

学認クラウド オンデマンド構築サービスの 利用方法と最新情報

2020年6月9日

佐賀 一繁

国立情報学研究所
クラウド基盤研究開発センター／クラウド支援室

「学認クラウド」は、クラウドの導入・利活用を支援する国立情報学研究所のサービス群の総称です。

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



導入検討 調達

活用

学認クラウド 導入支援サービス

- クラウド導入の検討
- 仕様策定・調達
- チェックリスト回答の検証
- 個別相談の実施 など
- チェックリスト回答の提供
- 大学・研究機関向け商品の提案



- チェックリスト回答の参照
 - 個別相談の依頼
 - スタートアップガイドの参照
 - クラウド利活用セミナー参加
 - その他 (情報共有、ワークショップ参加など)
- ※本学は参加機関のみ利用可能

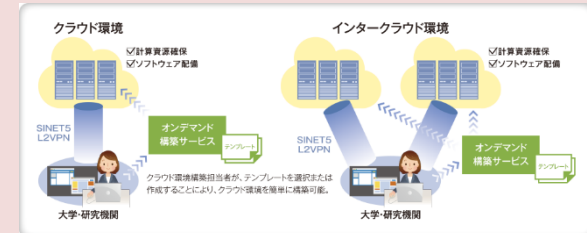
- 大学・研究機関にチェックリスト回答提供
 - 大学・研究機関のニーズ把握
 - その他 (情報共有、ワークショップへの参加など)
- ※すべて参加事業者のみ利用可能

学認クラウド ゲートウェイサービス



クラウドサービスにワンストップでアクセスするためのポータル機能

学認クラウド オンデマンド構築サービス



研究教育のためのクラウド環境構築を技術的に支援

- 初期導入支援
- オンデマンド構築機能
- 情報共有

選択の基準や、導入・活用に関わる情報を整備し、お伝えするサービス

サービス概要

■ クラウドの導入方法について

- 機関とクラウドの計算資源を安全に連携する方法がわからない
- 設定方法や用語がプロバイダ毎に異なるので分かりにくい

→ 1. 初期導入支援

■ クラウド上のアプリ環境構築について

- オンデマンドにアプリ環境を構築・再構築したい
- 構築・再構築の運用を簡単にしたい
- 構成変更に対し柔軟に対応できるようにしたい

→ 2. オンデマンド構築機能

■ クラウド上のアプリ環境構築のノウハウについて

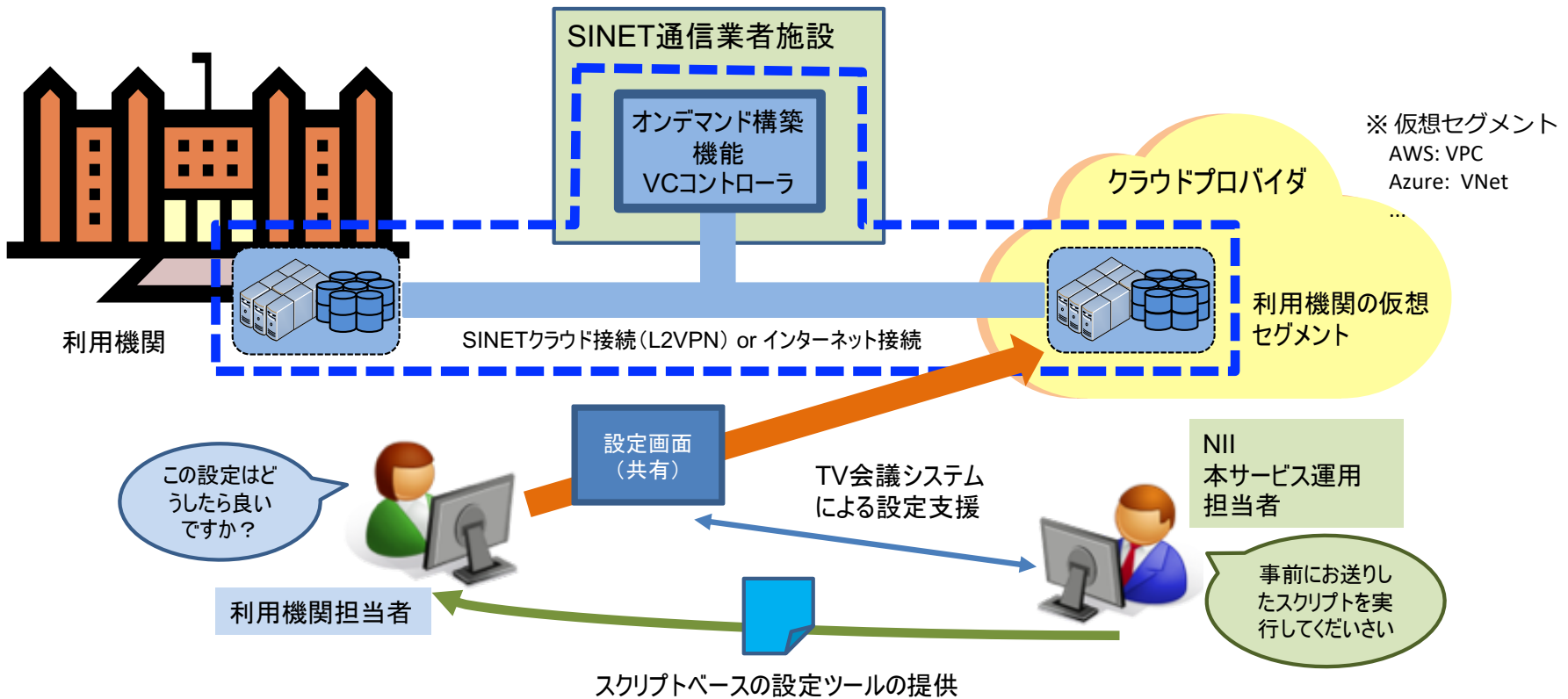
- 教育・研究用のアプリ環境構築のノウハウが広く流通していない

→ 3. 情報共有

オンデマンド構築サービスは、こんなお悩みに応えます

1. 初期導入支援

- 本サービス導入のための接続方法のご相談
- 本サービス導入時のクラウドプロバイダ設定支援 など



2. オンデマンド構築機能

- テンプレート(Jupyter Notebook形式)ベースのアプリ環境構築・再構築機能
- 一つのクラウド上のシンプルなアプリ環境から、複数のクラウドにまたがる複雑なアプリ環境まで構築可能
- クラウドプロバイダ毎に異なる計算資源操作インターフェースを抽象化



目的のテンプレートを用意すれば構築・再構築が容易に

1. ログイン
2. テンプレート選択
3. 構築実行

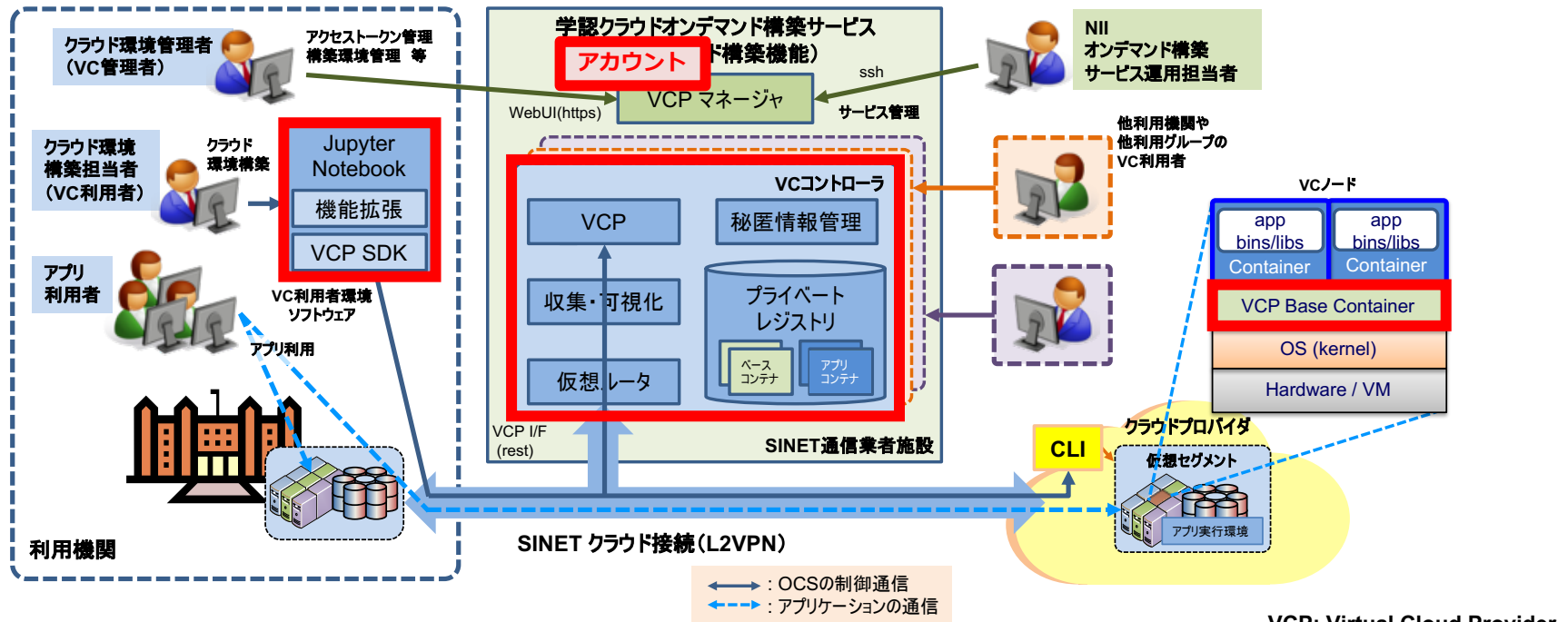
3. 情報共有

- 利用機関・利用グループ毎に専用Wikiページを提供
 - ドキュメント類
 - 運用情報
 - 質疑応答履歴 ...

- リポジトリ(一般公開)
 - チュートリアル(ハンズオン教材)
 - <https://github.com/nii-gakunin-cloud/handson/>
 - NIIやアプリコミュニティが作成した著名アプリの構築テンプレートとコンテナイメージ
 - OpenHPC、CoursewareHub、Moodle ...
 - <https://github.com/nii-gakunin-cloud/ocs-templates/>
 - 利用者間情報交換の場としても利用予定

利用方法

オンデマンド構築機能の構成と提供物



■ 提供物 (利用グループ単位)

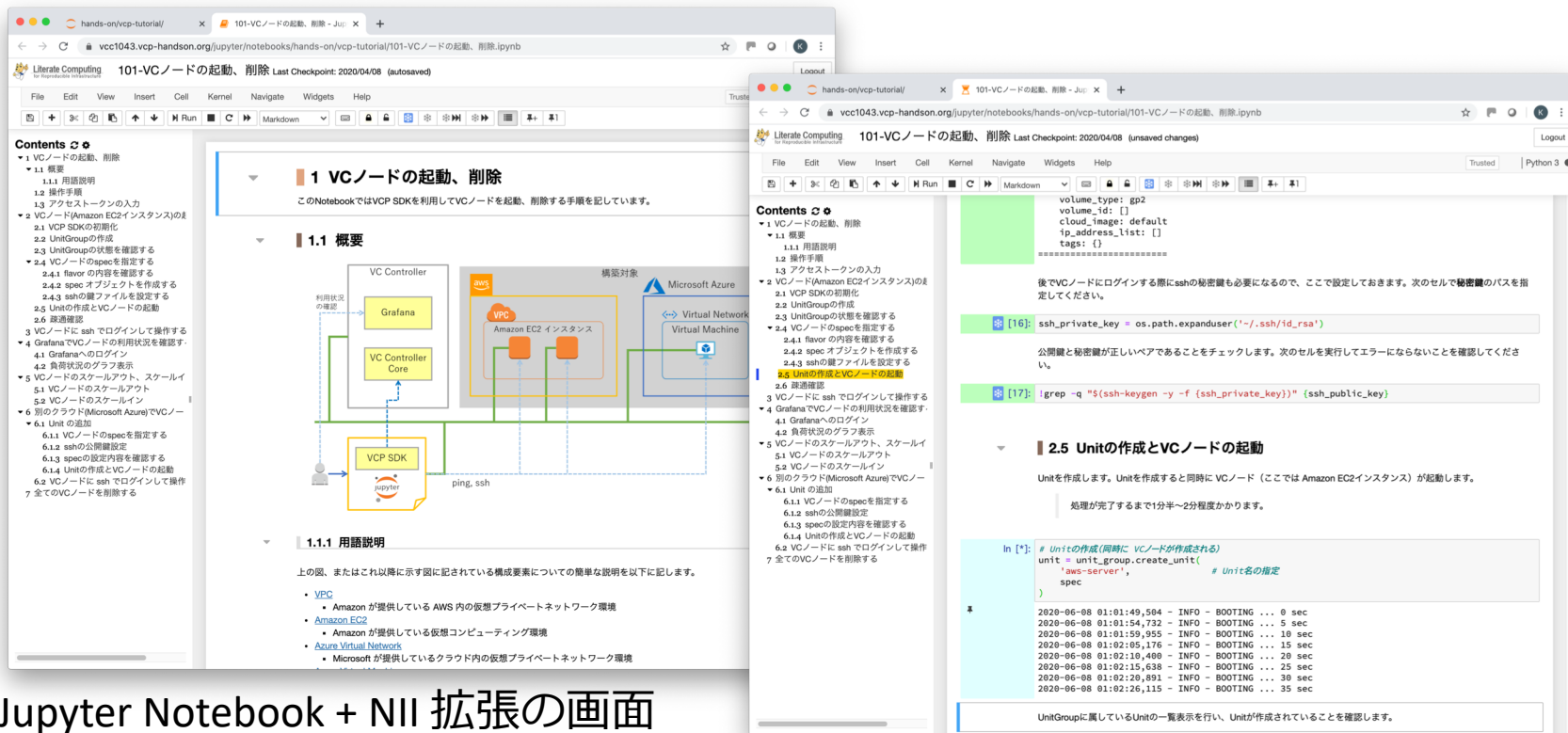
- クライアントソフトウェア環境 (利用グループ環境にインストール)
 - VCP SDK: 資源管理インターフェース。プロバイダの管理インターフェースを抽象化
 - Jupyter Notebook NII 拡張
- VCコントローラ (SINET通信業者施設の NII サーバにて動作)
 - クラウド上の資源の生成・削除・管理コントローラ
- VCノードのVCP Base Container
 - クラウド上の計算資源を管理するための、ベースコンテナ
- VCPマネージャのアカウント

VCP: Virtual Cloud Provider

テンプレート

■ アプリ環境構築ワークフローが記述された Jupyter Notebook 形式のファイル

■ 構築ドキュメント(文字、画像、動画...)、構築スクリプト、実行結果(記録)を1つのファイルに集約可能



The screenshot displays a Jupyter Notebook titled "101-VCノードの起動、削除" (Starting and Deleting 101-VC Nodes). The notebook is structured as follows:

- Contents:**
 - 1 VCノードの起動、削除
 - 1.1 概要
 - 1.2 操作手順
 - 1.3 アクセストークンの入力
 - 2 VCノード(Amazon EC2インスタンス)の構築
 - 2.1 VCP SDKの初期化
 - 2.2 UnitGroupの作成
 - 2.3 UnitGroupの状態を確認する
 - 2.4 VCノードのspecを指定する
 - 2.4.1 flavorの内容を確認する
 - 2.4.2 specオブジェクトを作成する
 - 2.4.3 sshの設定ファイルを設定する
 - 2.5 Unitの作成とVCノードの起動
 - 2.6 疎通確認
 - 2.7 VCノードに ssh でログインして操作する
 - 2.8 GrafanaでVCノードの利用状況を確認する
 - 2.8.1 Grafanaへのログイン
 - 2.8.2 負荷状況のグラフ表示
 - 2.9 VCノードのスケールアウト、スケールイン
 - 2.9.1 VCノードのスケールアウト
 - 2.9.2 VCノードのスケールイン
 - 2.10 別のクラウド(Microsoft Azure)でVCノードの追加
 - 2.10.1 VCノードのspecを指定する
 - 2.10.2 sshの公開鍵設定
 - 2.10.3 specの設定内容を確認する
 - 2.10.4 Unitの作成とVCノードの起動
 - 2.11 VCノードに ssh でログインして操作する
 - 2.12 全てのVCノードを削除する

The notebook includes a diagram illustrating the architecture, showing components like VCP Controller, Grafana, VCP SDK, Amazon EC2 instances, and Virtual Machines connected to Microsoft Azure. It also contains code cells for setting up the environment, such as:

```
unit = unit_group.create_unit(
    'aws-server',
    spec
)
```

The output shows the successful creation and booting of units, with a log of timestamps and status messages.

Jupyter Notebook + NII 拡張の画面

オンデマンド構築機能によるアプリ環境構築

- 準備：テンプレートとアプリケーションコンテナの作成（例）
 - アプリ環境構築ワークフローを記述したテンプレートの作成
 - 各種情報とVCPの初期化
 - VCノード(クラウドインスタンス)のspec設定(プロバイダ、フレーバ、数、ssh鍵ファイルなど)
 - VCノードの確保・起動
 - アプリ実行環境のコンテナ操作
 - 構築したアプリ環境の停止、再起動、削除テンプレートの作成
 - 再起動とコストを意識した停止操作
 - 構築した環境の削除操作
 - アプリケーション実行環境の構築とコンテナ化
 - アプリケーション実行環境構築とコンテナ化(Dockerコンテナイメージ)
 - 作成したコンテナイメージをプライベートレジストリに格納
(レジストリに格納しないで、構築テンプレート実行毎に構築することも可能)
 - 動作確認

オンデマンド構築機能によるアプリ環境構築(続)

■ アプリ環境運用

■ アプリ環境の構築

- アプリ環境が必要なときにJupyterNotebookにログイン
- 上記作成・確認済みテンプレートを実行して環境を構築・再構築、もしくは再起動テンプレートを実行して環境を再起動

■ アプリ環境の提供

- エンドユーザがアプリを実行

■ アプリ環境の停止もしくは削除

- アプリ環境の不使用时や不要時にJupyterNotebookにログイン
- 当該アプリ環境の停止や削除を行うテンプレートの実行

※ Jupyter Notebook ではなく、VCP SDK を利用する Python スクリプトによる操作も可能

例: OpenPose 環境構築

GPUインスタンスを使った OpenPose環境の構築例

まとめ実行を展開→

↓各ステップをまとめ実行表示

This block shows a vertical scroll of a document titled "GPUインスタンスの利用" (Using GPU Instances). The document is divided into five main sections: 1. OpenPose環境構築についての説明 (Description of OpenPose environment setup), 2. VCコントローラ アクセストークンの入手 (Obtaining VC controller access tokens), 3. VCノード (GPUインスタンス) の起動 (Starting VC nodes), 4. OpenPose環境の構築 (Building OpenPose environment), and 5. OpenPoseの利用 (Using OpenPose). Blue arrows point from the "まとめ実行を展開" (Expand summary execution) text to the "3.2 VCノードのspecを指定する" (Specify VC node spec) step in section 3. Red arrows point from the "↓各ステップをまとめ実行表示" (Display summary execution of each step) text to the "5.1 サンプルの画像を処理する" (Process sample image) step in section 5. On the left side, there are three vertical labels with arrows: "VCノード (計算資源) 確保・起動" (VC node (computing resource) acquisition and start) with an upward arrow, "アプリ環境構築" (App environment construction) with a bidirectional arrow, and "アプリ実行" (App execution) with a downward arrow.

This screenshot shows the "3.2 VCノードのspecを指定する" (Specify VC node spec) step. It includes sub-steps: 3.2.1 specの取得 (Obtain spec), 3.2.2 sshの鍵ファイルを設定する (Set ssh key file), 3.2.3 specの設定内容の確認 (Check spec settings), and 3.2.4 Unitの作成とVCノードの起動 (Create unit and start VC node). The code for 3.2.1 is:

```
spec = gpu_vcp_spec(
    'aws', # プロバイダ
    'gpu', # flavor
)
```

This screenshot shows the "4 OpenPose環境の構築" (Build OpenPose environment) step. It includes sub-steps: 4.1 準備 (Preparation), 4.2 OpenPoseコンテナのビルド (取得) (Build (Obtain) OpenPose container). The code for 4.2 is:

```
OpenPoseをアプリケーションコンテナとして実行できるように、
コマンドを実行します。
vcpのプライベートレジストリに既にビルドしたOpenPoseコンテナ
とイメージの取得が行われます。イメージの取得には1~3分程度か
かります。
```

This screenshot shows the "5 OpenPoseの利用" (Use OpenPose) step. It includes sub-steps: 5.1 サンプルの画像を処理する (Process sample image), 5.2 サンプルの動画を処理する (Process sample video), 5.3 GrafanaでVCノードの利用状況を確認する (Check VC node usage with Grafana), and 5.4 インターネット上の画像を処理する (Process image on internet). The code for 5.1 is:

```
! report tempfile
! log_file = tempfile.mkstemp()
! ssh root@[ip_address] "cd openpose && docker-compo
  o openpose --img_dir $log_file --write_images /root/result --
  write_images_format jpg"
```

This screenshot shows the "3.3 VCノードの起動" (Start VC node) step. It includes sub-steps: 3.3.1 Unitの作成とVCノードの起動 (Create unit and start VC node), 3.3.2 疎通確認 (Connectivity check), and 3.3.3 specの設定内容の確認 (Check spec settings). The code for 3.3.1 is:

```
unit_group = vcp.create_unit_group(
    'hands201'
)
```

This screenshot shows the "4.2 OpenPoseコンテナのビルド (取得)" (Build (Obtain) OpenPose container) step. It includes sub-steps: 4.2.1 OpenPoseのインストール (Obtain OpenPose), 4.2.2 OpenPoseコンテナのビルド (取得) (Build (Obtain) OpenPose container). The code for 4.2.1 is:

```
! ssh root@[ip_address] "cd openpose && docker-compo
  o openpose --img_dir $log_file --write_images /root/result --
  write_images_format jpg"
```

This screenshot shows the "5.1 サンプルの画像を処理する" (Process sample image) step. It includes sub-steps: 5.1.1 OpenPoseのインストール (Obtain OpenPose), 5.1.2 OpenPoseコンテナのビルド (取得) (Build (Obtain) OpenPose container), 5.1.3 OpenPoseの起動 (Start OpenPose), 5.1.4 OpenPoseの出力を確認する (Check OpenPose output). The code for 5.1.1 is:

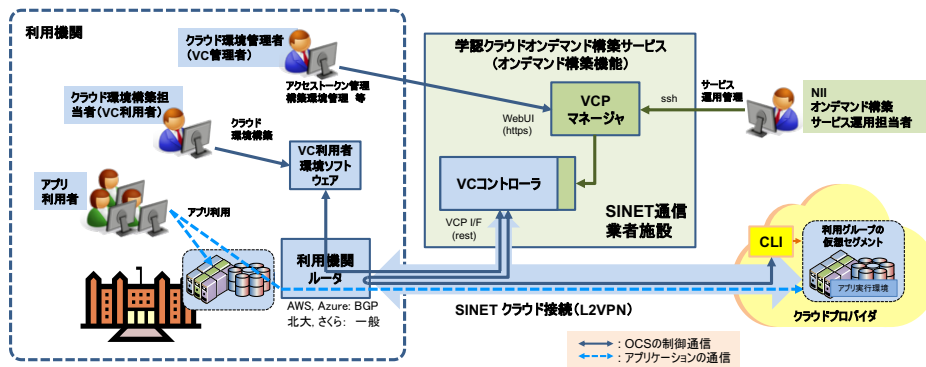
```
! report tempfile
! log_file = tempfile.mkstemp()
! ssh root@[ip_address] "cd openpose && docker-compo
  o openpose --img_dir $log_file --write_images /root/result --
  write_images_format jpg"
```

サービス利用環境

接続形態

■ SINET接続

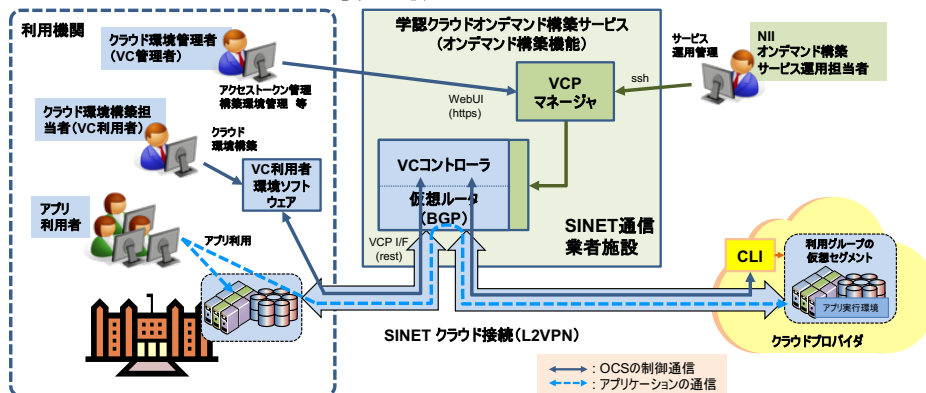
- 特徴：高速、低遅延、セキュア（インターネット接続比）
- 利用機関ルータ接続



■ 利用機関のルータを使用

- Azure, さくら, 北大
 - L3
- AWS, Oracle
 - L3+BGP

■ 仮想ルータ接続



■ BGPは仮想ルータを使用

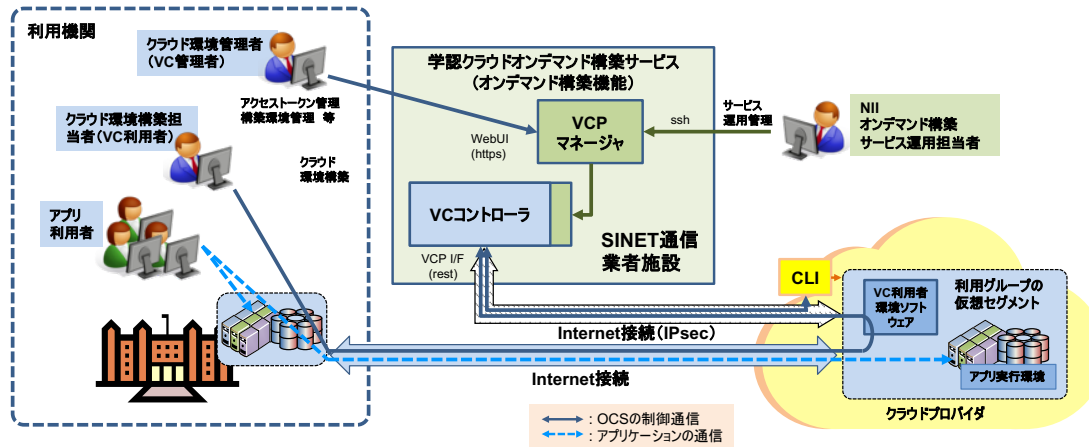
- AWS, Oracle
- BGPルータを所有しない機関向け形態

(2020年5月時点)

接続形態(続)

■ インターネット接続

- 特徴：本サービスを容易な利用開始可能(SINET接続比)
- 接続：
 - 本サービス・プロバイダ間：IPsec 接続
 - 利用機関・プロバイダ間：機関のポリシーに沿った接続



利用申請

サービス利用開始までの流れと時間

■ 利用開始までの流れ

- ① クラウドプロバイダとの契約
- ② 国立情報学研究所クラウド利活用サービスの利用申請
 - 利用機関につき1申請
- ③ 利用機関／利用グループ本サービス担当者と接続打合せ
- ④ SINET接続の場合：クラウドプロバイダへSINET接続申込み・VLAN-ID調整（プロバイダにより異なる）
- ⑤ 学認クラウドオンデマンド構築サービス接続申込み
 - 独立したVCコントローラが必要な利用グループ単位で申込み。複数申込み／利用機関可能
- ⑥ 接続設定
- ⑦ 利用開始

青：本サービスの申請・申込み
赤：初期導入支援による共同作業

■ 要する時間（契約、機関内調整時間、待ち時間等含まず）

- SINET接続： 新規：5週間程度、追加：3週間程度
- インターネット接続： 2週間程度

サービスの利用について

- 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
 - 学術クラウドプロバイダ
 - 北海道大学インタークラウドシステム／サーバサービス
 - オンプレミスシステム
 - VMware

- 対象利用者
 - 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

- 利用要件
 - 学術認証フェデレーション(学認)への参加が望ましい
 - NII OpenIdP でも利用可能

- 利用料金
 - 本サービスは無償です
 - クラウドプロバイダや接続回線など有償サービスは利用者負担です

試用環境の利用・ご質問について

- まずは試用環境で
 - 正式な申込みには、少なからぬ労力が必要です。まずは、簡単な申込みで利用できる、**試用環境**で有用性をご確認ください(無償)
 - ハンズオンセミナーはCovid-19の影響で開催を停止しています。再開は、メーリングリストや下記Webページでお知らせします

- 詳細は本サービスのWebページをご覧ください
 - <https://cloud.gakunin.jp/ocs/>

- 各種問い合わせは以下まで
 - NIIクラウド支援室 cld-office-support@nii.ac.jp

最新情報

昨年のオープンフォーラムからの差分

■ 2019/10

- さくらのクラウドのサポート
- 計算資源情報の収集・可視化機能のGPU対応 など

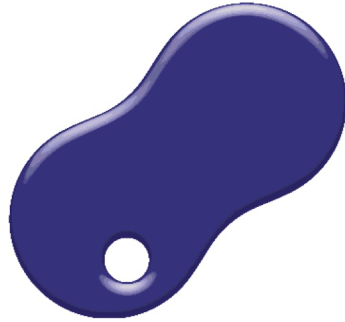
■ 2020/4

- Oracle Cloud のサポート
- 計算資源の停止・再起動機能追加
- アプリケーションテンプレートの正式提供
 - OpenHPC、CoursewareHub、Moodle
 - <https://github.com/nii-gakunin-cloud/ocs-templates/>

今後の予定

- 公開アプリケーションテンプレートの拡充
- オンデマンド構築機能の機能・利便性向上
- オンラインハンズオン環境の整備
- 利用者間の情報交換の場の整備 など

- ご意見・ご希望をお寄せください
 - NIIクラウド支援室 cld-office-support@nii.ac.jp
 - 例えば、
 - こんな機能があったら便利
 - こんなアプリケーションテンプレートが欲しい
 - ○○プロバイダをサポートしてほしい など



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