

データ収集・蓄積・解析クラウド基盤 学認クラウドオンデマンド構築サー ビス（OCS）を支えるソフトウェア

国立情報学研究所 クラウド基盤研究開発センター

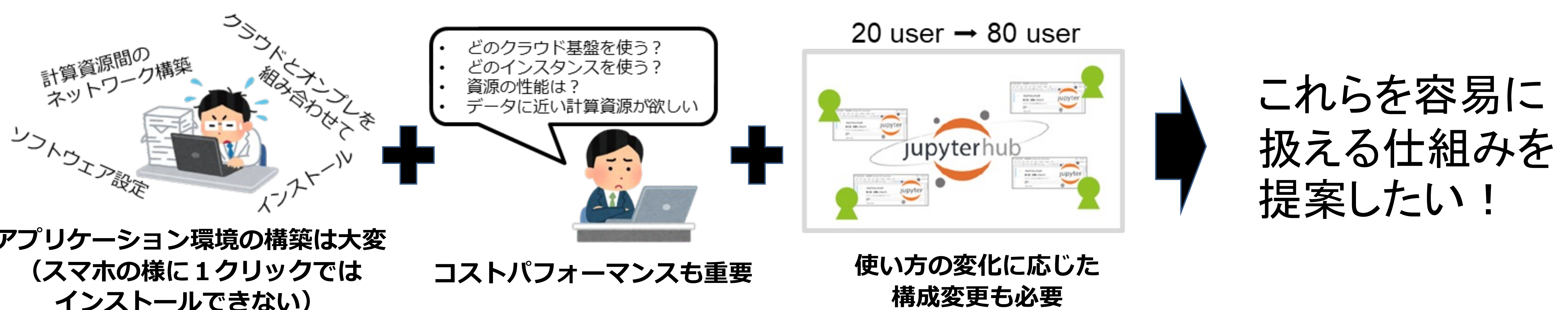
どんな研究？

- 複数のクラウドやサーバ（オンプレミス）をまとめて、同じように使えるようにします
- アプリの準備や操作を簡単にして、同じ設定で何度でも使える仕組みにします
- ユーザ数やデータの量が変わっても、それに合わせて環境を簡単に作り直せるようにします。

何がわかる？

- VCP (Virtual Cloud Provider)の仕組み
 - 複数の計算資源を同一方法で利用
 - ユーザからの操作方法
 - 主要なアプリケーションテンプレート
- 応用例
 - プログラミング演習システム（MCJ-CloudHub）
 - HPCクラスタ

背景・目的



研究内容（方法・結果・結論）

Virtual Cloud Provider (VCP)

方法:

- ◆ クラウドやサーバをつなげて、1つのコントローラからまとめて操作できるようにします
- ◆ 可読性の高いJupyter Notebookテンプレートを用いてVCコントローラを操作することで誰でも操作可能

結果:

- ◆ アプリケーション環境の構築は大変
 - ・ テンプレートを使えば、同じアプリの環境を何度でもすぐに作れます
- ◆ 適切な計算資源を使いたい
- ◆ アプリケーションの規模は変動する
 - ・ アプリを動かす場所をサーバやクラウドに自由に移動して、必要に応じて性能も変えられます

公開テンプレート:

- ◆ LMS, MCJ-CloudHub, OpenHPC, Open OnDemand, etc

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

JupyterNotebookの記述例

```
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IPアドレスは、ハブオンプレ環境の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_unitgroup(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_publickey(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 unit_group内の各VCノードのIPアドレスのリストを返します
ip_addresses = unit_group.find_ip_addresses(node_state='RUNNING')[0] # 今後は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_addresses)

# ssh 設定
!touch ~/.ssh/known_hosts
!ssh-keygen -f {ip_address} -N "" -i ~/.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

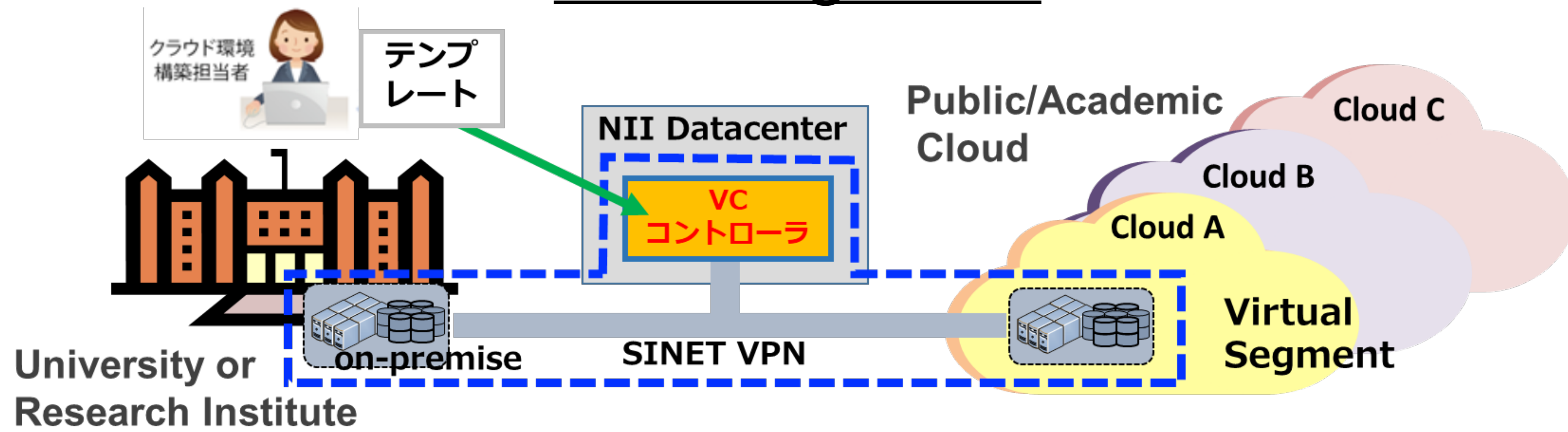
TLJH (The Littlest JupyterHub) 環境の構築
VCノード上に、本ハブオンプレ用に用意したThe Littlest JupyterHubのコンテナイメージを使用して環境を構築します。

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
- Pythonなどのプログラム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できる

データ収集・蓄積・解析クラウド基盤 学認クラウドオンデマンド構築サー ビス（OCS）を支えるソフトウェア

VCP configuration



利用方法:

- ◆ 国立情報学研究所 学認クラウドオンデマンド構築サービス (OCS) からサービスを提供

<https://cloud.gakunin.jp/ocs/>

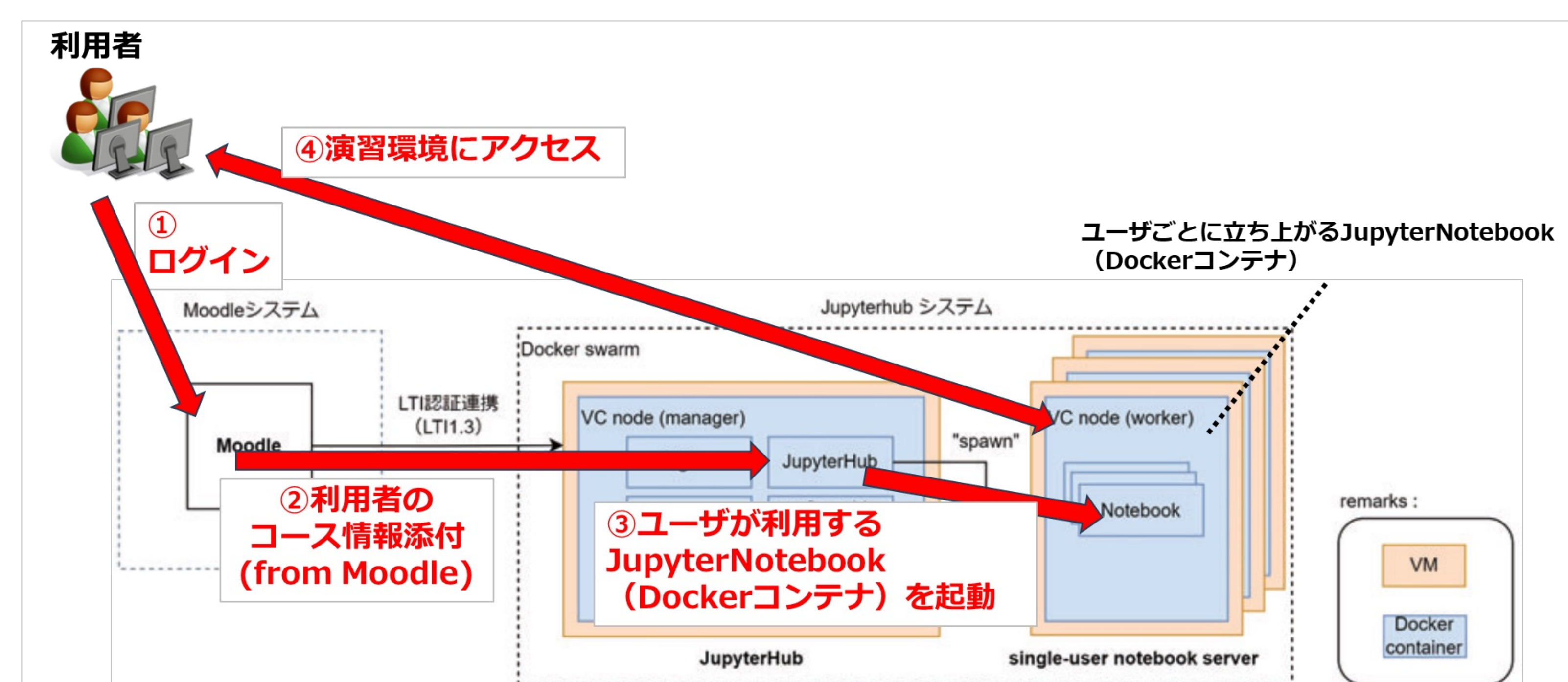
応用例（講義演習システム）

目的:

- ◆ 学校のサーバやクラウド上に、講義で使う演習用のシステムを簡単に作って使えるようにします
- ◆ 情報システム部門が手薄な大学においても運用可能とする

アプローチ:

- ◆ 山口大学の講義演習システムを汎用化した上で OCS から構築可能なアプリケーションテンプレートを開発(山口大学との共同研究)



講義演習システム (MCJ-CloudHub) の概要



進捗状況と今後の予定:

- ◆ 昨年度: 山口大学を含む複数の大学で運用開始。学習ログ分析 & 可視化機能などを実装。
- ◆ 今年度: 利用者からリクエストが多い機能の実装。利用者拡大に向けたハンズオンセミナー開催。

応用例（スパコン）

目的:

- ◆ Webブラウザから簡単にアプリを使える様にして、スーパーコンピュータ(スパコン)をもっと多くの人に使うことができるようにすること

アプローチ:

- ◆ 「OpenHPC」のテンプレートを使って、クラウド上にスパコン環境を作ります
- ◆ 「Open OnDemand」を使えば、Web画面からアプリ実行できます

今後の予定:

- ◆ 利用者拡大に向けたハンズオンセミナー開催。
- ◆ 想定利用者にフォーカスしたOpen OnDemandテンプレート拡充

