

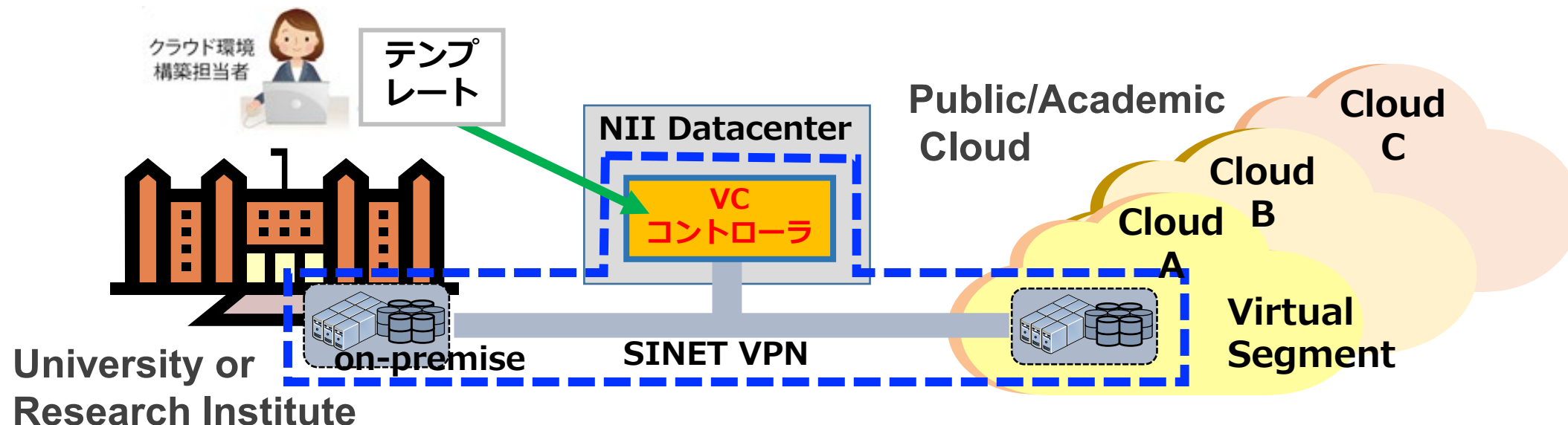
オンデマンド構築サービスの仕組みと環境構築概要

2025年4月23日
大江 和一

国立情報学研究所
クラウド基盤研究開発センター

オンデマンド構築サービスの仕組み

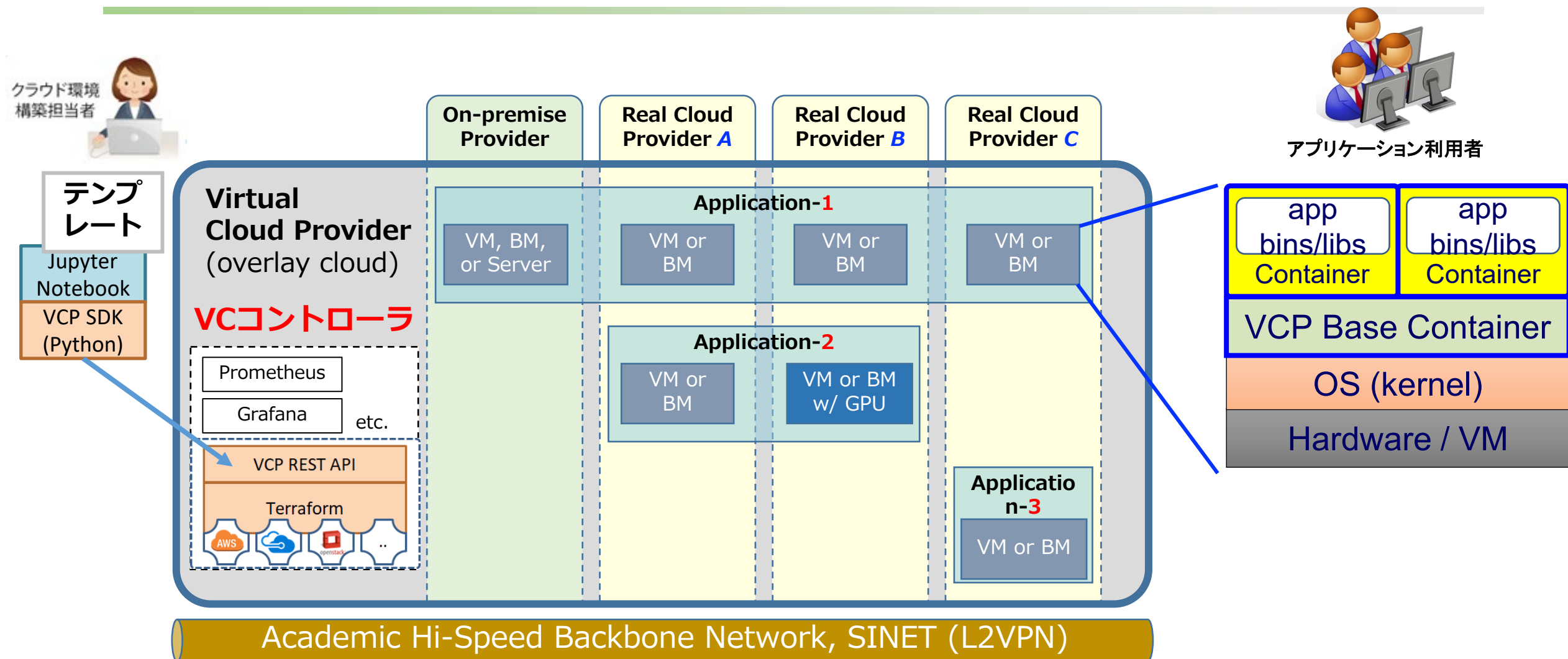
VCPの概要



■テンプレートを用いて、オンプレミスやクラウド(IaaS)上にアプリケーション実行環境を構築

- 仮想プライベートネットワーク（VPN）内に利用する資源を囲い込み、仮想コントローラ（VCコントローラ）から操作することで、全ての資源を統一的に利用できる。
- VCコントローラの操作は、可読性が高いテンプレート（JupyterNotebook）からの操作が可能。

VCPの概要



アプリケーションテンプレート

他者が作ったテンプレートの流用も可能

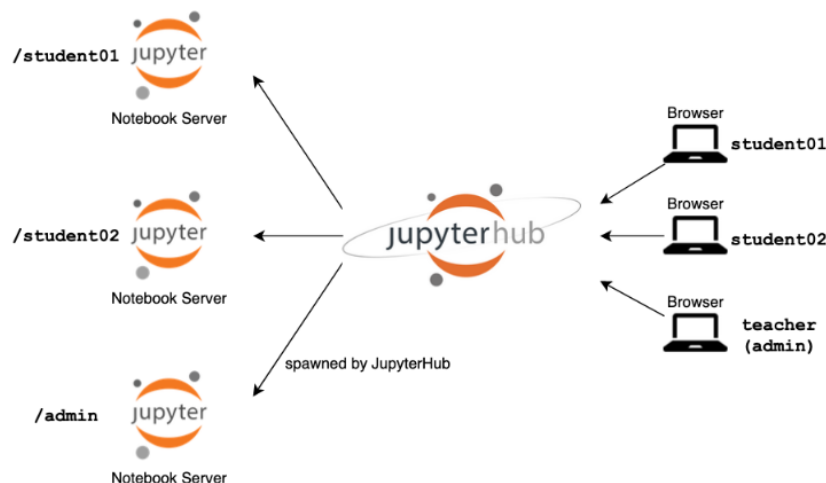
The Littlest JupyterHub による軽量Python実習環境の構築

JupyterHub は、Webブラウザからアクセス可能なマルチユーザ対応の認証機能付きJupyterNotebookサーバです。

JupyterHubを利用して管理者が用意したNotebookをユーザがブラウザからすぐに実行可能な環境を提供できるため、Pythonによるプログラミング研修やワークショップを開催したり、講義演習環境として活用したりするのに適しています。

本ハンズオンでは、JupyterHubを小規模なグループで手軽に利用することを想定し、単一のサーバで実行するために開発された「The Littlest JupyterHub」（以下「TLJH」と略）をVCPを用いて構築します。

ハンズオンご参加の皆様には、このテンプレートでTLJHによるVCPアプリケーション環境を構築していただきます。



構築環境情報の入力

TLJH環境の構築情報を入力します。必要に応じ、下記の情報を修正してください。

★ハンズオンでは以下のパラメータを変更しないでください★

```
#####
##### ハンズオンでは以下のパラメータを変更しないでください。 #####
#####
# UnitGroupの作成
ugroup_name = 'hands-on403'
# プロバイダ
vc_provider = 'aws'
```

VCノードのspecを指定する

TLJHを利用するのに十分な性能・容量のノードspecを指定します。

固定割り当てIPアドレスは、ハンズオン環境のNAT Proxyサーバに予め設定されているIPアドレスを使います。

```
In [ ]: # UnitGroup の作成
unit_group = vcp.create_ugroup(ugroup_name)

# VCノード spec
spec = vcp.get_spec(vc_provider, vcnode_flavor)

# spec オプション (ディスクサイズ 単位:GB)
spec.volume_size = volume_size

# spec オプション (固定割り当てIPアドレス)
spec.ip_addresses = [fixed_ipaddress]

# ssh keyfiles
import os
ssh_public_key = os.path.expanduser("~/ssh/id_rsa.pub")
spec.set_ssh_pubkey(ssh_public_key)
```

Unitの作成とVCノードの起動

Unitを作成します。Unitを作成すると同時にVCノード（ここではAmazon EC2インスタンス）が起動します。処理が完了するまで1分半～2分程度かかります。

```
In [ ]: # Unitの作成 (同時に VCノードが作成される)
unit = unit_group.create_unit('tljh-node', spec)
```

疎通確認

まず、sshのknown_hostsの設定を行います。

その後、VCノードに対してuname -aを実行し、ubuntu x86_64 linux が起動していることを確認します。起動していない場合は、spec.image に誤りがあります。本テンプレート下部にある「環境の削除」を実行、spec.image を修正、全てのセルを unfreeze してから、最初から再実行してください。

```
In [ ]: # unit_group.find_ip_addresses() は UnitGroup内の全VCノードのIPアドレスのリストを返します
ip_address = unit_group.find_ip_addresses(node_state='RUNNING')[0] # 今は1つのVCノードのみ起動しているので [0] で最初の要素を取り出す
print(ip_address)

# ssh 設定
!touch ~/.ssh/known_hosts
!ssh-keygen -R [ip_address] # ~/.ssh/known_hosts から古いホストキーを削除する
!ssh-keyscan -H [ip_address] >> ~/.ssh/known_hosts # ホストキーの登録

# システムの確認
!ssh [ip_address] uname -a
```

TLJH (The Littlest JupyterHub)

VCノード上に、本ハンズ

TLJHコンテナイ

VCノード上にコンテナイメージを取得するために docker pull を実行します。

スクリプトを組み込むことができ、ここから実行できる。実行結果を残すことも出来る。

図表を組み合わせた説明を挿入できる

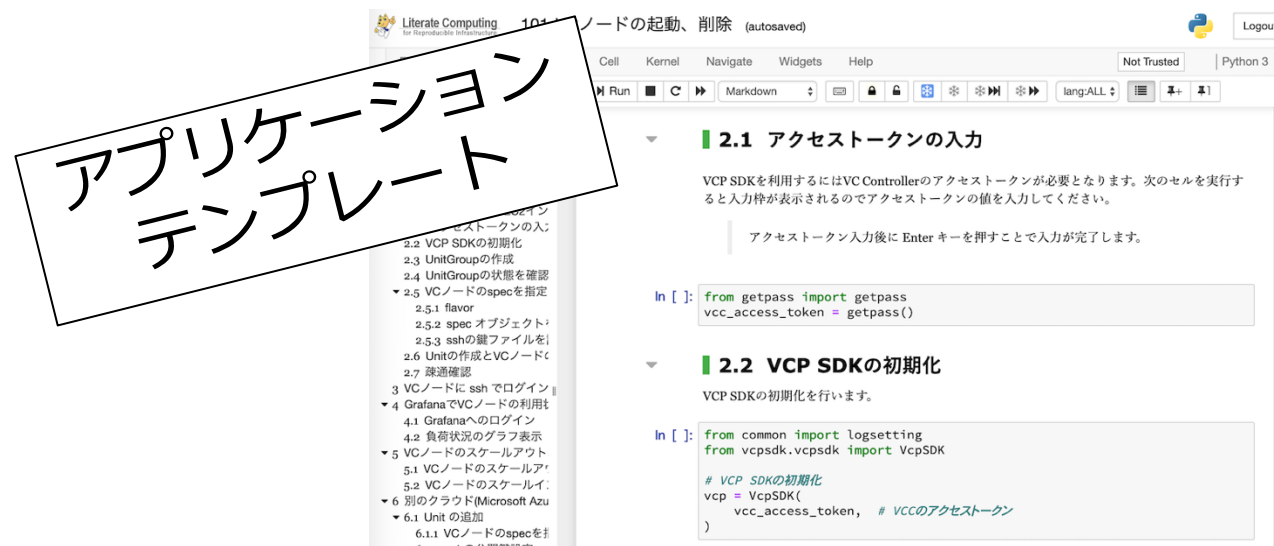
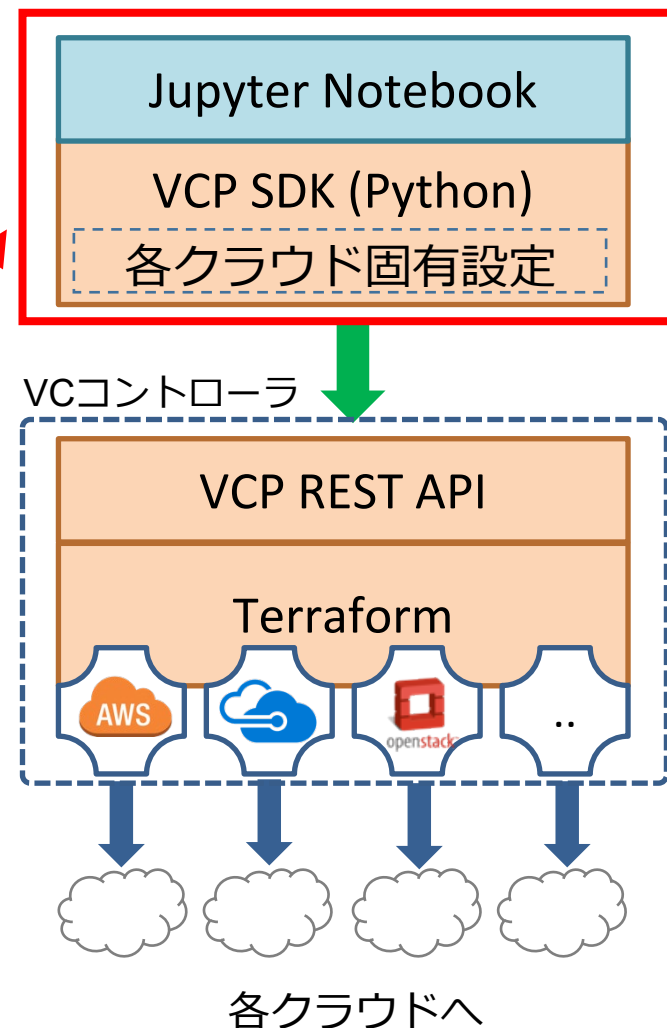
クライアント側からのアクセス方法

■VC利用者

- アプリケーションテンプレート*からVCP SDKを利用してVCP REST APIを操作

■VCコントローラ

- VCP REST API（プロバイダIFを抽象化）からTerraformを介して各クラウドを操作



*: JupyterNotebook (+NII拡張) で記述

VCノード構成

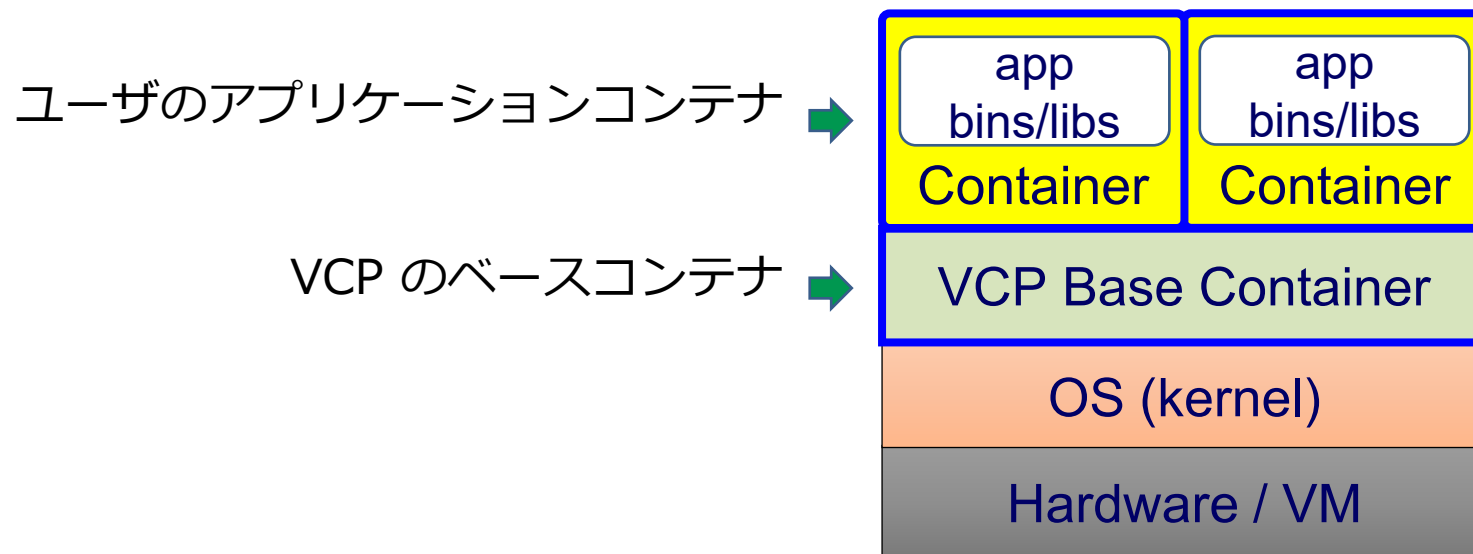
■ Docker in Docker構成

■ ベースコンテナ

- 死活監視やメトリクス収集などシステムの基本機能

■ アプリケーションコンテナ

- アプリケーションと関連ソフトウェアをベースコンテナ上に起動



モニタリング機能

- ベースコンテナ、アプリコンテナのモニタリング情報を提供
 - Prometheus + Grafanaで実現
- アプリケーションの収容設計を支援



VCノード（ベースコンテナ）毎の情報

アプリコンテナ毎の情報

環境構築概要

サービス版とポータブル版

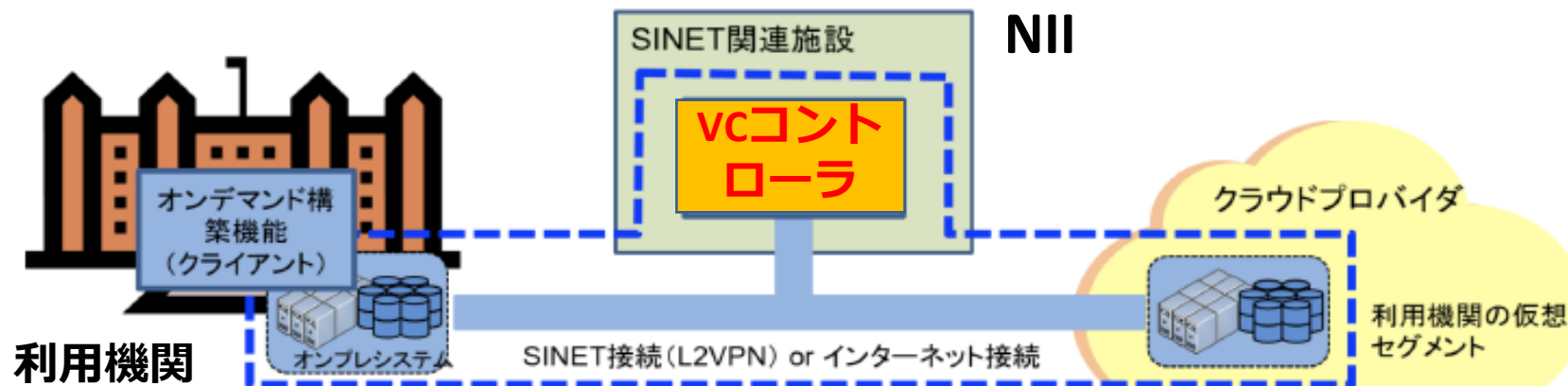
長所：

NII側でVCP運用・保守
仮想ルータが利用可能

短所：

NIIへのVCP構築申請
が必要

サービス版



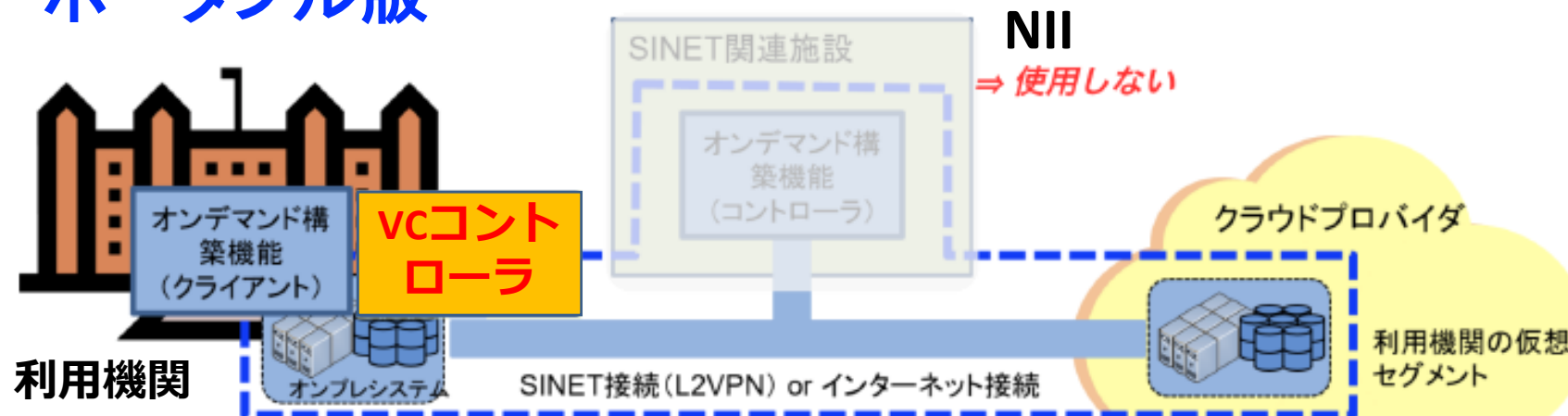
長所：

VCP構築申請が不要と
なり、すぐに利用可

短所：

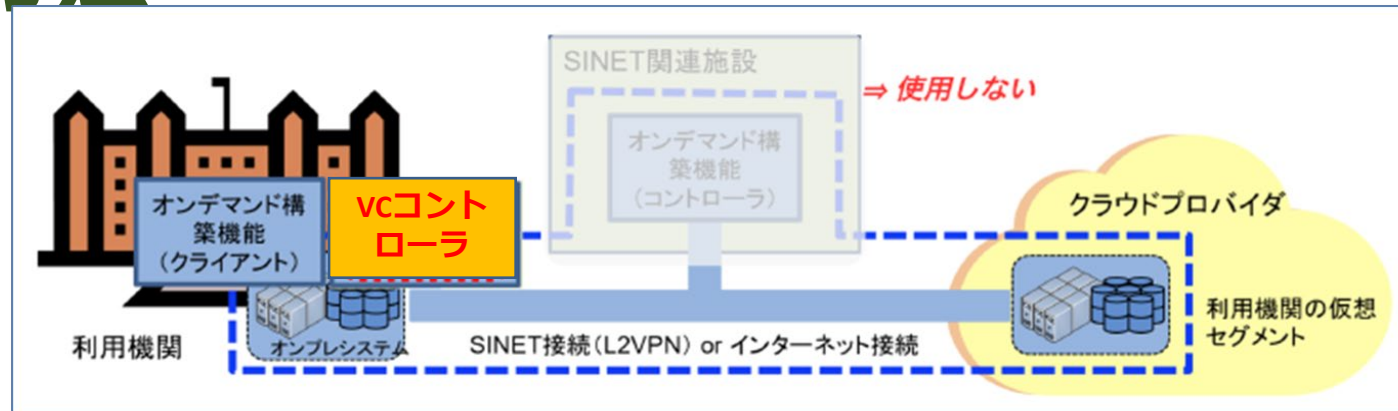
利用機関側でVCP構築・
運用・保守

ポータブル版

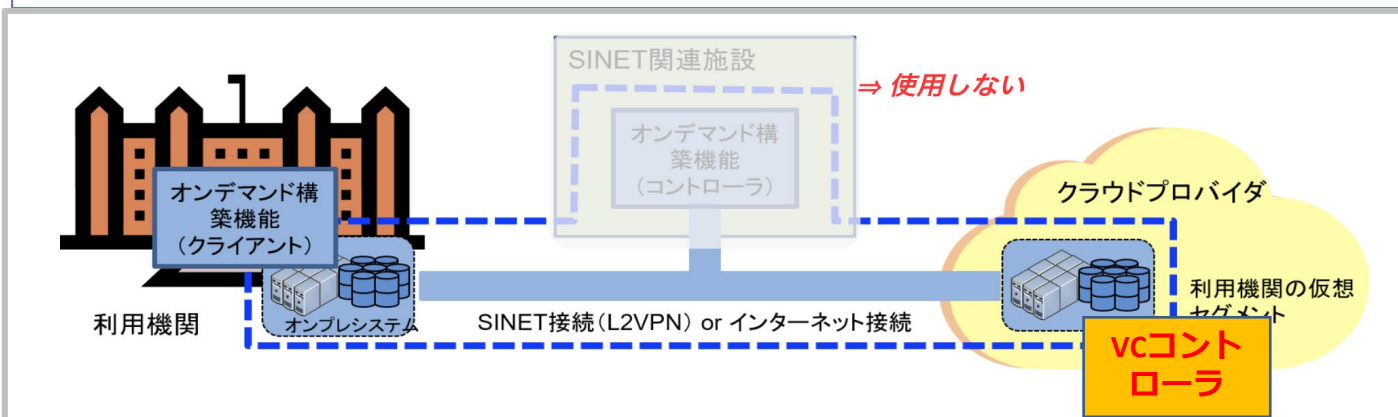


ポータブル版の構成方法

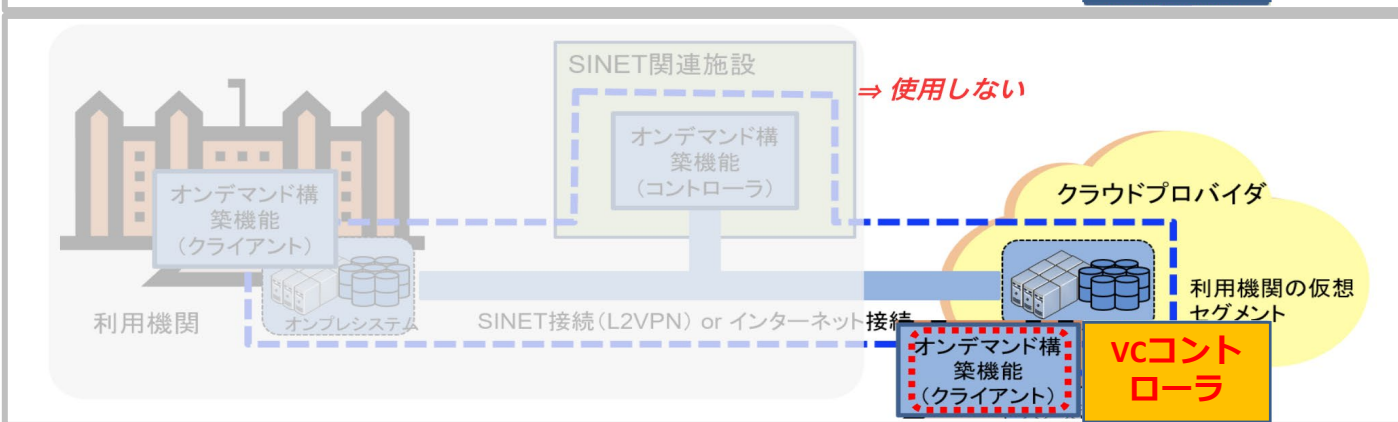
VCコントローラ： 利用機関
JupyterNotebook： 利用機関
(クライアント)



VCコントローラ： クラウド
JupyterNotebook： 利用機関
(クライアント)



VCコントローラ： クラウド
JupyterNotebook： クラウド
(クライアント)



■ 初期導入支援(サービス版)

- 利用機関とクラウドを安全に連携させるための、ネットワーク設定についての技術的相談
 - 含、クラウド設定用スクリプトの提供、画面共有による設定支援

■ オンデマンド構築機能

- 統一した利用方法で異なるクラウドの計算資源の確保、アプリケーションのインストール、及び監視を可能にするソフトウェアを提供

■ 情報共有

- ドキュメント、運用情報、個々の公開テンプレートに対する質疑応答等の情報共有
- リポジトリ(ポータブル版の提供、著名アプリの構築テンプレート・コンテナ、ハンズオンセミナーの教材)

利用について

■ サポートプロバイダ

■ 商用クラウドプロバイダ

■ Amazon Web Services、Microsoft Azure、さくらのクラウド、Oracle Cloud Infrastructure

■ 学術クラウドプロバイダ

■ 北海道大学ハイパフォーマンスインタークラウド サーバサービス、mdx

■ オンプレミスプロバイダ

■ VMware vSphere

■ 利用対象

■ 大学・研究機関などの研究室、学部、機関全体などの組織

■ 教職員個人では申込みません。研究室や所属課等でお申し込みください

■ 利用料金

■ 本サービスは無償です

■ クラウドプロバイダなどの有料サービスは利用者負担です

■ お試し環境

■ ハンズオンの実習参加者向けに1ヵ月間試用できる環境を準備しています

Jupyter Notebook 公開テンプレート

- LMSテンプレート (VCP SDK v20.04以降対応 (AWS、Azureで動作確認済み))
 - Moodleを用いた学習管理システムの構築テンプレート。パスワード認証、Shibboleth認証を利用したMoodleの構築とアップデート手順
- LMSテンプレート簡易構成版 (VCP SDK v20.04以降対応 (AWS、Azureで動作確認済み))
 - 上記LMSテンプレートより機能を絞ったシンプルな構成のMoodle環境の構築テンプレート。認証は手動設定アカウントかLDAP連携を用いた短期的な利用を想定。Shibboleth等のSSO連携や長期利用はカスタマイズが必要。また、VCPを利用せずにAWSまたはAzureに直接LMS環境を構築する手順も公開

Jupyter Notebook 公開テンプレート (続)

- MCJ-CloudHubテンプレート (VCP SDK v21.04以降対応 (AWS、mdxで動作確認済み))
 - 山口大学と共同開発したWeb型プログラミング教育支援システムMCJ-CloudHubの環境構築を行う。JupyterHubとnbgraderをベースとしている。運用には別途Moodle環境も必要。
- 軽量Python実習環境構築テンプレート (VCP SDK v20.04以降対応 (AWS、Azure、mdxで動作確認済み))
 - Pythonによるプログラムの共同開発や講義演習などを行うのに適したJupyterHubの中で小規模グループ用である「The Littlest JupyterHub」の環境構築をおこなう

Jupyter Notebook 公開テンプレート (続)

- 講義演習環境テンプレート (VCP SDK v21.04以降対応 (AWS、Azure、mdxで動作確認済み))
 - Jupyter Notebookを用いた講義演習環境の構築。基盤ソフトウェアには、JupyterHubを講義演習用に NII が拡張したCoursewareHubを使用。教材配布、課題の回答収集、操作履歴の収集等の機能を拡張

Jupyter Notebook 公開テンプレート (続)

- HPCテンプレート v1 (VCP SDK v20.04以降対応 (AWS、Azureで動作確認済み))
 - OpenHPC v1.xで配布されているパッケージを利用して、クラウド上にHPC環境を構築するテンプレート。Slurmを利用したジョブスケジューラやSingularityコンテナ利用環境の設定と、構築したHPC環境で動作可能なベンチマークプログラムも提供
- HPCテンプレート v2 (VCP SDK v21.04対応 (AWS、Azure、Oracle Cloud、mdx で動作確認済み))
 - OpenHPC v2.xで配布されているパッケージを利用して、クラウド上にHPC環境を構築するテンプレート。v1 の機能に加え、GPUノードの利用とNVIDIA社のNGCカタログのコンテナの実行が可能

Jupyter Notebook 公開テンプレート (続)

- HPCテンプレート v3 (VCP SDK v21.04対応 (AWS、Azure、Oracle Cloud、mdx で動作確認済み))
 - OpenHPC v3.1で配布されているパッケージを利用して、クラウド上にHPC環境を構築するテンプレート。
- Open OnDemand構築テンプレート (VCP SDK v21.04以降対応 (mdxで動作確認済み))
 - HPCテンプレートv2で構築したOpenHPC環境上にOpen OnDemand環境を構築する

Jupyter Notebook 公開テンプレート (続)

- 計算資源補完テンプレート (VCP SDK v20.04以降対応 (AWS、Azureで動作確認済み))
 - オンプレミスのバッチ型計算機システムの計算ノード不足時に、クラウド上に同じソフトウェア構成を持つ計算ノードを自動的に立ち上げ、バッチシステムに組み込むクラウドバースト機能を提供。Torque等クラウドに対応していないバッチシステムでも、簡単なプラグインを作成することでクラウドバーストが可能。なお、本テンプレートはipynb形式ではなく、Pythonならびにbashスクリプトで記述されている
- 手書き文字認識システム構築テンプレート (VCP SDK v21.04以降対応 (AWS、Azureで動作確認済み))
 - Open HPC v2テンプレートをベースにGPU ベースの学習システム(Tensorflowを使用)の構築とCPUベースの認識システム(独自仕様)の構築を行い、フロントエンドとしてJupyterNotebook上に手書き数字認識システムを動作させる

Jupyter Notebook 公開テンプレート (続)

URL: <https://github.com/nii-gakunin-cloud/ocs-templates/>

講義演習環境比較

講義演習環境比較

	構築・管理	構築方法	演習支援機能 (課題の配布・回収等)	複数科目情報 等の反映	GPU環境	費用等
Google Colaboratory	不要	不要	なし	なし	あり	無料 有料版を使うとGPU 等が強化される
CoursewareHub	サーバ上に 構築・管理	システム管理者 (教員等)が授業 単位に行う	あり (CUI)	準備中	授業ごとに準備	無料 (クラウド費用は ユーザが負担)
MCJ-CloudHub	サーバ上に 構築・管理	システム管理者が 1 回行う	あり (GUI) 自動採点・フィードバック 機能付き	あり (Moodleと連携)	システムにGPU 環境を準備	無料 (クラウド費用は ユーザが負担)

■ Colabは、自習演習向け

■ CoursewareHubは、**演習と研究を行う教員向け**

- 教員が実装等を理解した上で構築・運用を行う。配布imageのカスタマイズなども自由に行うことが可能。

- OSSなので、構築・運用の負担はそれなりに発生。

■ MCJ-CloudHubは、**学部や大学全体での運用向け**

- システム管理者と利用者（教員・学生）を分離し、利用者はGUIからの操作のみで利用可。
 - システムに詳しくない教員も利用対象
- システム管理者（情シス教員等スキルのある方を想定）も年度ごとに 1 度構築すれば運用可能とすることを目指している。

Thank You.

