CoursewareHubの概要

2023年1月

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

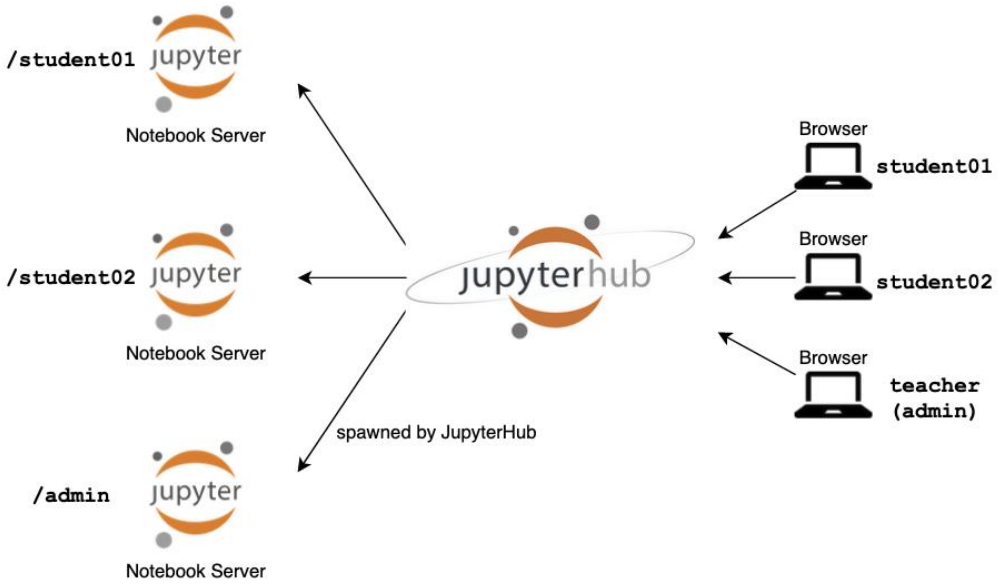
CoursewareHubの概要

CoursewareHub ＝
JupyterHub ＋
講義のためのモジュール群

►JupyterHub …
・Jupyter Notebook環境を複数のユーザーで使用する仕組みを提供
・認証機能
・ユーザーごとに隔離された実行環境

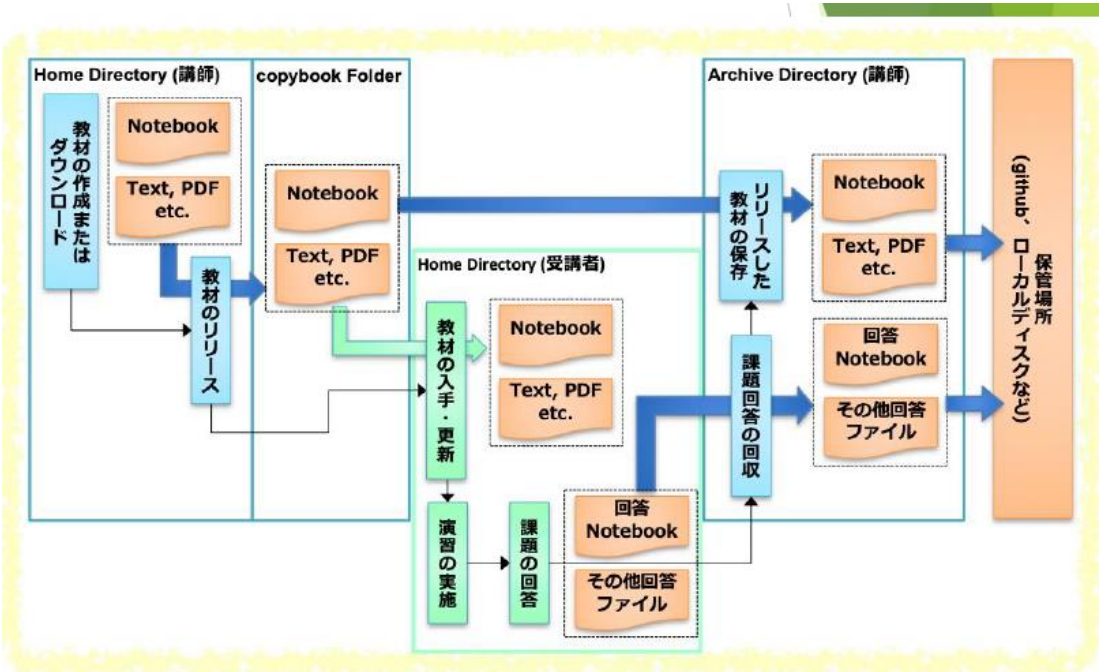
JupyterHubによる基本機能実現

- **1. Jupyter Notebook** を用いた講義演習を行うための環境です
  ⇒説明（画像や音声）、処理、処理結果(グラフ、画像や音声)を残せる
  ⇒すぐ、簡単に実行
- **2.** 受講生は Web ブラウザ経由で受講することができます
  ⇒統一された環境
- **3.** 講師は **Notebook** 教材を、**Web** ブラウザ経由で作成することができます



講義のためのCoursewareHub機能

4. 講師は**Notebook**教材で利用するアプリを**CoursewareHub**上に登録することができます
5.受講生を登録、教材を配布**(**講義途中で更新**)**、演習の回答を回収するなどの演習実施のためのワークフローをサポートしています



講義のためのCoursewareHub機能

6.受講生の作業履歴、演習実施履歴を収集することができます

・学生の理解度に応じた助言が重要

・疑問や分からないことを抱えたままに、次のステップに進み、ますます分からなくなりいよう進捗確認

・Jupyter Notebookを利用すれば、セルの実行回数やエラー数、利用者ごとの評価履歴などの情報がグラフで可視化できる

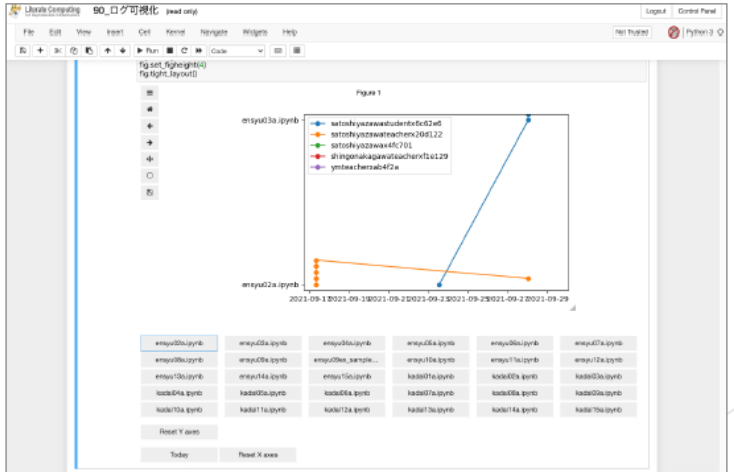
・これを元に教員やTA(ティーチング・アシスタント)が助言

・教材自体の改善に役立てる

（室蘭工業大学情報教育センター長 教授 桑田 喜隆氏）

作業履歴・演習実施履歴の収集

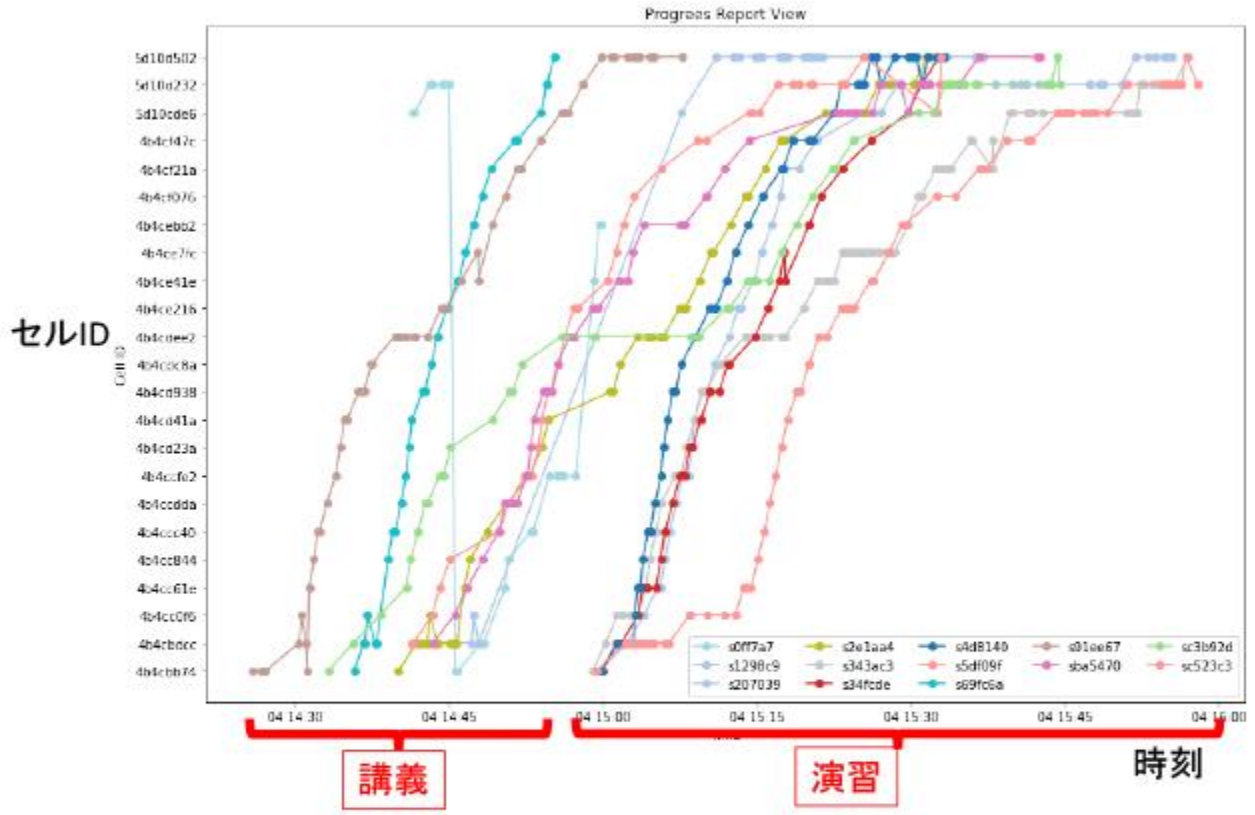
- 例: 進捗の可視化
- 実は…基礎編ではスタッフが講師として皆さんの進捗を集約・確認していました



講義のためのCoursewareHub機能

室蘭工業大学様の事例

・事前検証の際は数名～十数名
　→本番の授業では600名規模
・Pythonを学ぶための教材を
Jupyter Notebookで作成・配布
・課題を解くのに掛かった時間や進捗状況などのデータを収集・確認
（右グラフ）



収集データ例(進捗グラフ)

講義のためのCoursewareHub機能

7.安定稼働のためのリソース制御

受講生が演習環境に高負荷をかけるような行為を防止し、安定稼働させる
ことができます

ex.) Fork 爆弾
►学生一人あたりの環境で利用可能なメモリサイズ・ CPU サイズを制限
►一人の学生のミス・悪ふざけで環境全体が機能不全に陥らないように
►学生の環境の生存期間の制御
►しばらく使用されていない Jupyter Notebook サーバの停止
►大規模・複数クラス (200 名 の講義におけるリソース有効利用が可能

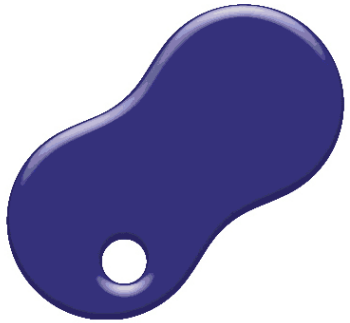
講義のためのCoursewareHub機能

7.既存システムとの連携性

受講生の認証に「学認」(Shibboleth) および LTI（LMSと相互**連携**を実現するために策定された、技術標準規格）を利用することができます

►学認 (Single Sign On) を使った認証
►学認クラウドゲートウェイサービスを利用した認可
►LMS(Moodle, Sakai, …) との連携
►LMS の特定コースの受講者のみが利用可能な JupyterHub 環境が実現可能

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

参考資料

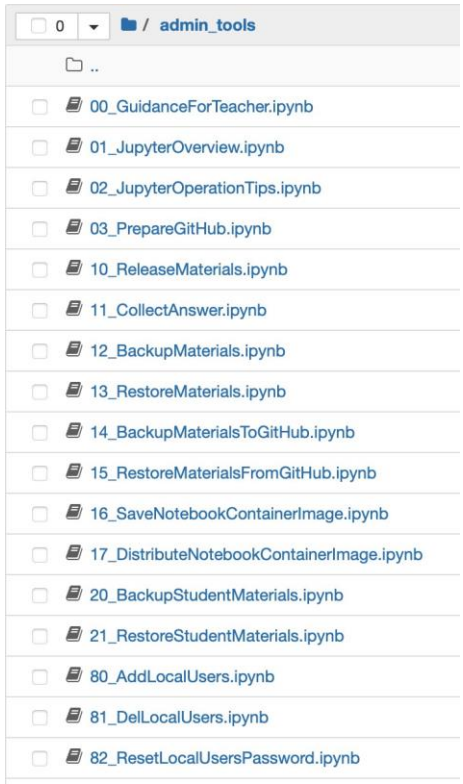
国立情報学研究所 学術基盤オープンフォーラム2021
2021年7月7日

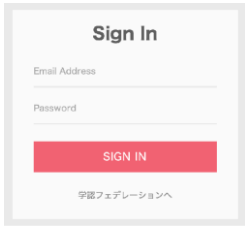
CoursewareHubを利用した授業: Python入門

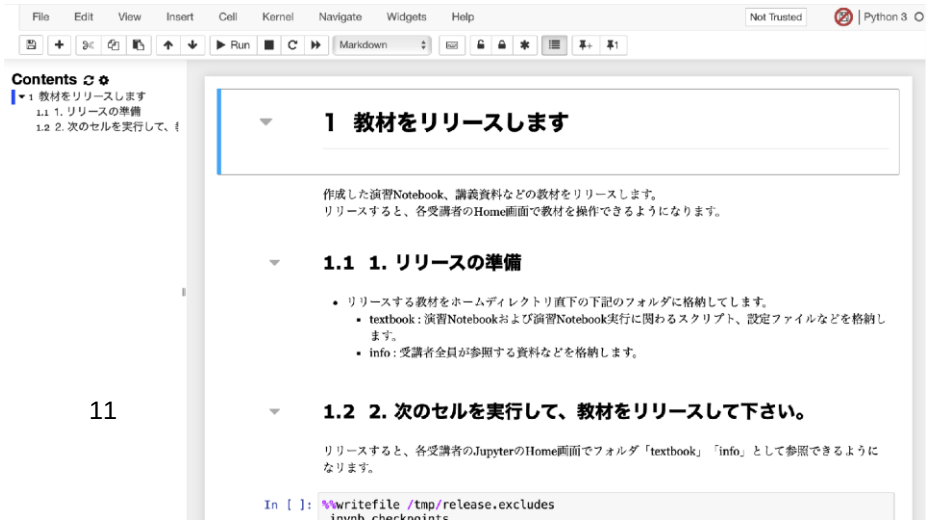
群馬大学・数理データ科学教育研究センター
井上　仁

CoursewareHubとは

- JupyterHubをベースに、NIIで機能拡張
  - 認証連携 (学認)
  - 教材配布
  - 課題回収
  - 操作履歴収集
  - … …



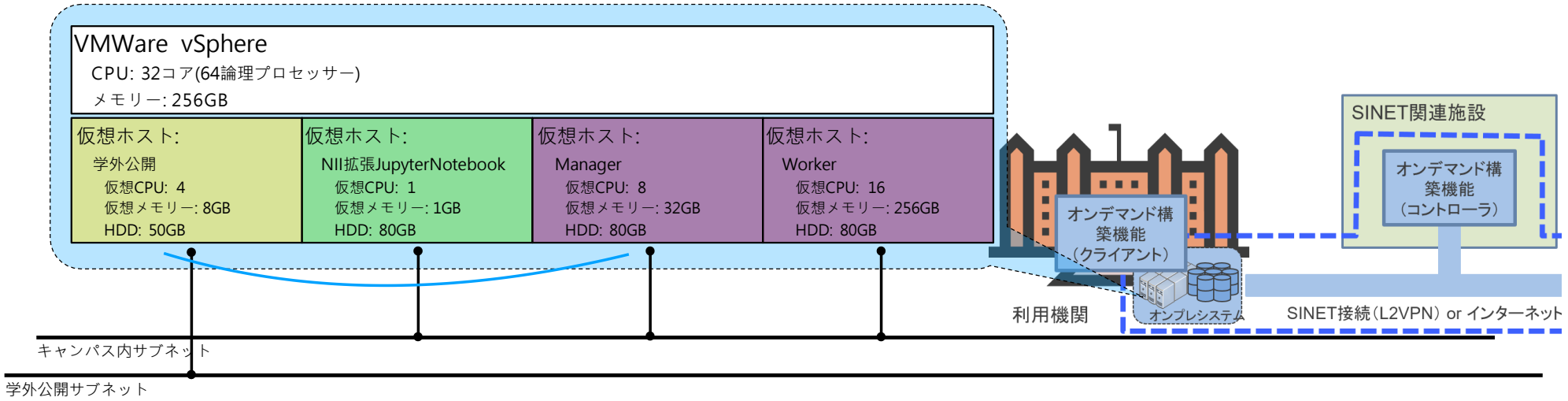




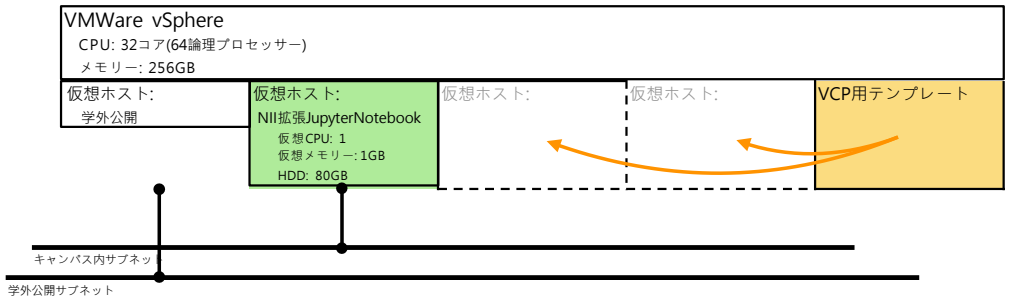
CoursewareHub利用の経緯

- 2020年度後学期から教養教育選択科目「Python入門」を新規開講
- 授業で使用するプラットフォームを検討 (2019年度)
  - 授業時間内
    - PC設置教室でJupyter Notebookの利用
    - Google Colaboratoryの利用
  - 授業時間外
    - 個人PCにJupyter Notebookをインストール
    - Google Colaboratoryの利用
- 群馬大学・横山教授 (現在、国立情報学研究所)からCoursewareHub利用の提案

ハードウェア構成


VMWare vSphere
CPU: 32コア(64論理プロセッサー)
メモリー: 256GB
仮想ホスト:
学外公開
仮想CPU: 4
仮想メモリー: 8GB
HDD: 50GB
仮想ホスト:
NII拡張JupyterNotebook
仮想CPU: 1
仮想メモリー: 1GB
HDD: 80GB
仮想ホスト:
Manager
仮想CPU: 8
仮想メモリー: 32GB
HDD: 80GB
仮想ホスト:
Worker
仮想CPU: 16
仮想メモリー: 256GB
HDD: 80GB
オンデマンド構築機能（クライアント）
利用機関
オンプレシステム
SINET関連施設
オンデマンド構築機能（コントローラ）
SINET接続（L2VPN） or インターネット
キャンパス内サブネット
学外公開サブネット

環境構築

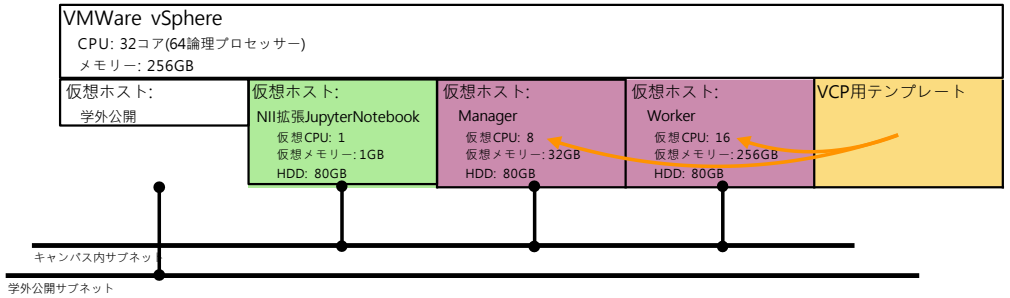


- NII拡張JupyterNotebook環境の作成
  - VCノードの作成
  - CousewareHub環境の構築

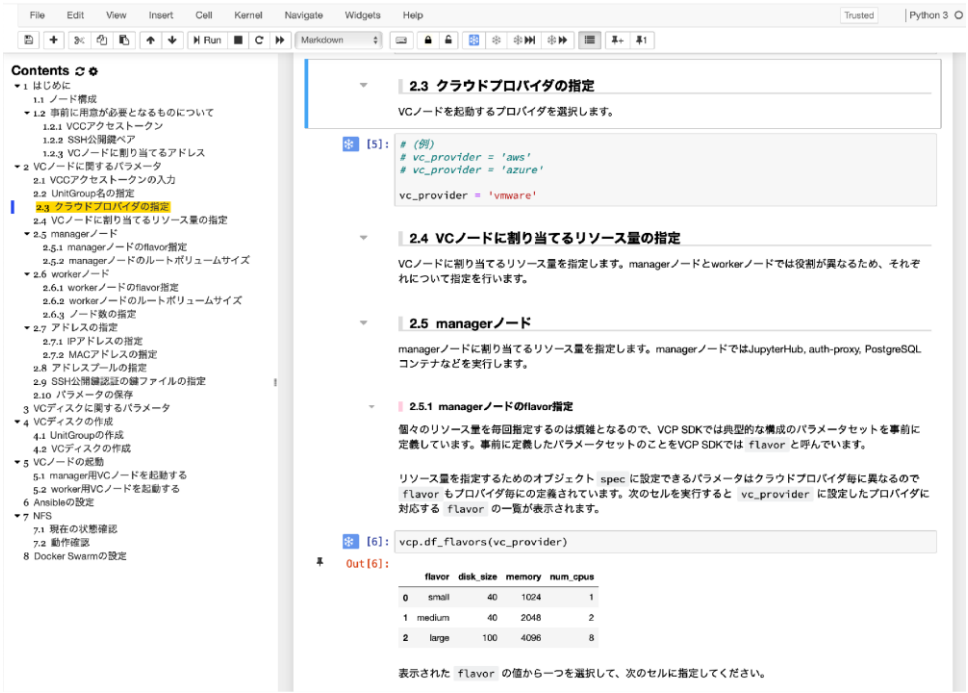
- VCP用テンプレートの作成

1. 仮想ホストの作成
   - ハードウェアのカスタマイズ
     - SCSI Controller
     - HDD (Disk Provisioning)
   - OS
     - Ubuntu 16.04
     - ssh
     - Docker
2. 仮想ホストのテンプレートへの変換

VCノードの作成

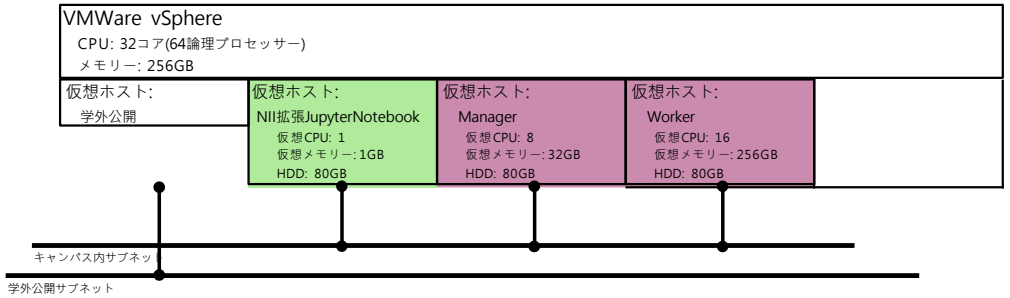

VMWare vSphere
CPU: 32コア(64論理プロセッサー)
メモリー: 256GB
仮想ホスト:
学外公開
仮想ホスト:
NII拡張JupyterNotebook
仮想CPU: 1
仮想メモリー: 1GB
HDD: 80GB
仮想ホスト:
Manager
仮想CPU: 8
仮想メモリー: 32GB
HDD: 80GB
仮想ホスト:
Worker
仮想CPU: 16
仮想メモリー: 256GB
HDD: 80GB
VCP用テンプレート
キャンパス内サブネット
学外公開サブネット

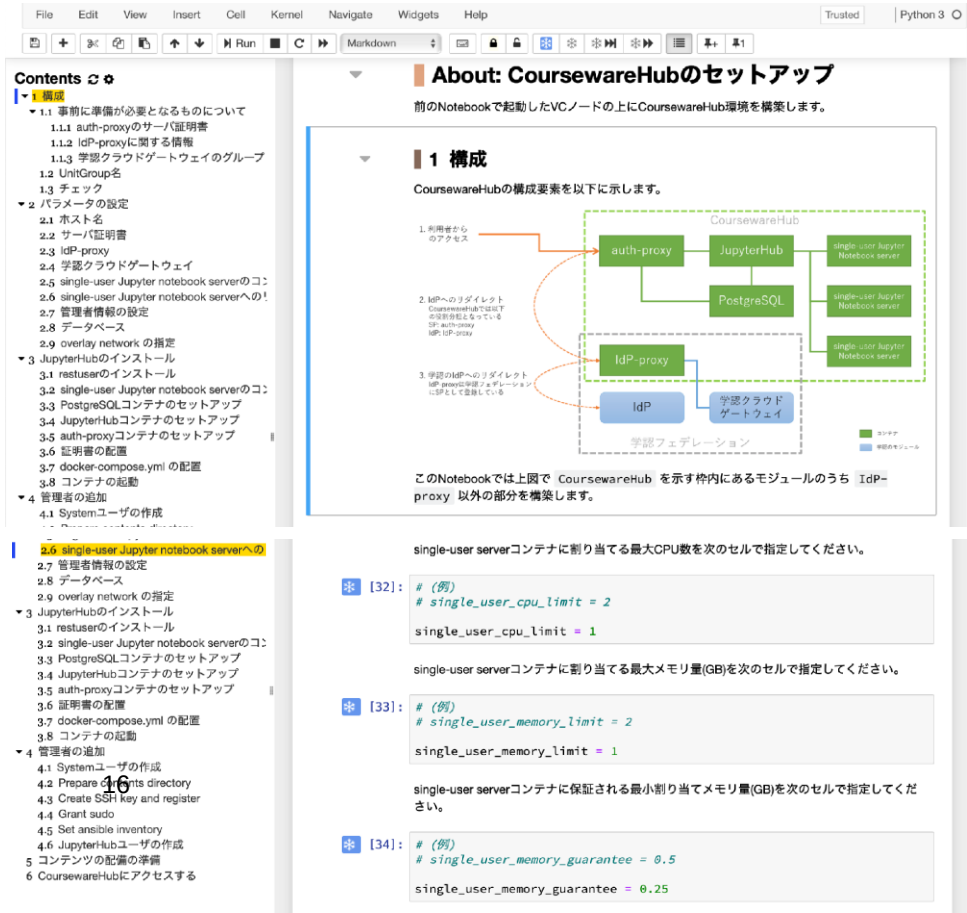

000-README.ipynb
001-VCノード作成-構成1.ipynb
001-VCノード作成-構成1_new.ipynb
021-CoursewareHubのセットアップ.ipynb
801-リソース可視化.ipynb
901-CoursewareHub環境の削除.ipynb


File Edit View Insert Cell Kernel Navigate Widgets Help
Trusted
Python 3
Contents
2.3 クラウドプロバイダの指定
VCノードを起動するプロバイダを選択します。
[5]:
# (例)
# vc_provider = 'aws'
# vc_provider = 'azure'
vc_provider = 'vmware'
2.4 VCノードに割り当てるリソース量の指定
VCノードに割り当てるリソース量を指定します。managerノードとworkerノードでは役割が異なるため、それぞれについて指定を行います。
2.5 managerノード
managerノードに割り当てるリソース量を指定します。managerノードではJupyterHub, auth-proxy, PostgreSQLコンテナなどを実行します。
2.5.1 managerノードのflavor指定
個々のリソース量を毎回指定するのは煩雑となるので、VCP SDKでは典型的な構成のパラメータセットを事前に定義しています。事前に定義したパラメータセットのことをVCP SDKでは flavor と呼んでいます。
リソース量を指定するためのオブジェクト spec に設定できるパラメータはクラウドプロバイダ毎に異なるので flavor もプロバイダ毎にの定義されています。次のセルを実行すると vc_provider に設定したプロバイダに対応する flavor の一覧が表示されます。
[6]:
vcp.df_flavors(vc_provider)
Out[6]:
flavor disk_size memory num_cpus
0 small 40 1024 1
1 medium 40 2048 2
2 large 100 4096 8
表示された flavor の値から一つを選択して、次のセルに指定してください。

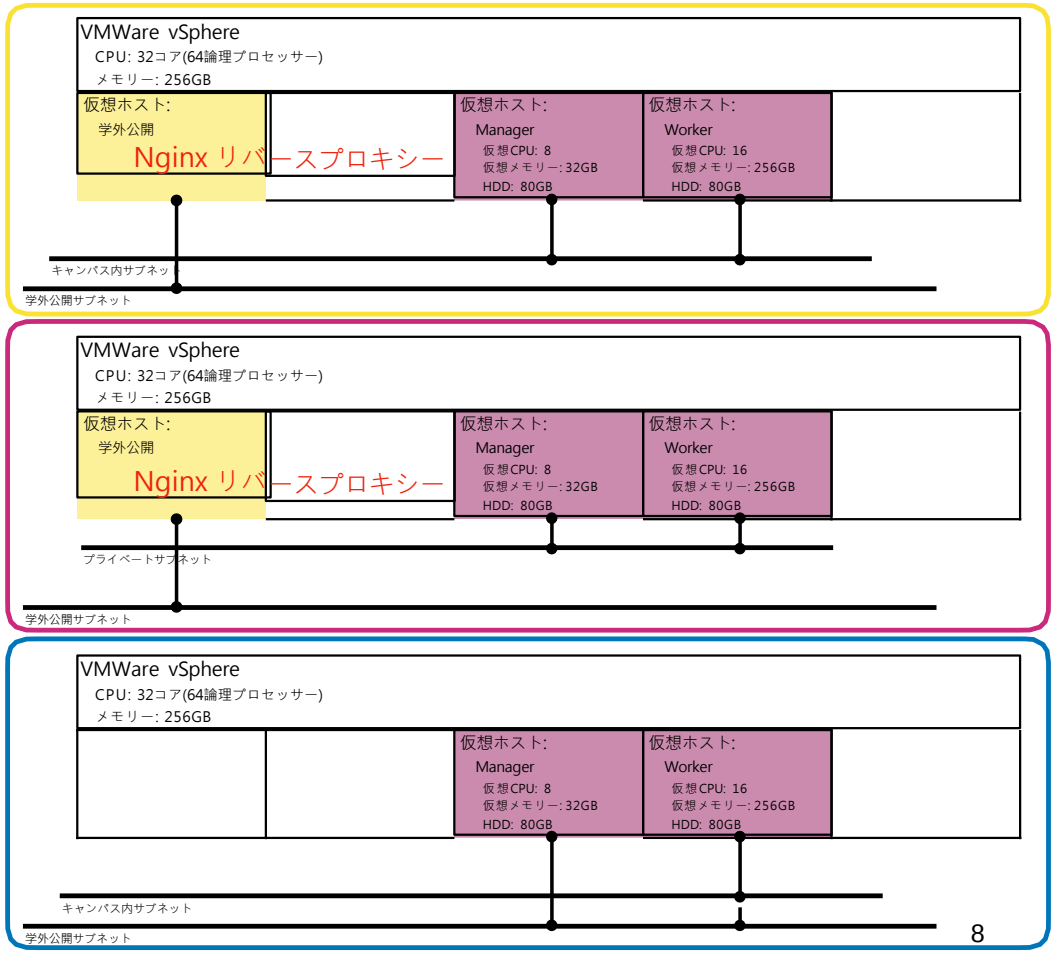

5 VCノードの起動
5.1 managerVCノードを起動する
5.2 workerVCノードを起動する
6 Ansibleの設定
7 NFS
7.1 現在の状態確認
7.2 動作確認
8 Docker Swarmの設定
5.1 managerVCノードを起動する

CousewareHub環境の構築





ネットワーク構成



- 本学のネットワークサブネット
  - キャンパス内
  - 学内公開
  - 学外公開
- セキュリティー
- 運用の柔軟性
  - CoursewareHubの切り替え
  - 学内ネットワーク申請の簡素化

授業での利用

- 2020年度後学期「Python入門」
- 受講者 119名＋α (PC教室だと定員60名弱)
- オンライン(Zoom)で授業

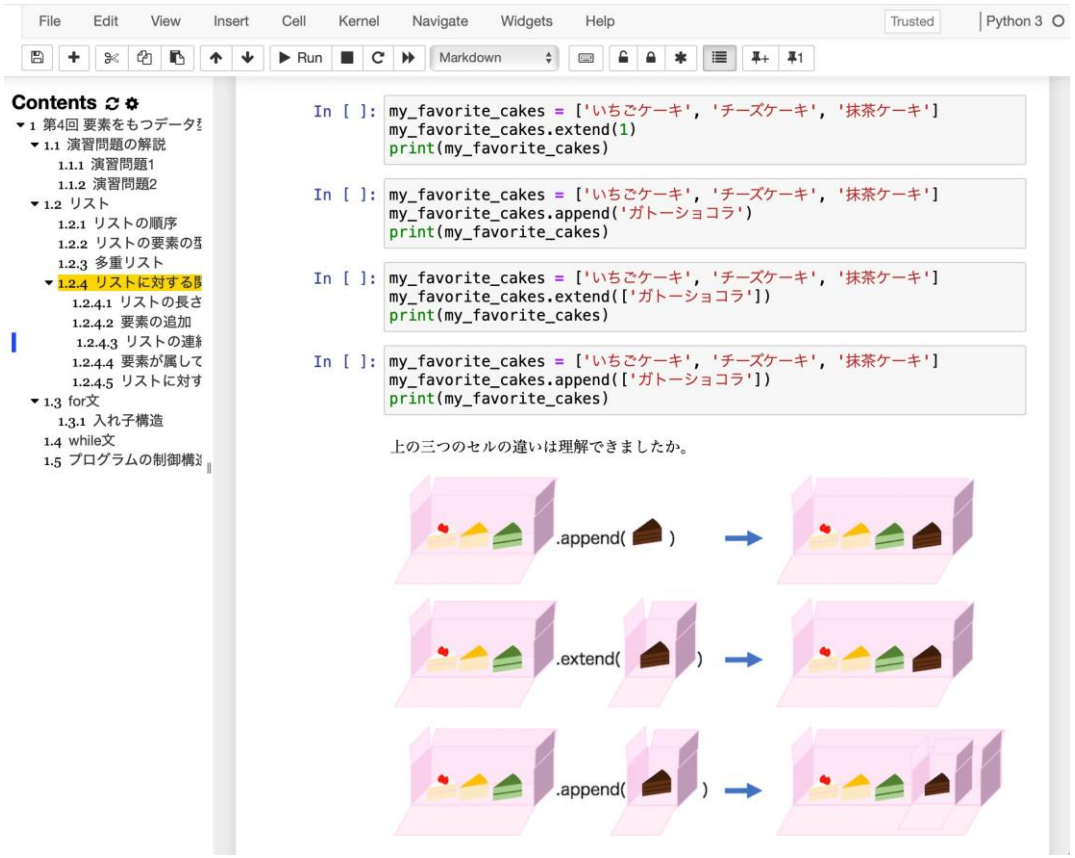
授業スケジュール

- 第1回　導入 (講義)
- 第2回　基本的なプログラムと演算子を用いた式の表現 (講義と演習)
- 第3回　要素をもつデータ型(1) (講義と演習)
- 第4回　選択型のプログラム (講義と演習)
- 第5回　反復型のプログラム (講義と演習)
- 第6回　要素をもつデータ型(2) (講義と演習)
- 第7回　関数の利用 (講義と演習)
- 第8回　ライブラリの利用 (講義と演習)
- 第9回　ファイル処理 (講義と演習)
- 第10回　データ整形処理 (講義と演習)オ
- 第11回　ブジェクト指向 (講義と演習)
- 第12回　NumPyライブラリを利用したデータ処理 (講義と演習)
- 第13回　pandasライブラリを利用したデータ処理(1) (講義と演習)
- 第14回　pandasライブラリを利用したデータ処理(2) (講義と演習)
- 第15回　Matplotlibを利用したグラフ描画 (講義と演習)

授業の流れ

- 教材の作成
  - 教材配布用フォルダー(textbook/)に保存
- 教材の配布
  - admin_tools/10_ReleaseMaterials.ipynb
- 授業
- 課題の提出 (学生)
  - 指定したフォルダー(report/)に保存
- 課題の回収
  - admin_tools/11_CollectAnswer.ipynb

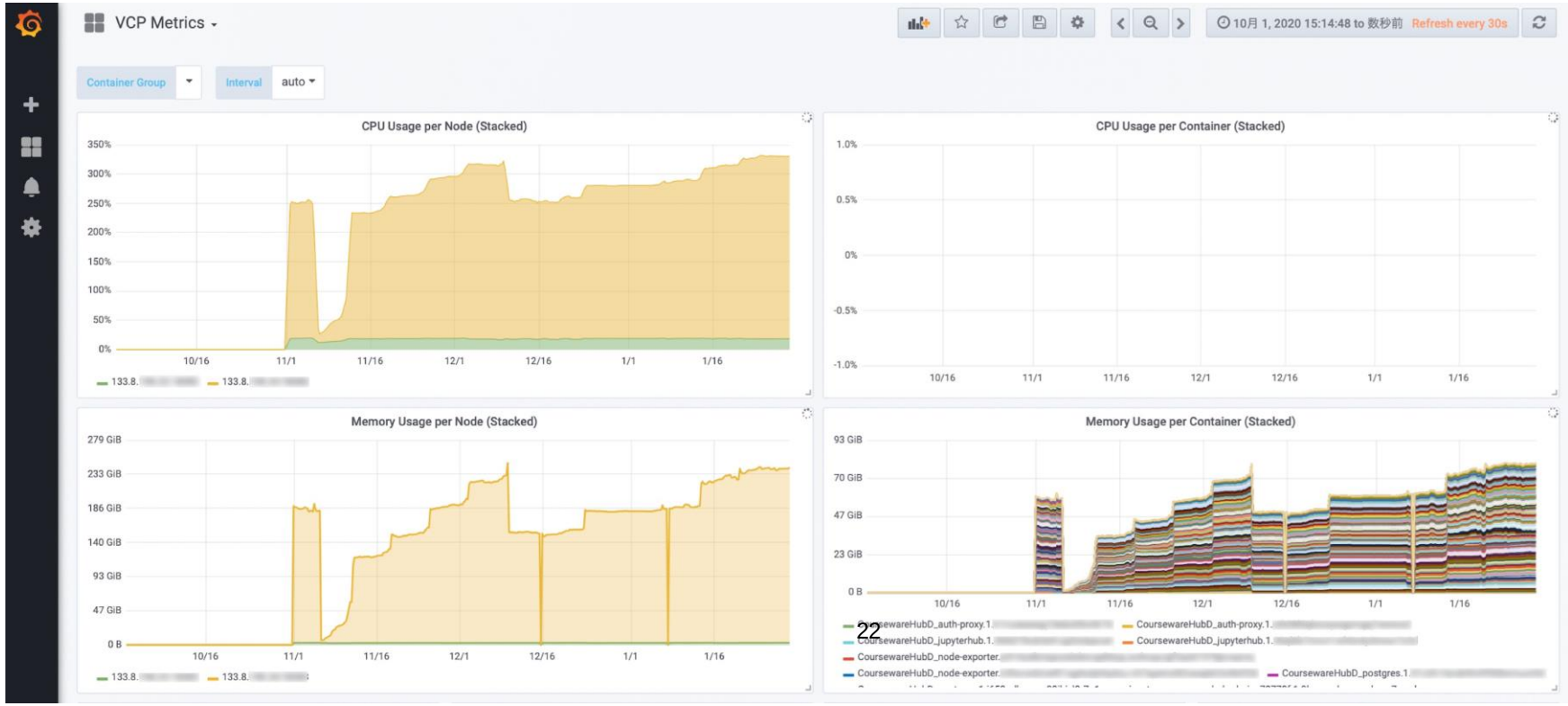
教材の例


File Edit View Insert Cell Kernel Navigate Widgets Help
Trusted
Python 3
Run
Markdown
Contents
1 第4回 要素をもつデータ
1.1 演習問題の解説
1.1.1 演習問題1
1.1.2 演習問題2
1.2 リスト
1.2.1 リストの順序
1.2.2 リストの要素の
1.2.3 多重リスト
1.2.4 リストに対する
1.2.4.1 リストの長さ
1.2.4.2 要素の追加
1.2.4.3 リストの連
1.2.4.4 要素が属して
1.2.4.5 リストに対す
1.3 for文
1.3.1 入れ子構造
1.4 while文
1.5 プログラムの制御構
In [ ]:
my_favorite_cakes = ['いちごケーキ', 'チーズケーキ', '抹茶ケーキ']
my_favorite_cakes.extend(1)
print(my_favorite_cakes)
In [ ]:
my_favorite_cakes = ['いちごケーキ', 'チーズケーキ', '抹茶ケーキ']
my_favorite_cakes.append('ガトーショコラ')
print(my_favorite_cakes)
In [ ]:
my_favorite_cakes = ['いちごケーキ', 'チーズケーキ', '抹茶ケーキ']
my_favorite_cakes.extend(['ガトーショコラ'])
print(my_favorite_cakes)
In [ ]:
my_favorite_cakes = ['いちごケーキ', 'チーズケーキ', '抹茶ケーキ']
my_favorite_cakes.append(['ガトーショコラ'])
print(my_favorite_cakes)
上の三つのセルの違いは理解できましたか。
.append( )
.extend( )
.append( )


同じことを繰り返して実行したいことがあります。
簡単なのは、繰り返したい回数だけ同じ文を書くことです。
In [ ]:
print('小麦粉をふるいにかける')
In [ ]:
# 小麦粉は数回ふるいにかけたほうがだまにならない
print('小麦粉をふるいにかける')
print('小麦粉をふるいにかける')
print('小麦粉をふるいにかける')
print('小麦粉をふるいにかける')
print('小麦粉をふるいにかける')
for n in range(5):
In [ ]:
for n in range(5):
print('小麦粉をふるいにかける')
1.3.1 入れ子構造
for 文の中(下)には実行文を書きますが、for 文も実行文の一つですから、for 文の中(下)に別の for 文を書くことができます。
※ if 文でも同様のことが可能でした。
このような構造を「入れ子構造」といいます。
for ……:
for ……:
for ……:
for ……:
寿永腹頓
ダルマトリョーシカ
for ……:
for ……:
for ……:
for ……:

リソース監視

- 可視化ツールの提供 (Grafana)



今後の予定

- LTI (Learning Tools Interoperability)
  - IMS Global Learning Consortiumが提唱している標準規格
  - LMS等の機能の一部を他のLMSから利用することが可能