

学認クラウドオンデマンド構築サービス (OCS)の概要

2023年8月30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

はじめに

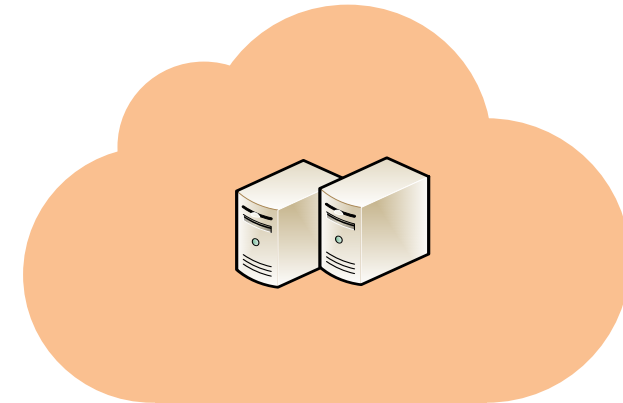
今回の学認クラウドオンデマンド構築サービスセミナー実施に当たり、実習及び実習後のお試し環境利用に必要なmdx計算資源は、東京大学情報基盤センター様よりご提供頂いております。

OCSとは

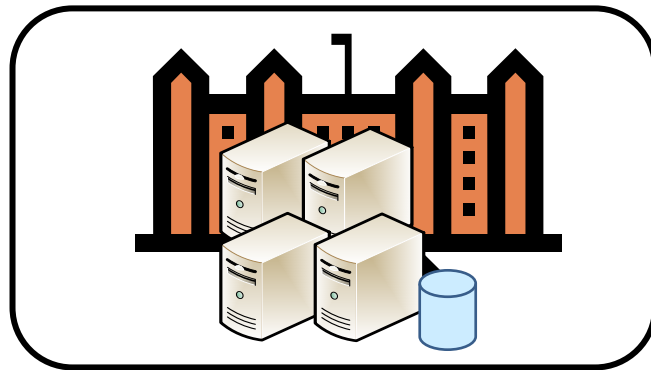
OCS提供の背景(1)



クラウドA



クラウドB



オンプレミス

OCS提供の背景(2)

どの環境を選ぶべきか？

JupyterHubを用いて
Python演習環境を
立ち上げたい



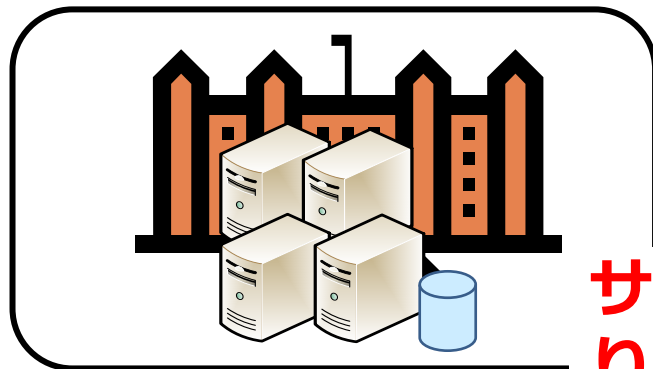
高速、だけど
単価も高い..

クラウドA



単価は安い、が
CPUが旧世代..

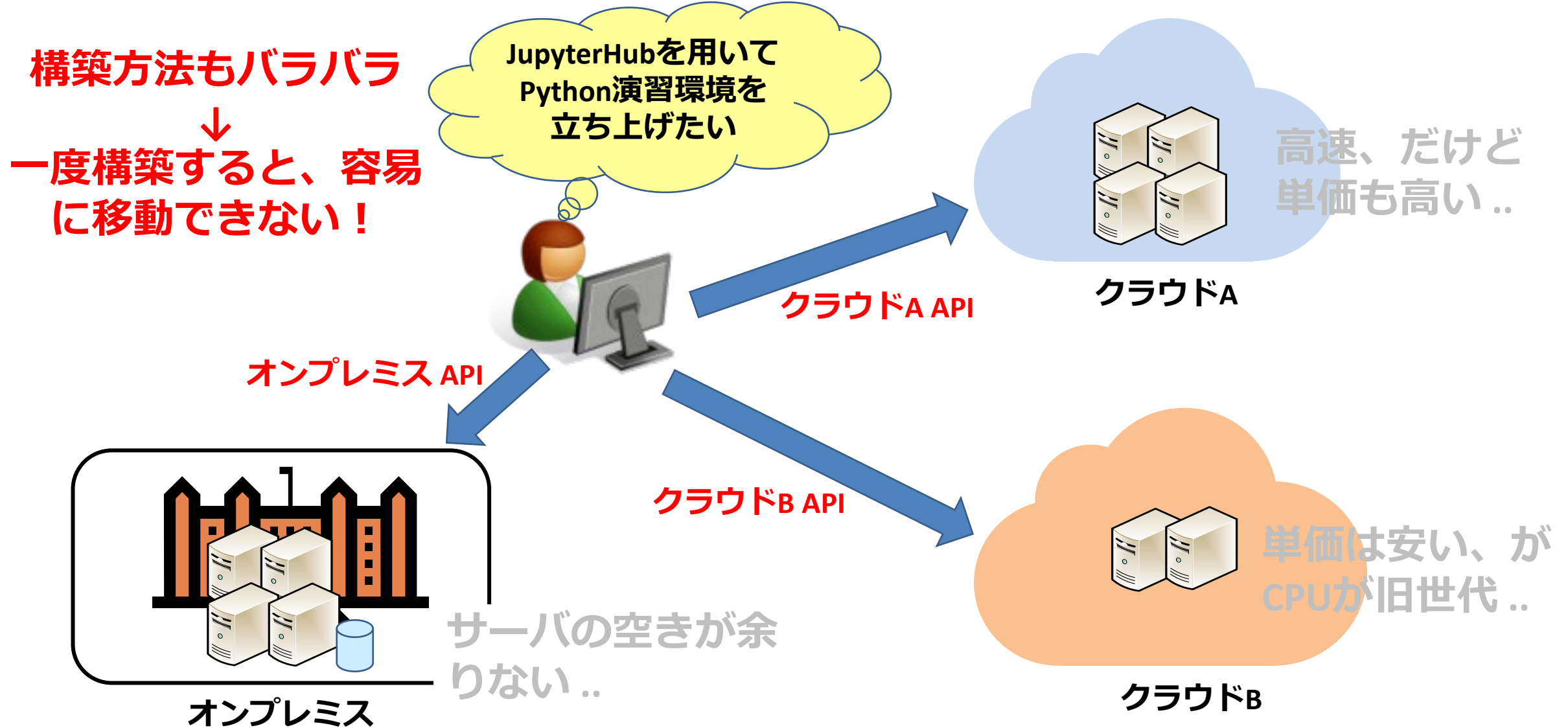
クラウドB



オンプレミス

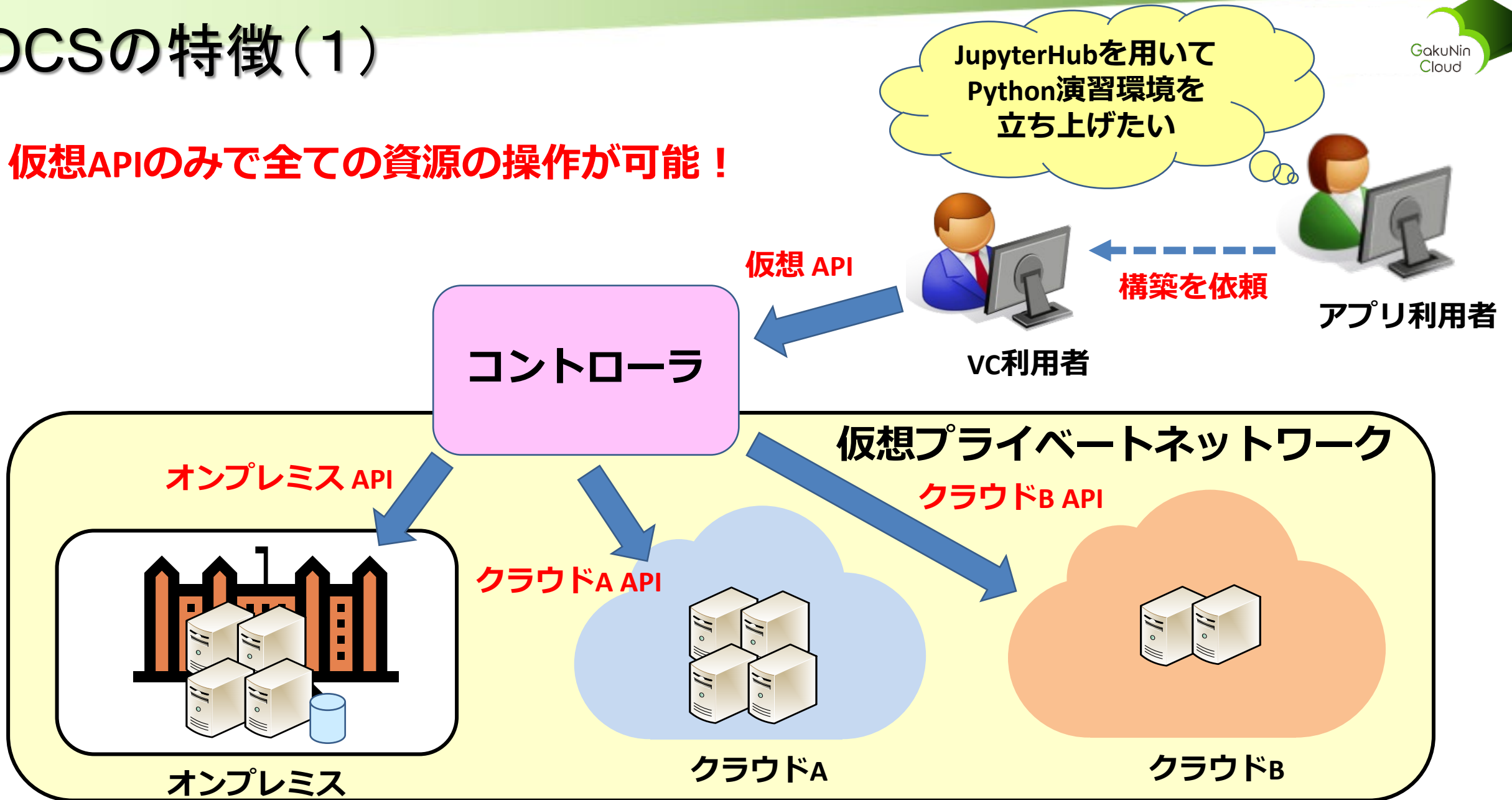
サーバの空きが余
りない..

OCS提供の背景(3)



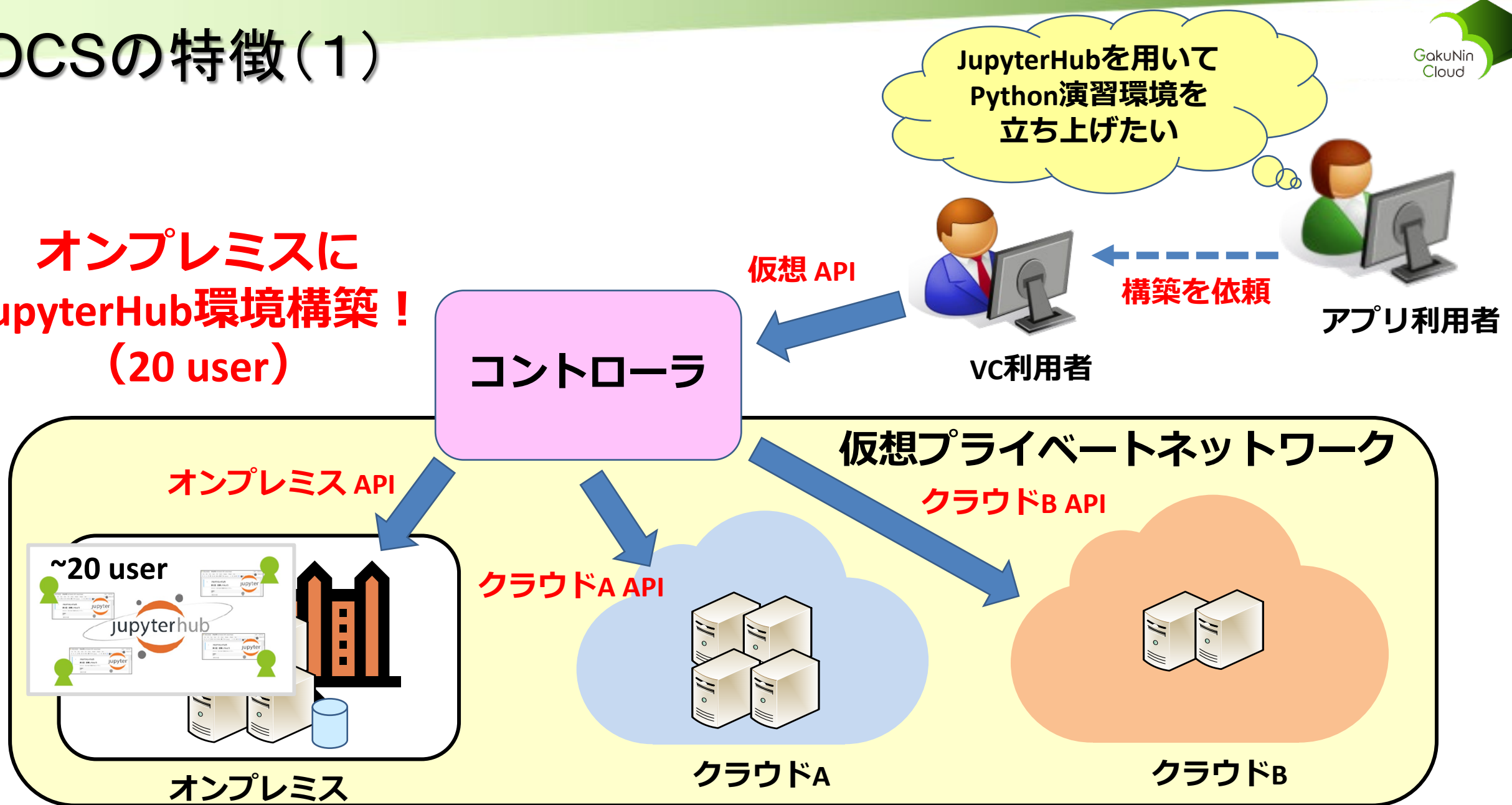
OCSの特徴(1)

仮想APIのみで全ての資源の操作が可能！



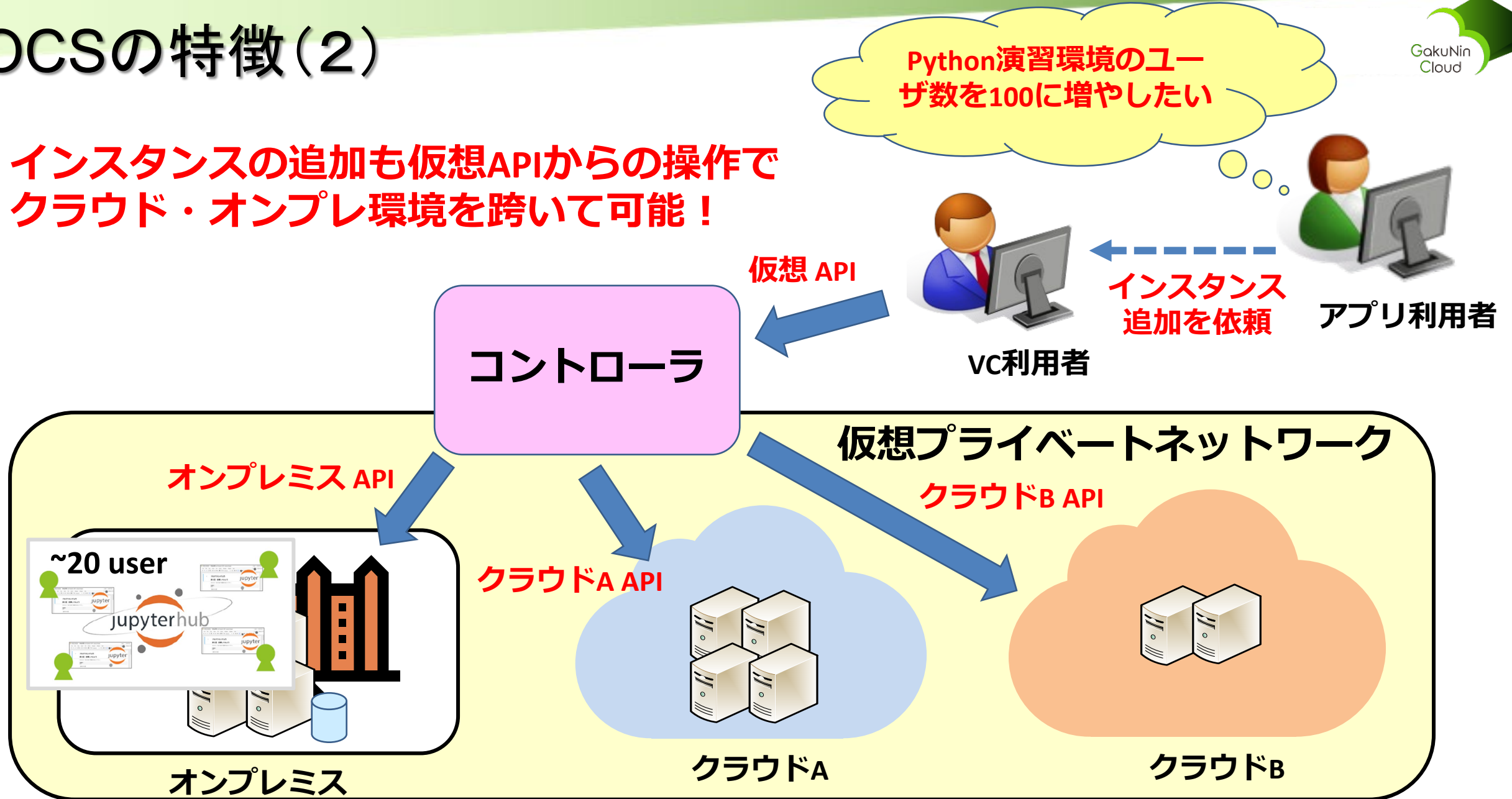
OCSの特徴(1)

**オンプレミスに
JupyterHub環境構築！
(20 user)**



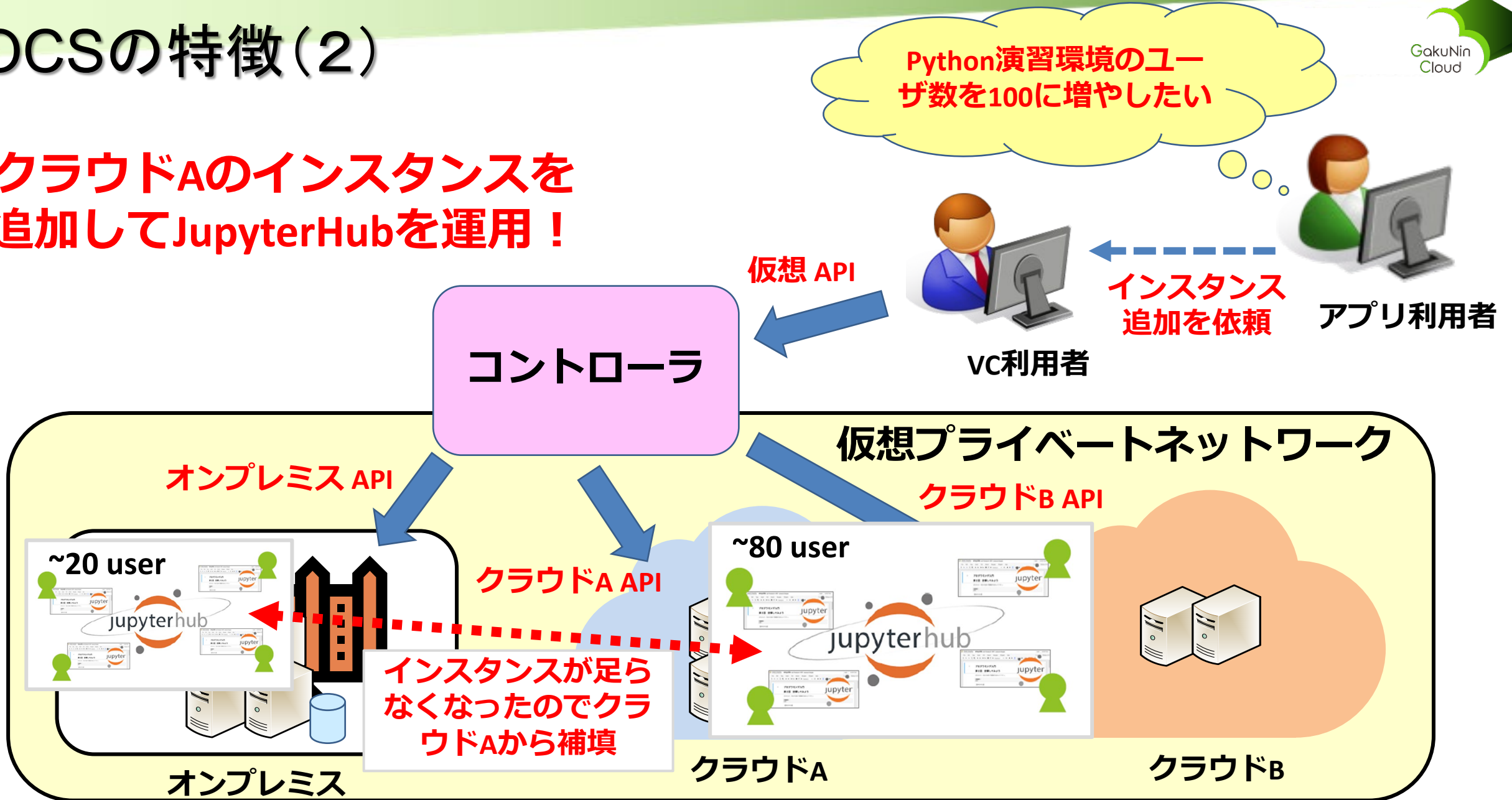
OCSの特徴(2)

インスタンスの追加も仮想APIからの操作で
クラウド・オンプレ環境を跨いで可能！



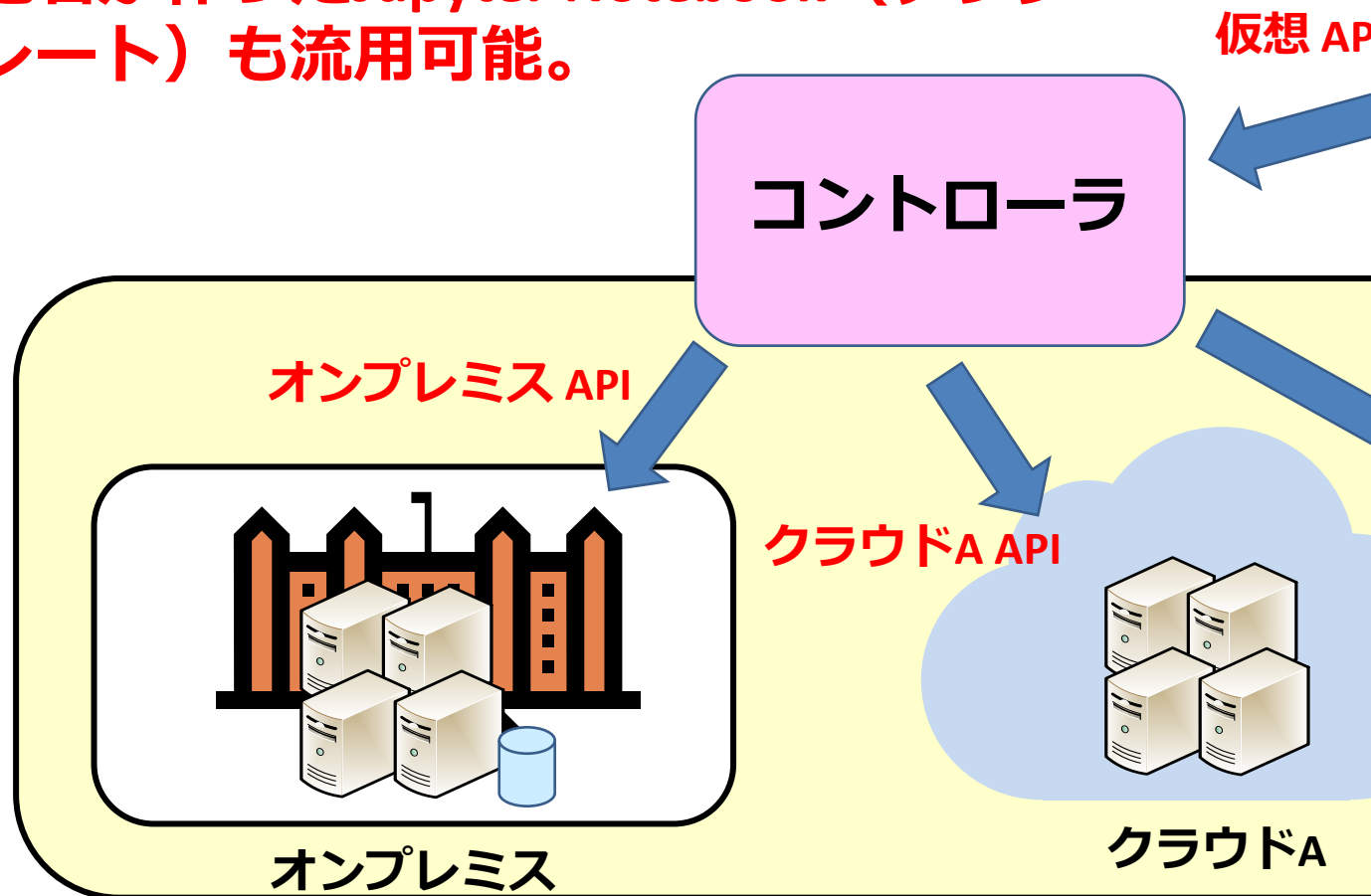
OCSの特徴(2)

クラウドAのインスタンスを追加してJupyterHubを運用！



OCSの特徴(3)

仮想APIはJupyter Notebookを介してアクセスするため、構築作業の再現性が高い！
他者が作ったJupyter Notebook（テンプレート）も流用可能。



1.1 初期化 Jupyter Notebookの記述例

```
[1]: parameters
1 vcc_access_token = "..."
2 testname = "TEST-2022-03-15"

[2]:
1 from common import logsetting
2 from vcpsdk.vcpsdk import VcpSDK
3
4 #
5 # VCP SDK の初期化
6 #
7
8 sdk = VcpSDK(vcc_access_token)
9
10 # VCP SDK バージョン確認
11 sdk.version()
12
13 # UnitGroup作成
14 my_ugroup_name = "03_sample" + testname
15
16 ugroup = sdk.get_ugroup(my_ugroup_name)
17 if ugroup is None:
18     ugroup = sdk.create_ugroup(my_ugroup_name)

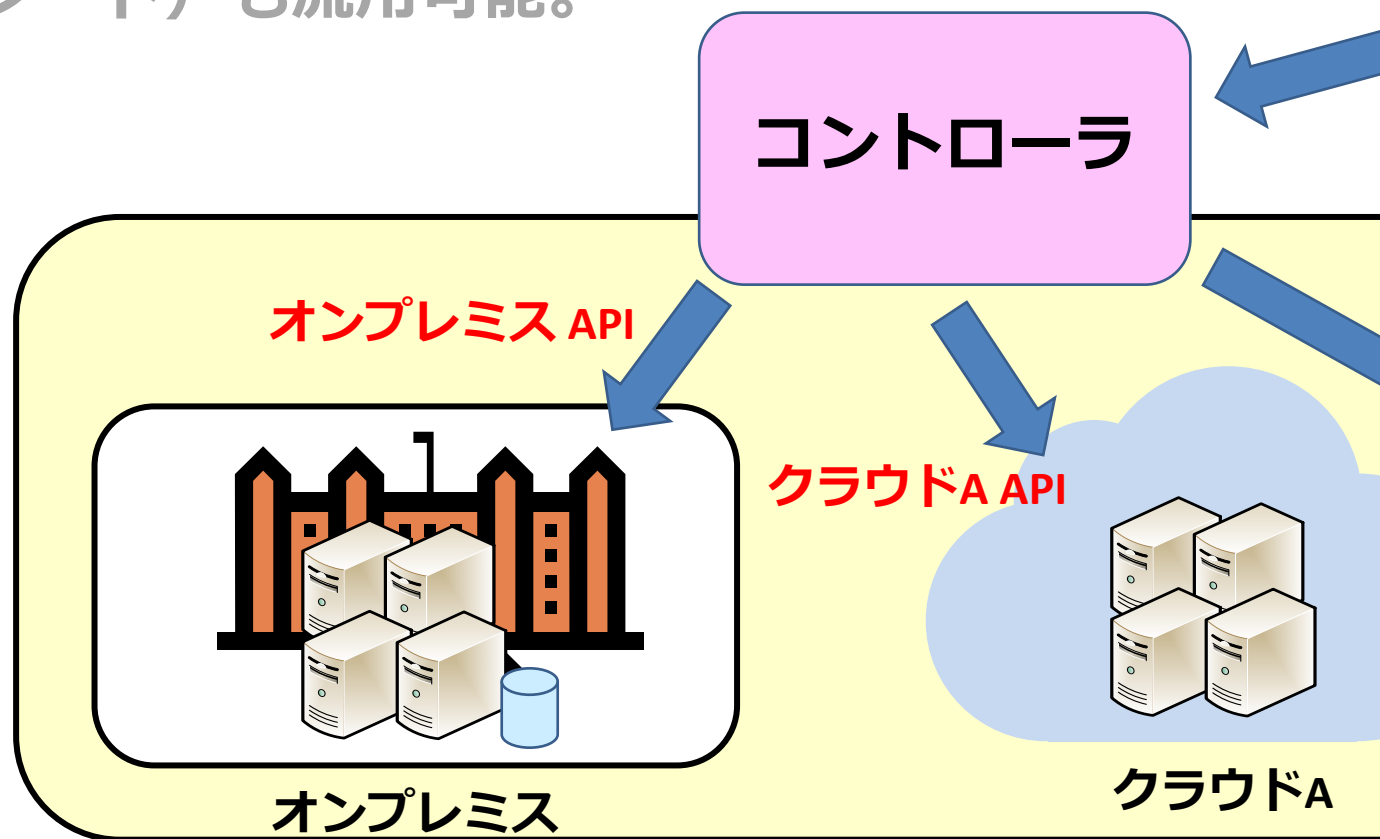
vcplib:
  filename: /home/jovyan/vcpsdk/vcplib/occtr.py
  version: 20.10.0+20201001

vcpsdk:
```

OCSの特徴(3)

仮想APIはJupyter Notebookを介してアクセスするため、構築作業の再現性が高い！
他者が作ったJupyter Notebook（テンプレート）も流用可能。

VC利用者となる敷居は低いです！



1.1 初期化 Jupyter Notebookの記述例

```

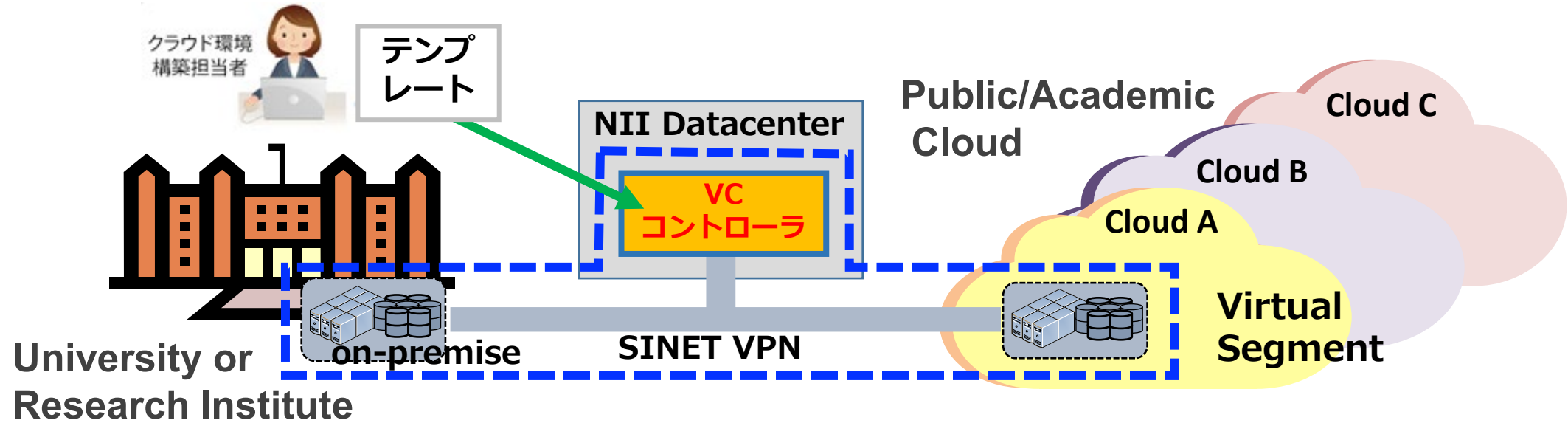
[1]: parameters
1 vcc_access_token = "..."
2 testname = "TEST-2022-03-15"

[2]:
1 from common import logsetting
2 from vcpsdk.vcpsdk import VcpSDK
3
4 #
5 # VCP SDK の初期化
6 #
7
8 sdk = VcpSDK(vcc_access_token)
9
10 # VCP SDK バージョン確認
11 sdk.version()
12
13 # UnitGroup作成
14 my_ugroup_name = "03_sample" + testname
15
16 ugroup = sdk.get_ugroup(my_ugroup_name)
17 if ugroup is None:
18     ugroup = sdk.create_ugroup(my_ugroup_name)

vcplib:
  filename: /home/jovyan/vcpsdk/vcplib/occtr.py
  version: 20.10.0+20201001

vcpsdk:
  
```

OCSの特徴(まとめ)



- 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するサービス
 - 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。

OCSの特徴(テンプレート)

他者が作ったテンプレートの流用も可能



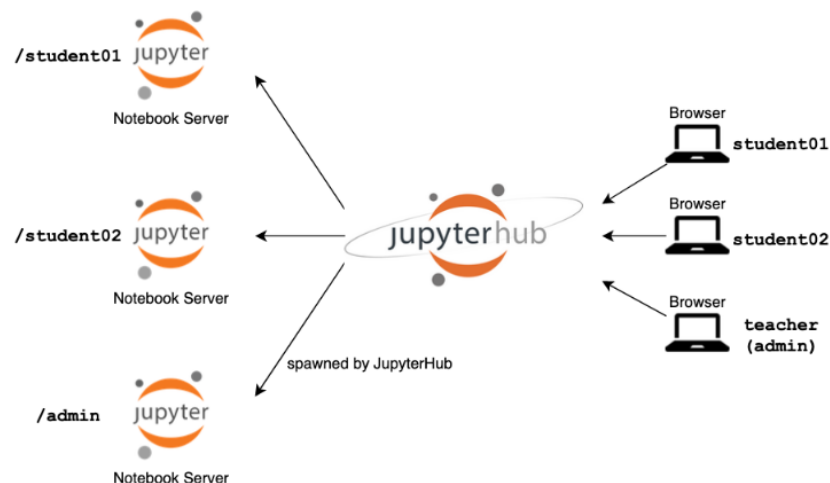
The Littlest JupyterHub による軽量Python実習環境の構築

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Jupyter Notebook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
ugroup_name = "hands003"

# プロバイダ
vc_provider = "aws"
```

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。

VCノードのspecを指定

TLJH を利用するのに十分な性能

固定割当IPアドレスは、ハンズオンでは必要ありません。

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

# VCノード spec
spec = vcp.get_spec(vo_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 (固定割当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分程度がかかります。

```
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」を実行し、`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) 環境の構築

VCノード上に、本ハンズオン用に用意したThe Littlest JupyterHubのコンテナイメージを使用して環境を構築します。

TLJHコンテナイメージの取得

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。

図表を組み合わせた説明を挿入できる

利用例 (Pythonプログラミング教育)

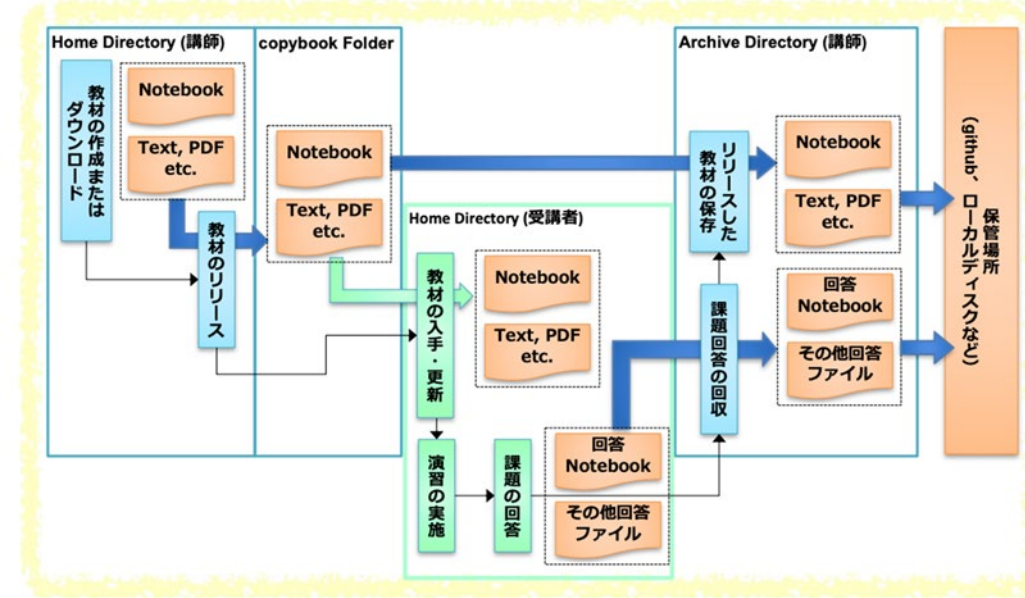
■ CoursewareHubとは

■ NIIで開発を進めている講義・演習システム

- JupyterHub + 講義のためのモジュール群
- OCS用テンプレートは以下から公開
 - <https://github.com/nii-gakunin-cloud/ocs-templates/tree/master/CoursewareHub>

■ OCS+CoursewareHubの特徴

- 授業内容に応じて計算資源の増減が容易
 - 20人の授業: 2 VMで環境を構築
 - 100人の授業: 10 VMで環境を構築
- 特定クラウドにロックインされない
- オンプレとクラウドを跨った環境を作れる
 - 例、オンプレ資源が枯渇したときのみクラウドを利用



CoursewareHubでのデータ集約・配信
(2021年度 NII情報処理技術セミナー
(クラウド編) より引用)

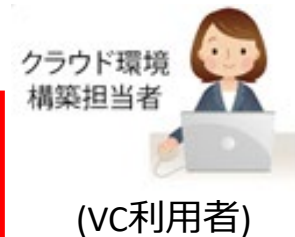
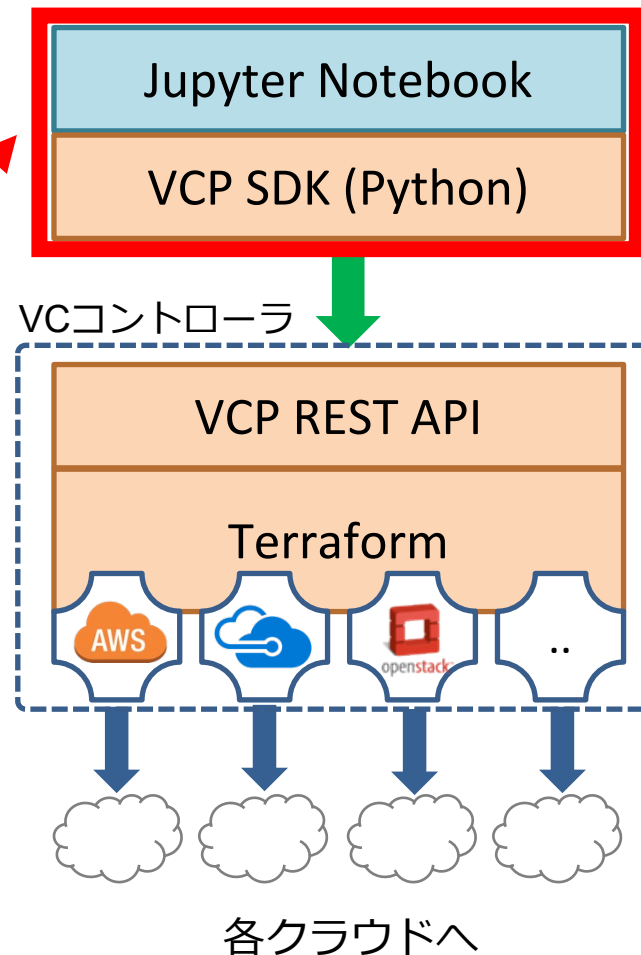
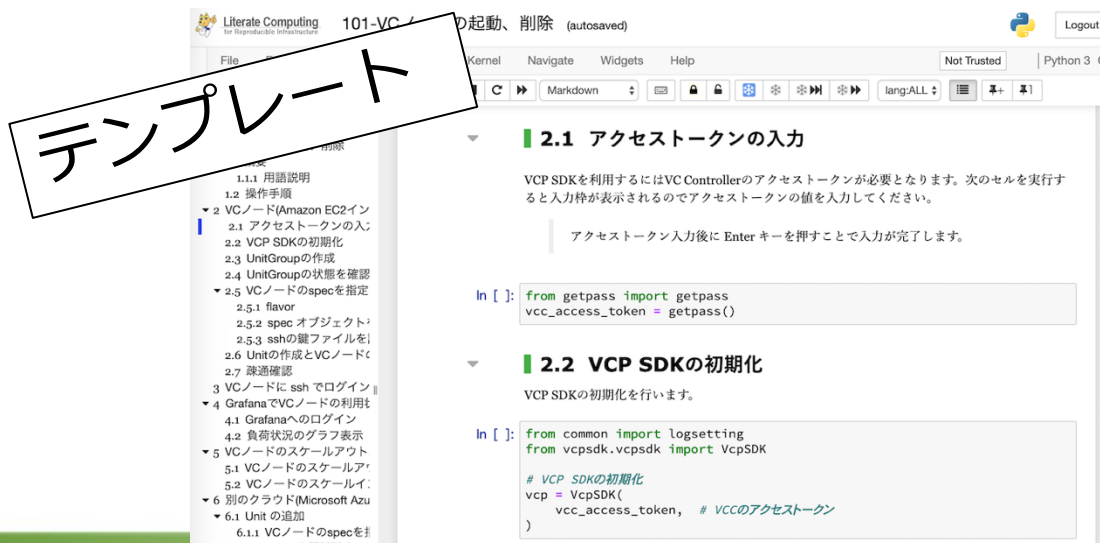
OCSを支えるVCPの仕組み

管理ソフトウェアの概要(1)

■ Virtual Cloud Provider (VCP)

- 本機能の中心ソフトウェア
- プロバイダI/Fを抽象化したREST API
- VCPの利用を容易にするPythonライブラリ VCP SDK

■ Jupyter Notebook(+NII拡張)からVCP SDKを利用して操作



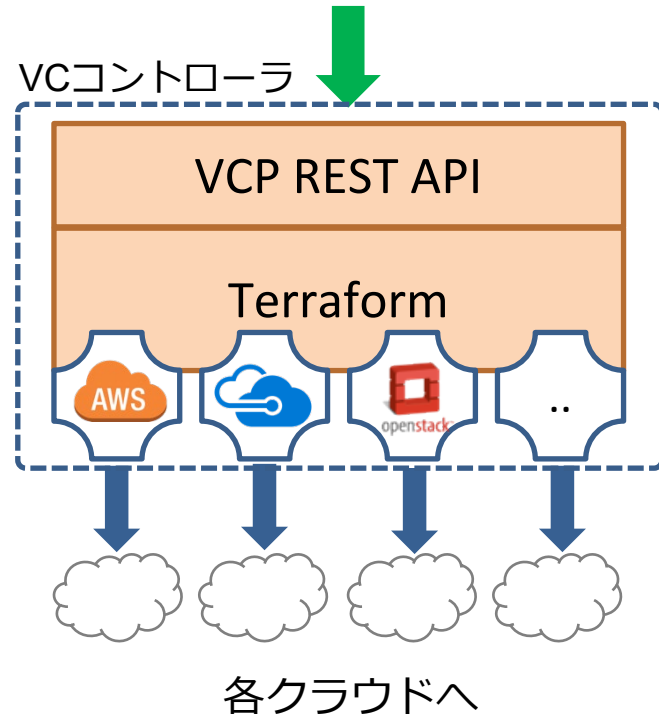
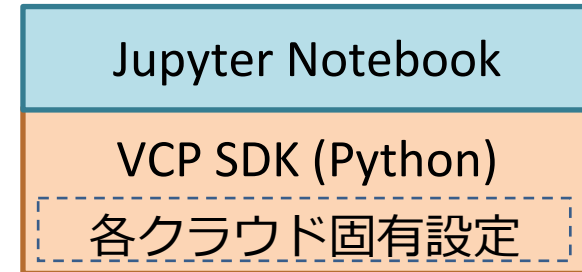
管理ソフトウェアの概要(2)

■ VCP SDK

- 各クラウドの固有設定をSDK内に隠蔽することで、Jupyter Notebookを変更することなくクラウド間での使い回しを実現



クラウド環境
構築担当者



VCP SDKの中で各クラウドのmediumを定義

```
1.2 新規 server を作成

[5]:
1 #
2 # 作成するserverのspec情報を作成
3 #
4 spec = sdk.get_spec("aws", "medium")
5
6 #
7 # 変更できること
8 #
9 # spec.num_nodes = 1
10 spec.num_nodes = 2
11 #spec.instance_type = 't2.medium'
12 # spec.params_v = ['/opt:/opt']
13 # spec.volume_size = 40
14 # spec.volume_type = "standard" # standard/io1/gp2/sc1/st1
15 # spec.ip_addresses = ['起動するnodeの静的なIPアドレス']
16 # spec.image = 'vcp/base:1.5' # base container
17 # 追加で使用するVolume
18 # spec.disks = ['vol-08cbb04b35c8c9545']
19
20 # cloud上のタグ設定
21 spec.set_tag('key1', 'value1')
22 spec.set_tag('key2', 'value2')
23
```

計算インスタンス (VCノード)

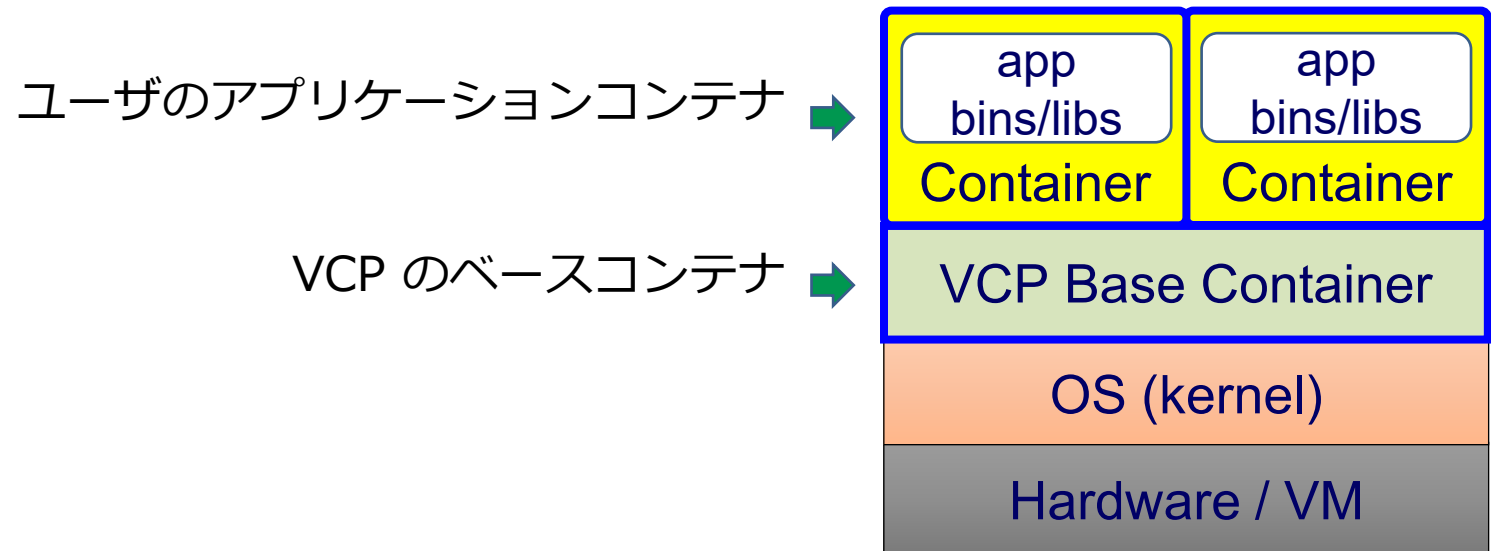
■ Docker in Docker 構成

■ 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
- Dockerのエコシステムが利用可能



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
- アプリケーションの収容設計を支援

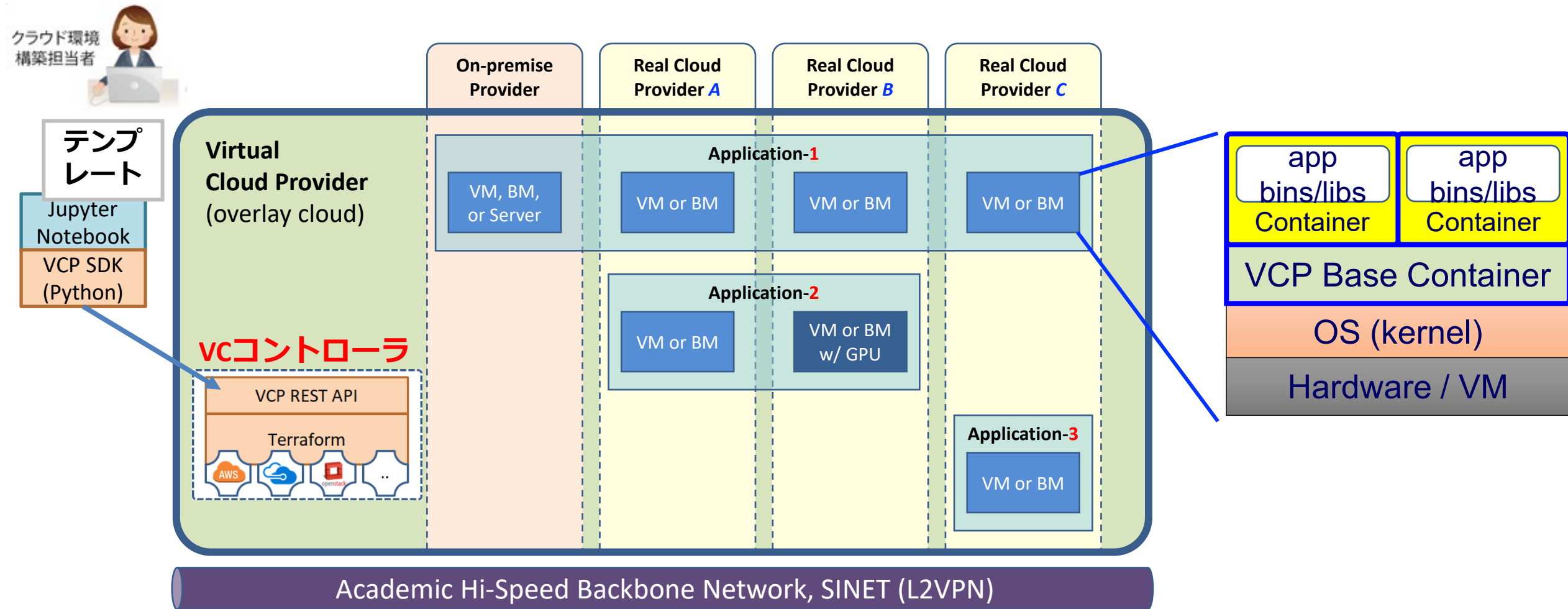


VCノード（ベースコンテナ）毎の情報

アプリコンテナ毎の情報

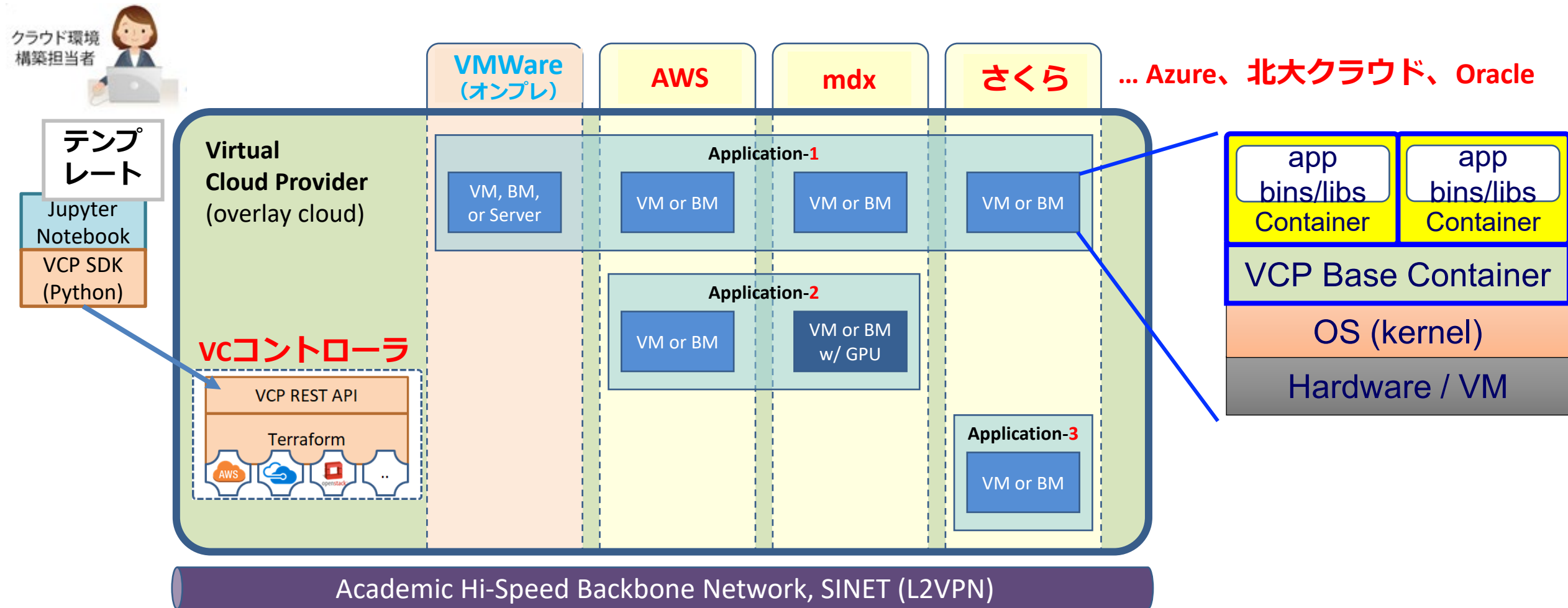
OCSを利用したアプリケーション配備例

- オンプレ・複数の実クラウドを跨ってのアプリケーション配備が可能！



OCSを利用したアプリケーション配備例

- オンプレ・複数の実クラウドを跨ってのアプリケーション配備が可能！



OCSの提供形態

サービス版とポータブル版

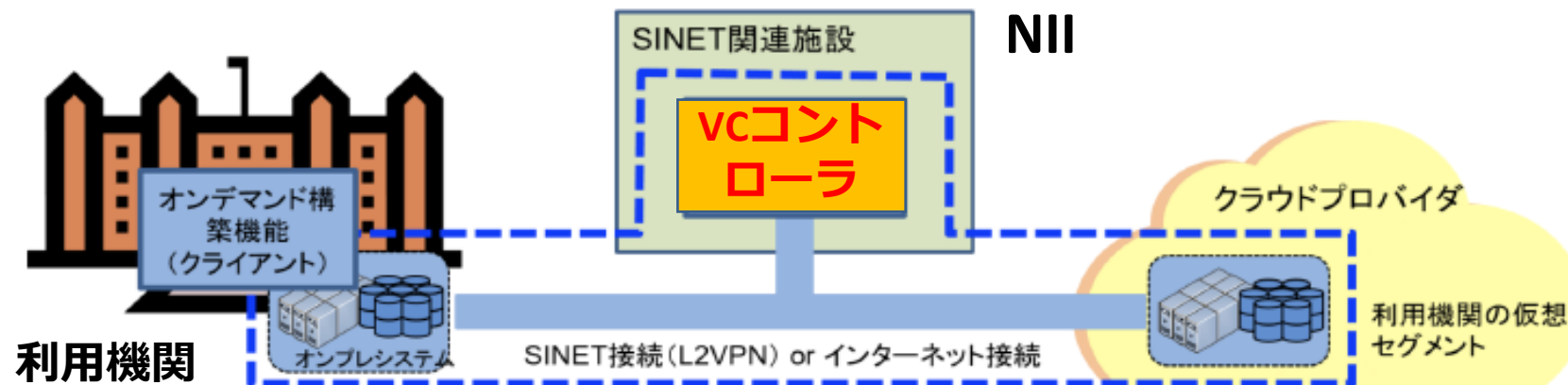
長所：

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



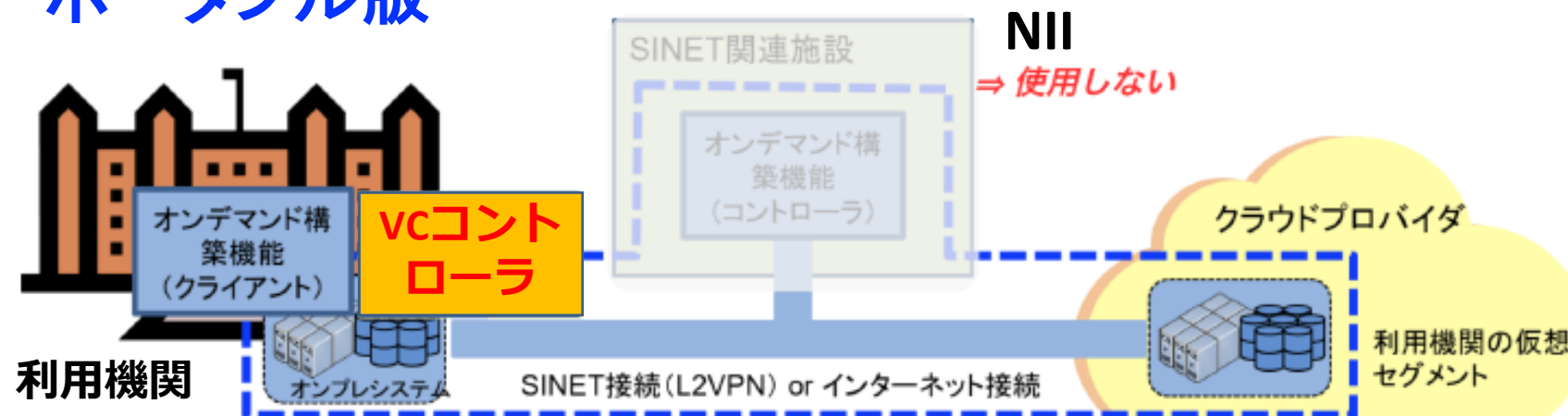
長所：

VCP構築申請が不要となり、すぐに利用可

短所：

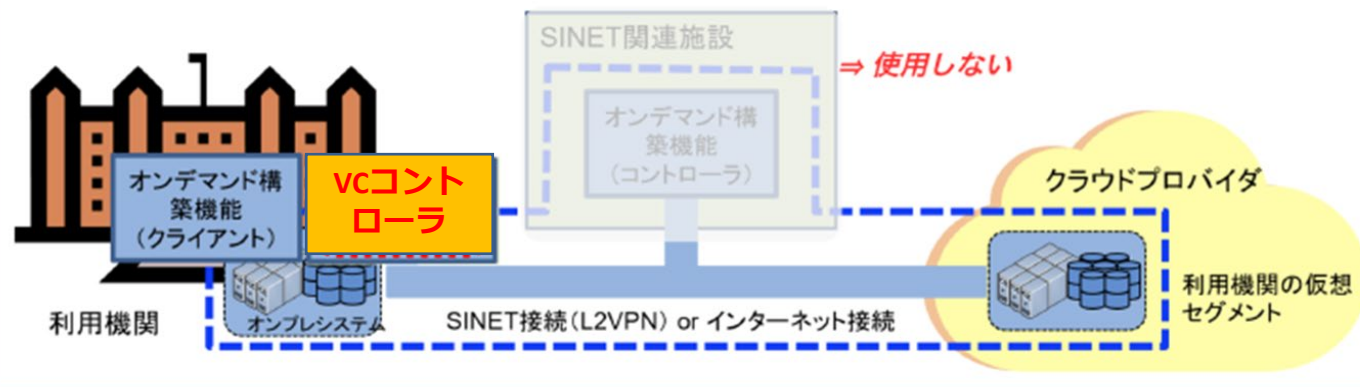
利用機関側でVCP構築・運用・保守

ポータブル版

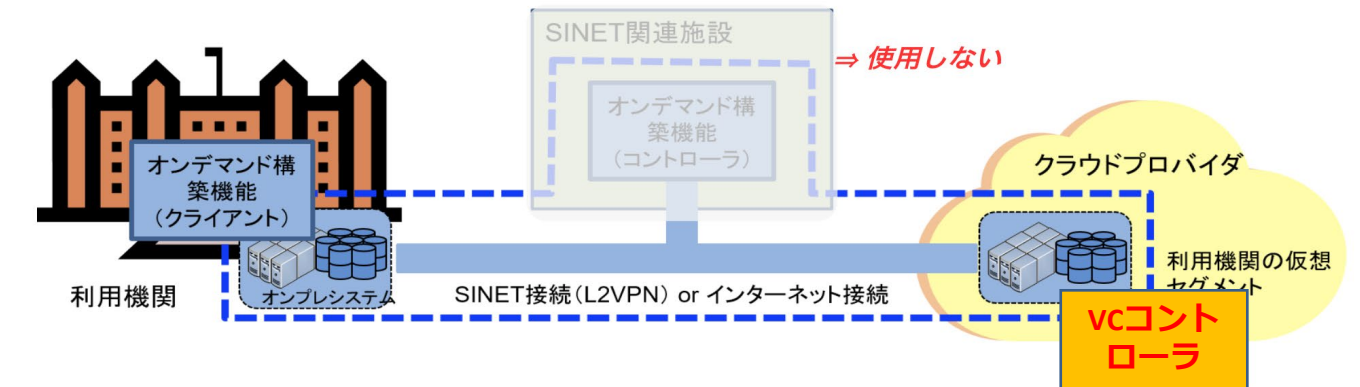


ポータブル版の構成方法

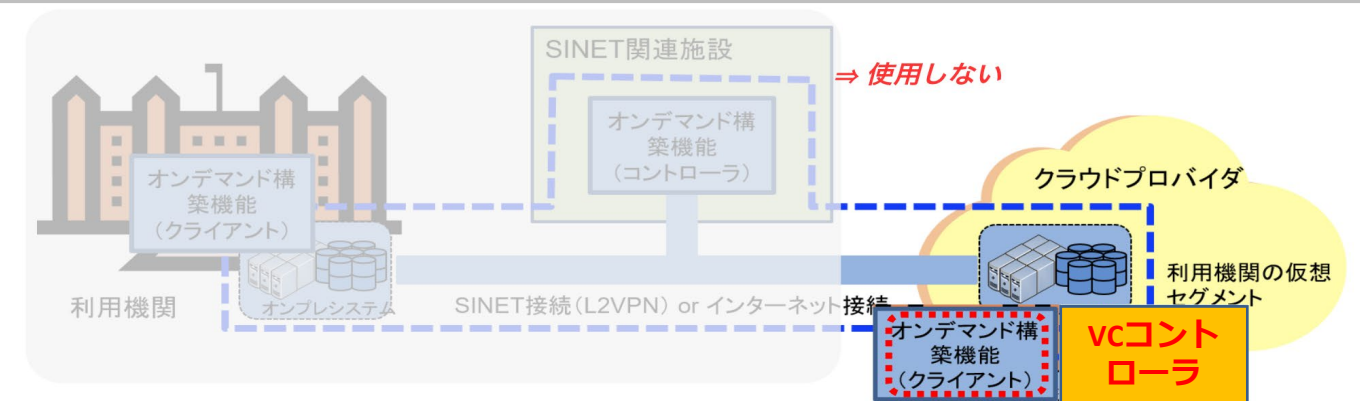
vcコントローラ：利用機関
JupyterNotebook：利用機関
(クライアント)



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)



サービス構成

■ 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

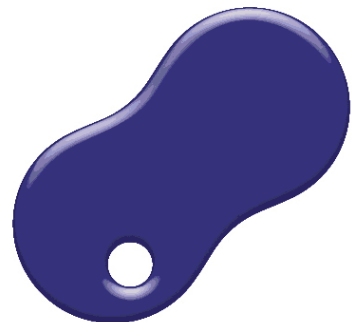
■ 本サービスは無償です

■ 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

各種お問い合わせは、
NIIクラウド支援室 [cld-office-
support@nii.ac.jp](mailto:cld-office-support@nii.ac.jp)
までお願いいたします！



大学共同利用機関法人 情報・システム研究機構

国立情報学研究所

National Institute of Informatics