

学認クラウドオンデマンド構築サービス (OCS) とMCJ-CloudHubの概要

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

OCSとは

OCS提供の背景（１）

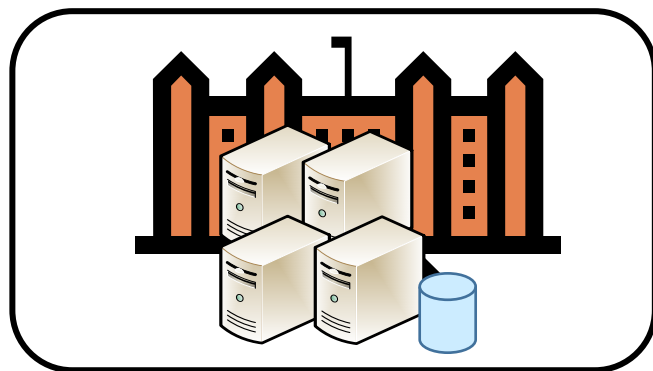
JupyterHubを用
いてPython演習環
境を立ち上げたい



クラウドA



クラウドB

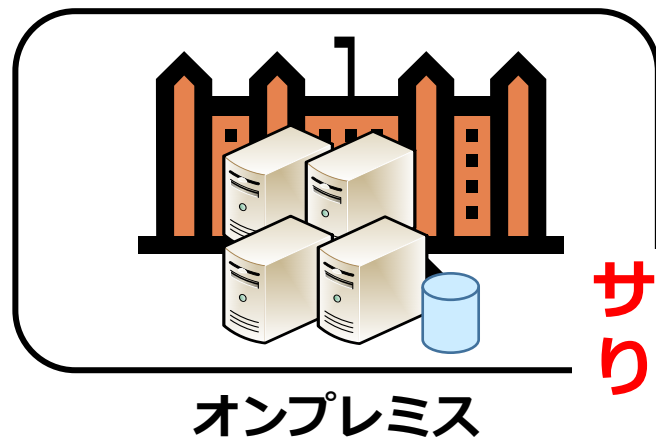
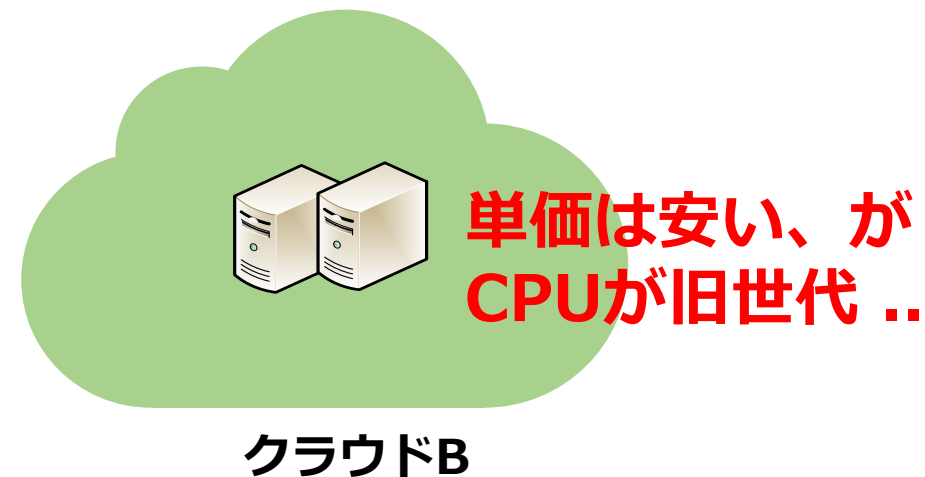
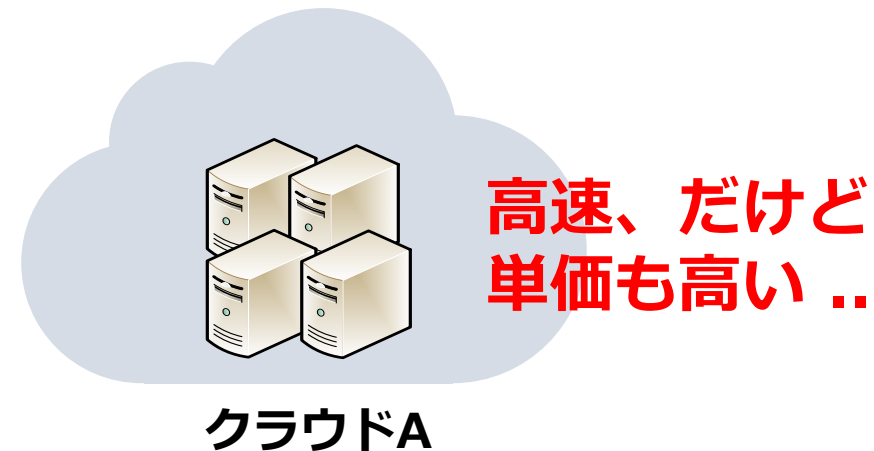


オンプレミス

OCS提供の背景（2）

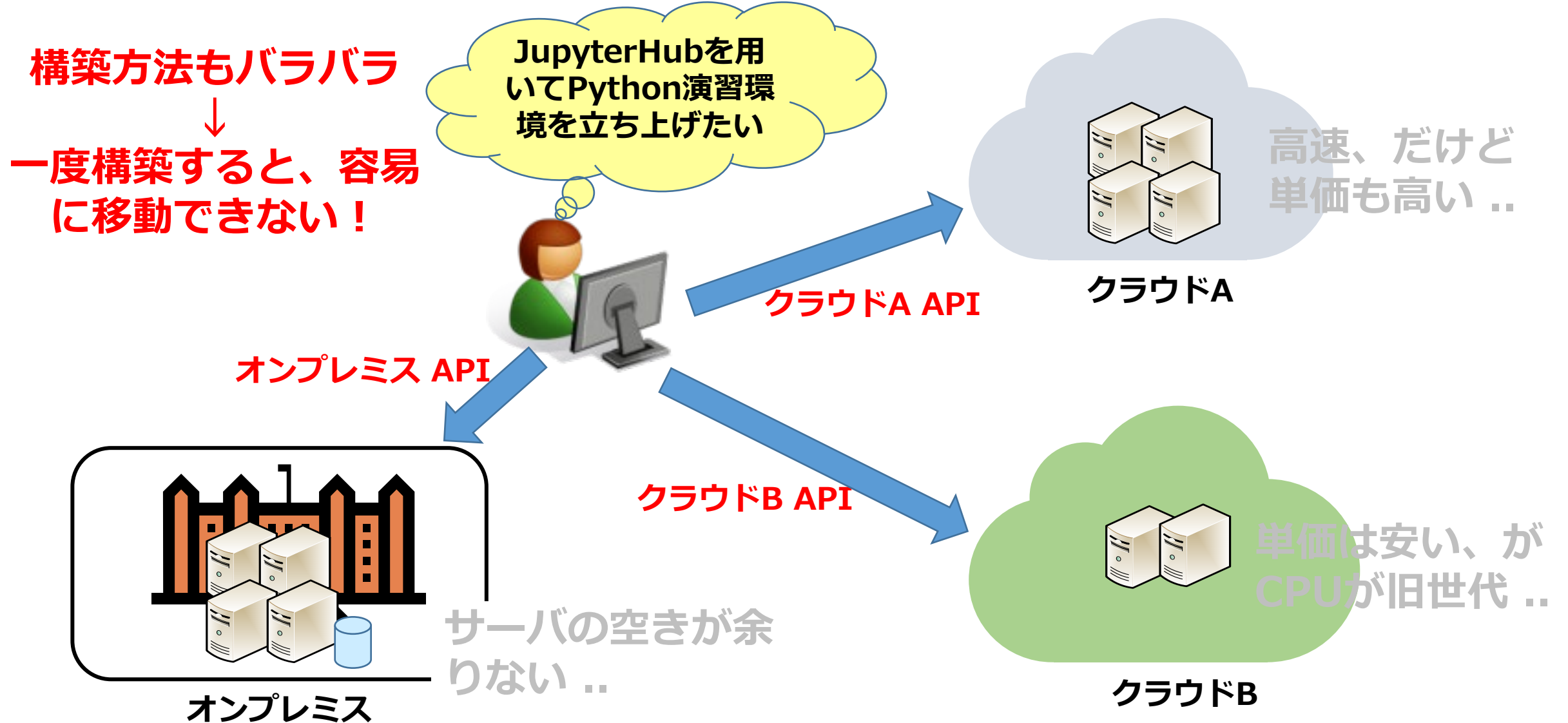
どの環境を選ぶべきか？

JupyterHubを用いてPython演習環境を立ち上げたい



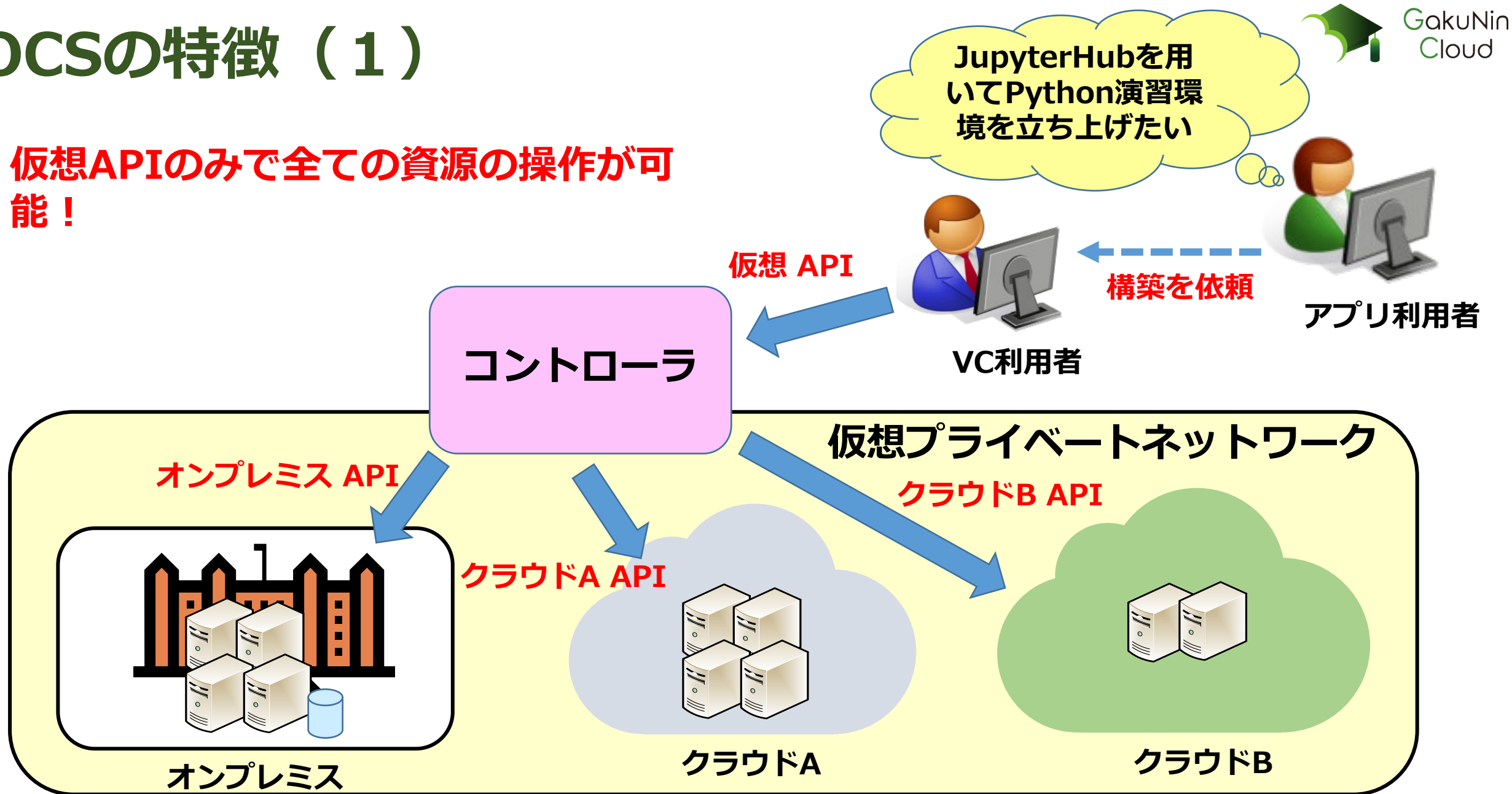
サーバの空きが余りない..

OCS提供の背景（3）



OCSの特徴（１）

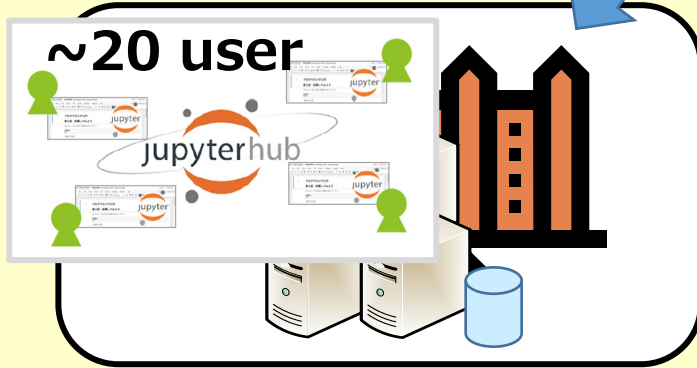
仮想APIのみで全ての資源の操作が可能！



OCSの特徴（１）

オンプレミスに
JupyterHub環境構
築！
(20 user)

オンプレミス API



オンプレミス

コントローラ

仮想 API

JupyterHubを用
いてPython演習環
境を立ち上げたい



VC利用者

構築を依頼



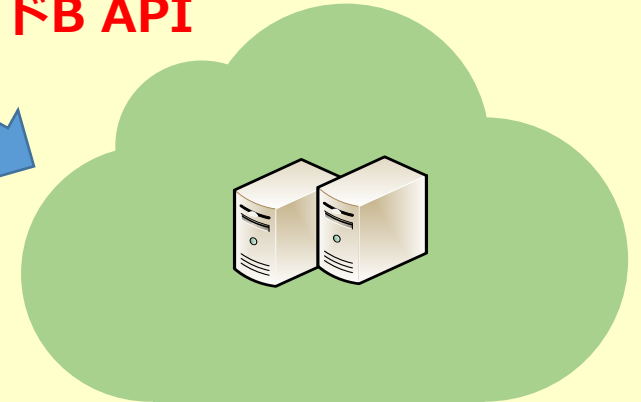
アプリ利用者

仮想プライベートネットワーク
クラウドB API

クラウドA API



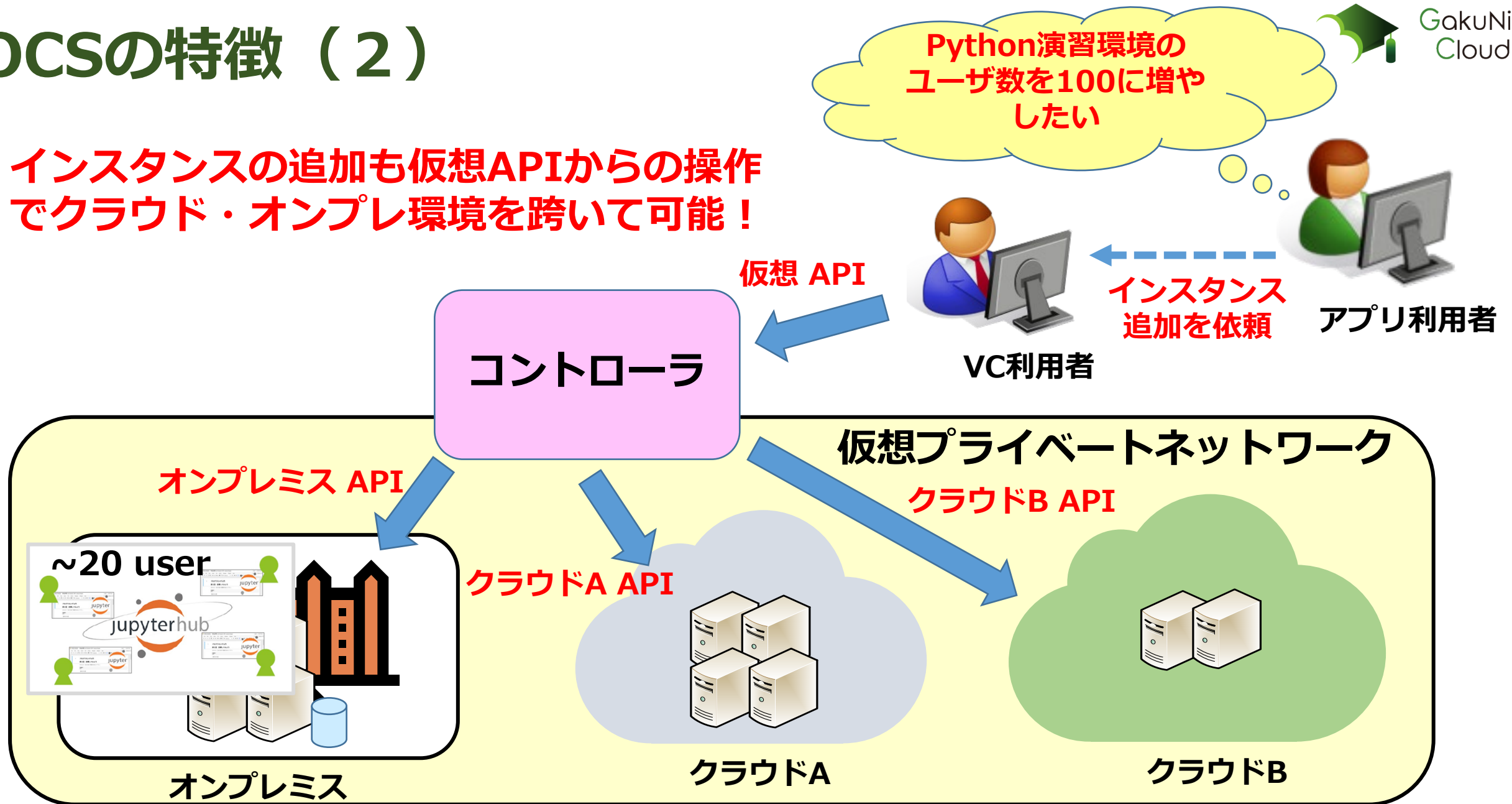
クラウドA



クラウドB

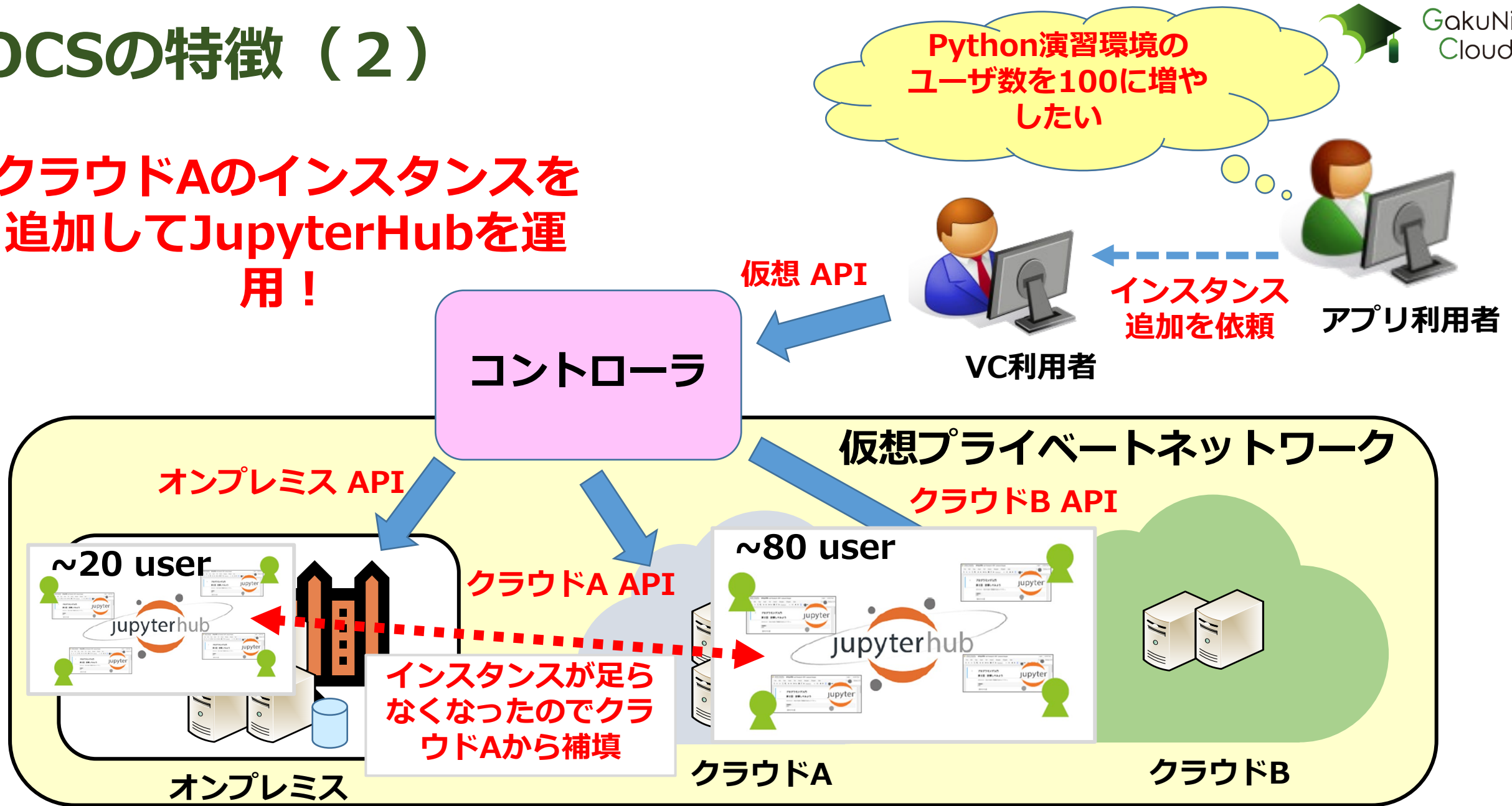
OCSの特徴（2）

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）

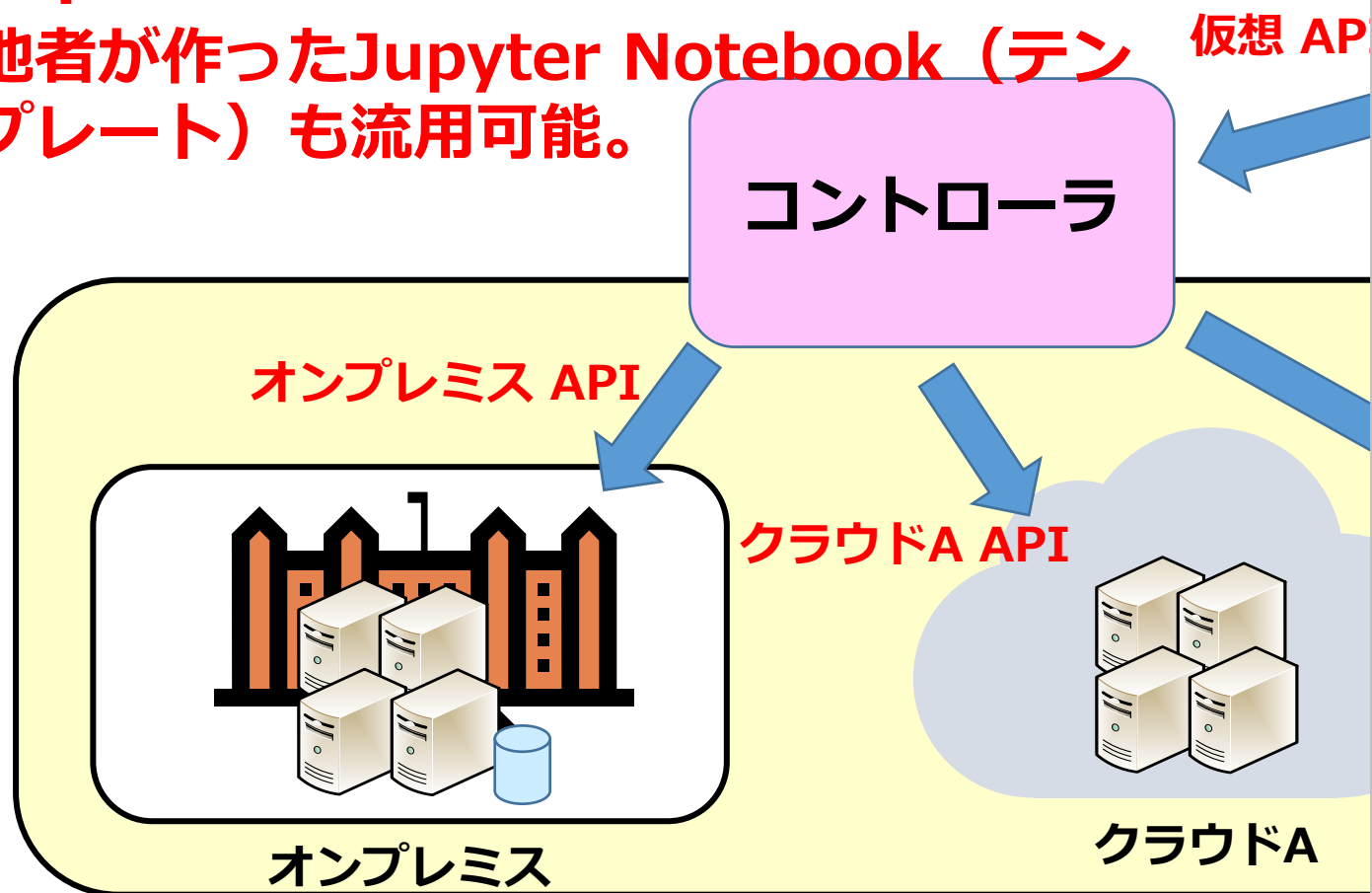
クラウドAのインスタンスを追加してJupyterHubを運用！



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）

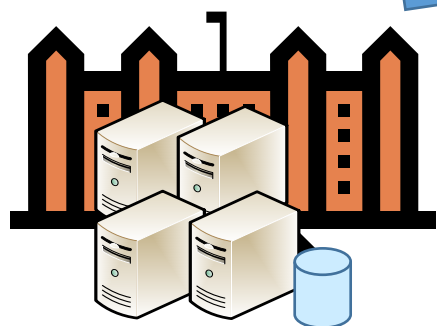
仮想APIはJupyter Notebookを介してアクセスするため、構築作業の再現性が高い！

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！

コントローラ

オンプレミス API



オンプレミス

クラウドA API



クラウドA

仮想 API

1.1 初期化 Jupyter Notebookの記述例

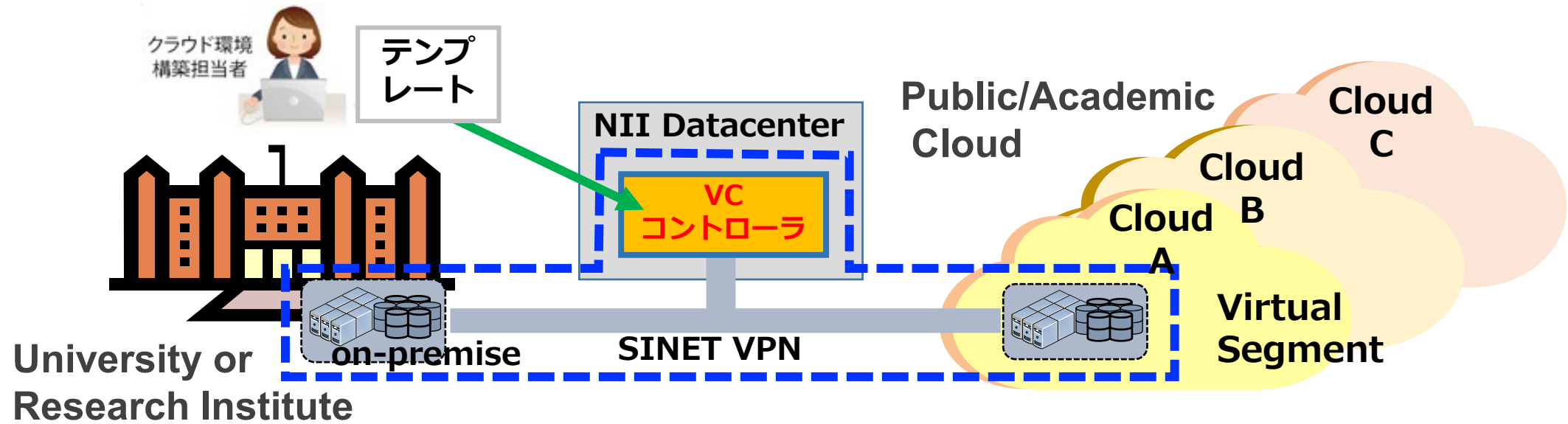
```
[1]: parameters
1 vcc_access_token = "..."
2 testname = "TEST-2022-03-15"

[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。

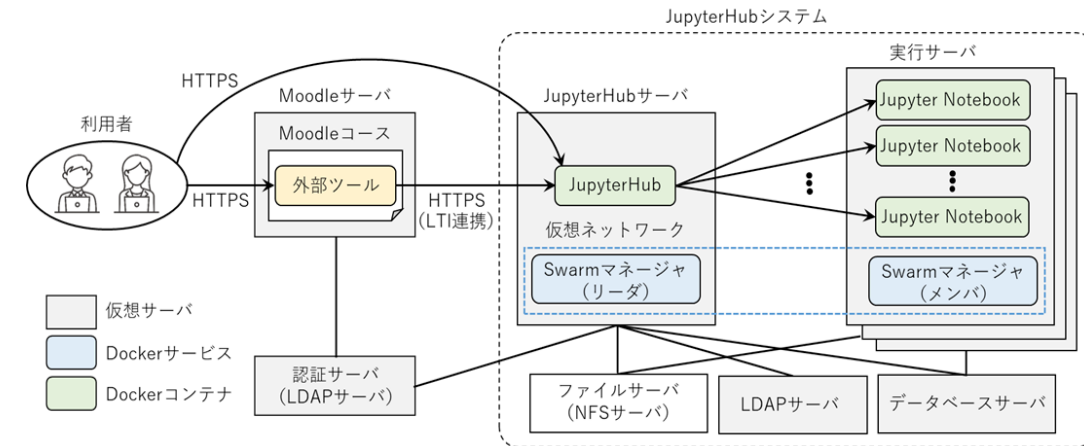
利用例（講義演習環境）

■ MCJ-CloudHubとは

- 山口大学とNIIで共同開発した講義・演習システム
 - 山口大学で運用していたオンプレミスシステムをOCSテンプレートから構築・運用出来るように拡張
 - 山口大学固有設定等の一般化

■ OCS+MCJ-CloudHubの特徴

- システム管理者と利用者（教員・学生）を分離
 - 利用者は、GUI操作のみで演習可能
 - システム管理者は、障害等が発生しない限り特別なサポートは不要（年度初めに構築するのみ）
- 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



MCJ-CloudHubの概要

（「複数科目で共同・同時利用可能なWeb型プログラミング教育支援システムのアプリケーションテンプレート開発」より引用）

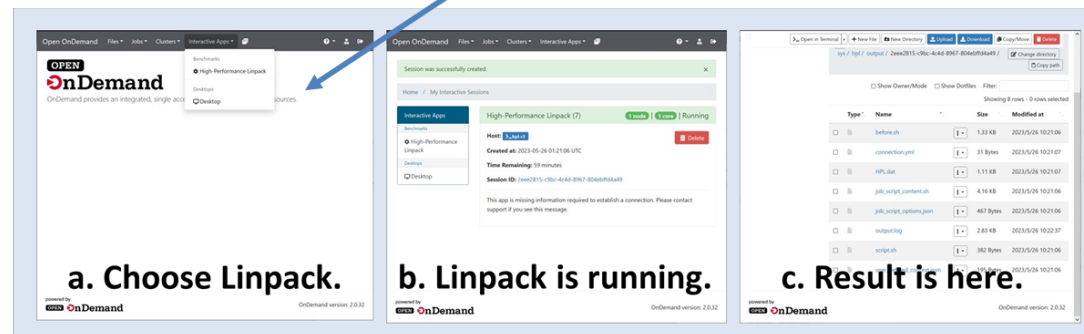
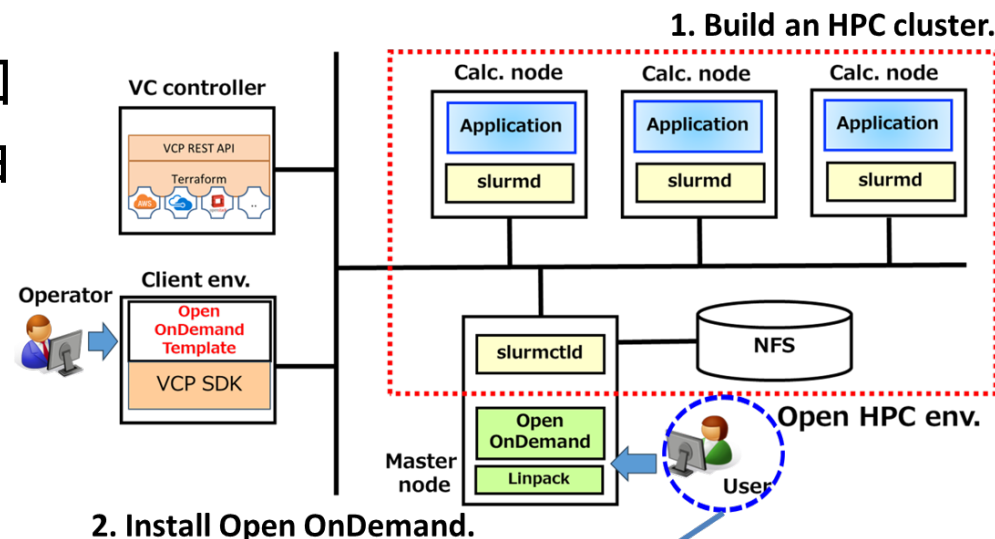
利用例（HPCクラスタ）

■ Open OnDemandとは

- 概要： 初心者がHPCクラスタの前提知識なしに機械学習などのアプリケーション実行を可能とするシステム
- 課題： 構築・運用が容易でない

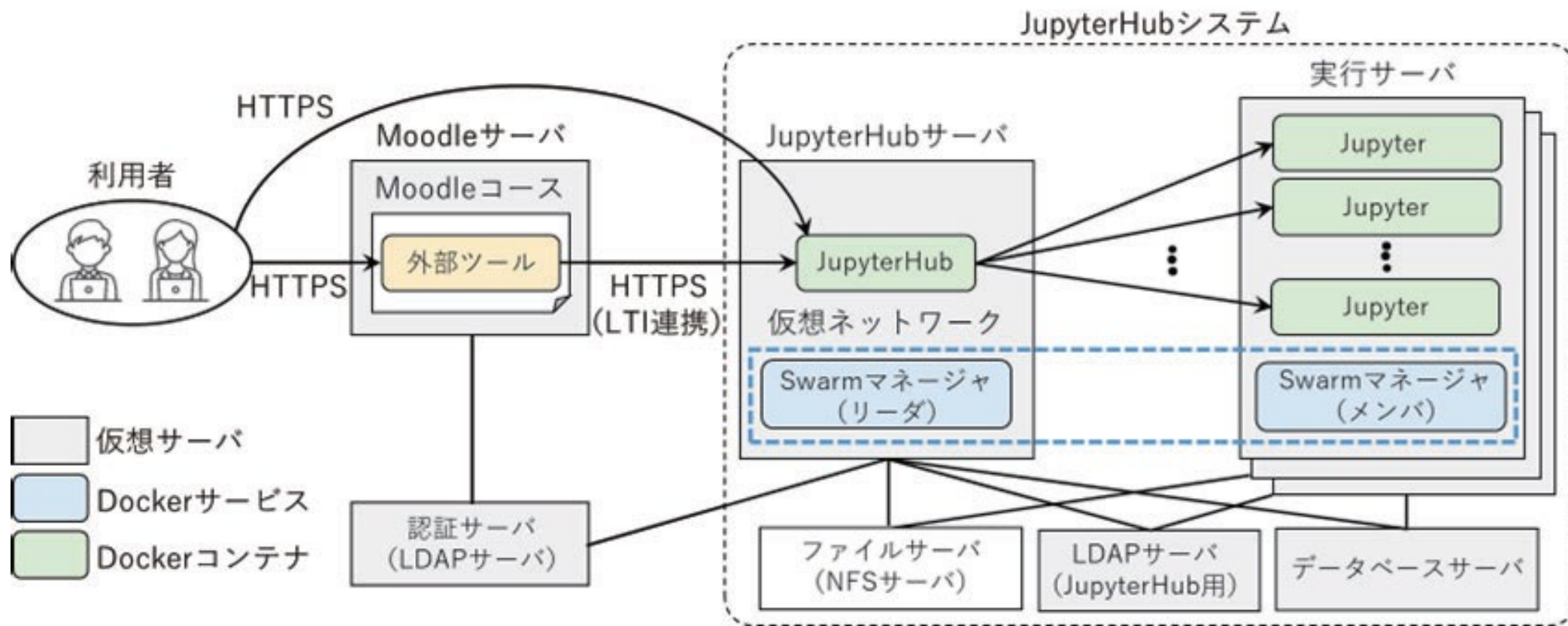
■ OCS + Open OnDemandの特徴

- OCSテンプレート化することでOpen OnDemand環境を容易に構築できる手段を提供
- 従来から提供していたOpen HPC テンプレートを用いて構築したHPCクラスタが前提



MCJ-CloudHubの概要

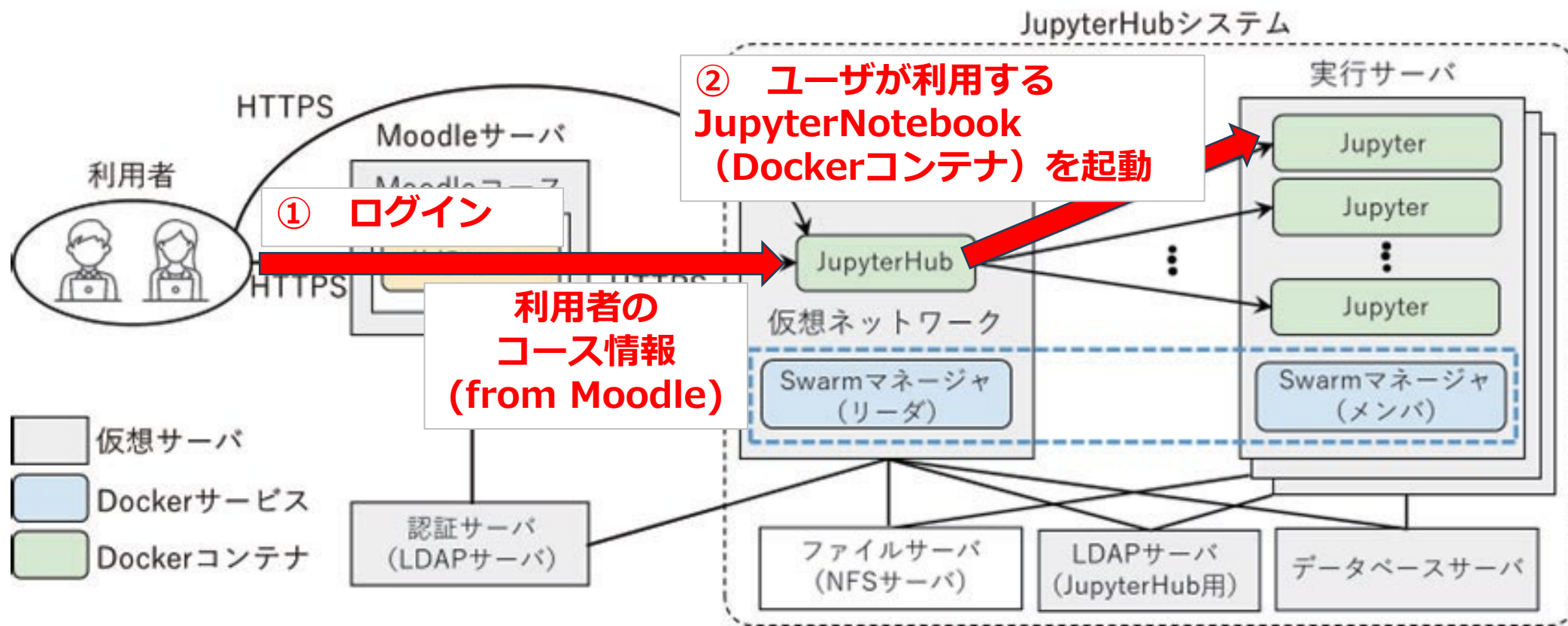
山口大学で運用中の講義演習システム



JupyterHub + nbgrader + 山口大学独自拡張

山口大学オンプレミスシステムで運用中

山口大学で運用中の講義演習システム



JupyterHub + nbgrader + 山口大学独自拡張

山口大学オンプレミスシステムで運用中

山口大学で運用中の講義演習システム

■特徴

- コンピュータシステムに詳しくない文系学部の教員・学生でも容易に利用可能で、且つ、管理者の負担も少ない
 - 利用者（教員・学生）は、GUIからの操作のみで演習を行える
 - 課題の配布・回収・採点は、GUIから全ての操作が可能な**nbgrader**を採用
 - システム管理者は情シス教員等スキルのある方を想定し、年度ごとに構築すれば運用可能な設計

■他機関で運用する上での課題

- 各機関のご事情に応じてオンプレミス and/or クラウド上で講義演習システムの構築・運用を可能に
- 山口大学オンプレミス環境依存設定や実装の分離
- ユーザ認証や属性情報取得方法の汎用化
- 学習管理システムからの履歴者情報一覧取得方法の汎用化
- 古い版数のOSSを前提にしたシステム構築

MCJ-CloudHubにおける課題の解決方法

- 各機関のご事情に応じてオンプレミス and/or クラウド上で講義演習システムの構築・運用を可能に
 - ➡ **構築・運用手順のOCSアプリケーションテンプレート化**
- 山口大学オンプレミス環境依存設定や実装の分離
 - ➡ **山口大学固有設定・実装を分離した上でOCSアプリケーションテンプレート化**
- ユーザ認証や属性情報取得方法の汎用化
 - ➡ **MoodleとLTI認証連携で得られる情報のみを用いる**
- 学習管理システムからの履歴者情報一覧取得方法の汎用化
 - ➡ **MoodleとLTI認証連携で利用できるNRPSを用いる**
- 古い版数のOSSを前提にしたシステム構築
 - ➡ **JupyterHub, nbgrader等を最新版にアップグレード、設定ファイルを更新**

MCJ-CloudHubの概要

利用者



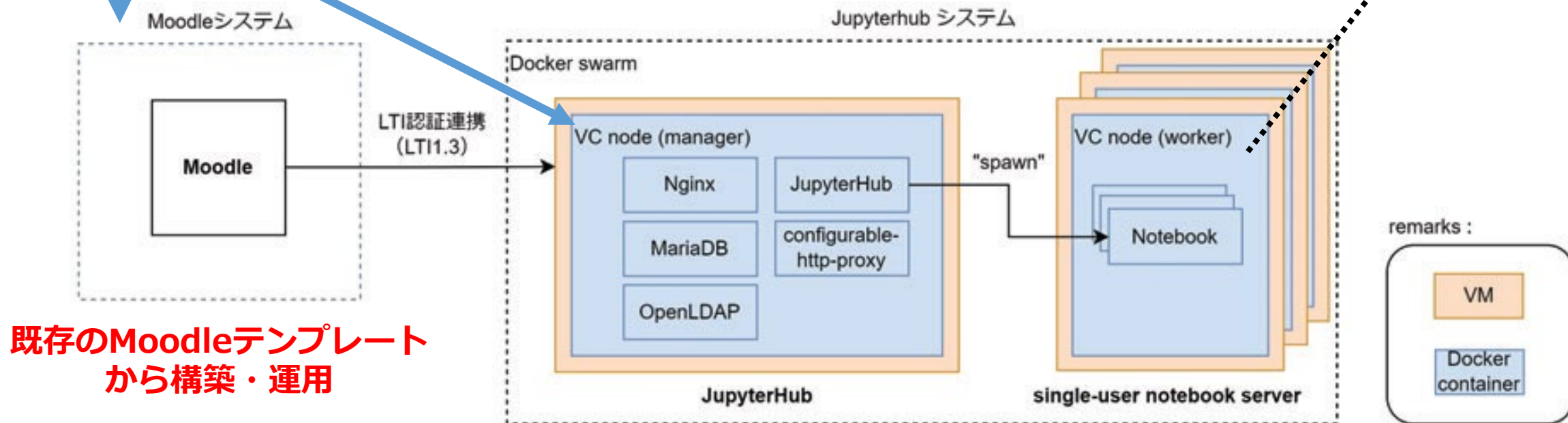
<https://github.com/nii-gakunin-cloud/ocs-templates/tree/master/Moodle-Simple>
<https://github.com/nii-gakunin-cloud/mcj-cloudhub>

両テンプレートは上記からダウンロードできます！

①
ログイン

②演習環境に
アクセス

ユーザごとに立ち上がるJupyterNotebook
(Dockerコンテナ)



既存のMoodleテンプレート
から構築・運用

新たに開発したMCJ-CloudHubテンプレートから構築・運用

MCJ-CloudHubへのログイン

① Moodleログイン

Example University Moodle Demo
にログインする

student01

.....

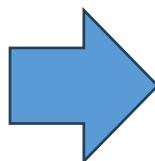
ログイン

パスワードを忘れましたか?

いくつかのコースにはゲストアクセスできます

ゲストとしてアクセスする

日本語 (ja) クッキー通知



② Moodleで受講しているコースを選択

demo moodle Home ダッシュボード マイコース

×

一般

アナウンスメント

○ MCJ-CloudHub

トピック 1

トピック 2

トピック 3

トピック 4

mcj-handson

コース 参加者 評価 コンビデンシ

一般

すべてを折りたたむ

アナウンスメント

MCJ-CloudHub

トピック 1

トピック 2

教師・もしくはMoodle管理者が設定した外部ツールリンクから、MCJ-CloudHubにログイン



③ ユーザ用のDockerコンテナ起動を待つ

jupyterhub Home Token

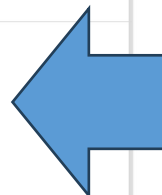
Your server is starting up.

You will be redirected automatically when it's ready for you.

Spawning server...

Event log

④ ログイン後の画面



ファイル 編集 表示 実行 カーネル Nbgrader タブ 設定 ヘルプ

名前前でファイルをフィルタ

名前

mcjh

最終更新

2 hours ago

Notebook

Python 3 (ipykernel)

コンソール

Python 3 (ipykernel)

その他

ターミナル

テキストファイル

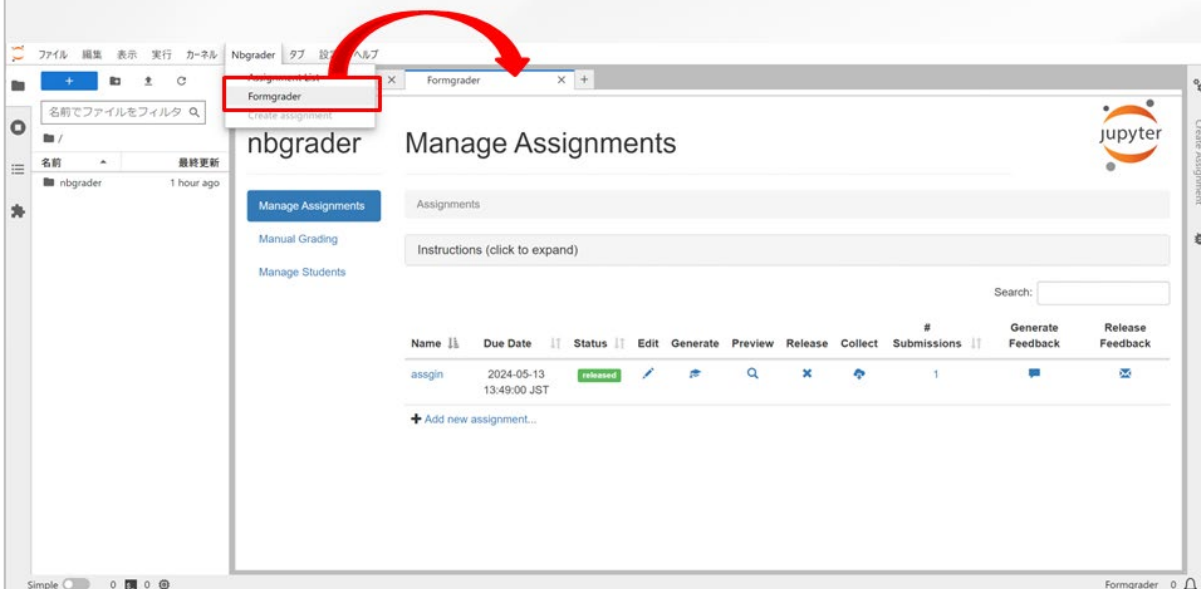
Markdownファイル

Pythonファイル

コンテキストヘルプを表示

課題の配布（教師側）

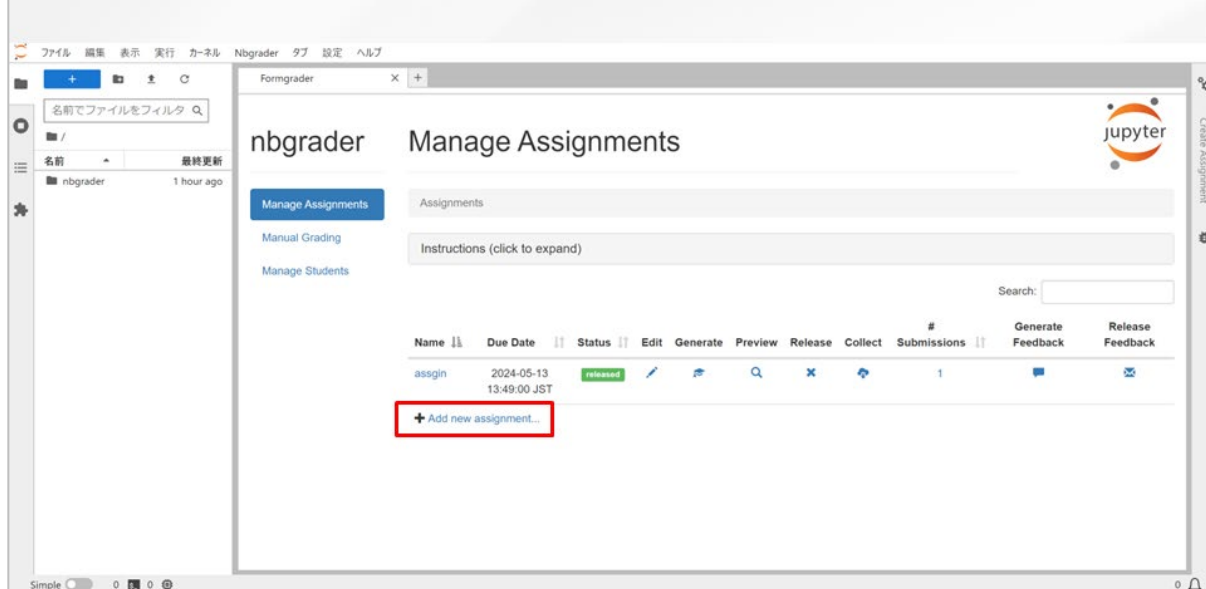
課題管理画面を開く



The screenshot shows the nbgrader web interface. The top navigation bar includes 'ファイル', '編集', '表示', '実行', 'カーネル', 'Nbgrader', 'タブ', '設定', and 'ヘルプ'. The 'Formgrader' tab is highlighted with a red box and a red arrow pointing to it. Below the navigation bar, the 'Manage Assignments' page is visible, featuring a sidebar with 'Manage Assignments', 'Manual Grading', and 'Manage Students'. The main content area shows a table of assignments with columns for Name, Due Date, Status, Edit, Generate, Preview, Release, Collect, Submissions, Generate Feedback, and Release Feedback. A single assignment named 'assgin' is listed with a due date of 2024-05-13 13:49:00 JST and a status of 'released'.

© ASCADE, Inc. 7

Formgraderにて、課題の作成を行う

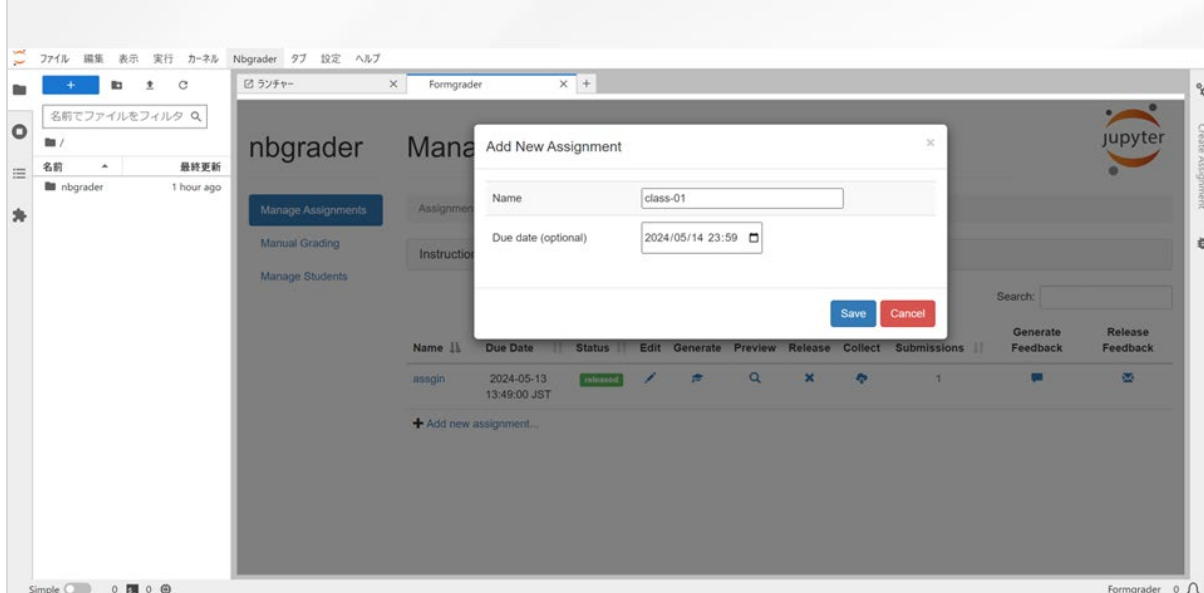


The screenshot shows the nbgrader web interface, similar to the previous one. The 'Formgrader' tab is selected. The 'Manage Assignments' page is displayed. At the bottom of the assignment table, the '+ Add new assignment...' button is highlighted with a red box, indicating the next step in the process.

© ASCADE, Inc. 8

課題の配布（教師側）

課題名と提出期限を入力



nbgrader Manage Assignments

Add New Assignment

Name: class-01

Due date (optional): 2024/05/14 23:59

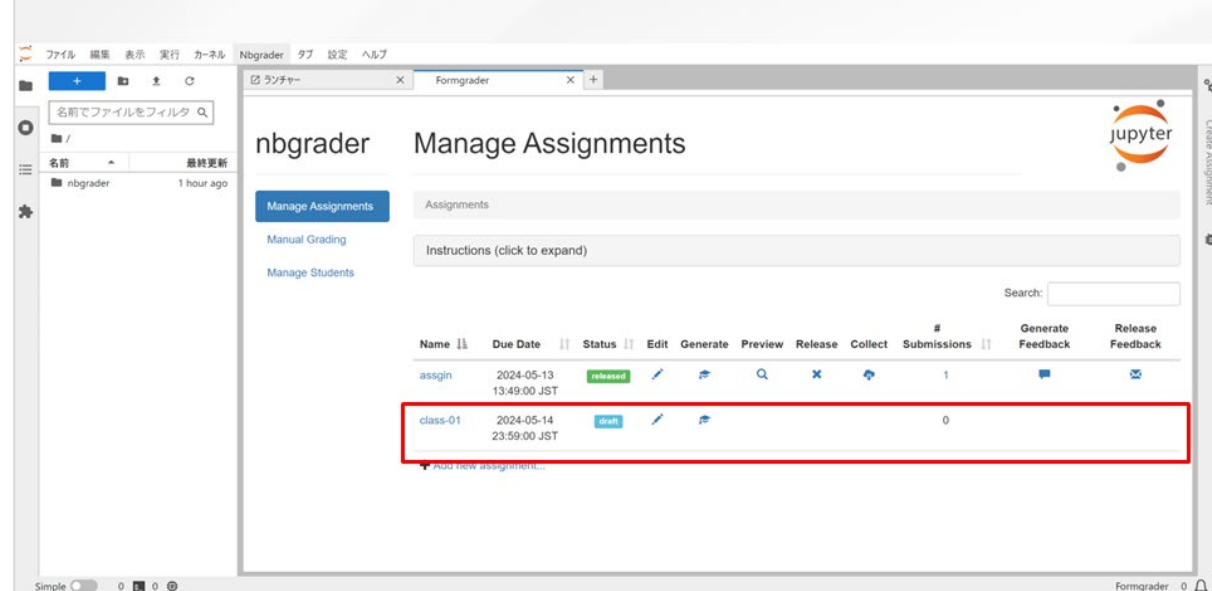
Save Cancel

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	Submissions	Generate Feedback	Release Feedback
assign	2024-05-13 13:49:00 JST	released						1		

+ Add new assignment...

© ASCADE, Inc. 9

課題が追加される



nbgrader Manage Assignments

Assignments

Instructions (click to expand)

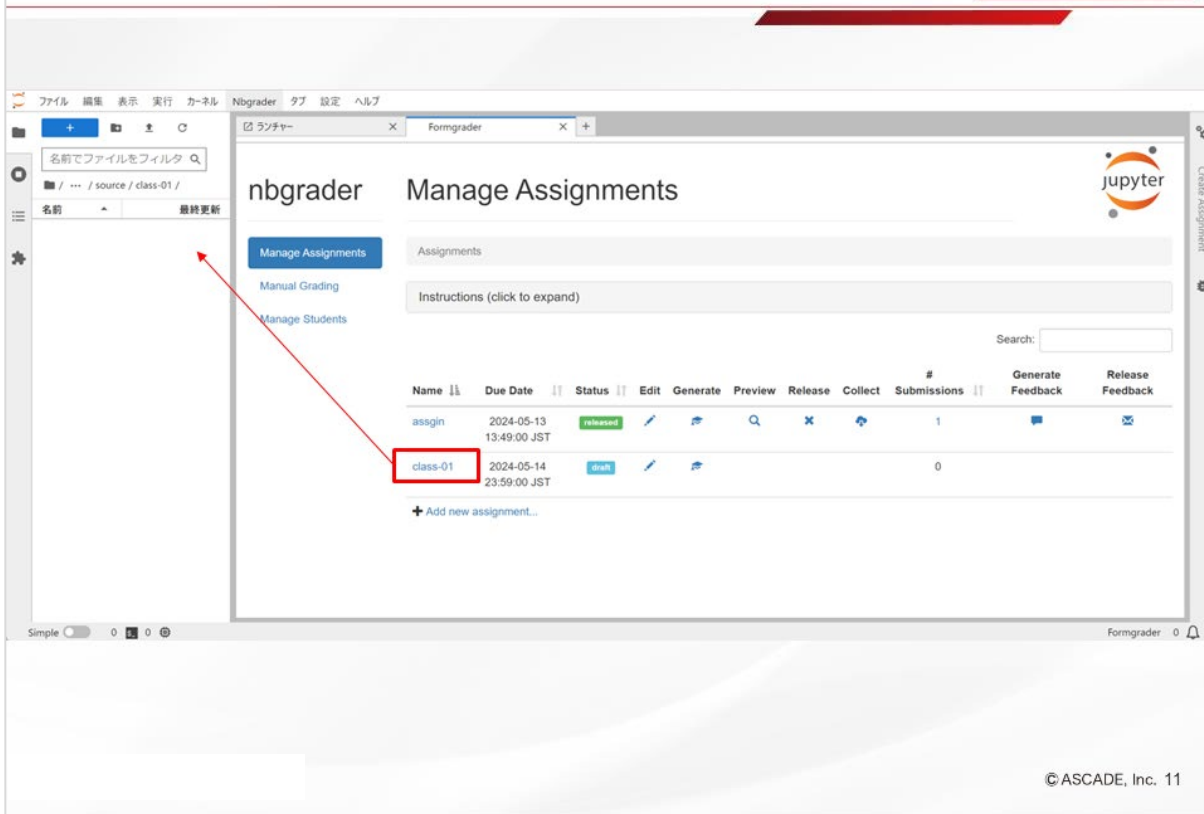
Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
assign	2024-05-13 13:49:00 JST	released						1		
class-01	2024-05-14 23:59:00 JST	draft						0		

+ Add new assignment...

© ASCADE, Inc. 10

課題の配布（教師側）

課題名をクリックすると、課題を作成するディレクトリが開く



nbgrader Manage Assignments

Assignments

Instructions (click to expand)

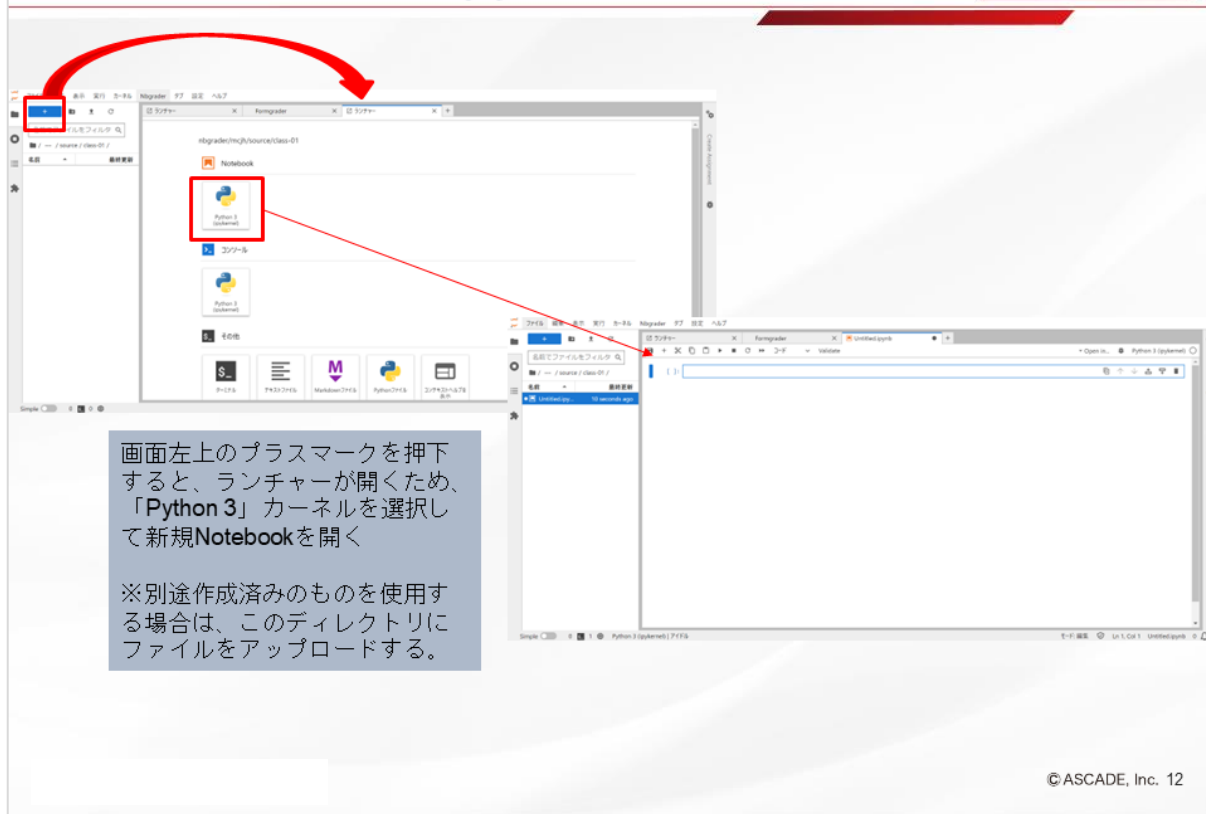
Search:

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	Submissions	Generate Feedback	Release Feedback
assgin	2024-05-13 13:49:00 JST	released						1		
class-01	2024-05-14 23:59:00 JST	draft						0		

+ Add new assignment...

© ASCADE, Inc. 11

課題ファイル（.ipynb）を作成する



画面左上のプラスマークを押下すると、ランチャーが開くため、「Python 3」カーネルを選択して新規Notebookを開く

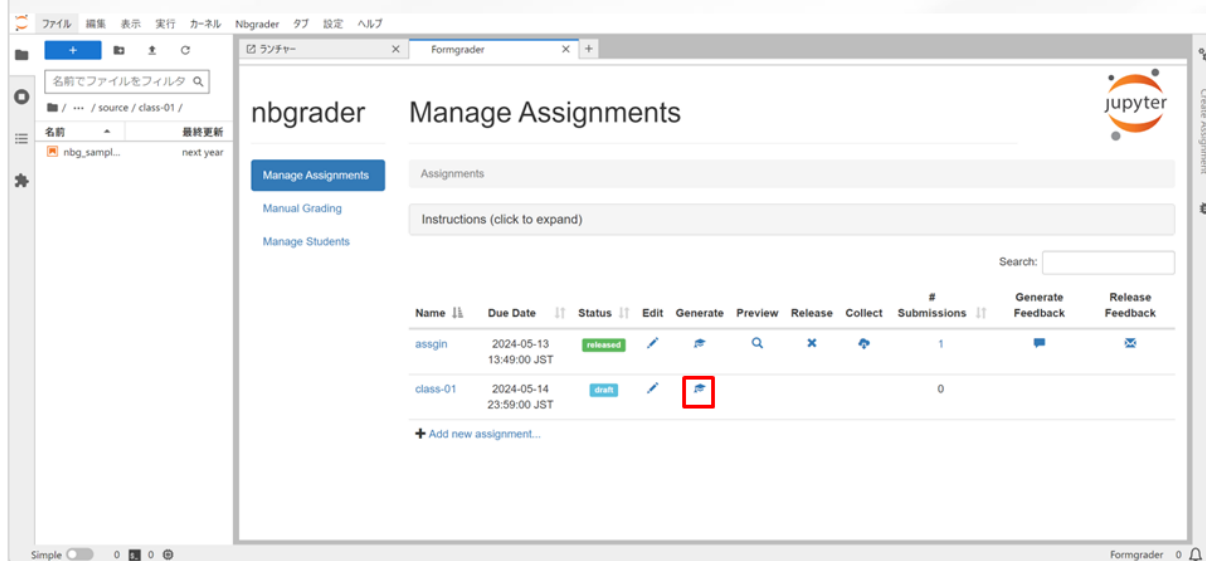
※別途作成済みのものを使用する場合は、このディレクトリにファイルをアップロードする。

© ASCADE, Inc. 12

課題の配布（教師側）

課題ファイル作成後、リリース用ファイルを作成

Nbgraderの機能によって、課題ファイルの解答例が記述された部分等を削除した、学生用のNotebookを生成することができる。



© ASCADE, Inc. 13

学生用Notebook比較

nbgrader サンプルノートブック

```
[ ]: def plz_ans(a, b):
    ### BEGIN SOLUTION
    return a + b
    # END SOLUTION
[ ]: assert plz_ans(1, 1) == 2
```

nbgrader サンプルノートブック

```
[ ]: def plz_ans(a, b):
    # YOUR CODE HERE
    raise NotImplementedError()
[ ]: assert plz_ans(1, 1) == 2
```

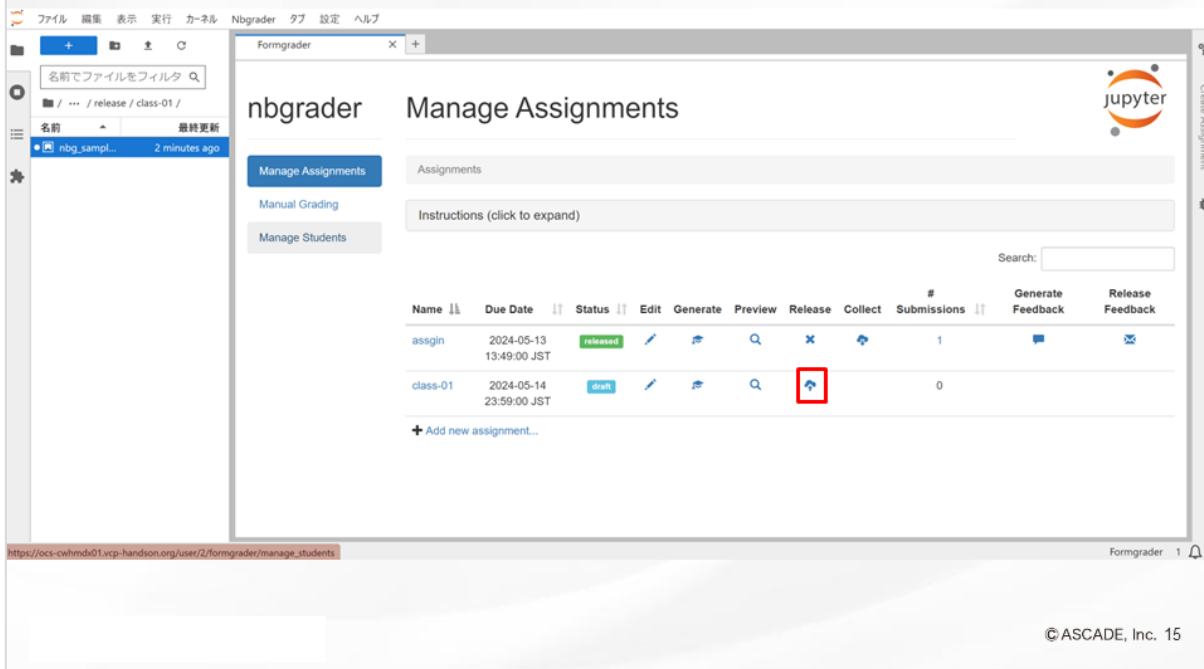
学生用Notebook

14

課題の配布（教師側）

課題ファイルのリリース

学生向けのNotebook生成が完了すると、そのNotebookを学生向けにリリースすることができるようになる。



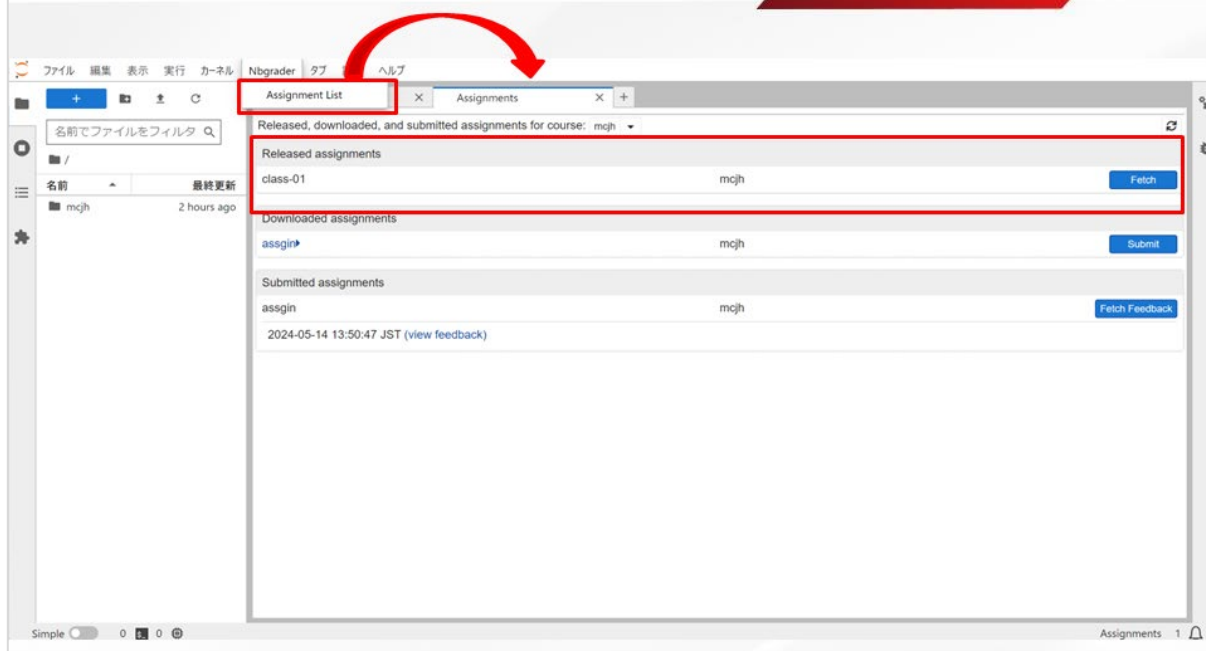
The screenshot shows the nbgrader 'Manage Assignments' interface. The table lists two assignments: 'assgin' and 'class-01'. The 'class-01' assignment is in 'draft' status and has a red box around the 'Release' button in the 'Release' column.

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
assgin	2024-05-13 13:49:00 JST	released						1		
class-01	2024-05-14 23:59:00 JST	draft						0		

課題の回収・採点（自動・手動）もGUIからの操作で行えます！

課題の取得と提出（学生側）

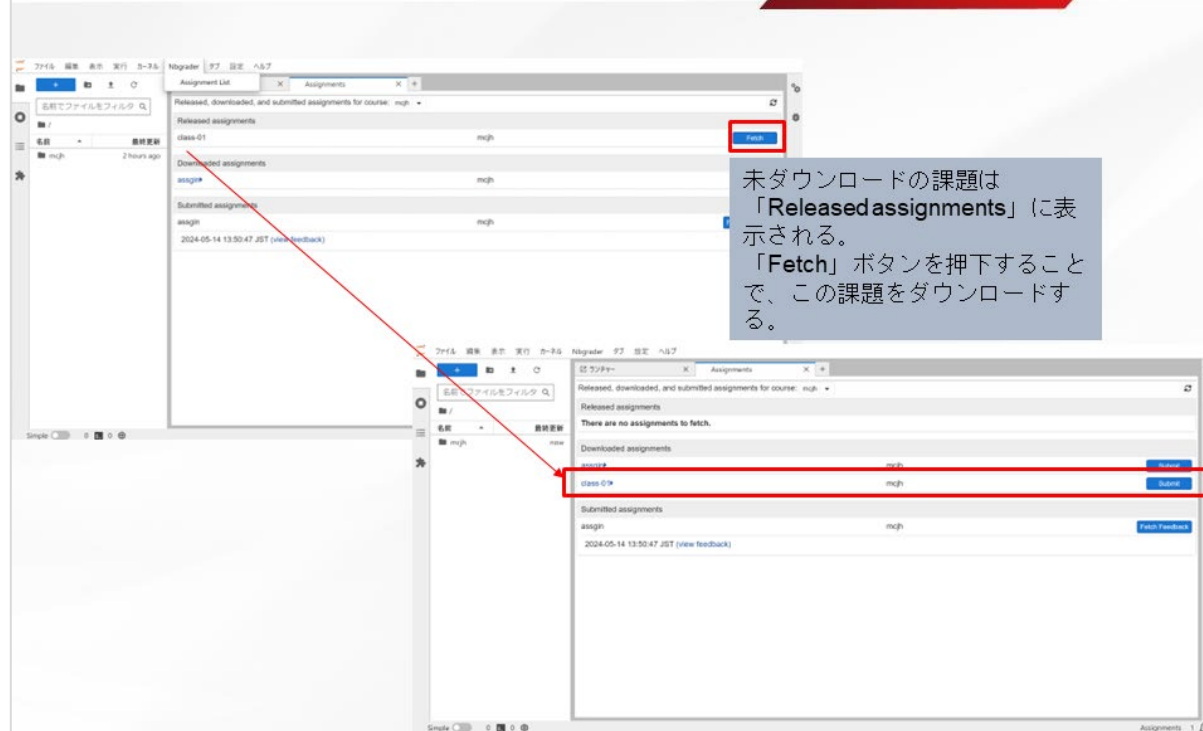
配布された課題一覧を確認する



The screenshot shows the Nbgrader web interface. The 'Assignment List' tab is active, displaying a table of assignments for the course 'mcjh'. The table is divided into three sections: 'Released assignments', 'Downloaded assignments', and 'Submitted assignments'. The 'Released assignments' section contains one entry, 'class-01', with a 'Fetch' button next to it. The 'Downloaded assignments' section contains one entry, 'assgin', with a 'Submit' button. The 'Submitted assignments' section contains one entry, 'assgin', with a 'Fetch Feedback' button. A red box highlights the 'Fetch' button for 'class-01'. A red arrow points from the 'Fetch' button to the 'Downloaded assignments' section.

© ASCADE, Inc. 25

配布された課題を取得する

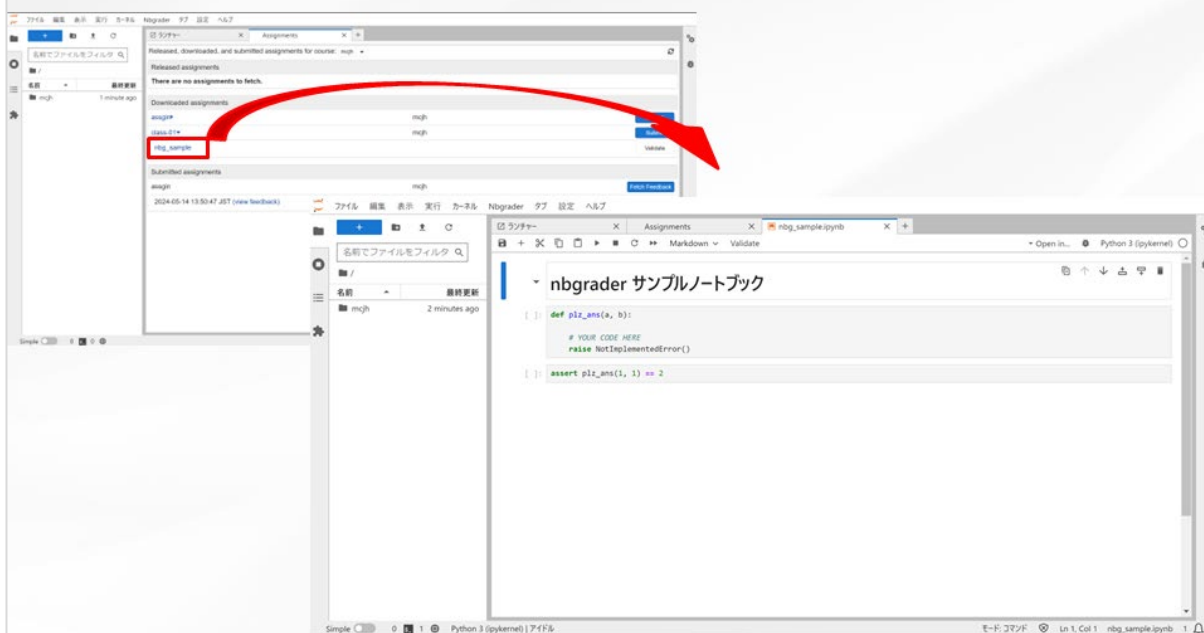


The screenshot shows the Nbgrader web interface. The 'Assignment List' tab is active, displaying a table of assignments for the course 'mcjh'. The table is divided into three sections: 'Released assignments', 'Downloaded assignments', and 'Submitted assignments'. The 'Released assignments' section contains one entry, 'class-01', with a 'Fetch' button next to it. The 'Downloaded assignments' section contains one entry, 'assgin', with a 'Submit' button. The 'Submitted assignments' section contains one entry, 'assgin', with a 'Fetch Feedback' button. A red box highlights the 'Fetch' button for 'class-01'. A red arrow points from the 'Fetch' button to the 'Downloaded assignments' section. A text box on the right explains the process: '未ダウンロードの課題は「Released assignments」に表示される。「Fetch」ボタンを押下することで、この課題をダウンロードする。' (Tasks not yet downloaded are displayed in 'Released assignments'. By clicking the 'Fetch' button, you can download this task.)

© ASCADE, Inc. 26

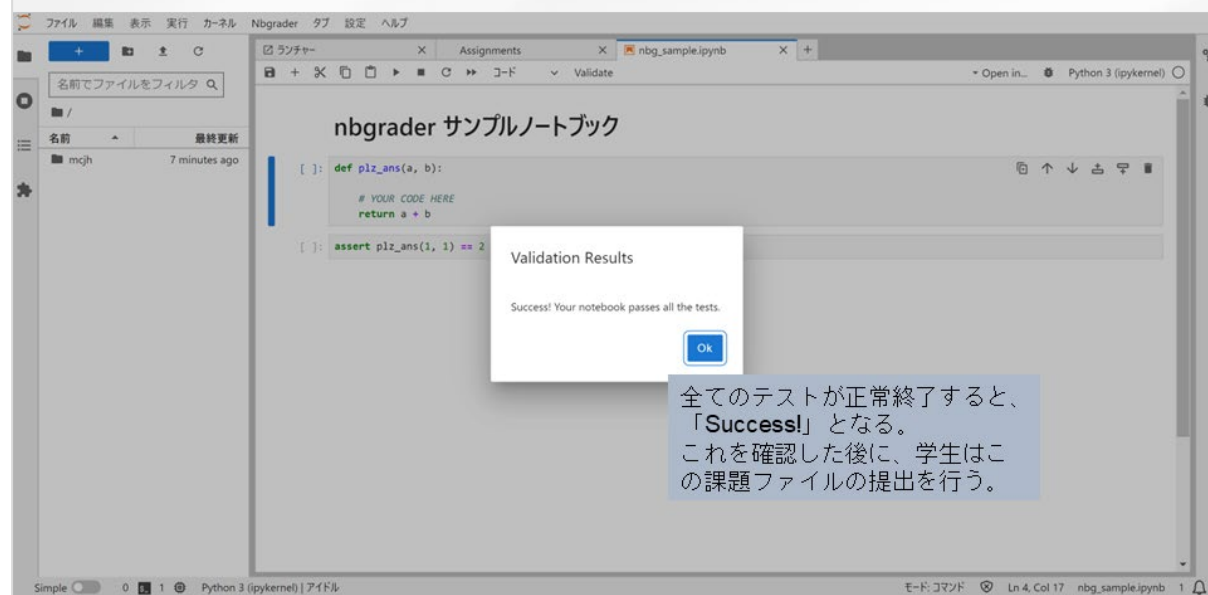
課題の取得と提出（学生側）

課題ファイル名が、ファイルへのリンクとなっている。



© ASCADE, Inc. 27

【検証成功ケース】課題を解き、nbgraderのValidateを用いて検証する

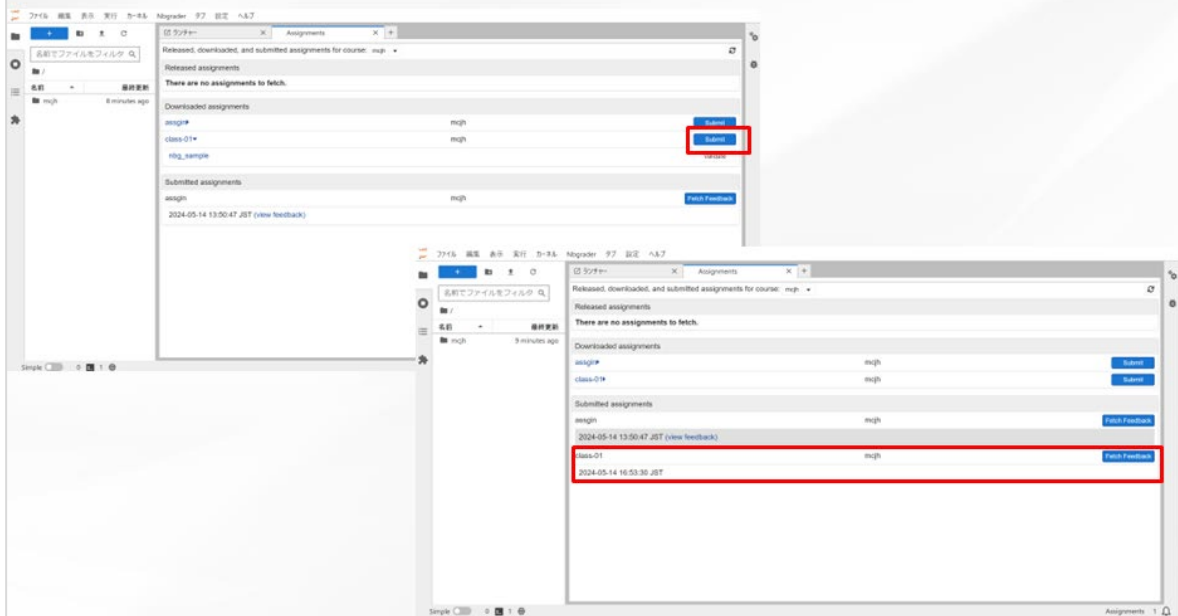


© ASCADE, Inc. 29

課題の取得と提出（学生側）

課題ファイルの提出

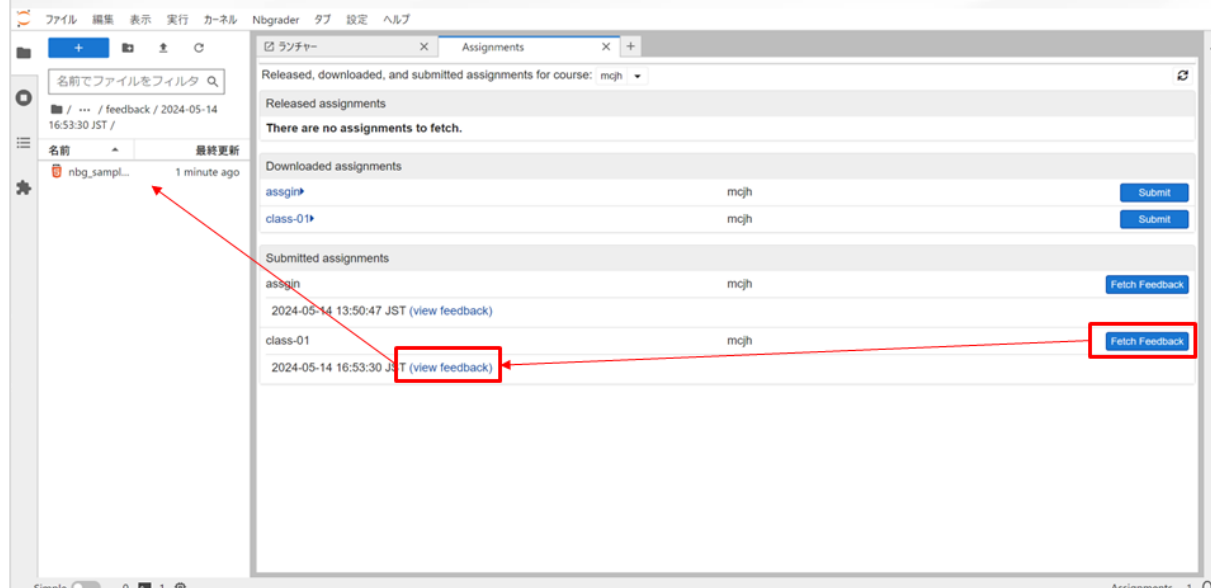
課題一覧画面にて、解き終わった課題ファイルを提出する。
提出が完了すると、教師は、この課題ファイルの回収が可能になる。



© ASCADE, Inc. 30

教師からのフィードバック閲覧

教師が採点・フィードバックの共有を行うと、学生側でそれらを確認可能になる。
「Fetch Feedback」ボタンの押下をすると、フィードバックの共有がされていれば、これを取得する。
フィードバックが閲覧可能である場合、(view feedback)というリンクが表示される。
このリンクを押下すると、画面左のディレクトリツリーにて、そのファイルが表示される。



© ASCADE, Inc. 31

まとめ

■MCJ-CloudHubの特徴

- 受講生が選択した科目ごとに統一した実行環境を提供
- nbgraderからの課題の配布・回収が可能
- Web 型であるため授業時間以外でも課題への取り組みが可能
- 利用者（教員・学生）は全ての機能をGUI からのみで操作可能
- OCSから他機関のオンプレミス環境やクラウド環境での利用も可能

■今後の予定

- MCJ-CloudHubの機能強化と試運用による品質向上
- ハンスオンセミナー等による他機関への利用促進活動の実施
- NII オープンフォーラム2025（6/16-18）の以下のトラックにてMCJ-CloudHubも取り上げる予定です。ご参加をご検討頂けると幸いです。
 - 6/18 12:30-14:00 大学の教育研究DXを支える基盤サービス

Thank You.



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

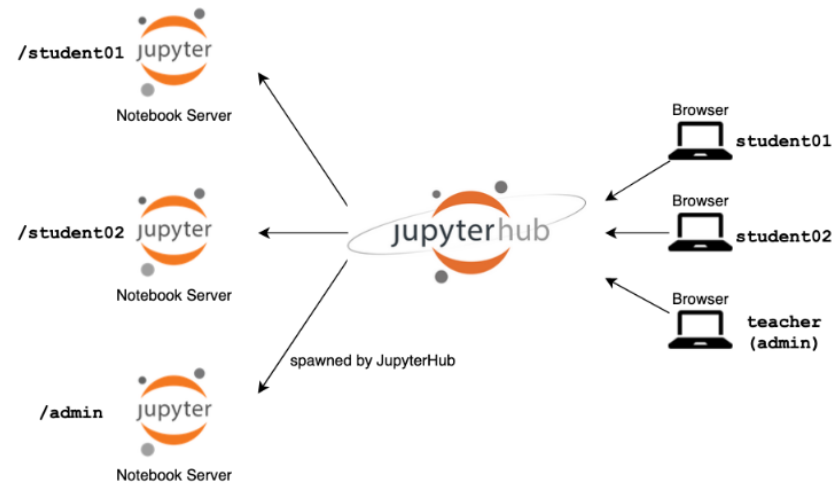
The Littlest JupyterHub による軽量Python実習環境の構築

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★

```
#####
### 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 = "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、ハンズオンでは指定していません。

```
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

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！

