学認クラウドオンデマンド構築サービス（OCS）の概要

2022年12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

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計算資源は、さくらのクラウド様よりご提供頂いております。

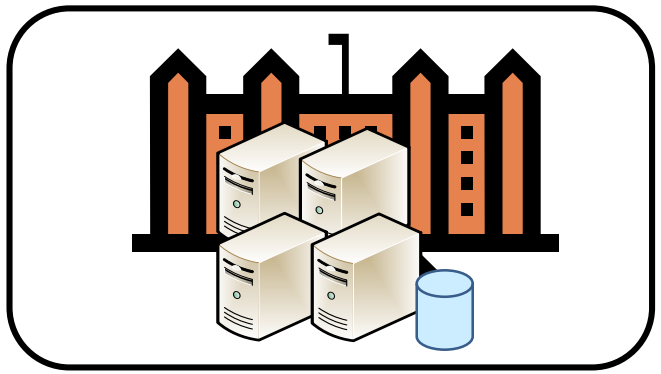
Copyright 2022 National Institute of Informatics

OCSとは

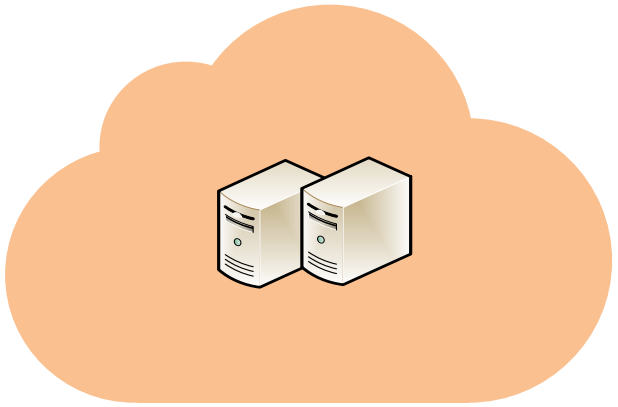
Copyright 2022 National Institute of Informatics

OCS提供の背景（1）

クラウドA



オンプレミス

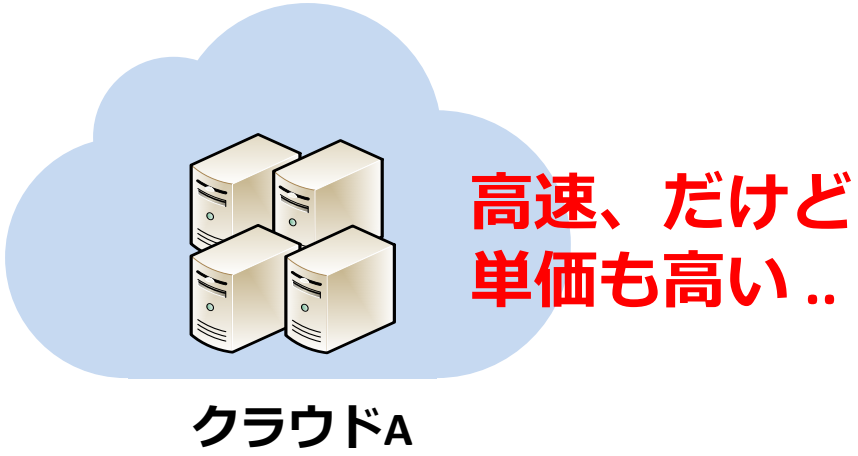


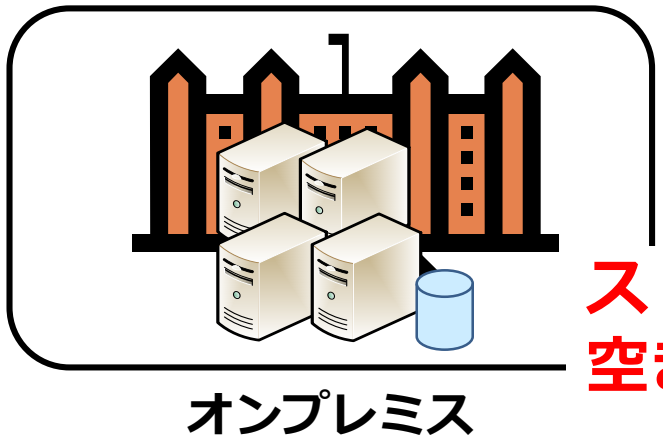
クラウドB

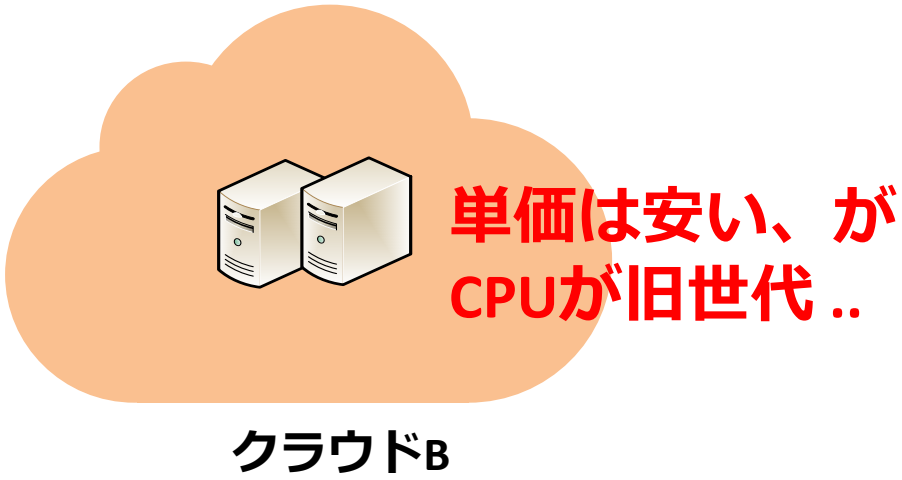
Copyright 2022 National Institute of Informatics

OCS提供の背景(2)

どの環境を選ぶべきか？







Copyright 2022 National Institute of Informatics

OCS提供の背景(3)

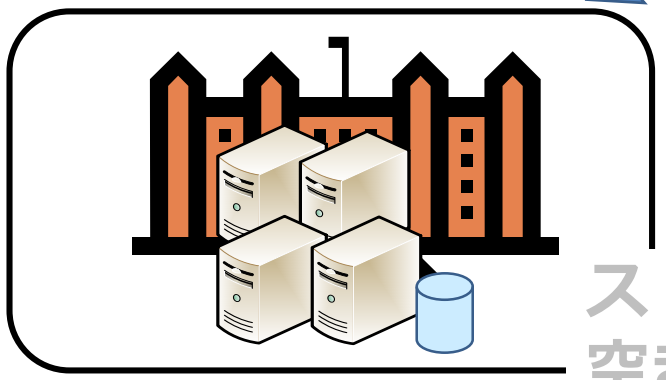
構築方法もバラバラ
↓
一度構築すると、容易に移動できない！

クラウドA API

高速、だけど単価も高い..

クラウドA

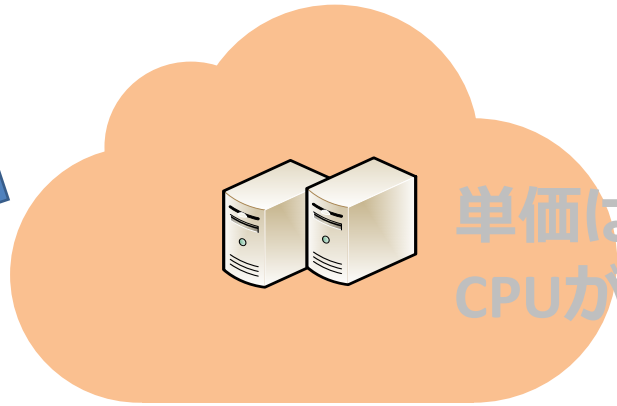
オンプレミス API



ストレージ容量の空きが余りない..

オンプレミス

クラウドB API



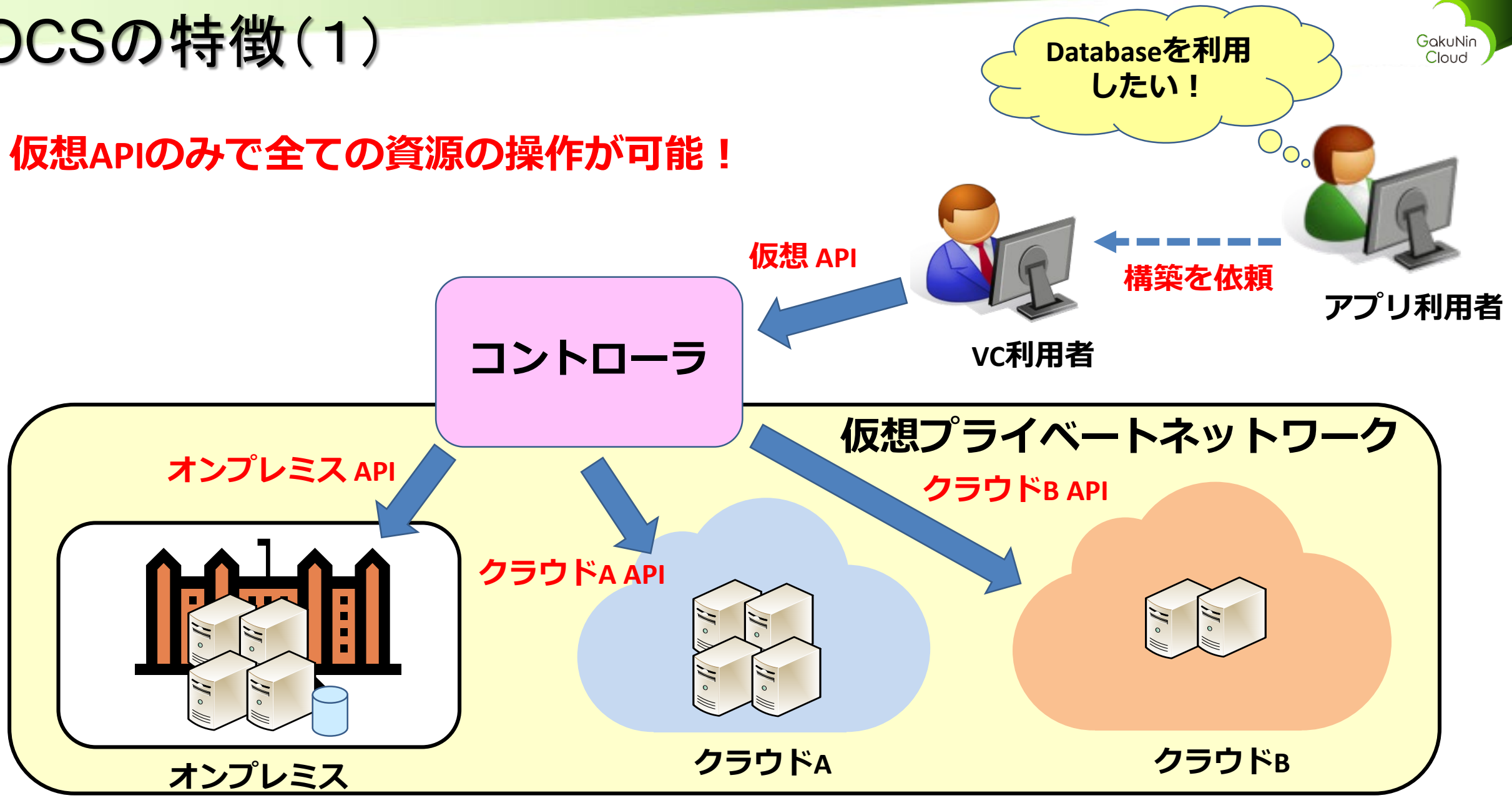
単価は安い、がCPUが旧世代..

クラウドB

Copyright 2022 National Institute of Informatics

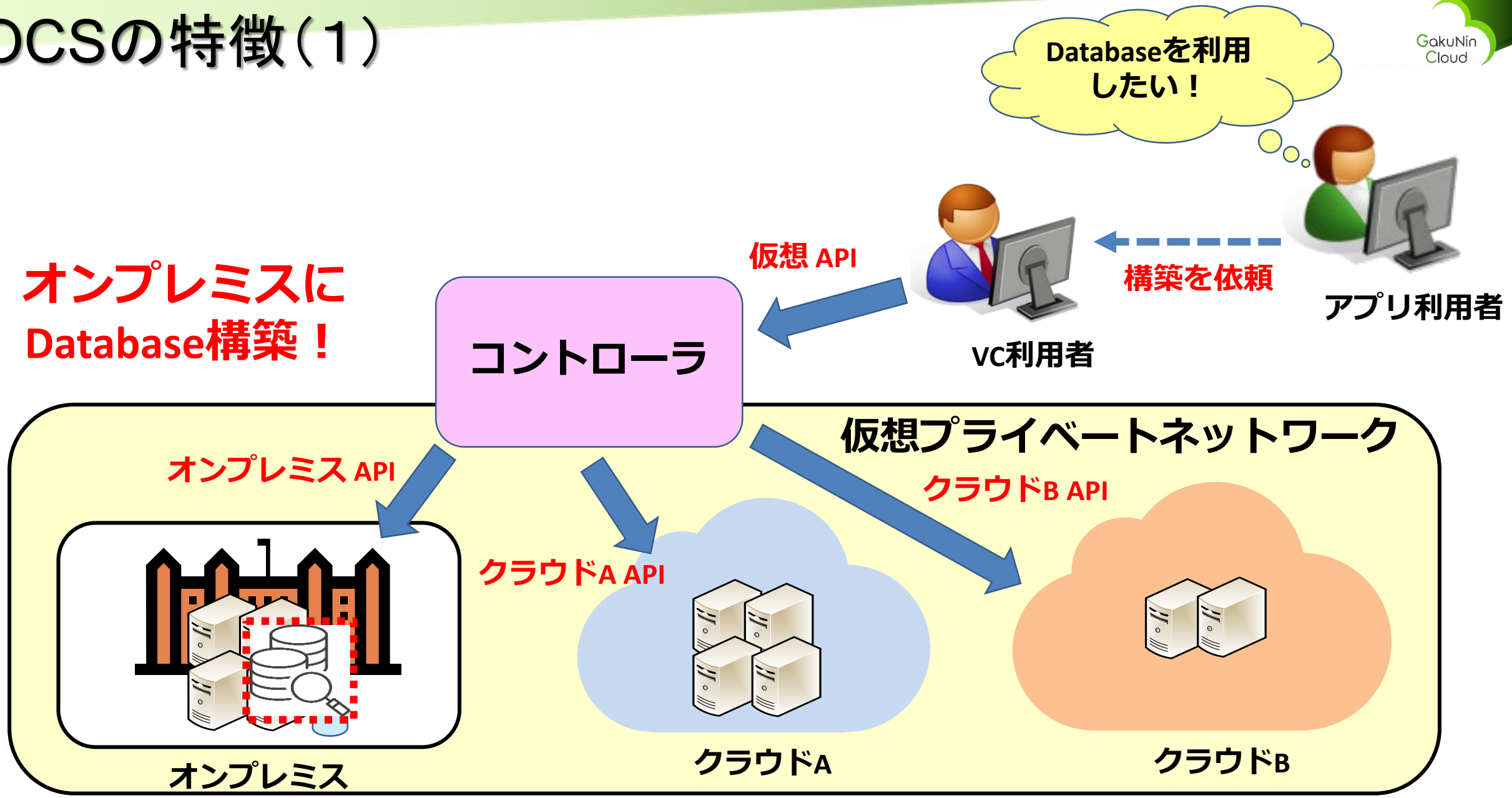
OCSの特徴（1）

仮想APIのみで全ての資源の操作が可能！



Copyright 2022 National Institute of Informatics

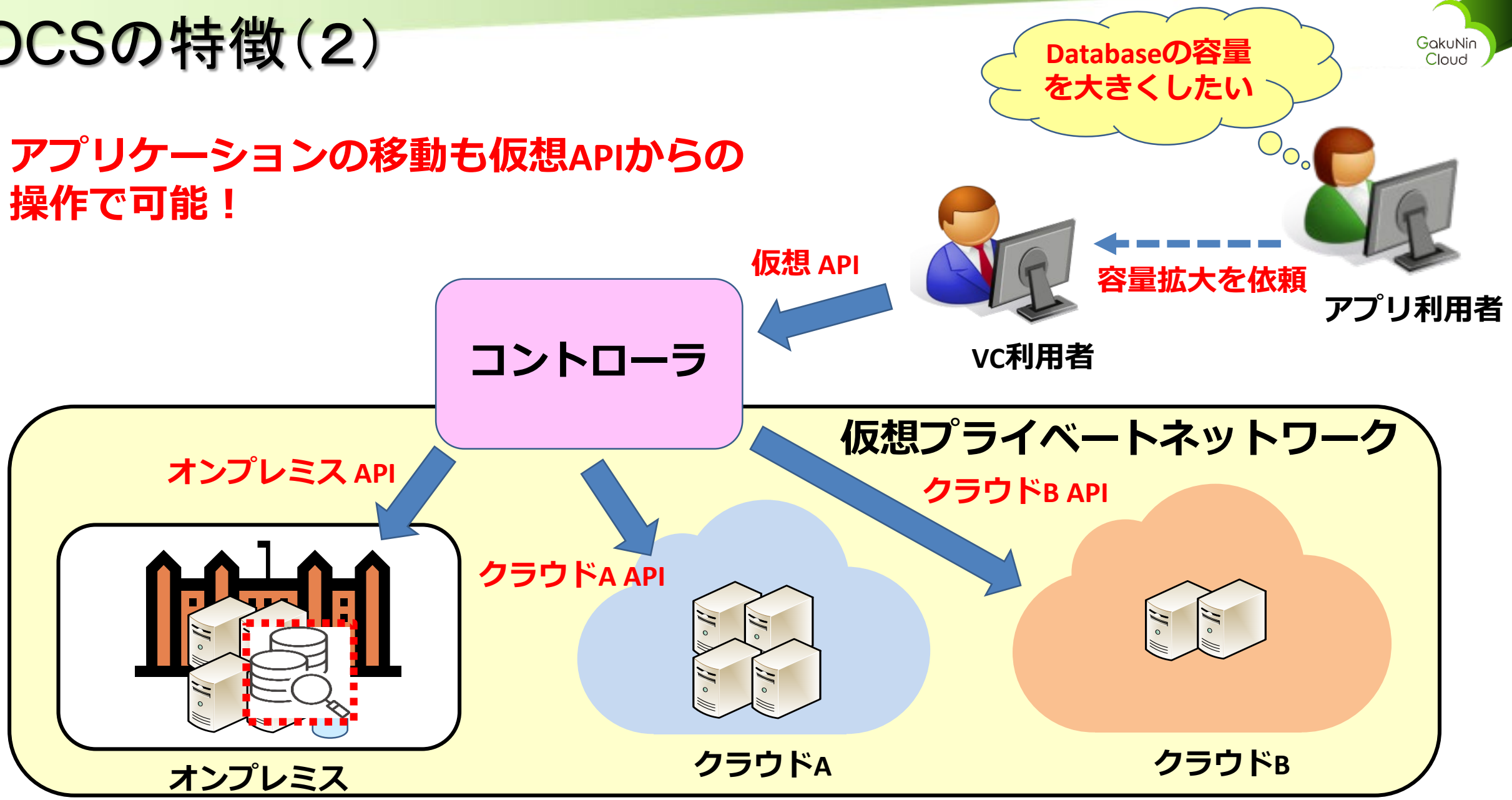
OCSの特徴（1）



Copyright 2022 National Institute of Informatics

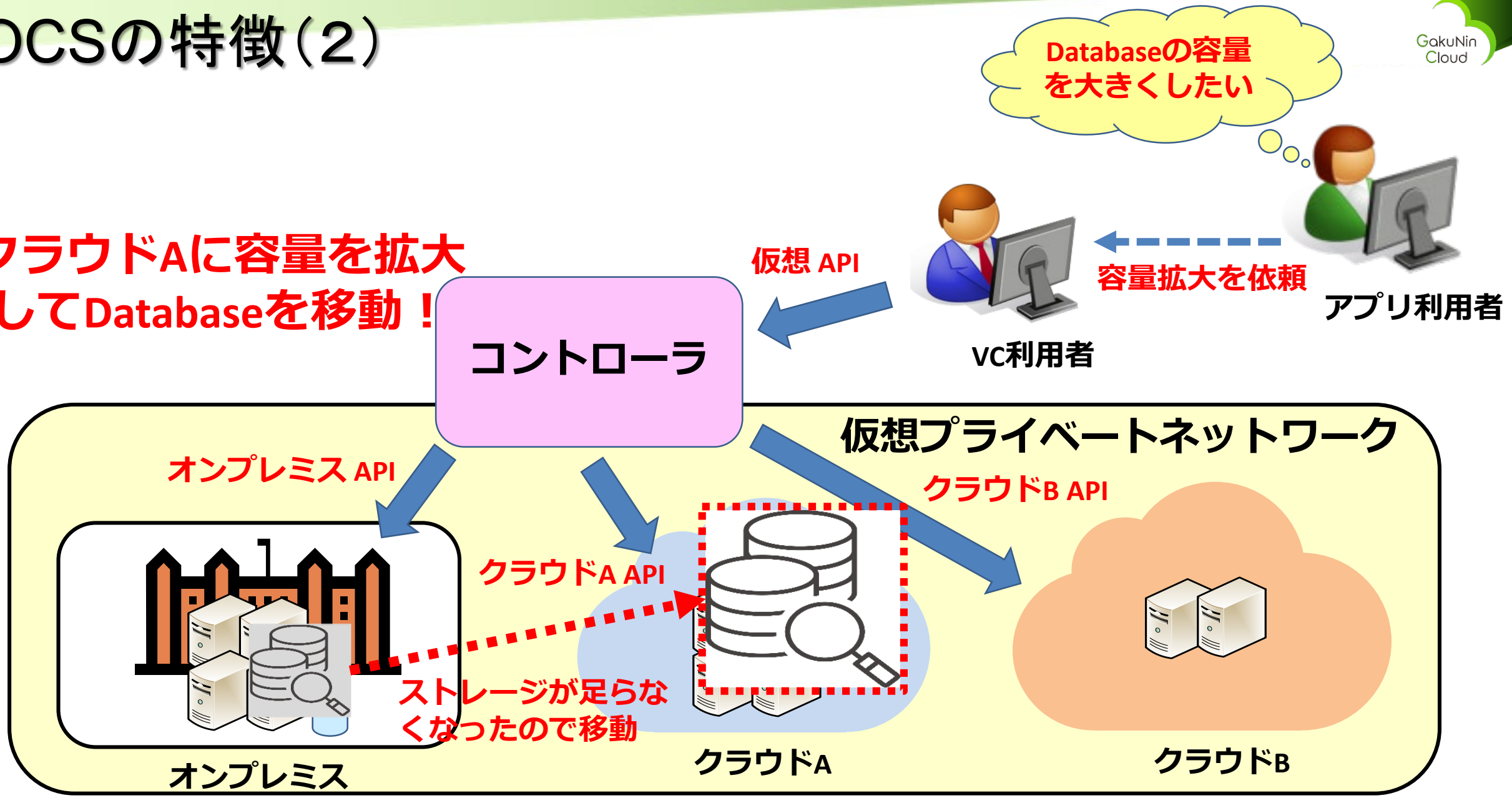
OCSの特徴(2)

アプリケーションの移動も仮想APIからの操作で可能！



Copyright 2022 National Institute of Informatics

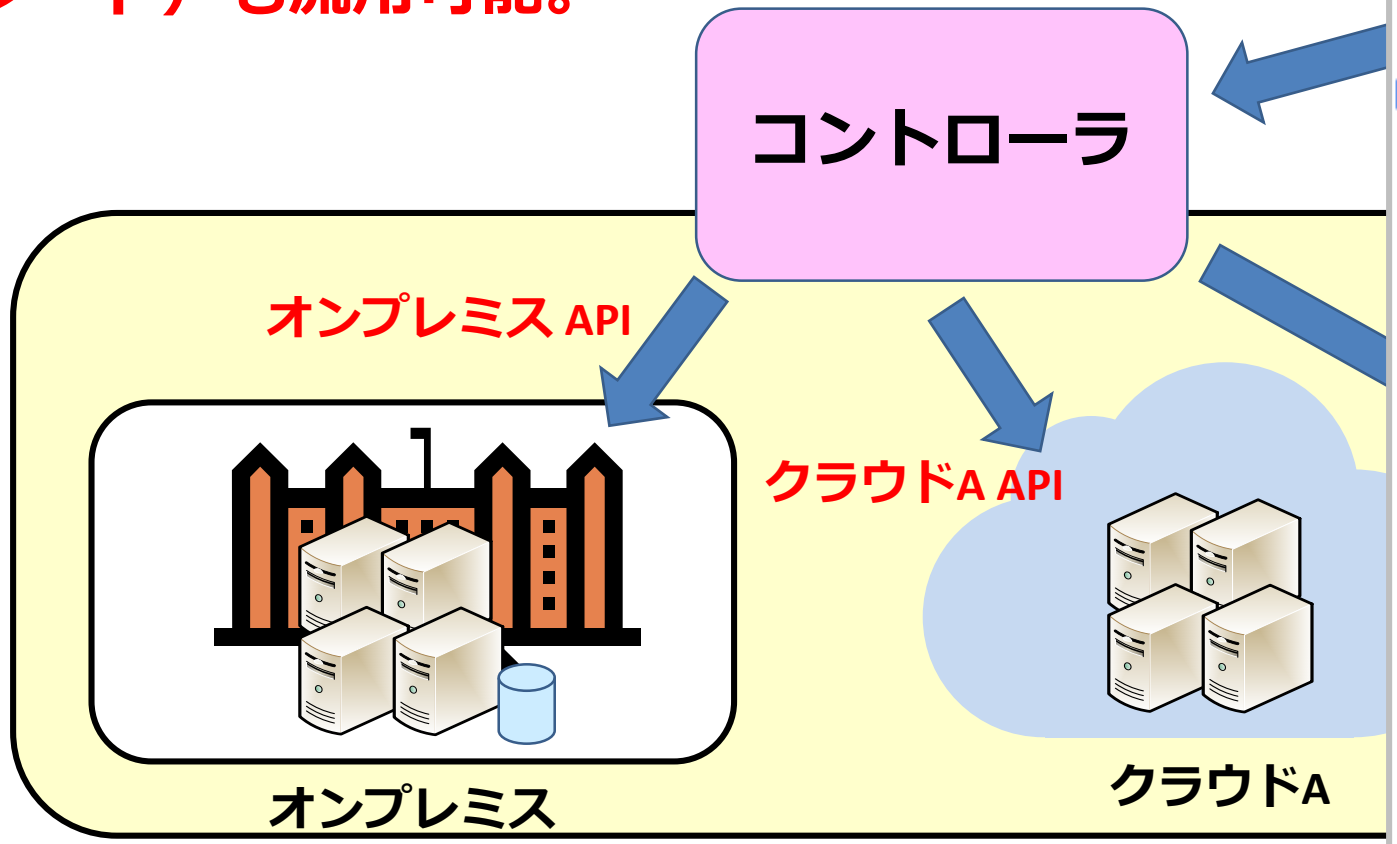
OCSの特徴(2)



Copyright 2022 National Institute of Informatics

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
vcc_access_token = "
testname = "TEST-2022-03-15"
```

[2]:
```
from common import logsetting
from vcpsdk.vcpsdk import VcpSDK

#
# VCP SDK の初期化
#

sdk = VcpSDK(vcc_access_token)

# VCP SDK バージョン確認
sdk.version()

# UnitGroup作成
my_ugroup_name = "03_sample" + testname

ugroup = sdk.get_ugroup(my_ugroup_name)
if ugroup is None:
    ugroup = sdk.create_ugroup(my_ugroup_name)
```

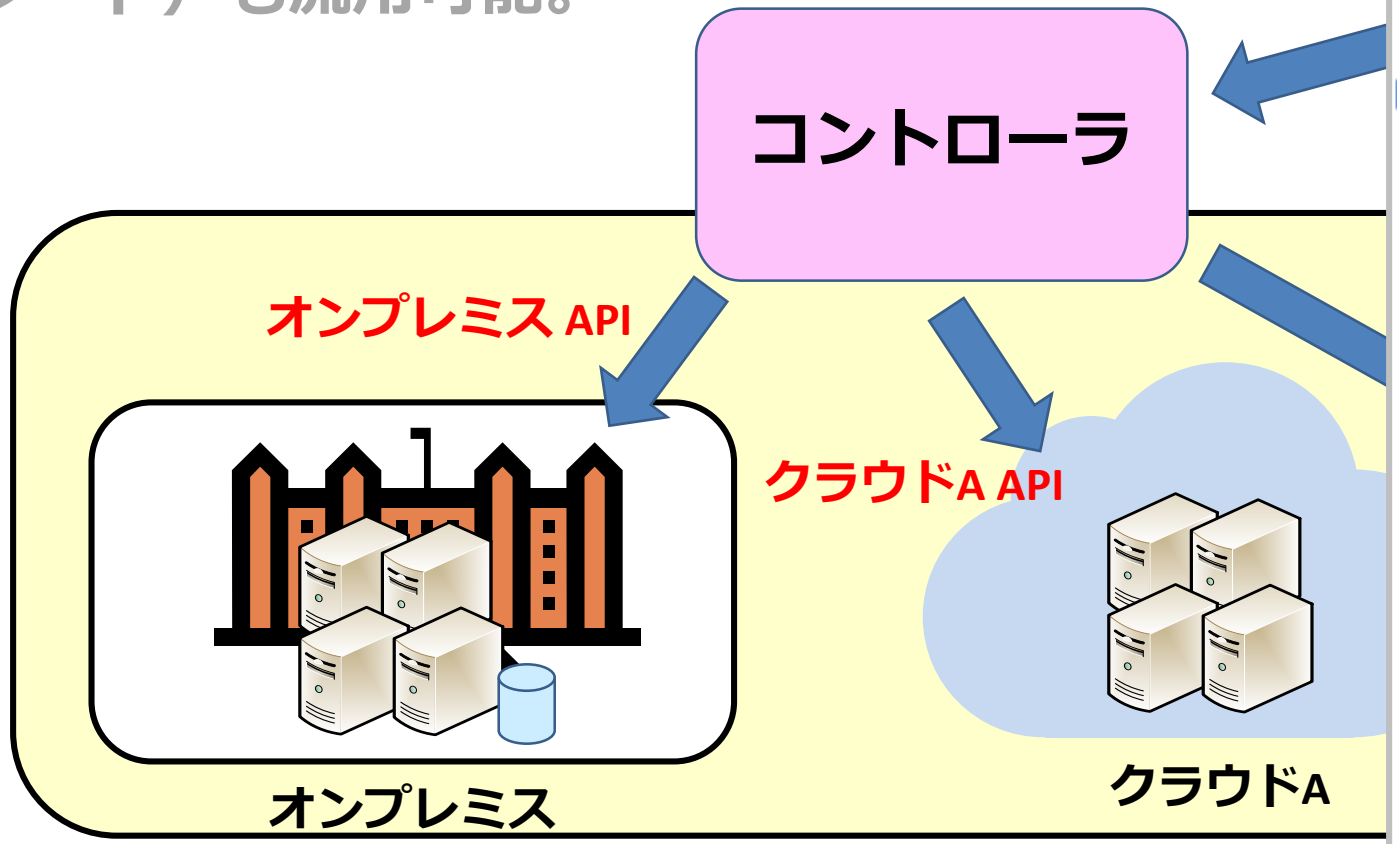
```
vcplib:
  filename: /home/jovyan/vcpsdk/vcplib/occtr.py
  version: 20.10.0+20201001

vcpsdk:
```


Copyright 2022 National Institute of Informatics

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

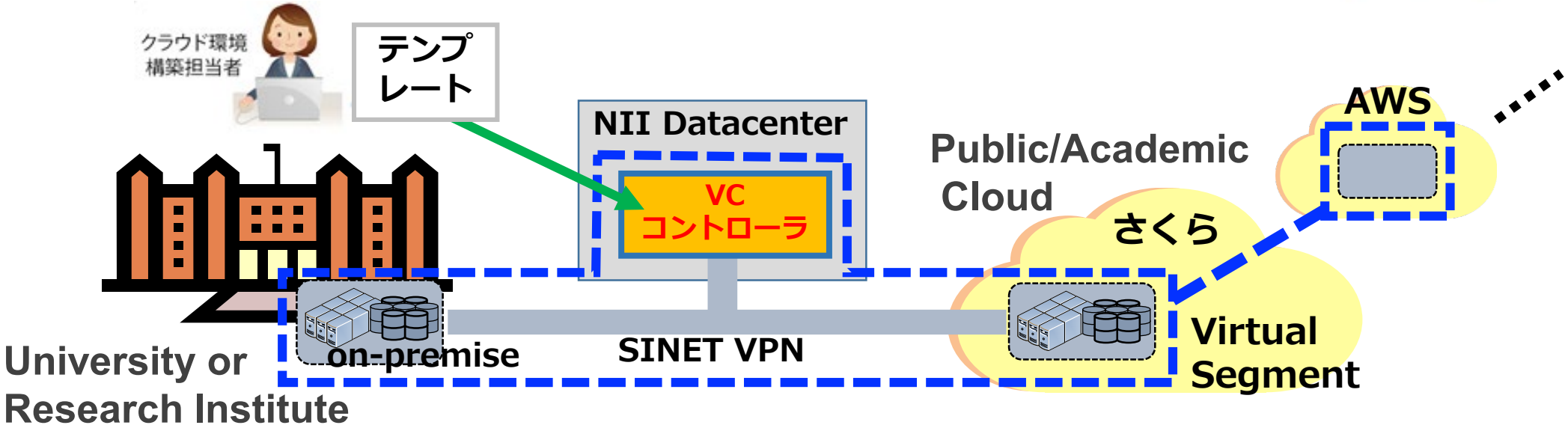
[1]:
```
parameters
vcc_access_token = "
testname = "TEST-2022-03-15"
```

[2]:
```
from common import logsetting
from vcpsdk.vcpsdk import VcpSDK

#
# VCP SDK の初期化
#

sdk = VcpSDK(vcc_access_token)

# VCP SDK バージョン確認
sdk.version()

# UnitGroup作成
my_ugroup_name = "03_sample" + testname

ugroup = sdk.get_ugroup(my_ugroup_name)
if ugroup is None:
    ugroup = sdk.create_ugroup(my_ugroup_name)
```

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

vcpsdk:
```


Copyright 2022 National Institute of Informatics

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。

Copyright 2022 National Institute of Informatics

OCSの特徴（テンプレート）

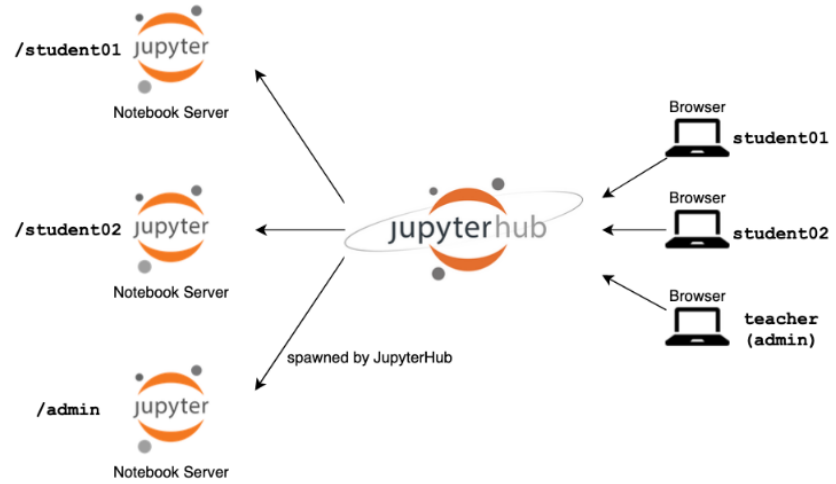
他者が作ったテンプレートの流用も可能

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★

```
In [ ]:
##################################################
### 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名
ugroup_name = 'handson403'

# プロバイダ
vc_provider = 'aws'
```

図表を組み合わせた説明を挿入できる

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

```
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) 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。

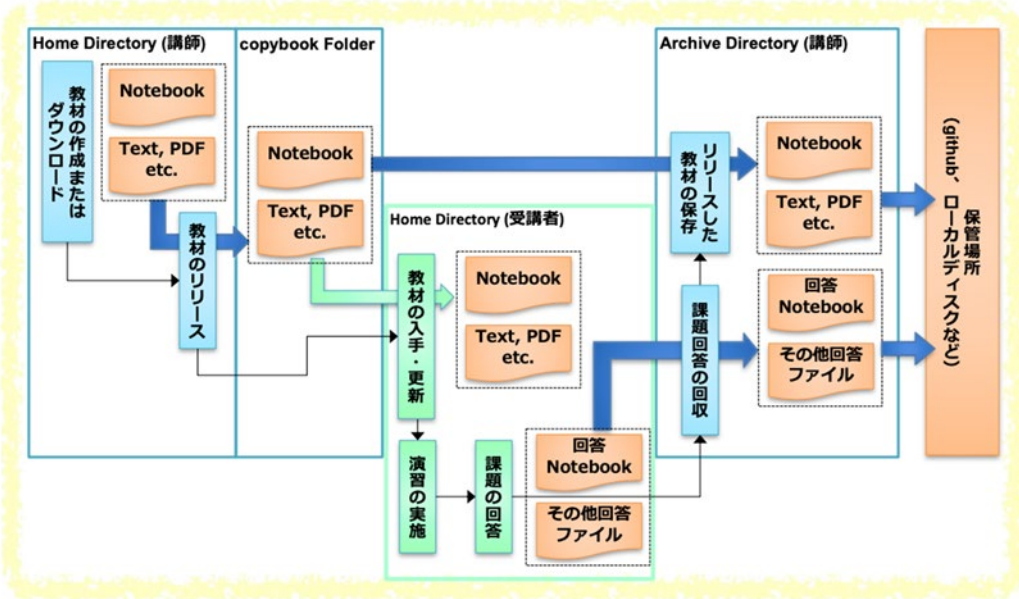
TLJHコンテナイメージの取得

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。

Copyright 2022 National Institute of Informatics

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）

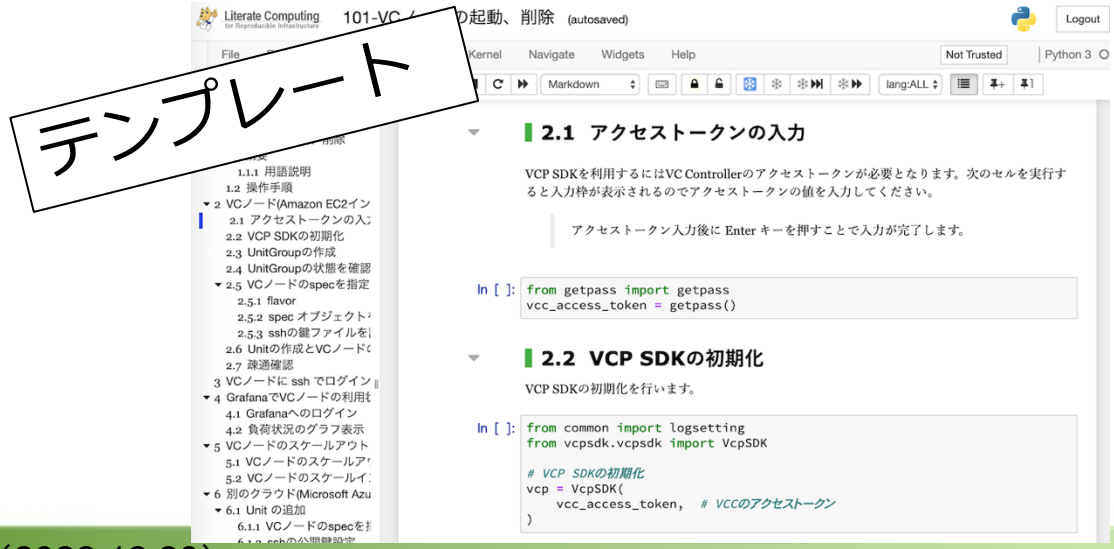
Copyright 2022 National Institute of Informatics

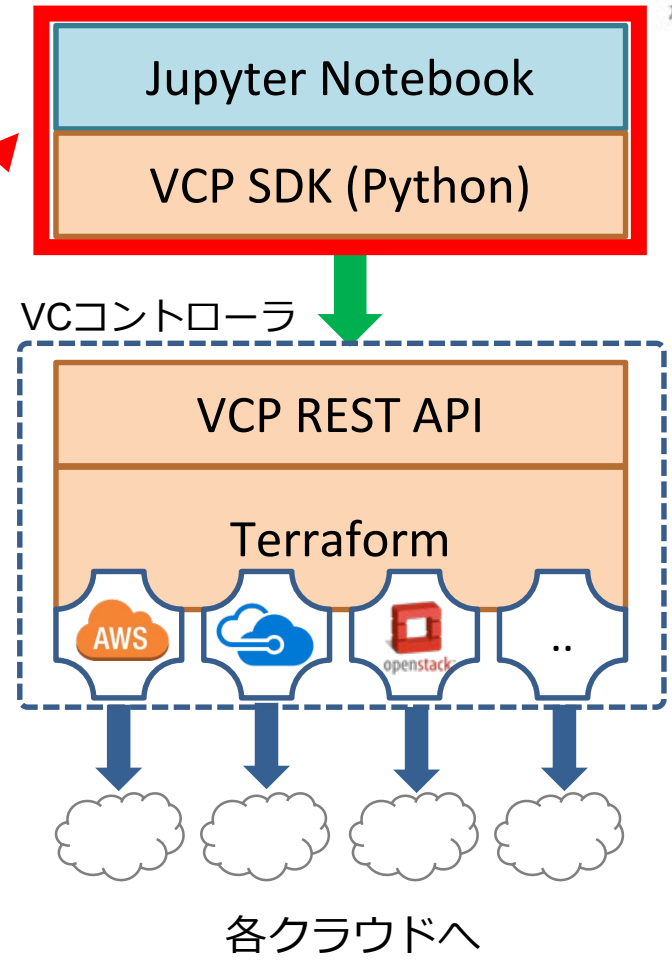
OCSを支えるVCPの仕組み

Copyright 2022 National Institute of Informatics

管理ソフトウエアの概要(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





(VC利用者)

Copyright 2022 National Institute of Informatics

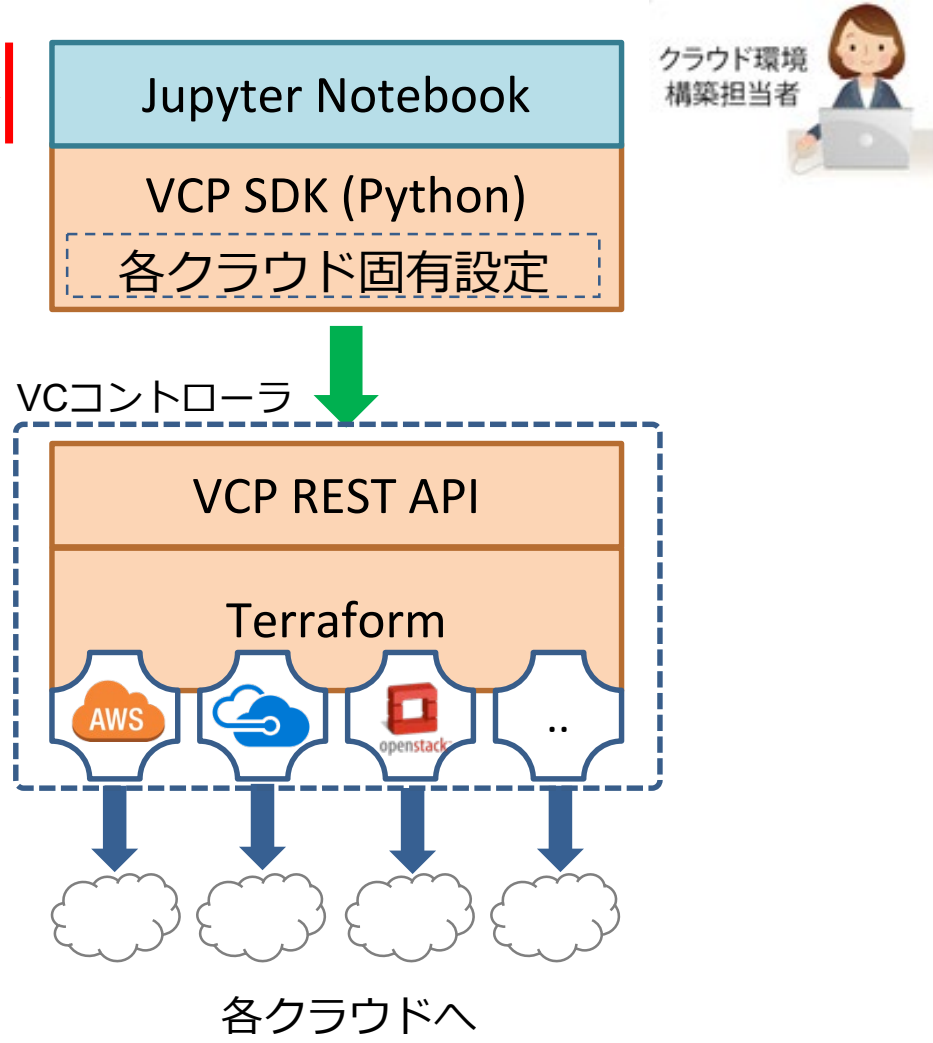
管理ソフトウエアの概要（2）

- 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

1.2 新規 server を作成

[5]:

```
#
# 作成するserverのspec情報を作成
#
spec = sdk.get_spec("aws", "medium")

#
# 変更できること
#
# spec.num_nodes = 1
spec.num_nodes = 2
#spec.instance_type = 't2.medium'
# spec.params_v = ['/opt:/opt']
# spec.volume_size = 40
# spec.volume_type = "standard"  # standard|io1|gp2|sc1|st1
# spec.ip_addresses = ['起動するnodeの静的なIPアドレス']
# spec.image = 'vcp/base:1.5'   # base container
# 追加で使用するVolume
# spec.disks = ['vol-08cbb04b35c8c9545']

# cloud上のタグ設定
spec.set_tag('key1', 'value1')
spec.set_tag('key2', 'value2')
```

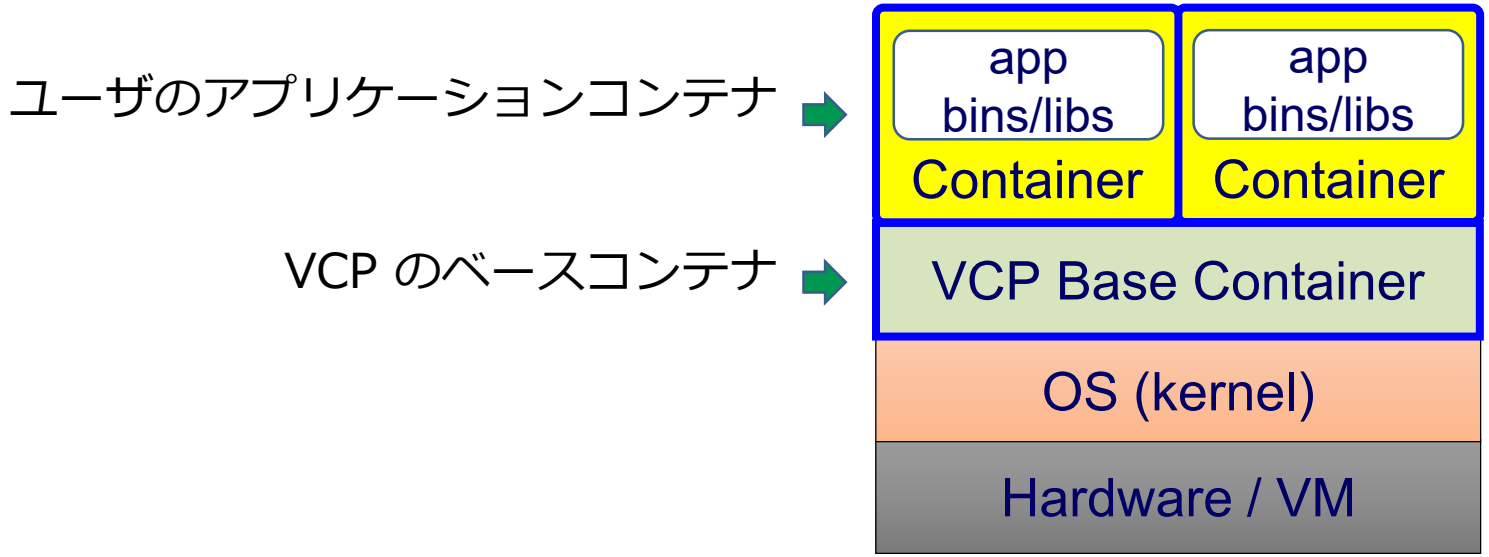


VCP SDKの中で各クラウドのmediumを定義

Copyright 2022 National Institute of Informatics

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



Copyright 2022 National Institute of Informatics

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

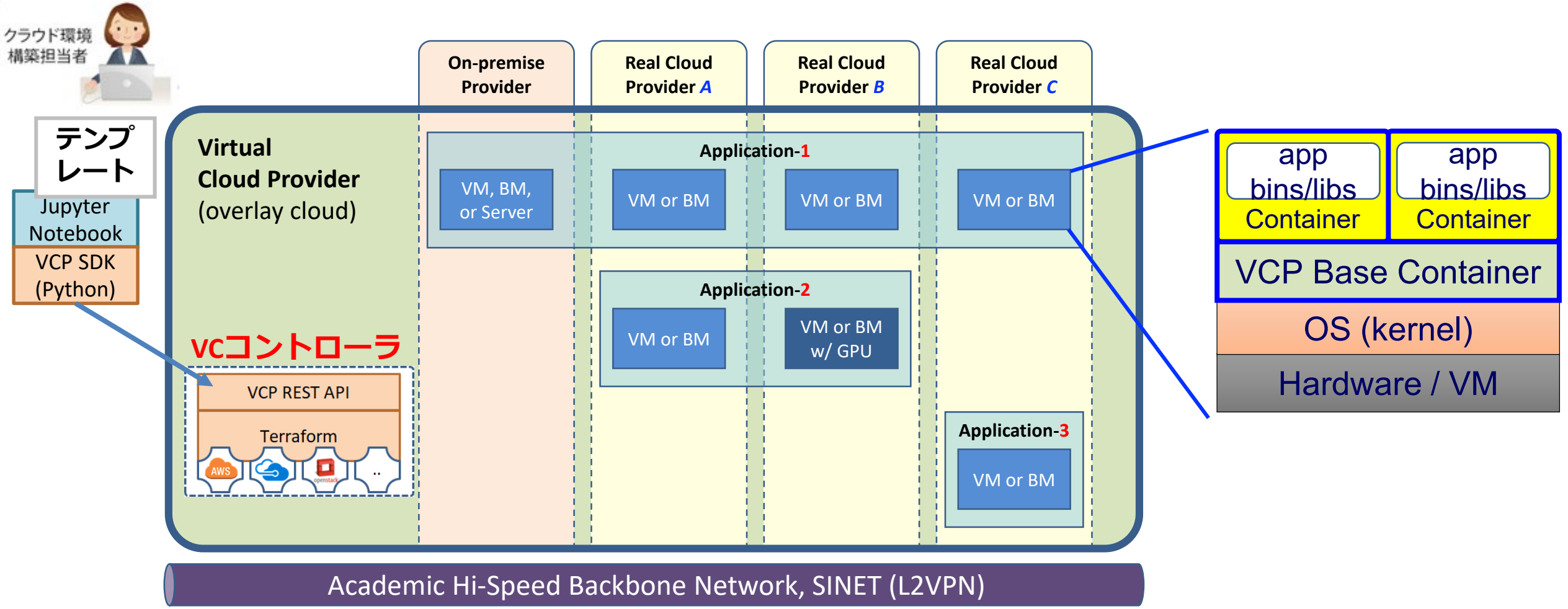
VCノード（ベースコンテナ）毎の情報

アプリコンテナ毎の情報

Copyright 2022 National Institute of Informatics

OCSを利用したアプリケーション配備例

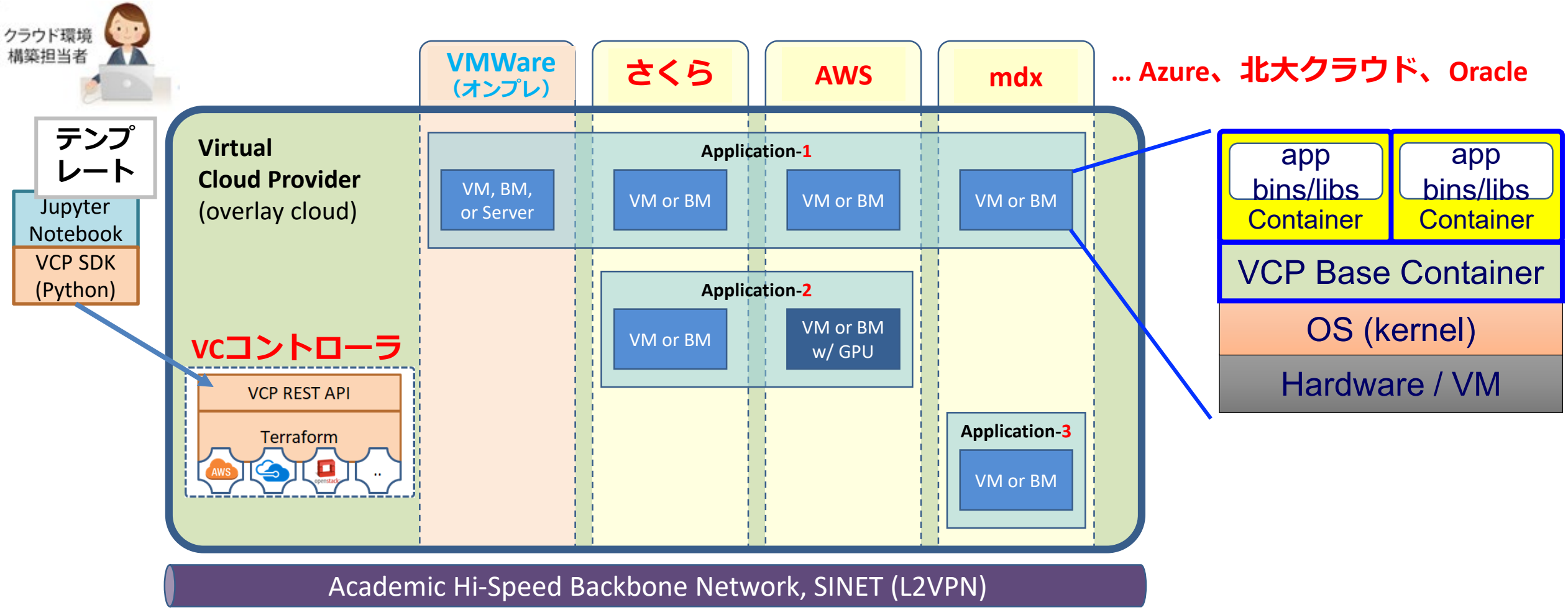
■ オンプレ・複数の実クラウドを跨ってのアプリケーション配備が可能！



Copyright 2022 National Institute of Informatics

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！

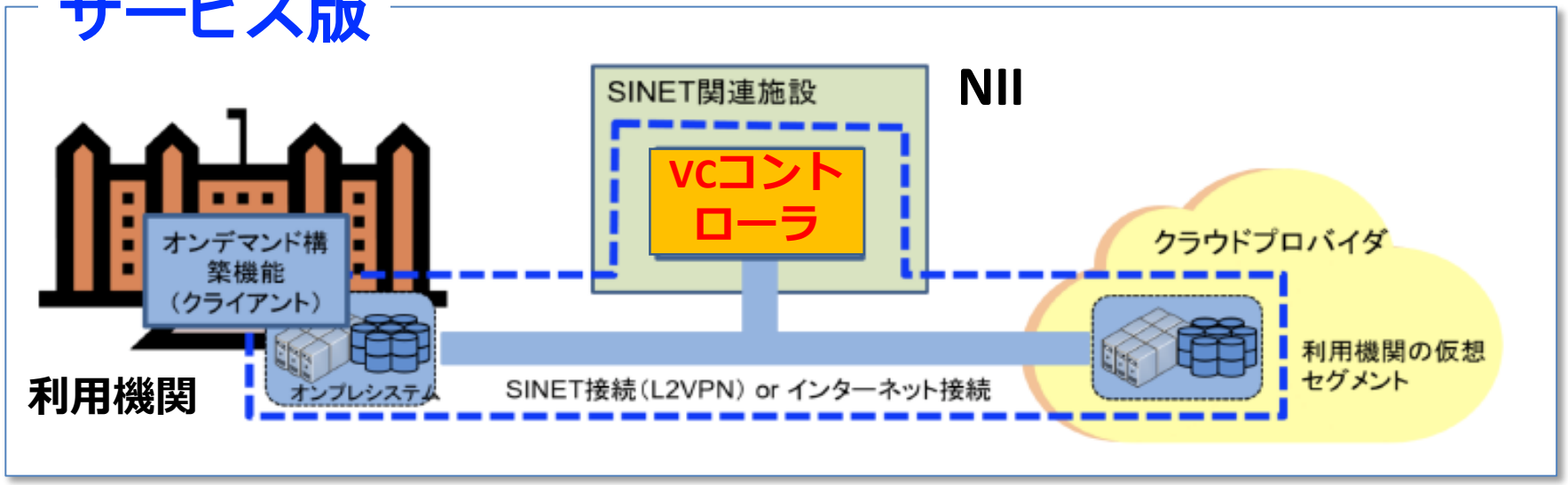


Copyright 2022 National Institute of Informatics

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

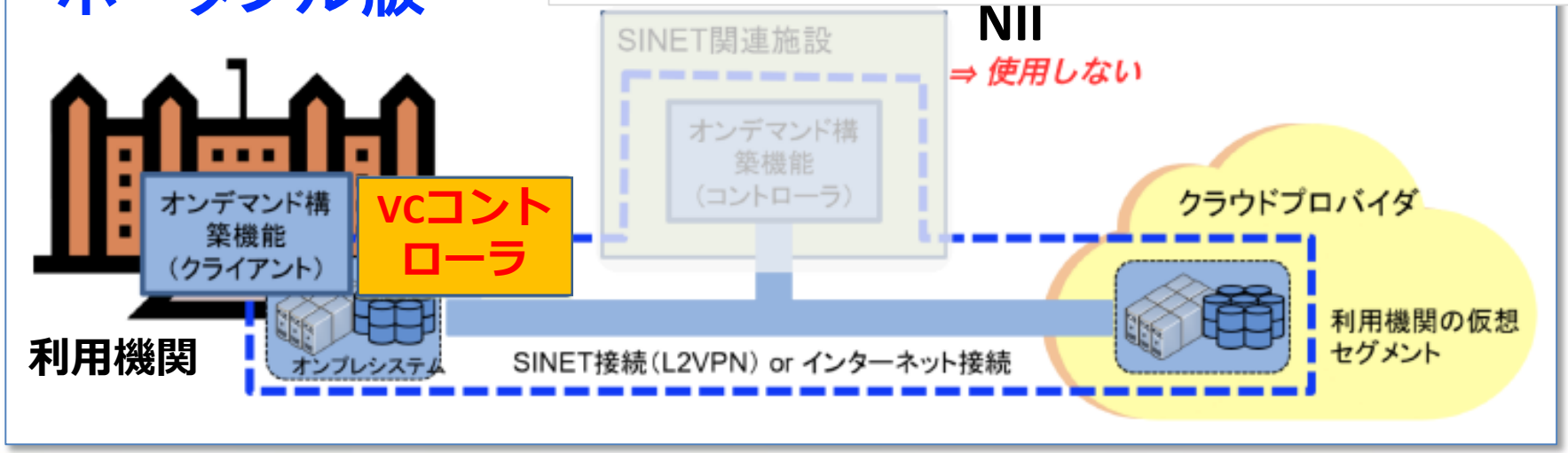
サービス版



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

本日のハンズオンはさくら上でポータブル版を使用



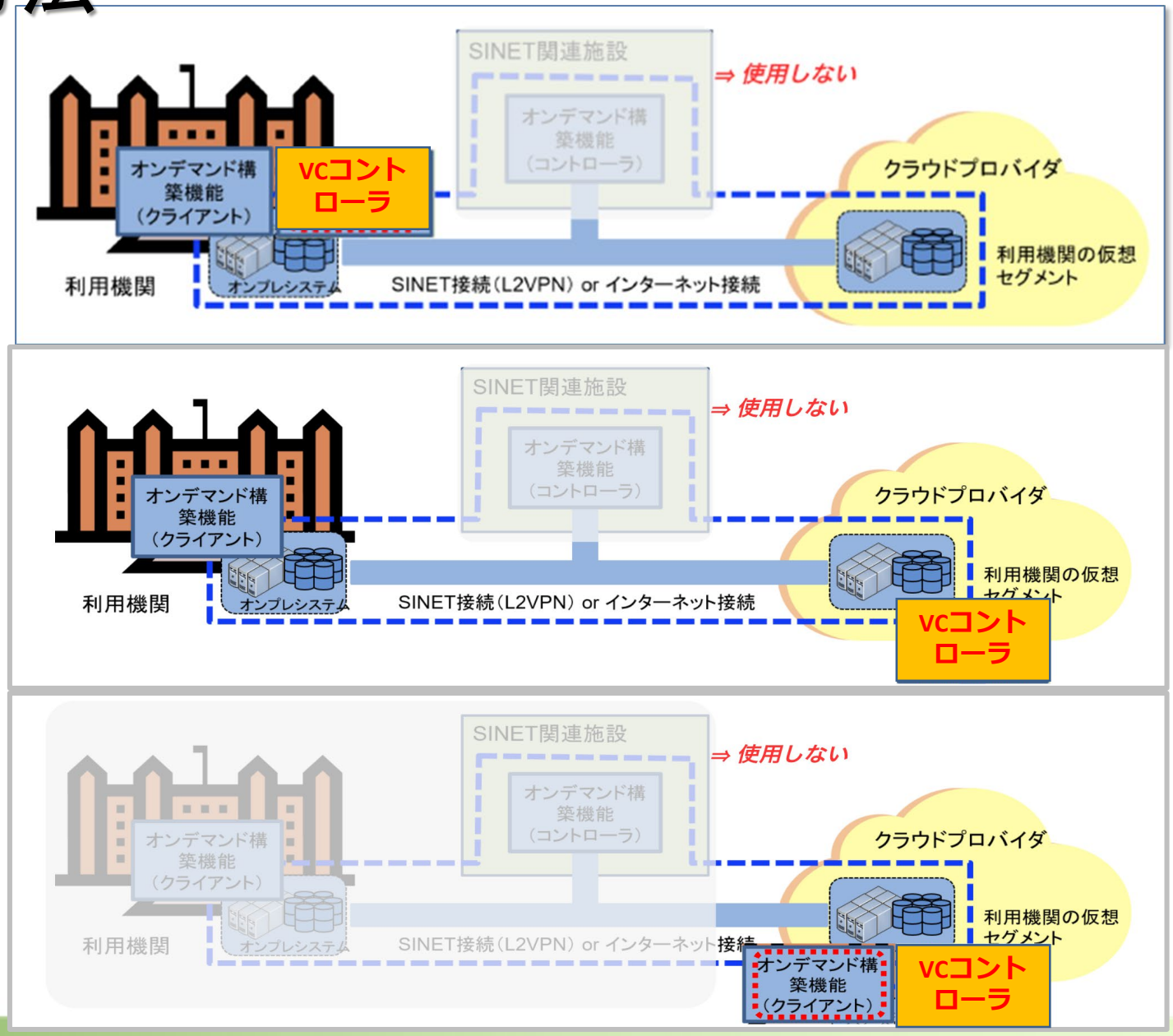
Copyright 2022 National Institute of Informatics

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）



Copyright 2022 National Institute of Informatics

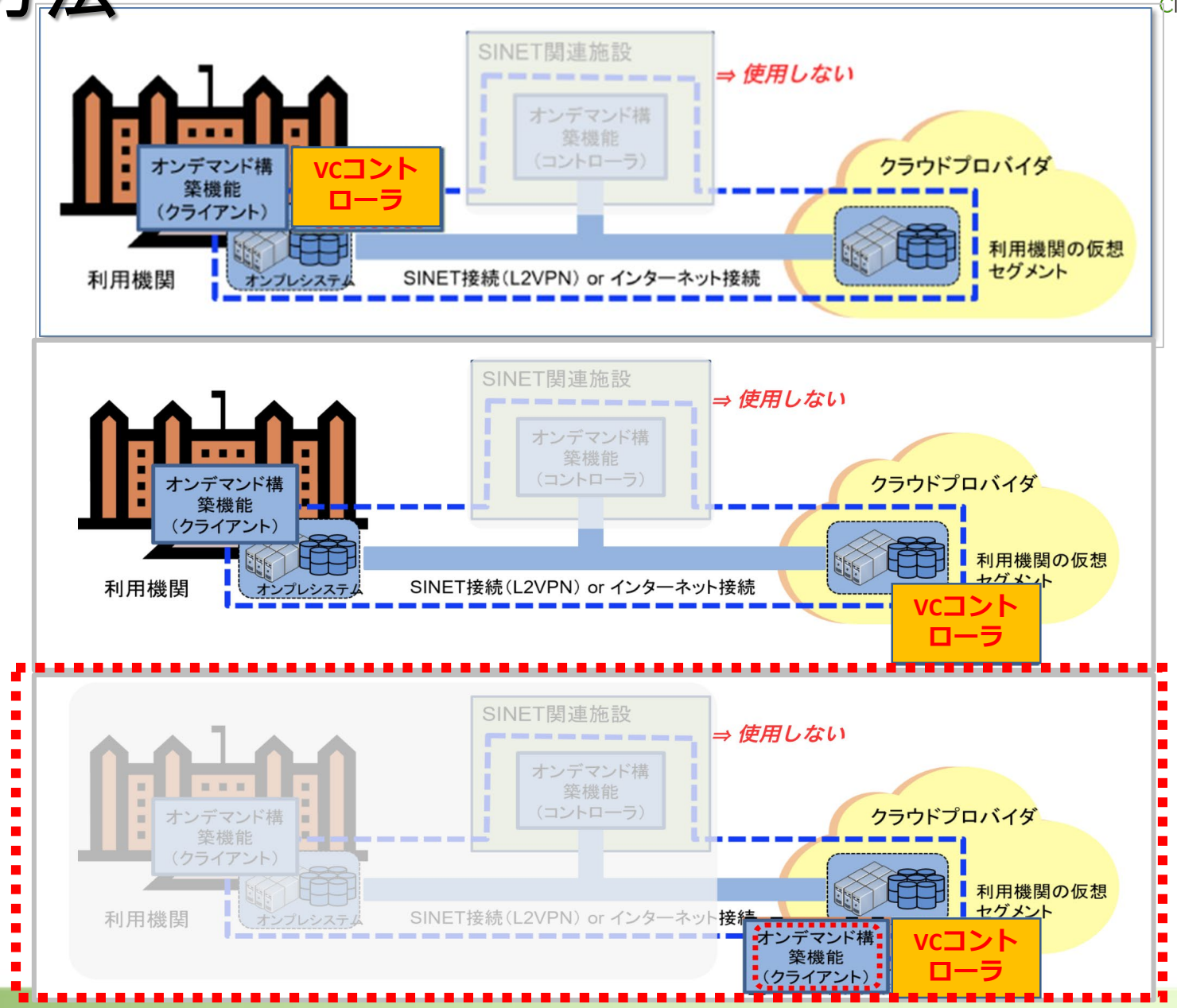
ポータブル版の構成方法

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）

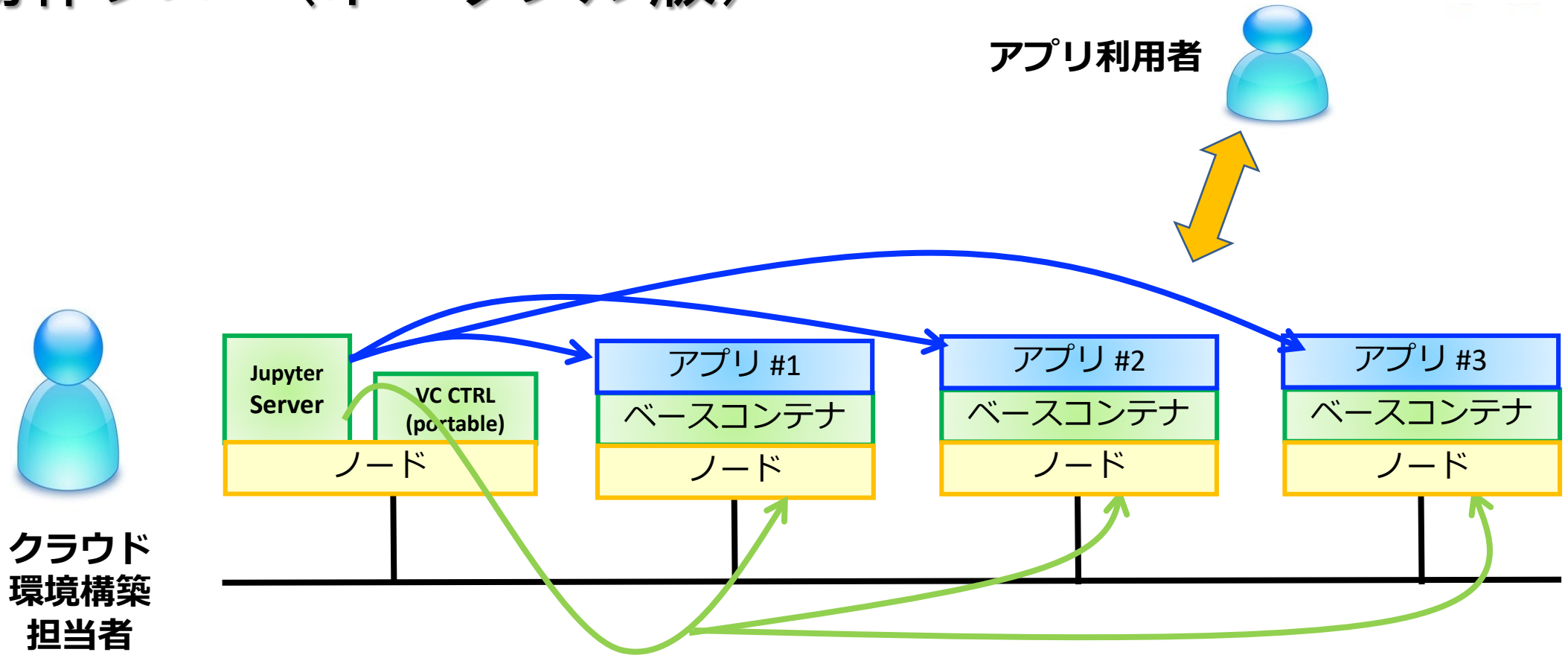
vcコントローラ： クラウド
JupyterNotebook： クラウド
（クライアント）

本日の実習はこの構成！



Copyright 2022 National Institute of Informatics

動作フロー（ポータブル版）



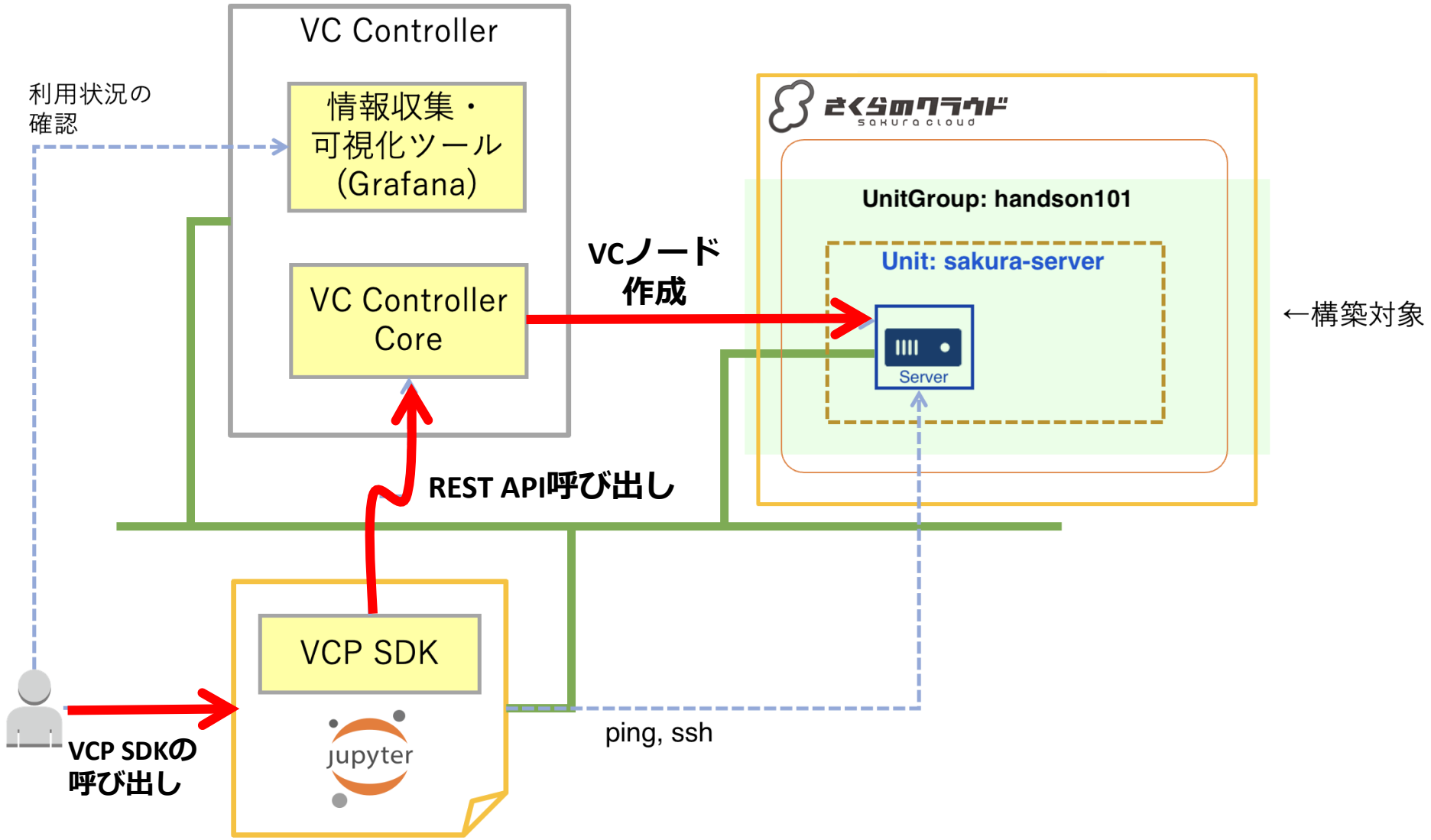
① OCSイメージ（Jupyter + VC CTRL）を起動

② VC CTRL経由でノードとベースコンテナを起動

③ ベースコンテナ上でアプリケーション環境構築

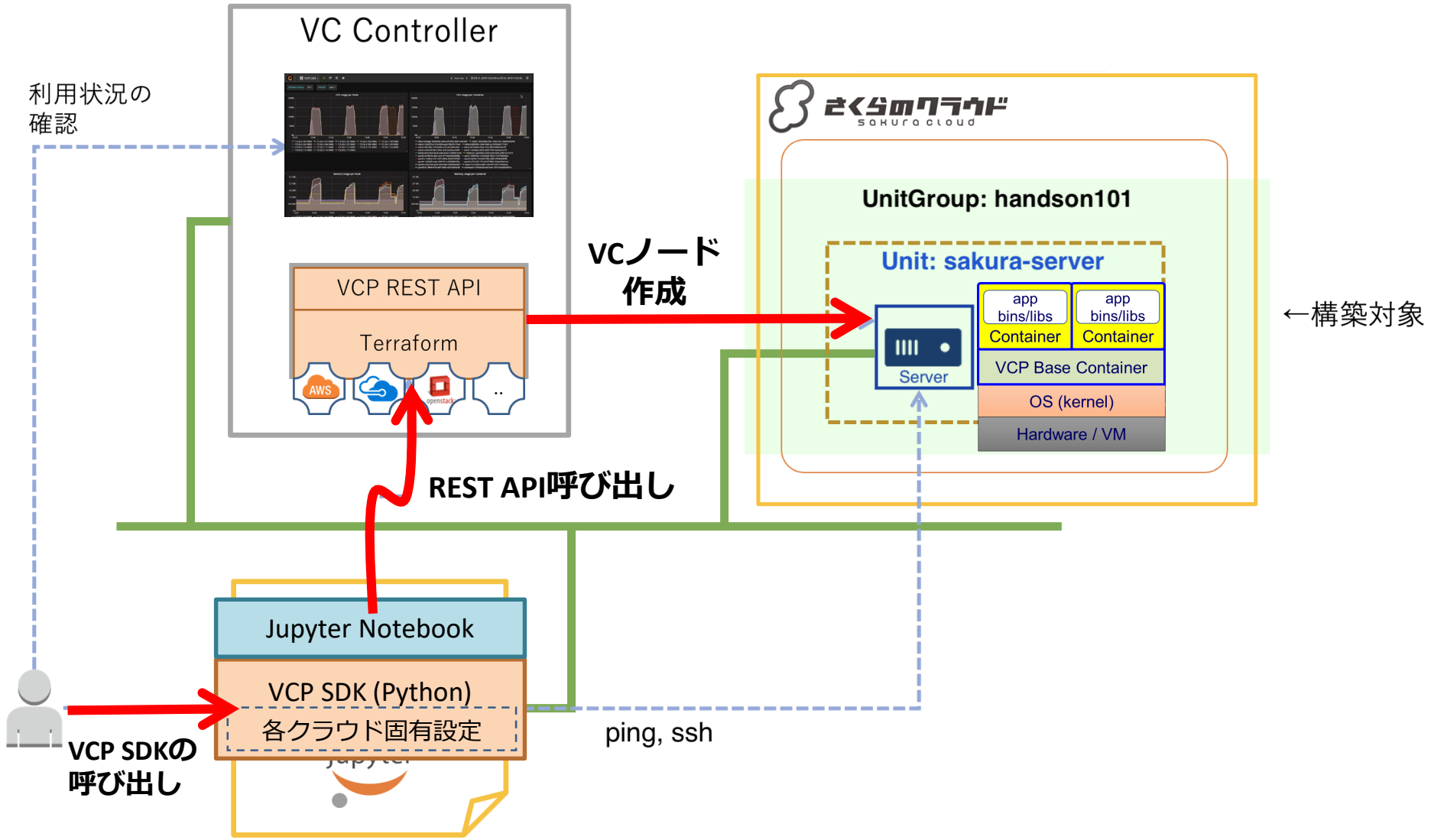
Copyright 2022 National Institute of Informatics

ハンズオン教材とのマッピング



Copyright 2022 National Institute of Informatics

ハンズオン教材とのマッピング

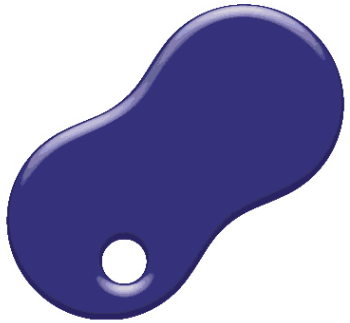


Copyright 2022 National Institute of Informatics

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

Copyright 2022 National Institute of Informatics

GakuNin Cloud



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