作履歴取得（プロジェクト毎、仮想マシン毎）
  - Allow ACL、DNAT 設定

- 2022/06 現在、REST API を利用するためにはプロジェクト単位での申請が必要

mdx REST API 認証トークンの取得方法

REST API の認証トークンはユーザポータルから取得可能





mdx REST API 使用例（仮想マシンのデプロイ）

POST Request `/api/vm/deploy`

```
curl https://oprpl.mdx.jp/api/vm/deploy/ \
  -H 'Content-Type: application/json' \
  -H 'Authorization: JWT eyJ0eXAiO...' \
  -d '
{
  "catalog": "16a410...",
  "pack_type": "cpu",
  "pack_num": "4",
  "disk_size": 50,
  "gpu": "0",
  "network_adapters": [
         {
         "adapter_number": 1,
         "segment": "e6dcf..."
         }
  ],
  "os_type": "Linux",
  "power_on": true,
  "project": "03c87...",
  "shared_key": "ssh-ed25519 ...",
  "storage_network": "portgroup",
  "template_name": "UT-20220123-1511-ubuntu-2004-server",
  "vm_name": "testvm01"
}'
```

← 認証トークン

← 仮想マシンテンプレート

← パック数でCPU, メモリ指定

← 接続先ネットワーク

← SSH 公開鍵

Response

```
{
  "task_id": [
    "42acf275-fb10-44d4-aee3-cda91a19d1c0"
  ]
}
```

※ デプロイの進行状況は、返却された `"task_id"` をキーに操作履歴 API `/history/project/` を用いて確認する。

VCP 対応 mdx Extension ライブラリの利用手順

1. 初期化
   - VCP SDK の mdx 拡張モジュールを import
   - 認証トークン文字列を引数に指定して初期化

```
from vcpsdk.plugins.mdx_ext import MdxResourceExt
mdx = MdxResourceExt(mdx_token)
```

2. VMのデプロイに必要なパラメータを用意
   - ネットワークセグメント ID
   - CPUパック数
   - ディスクサイズ
   - SSH公開鍵

3. 操作対象のプロジェクト設定
   - 認証ユーザのプロジェクト情報を取得
   - 操作対象のプロジェクト IDを設定

```
projects = mdx.get_assigned_projects()
id = projects[0]["projects"][0]["uuid"] #先頭IDを利用
mdx.set_current_project_id(id)
```

4. mdx VM デプロイ実行
   - VM名、VM spec (dict型) を引数に指定
   - IPアドレス割当完了まで待つには "wait_for" 指定

```
vm = mdx.deploy_vm(name, spec, wait_for=True)

# VMに付与されたIPv4プライベートアドレス
ip_addr = vm["service_networks"][0]["ipv4_address"][0]
```

mdx VM を VC Node として利用するには

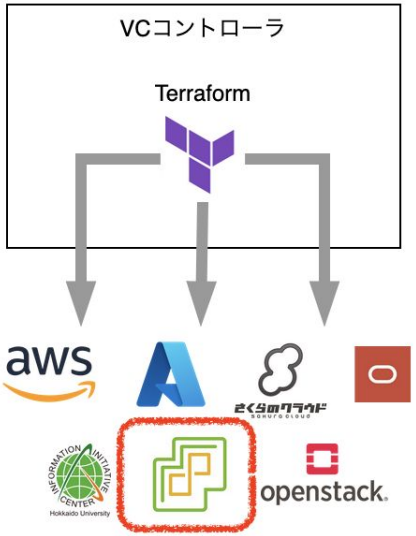
やりたいこと

- mdx 以外のクラウドと同様にVC (Virtual Cloud) として扱いたい
- VC Node = VCP SDK から制御可能なクラウドインスタンス

mdx における制約

- mdx REST API ≠ vSphere REST API
  - mdx の仮想化基盤ソフトウェアは VMware vSphere だが、vSphere REST APIの（mdx ユーザからの）直接利用は不可
  - *cf.* github.com/hashicorp/terraform-provider-vsphere

- VCコントローラ/ mdx VM 間のプライベートネットワーク接続
  - SINET L2VPN または IPsec 接続の環境が前提



VMware vSphere
オンプレミスシステム向けにVCP 対応済

mdx VM を VC Node として利用するには

- VCP 既存サーバ(SSH) モード
  - Terraform Provider Plugin 非対応の環境でも VCP を利用可能
  - 稼働中の VM やベアメタルサーバに対して、VC コントローラから SSH で VC Node をデプロイする

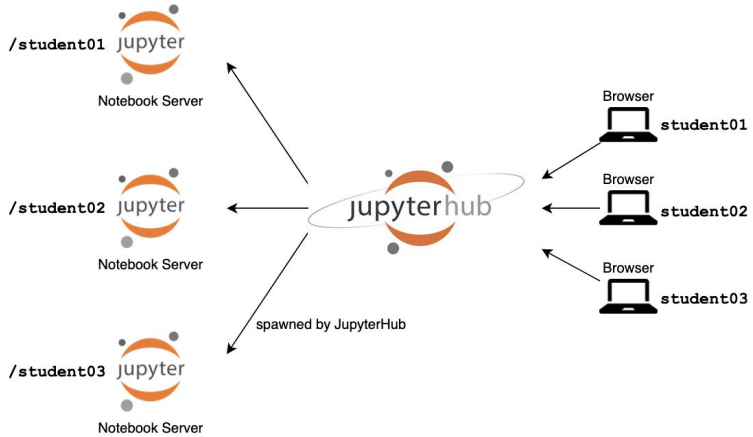
- mdx VM の場合
  - mdx VM をデプロイ ⇒ VCP 対応 mdx Extension 活用
  - VCP 既存サーバ (SSH) モードが使える状態に mdx VM をセットアップ
    - 具体的には、SSH 設定と Docker CE のインストールで OK
  - VCP 既存サーバ (SSH) モードで VC Node 作成
    - ポータブル版 VCP (※) 用 mdx VM を事前に準備しておく

      (※) OCS を利用せず、自ネットワーク内でVC コントローラを運用可能なツールセット

mdx での VCP アプリケーション構築例

CoursewareHub = JupyterHub + 講義のためのモジュール群

- Jupyter Notebook を用いた講義演習環境
  - 演習実施ワークフローをサポート
    （受講生登録、教材配布、演習回答の回収）
  - 受講生の作業履歴、演習実施履歴収集機能
- Pythonプログラミング実習などでの活用実績
  - 群馬大学、室蘭工業大学において 100名規模の
    授業で利用されている
- ソースコード公開
  - github.com/NII-cloud-operation/CoursewareHub-LC_platform

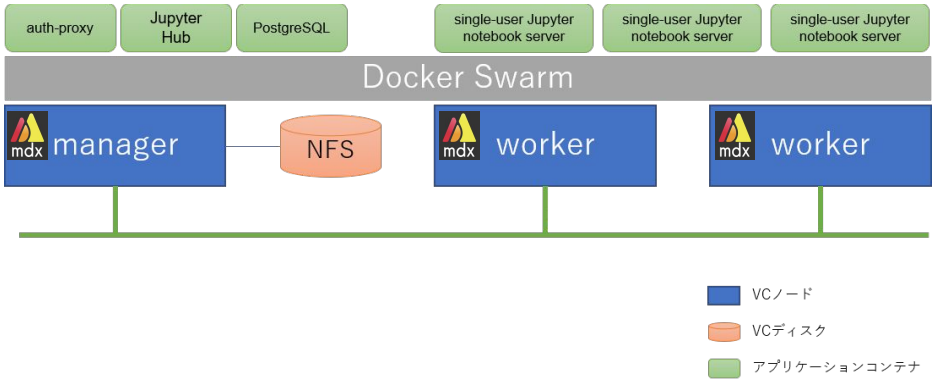


mdx での VCP アプリケーション構築例

CoursewareHub テンプレート

- OCS の公開テンプレートを利用
  - github.com/nii-gakunin-cloud/ocs-templates
  - Jupyter Notebook から VCP SDK を利用

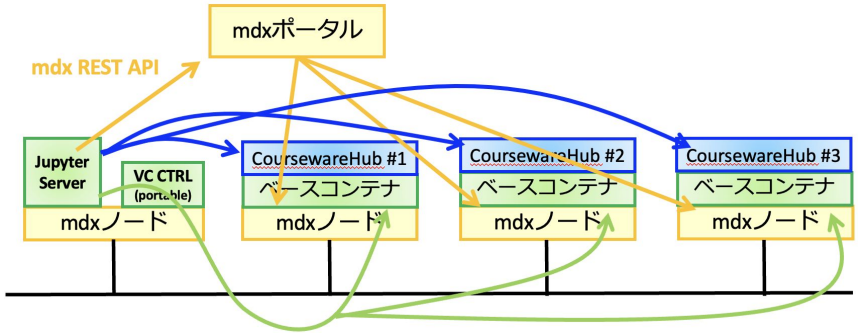
- CoursewareHub ノードの役割に応じて mdx VM を配備
  - manager
    JupyterHub などの System コンテナ実行
  - worker
    ユーザ毎の Notebook server コンテナ実行



デモ：CoursewareHubテンプレートによる環境構築

構築の流れ

1. ポータブル版VCP 用 VM デプロイ
   - ユーザポータルにログイン、VM 1 個作成
   - mdx VM 上でセットアップスクリプト実行
   - Jupyter Notebook から VCP SDK 初期設定
2. CoursewareHub ノード用 VM デプロイ
   - VCP SDK の mdx Extension Library を利用
   - manager + worker * 2 の合計3ノード
3. CoursewareHub 公開テンプレート実行
   - VC 作成（mdx VM を VC Node として管理）
   - JupyterHub など CoursewareHub を構成する各コンテナのセットアップ
4. CoursewareHub の管理作業
   - 管理者アカウントで CoursewareHub にログイン
   - 演習実施ワークフローに沿った作業



まとめ

- mdx REST API に対応したPython ライブラリを実装し、CLIプログラムからmdx の各種操作が可能となった。

- ポータブル版VCP を活用し、mdx の仮想マシンをVC Node（= VCP SDK で制御可能な計算資源）として扱えることを確認した。
  - サービス版 VCP での利用について、mdx プロジェクト VLAN の SINET 接続 (L2VPN) により動作確認予定

- CoursewareHub（Jupyter Notebook を用いた講義演習環境）のOCS テンプレートをmdx 上で実行し、再現性のある環境構築が可能なことを確認した。
  - 次週 6/9 開催 OCS セミナーのハンズオン環境として活用