

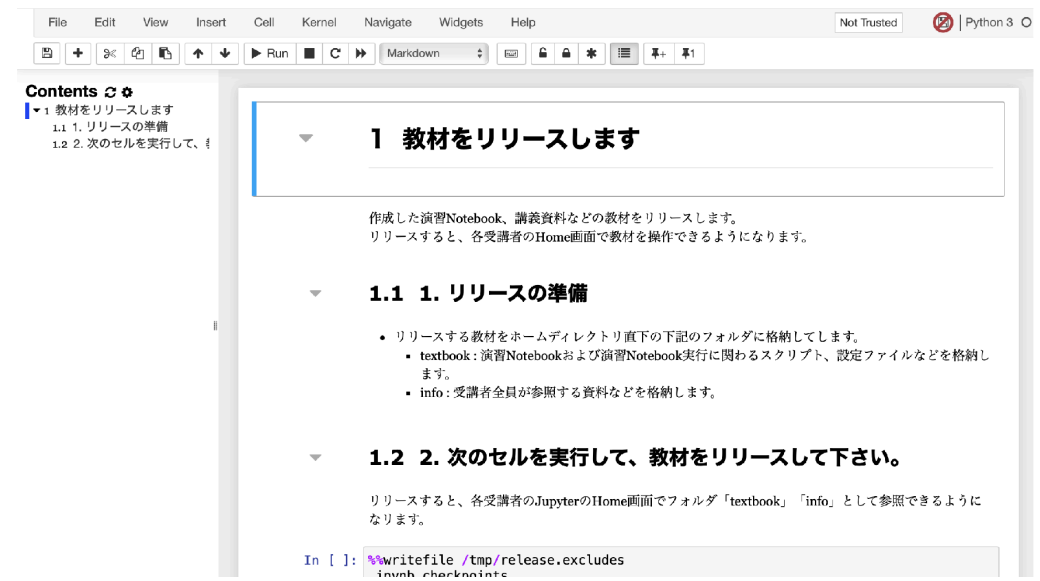
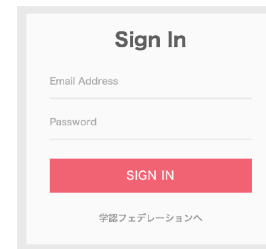
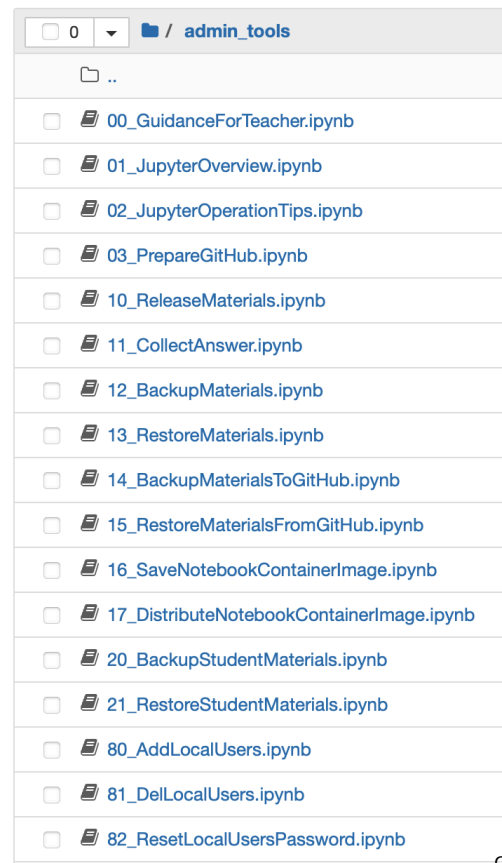
# CoursewareHubを利用した授業: Python入門

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

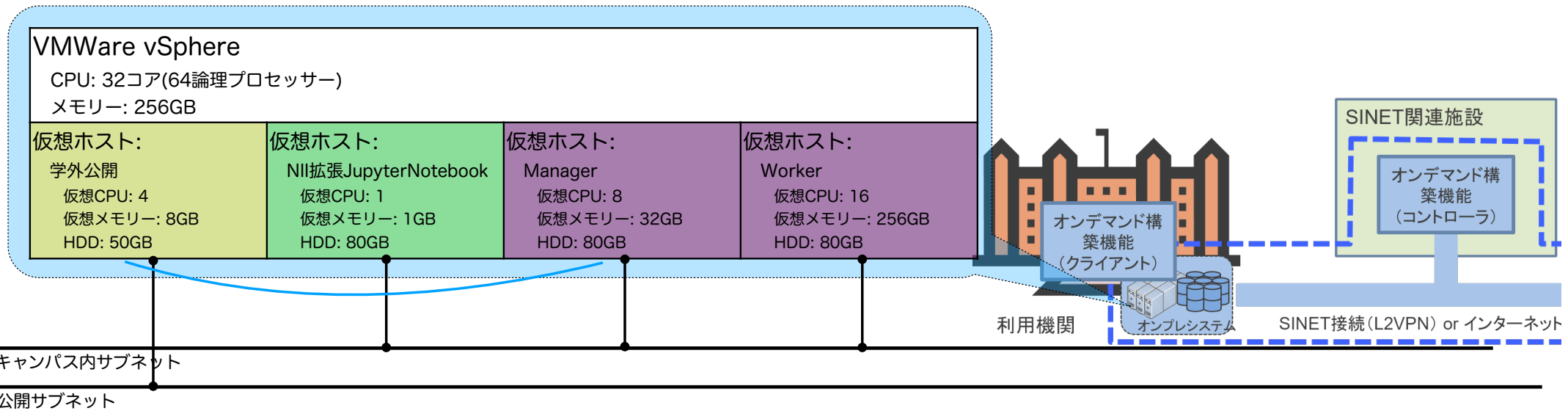
- 認証連携 (学認)
- 教材配布
- 課題回収
- 操作履歴収集
- .....



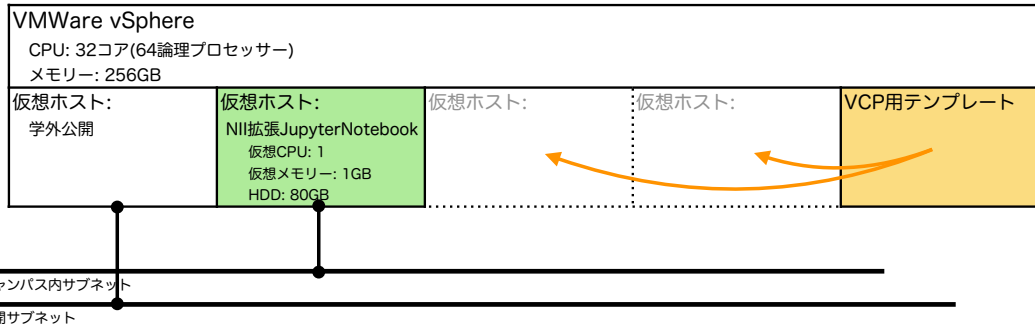
# CoursewareHub利用の経緯

- 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



# 環境構築



## • NII拡張JupyterNotebook環境の作成

- VCノードの作成
- CoursewareHub環境の構築

- 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

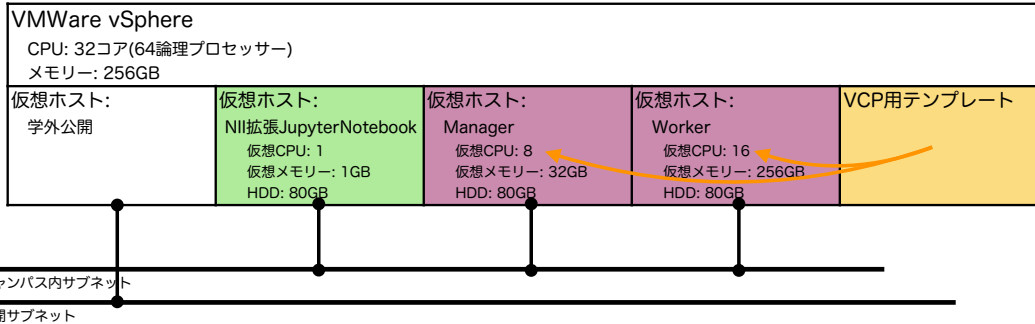
## • VCP用テンプレートの作成

### 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



- 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

Contents

- 1 はじめに
  - 1.1 ノード構成
  - 1.2 事前に用意が必要となるものについて
    - 1.2.1 VCCアクセストークン
    - 1.2.2 SSH公開鍵ペア
    - 1.2.3 VCノードに割り当てるアドレス
- 2 VCノードに関するパラメータ
  - 2.1 VCCアクセストークンの入力
  - 2.2 UnitGroup名の指定
  - 2.3 クラウドプロバイダの指定
  - 2.4 VCノードに割り当てるリソース量の指定
  - 2.5 managerノード
    - 2.5.1 managerノードのflavor指定
  - 2.6 workerノード
    - 2.6.1 workerノードのflavor指定
    - 2.6.2 workerノードのルートボリュームサイズ
    - 2.6.3 ノード数の指定
  - 2.7 アドレスの指定
    - 2.7.1 IPアドレスの指定
    - 2.7.2 MACアドレスの指定
  - 2.8 アドレスプールの指定
  - 2.9 SSH公開鍵認証の鍵ファイルの指定
  - 2.10 パラメータの保存
- 3 VCディスクに関するパラメータ
- 4 VCディスクの作成
  - 4.1 UnitGroupの作成
  - 4.2 VCディスクの作成
- 5 VCノードの起動
  - 5.1 manager用VCノードを起動する
  - 5.2 worker用VC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

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]:
```

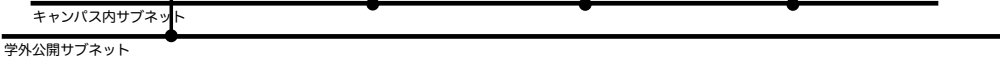
	flavor	disk_size	memory	num_cpus
0	small	40	1024	1
1	medium	40	2048	2
2	large	100	4096	8

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。

5.1 manager用VCノードを起動する

# CousewareHub環境の構築

VMWare vSphere				
CPU: 32コア(64論理プロセッサ) メモリ: 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: 1GB HDD: 80GB	仮想ホスト: Manager 仮想CPU: 8 仮想メモリ: 32GB HDD: 80GB	仮想ホスト: Worker 仮想CPU: 16 仮想メモリ: 256GB HDD: 80GB	



- 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

**Contents**

- 1 構成
  - 1.1 事前に準備が必要となるものについて
    - 1.1.1 auth-proxyのサーバ証明書
    - 1.1.2 IdP-proxyに関する情報
    - 1.1.3 学認クラウドゲートウェイのグループ
  - 1.2 UnitGroup名
  - 1.3 チェック
  - 2 パラメータの設定
    - 2.1 ホスト名
    - 2.2 サーバ証明書
    - 2.3 IdP-proxy
    - 2.4 学認クラウドゲートウェイ
    - 2.5 single-user Jupyter notebook serverのコ
    - 2.6 single-user Jupyter notebook serverへの!
    - 2.7 管理者情報の設定
    - 2.8 データベース
    - 2.9 overlay network の指定
  - 3 JupyterHubのインストール
    - 3.1 restuserのインストール
    - 3.2 single-user Jupyter notebook serverのコ
    - 3.3 PostgreSQLコンテナのセットアップ
    - 3.4 JupyterHubコンテナのセットアップ
    - 3.5 auth-proxyコンテナのセットアップ
    - 3.6 証明書の配置
    - 3.7 docker-compose.yml の配置
    - 3.8 コンテナの起動
  - 4 管理者の追加
    - 4.1 Systemユーザの作成
    - 4.2 Prepare contents directory
    - 4.3 Create SSH key and register
    - 4.4 Grant sudo
    - 4.5 Set ansible inventory
    - 4.6 JupyterHubユーザの作成
    - 5 コンテンツの配備の準備
    - 6 CoursewareHubにアクセスする

**About: CoursewareHubのセットアップ**

前のNotebookで起動したVCノードの上にCoursewareHub環境を構築します。

### 1 構成

CoursewareHubの構成要素を以下に示します。

このNotebookでは上図で CoursewareHub を示す枠内にあるモジュールのうち IdP-proxy 以外の部分を構築します。

single-user serverコンテナに割り当てる最大CPU数を次のセルで指定してください。

```
[*] [32]: # (例)
# single_user_cpu_limit = 2
single_user_cpu_limit = 1
```

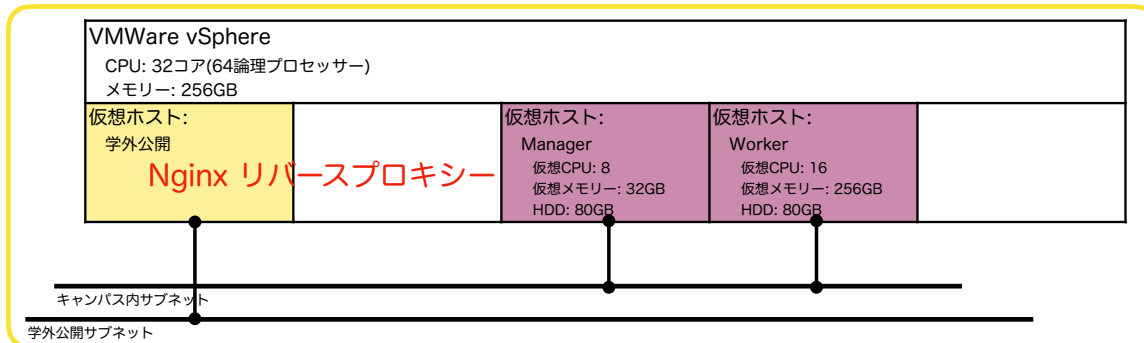
single-user serverコンテナに割り当てる最大メモリ量(GB)を次のセルで指定してください。

```
[*] [33]: # (例)
# single_user_memory_limit = 2
single_user_memory_limit = 1
```

single-user serverコンテナに保証される最小割り当てメモリ量(GB)を次のセルで指定してください。

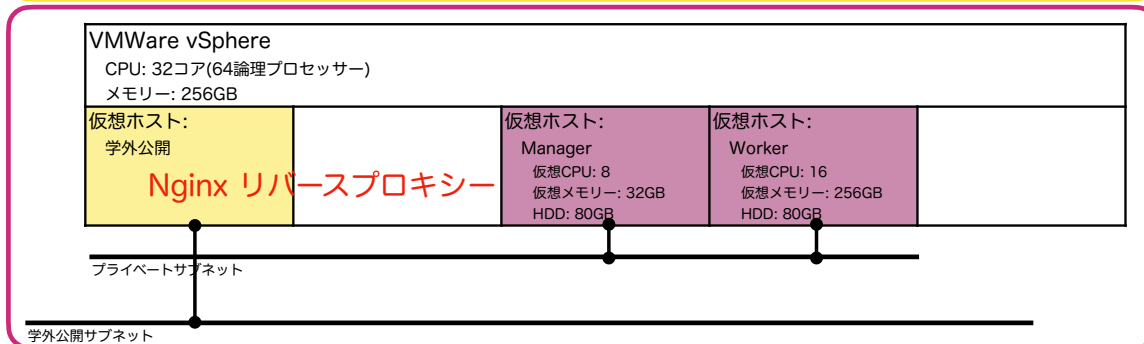
```
[*] [34]: # (例)
# single_user_memory_guarantee = 0.5
single_user_memory_guarantee = 0.25
```

# ネットワーク構成



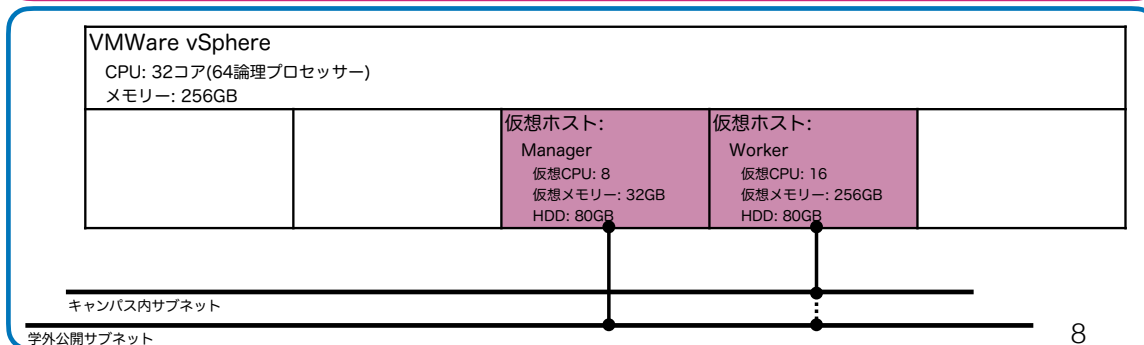
今回採用

- 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



当初希望

- 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



標準構成

# 授業での利用

- 2020年度後学期「Python入門」
- 受講者 119名+ $\alpha$  (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)
- ・ 第11回 オブジェクト指向 (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

The screenshot shows a Jupyter Notebook with a sidebar on the left containing a 'Contents' table of contents. The main area displays four code cells, each starting with 'In [ ]:' and containing Python code that manipulates a list named 'my\_favorite\_cakes'. The code demonstrates the use of 'extend()' and 'append()' methods. Below the code cells, there is a question in Japanese: '上の三つのセルの違いは理解できましたか。' (Did you understand the difference between the three cells above?).

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 リストの要素の型
    - 1.2.3 多重リスト
    - ▼ 1.2.4 リストに対する操作
      - 1.2.4.1 リストの長さ
      - 1.2.4.2 要素の追加
      - 1.2.4.3 リストの連結
      - 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造

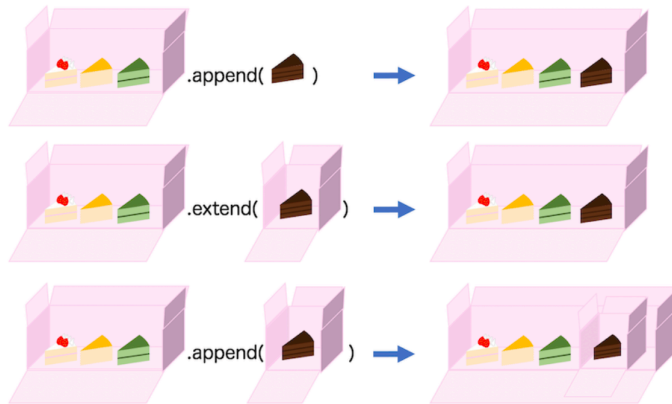
```
In [ ]: my_favorite_cakes = ['いちごケーキ', 'チーズケーキ', '抹茶ケーキ']
my_favorite_cakes.extend(1)
print(my_favorite_cakes)

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。

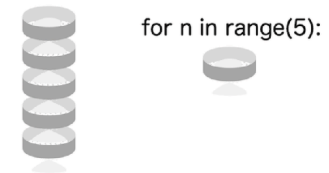


同じことを繰り返して実行したいことがあります。  
簡単なのは、繰り返したい回数だけ同じ文を書くことです。

```
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')
```



```
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。

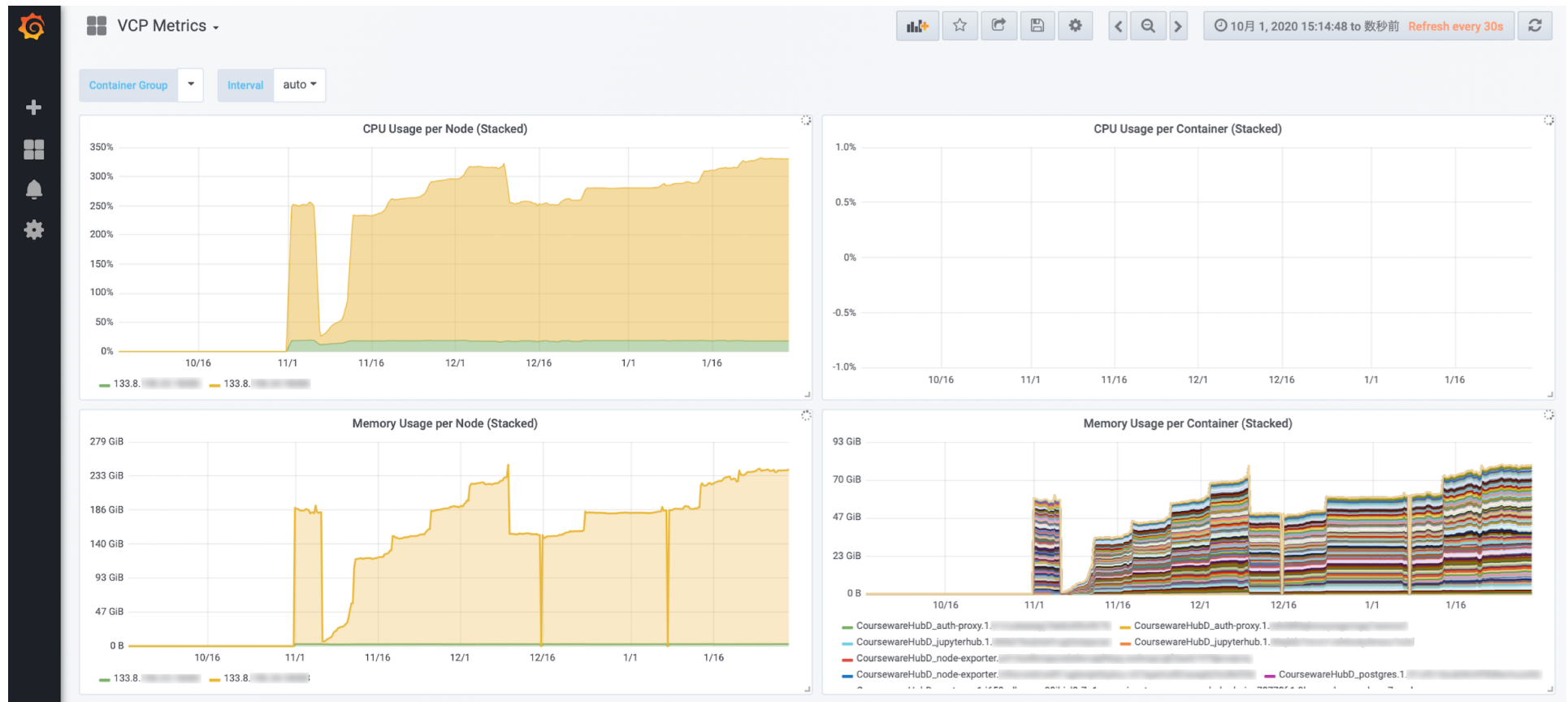
※ if 文でも同様のことが可能でした。

このような構造を「入れ子構造」といいます。



# 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

