

# 2023.9.1 情報処理技術セミナー 午後の演習

NII クラウド担当 谷沢

# 本日の流れ

- ▶ OperationHubを使ってみよう - 90min.
  - ▶ OperationHubでJupyter Notebookの操作を習得 - 50min.
  - ▶ Jupyter Notebookの運用への適用 - LC4RIについて - 40min.
- ▶ Etherpadを構築・運用してみよう - 90min.
  - ▶ LC4RIを実践してみよう - MinIOを構築・運用 - 30min.
  - ▶ LC4RIで実際に環境を作ってみよう - Etherpadを構築・運用 - 60min.
  - ▶ まとめ: Jupyter Notebookによる情報システムの構築・運用

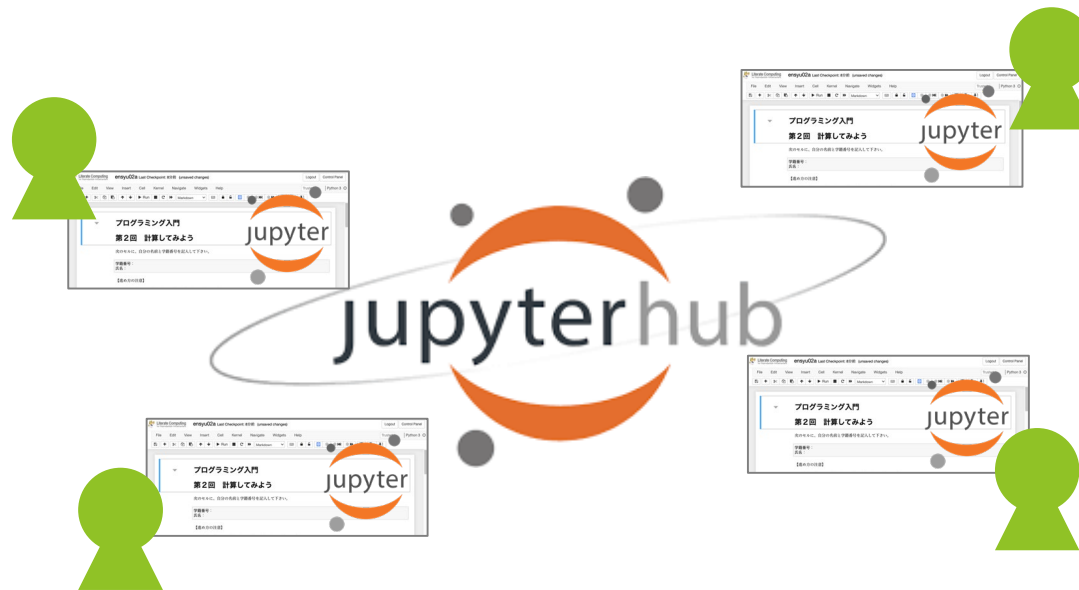
# 午後の内容

- ▶ 座学＋演習を交互に行っていくスタイルで進めていきます
- ▶ 演習はこちらがご用意したOperationHubに対して、あらかじめお送りしたアカウント情報でログインしていただく形で実施します
- ▶ 操作上のトラブル・質問はSlackチャンネル **#2023\_cloud** をお願いします
  - ▶ スタッフがサポートさせていただきます
  - ▶ 必要に応じて、スクリーンショットなどをお願いする場合があります
- ▶ (質問) 会議アプリと作業環境は同一PCでしょうか？
  - ▶ Slack **#2023\_cloud** にご回答ください

# OperationHubでJupyter Notebook の操作を習得

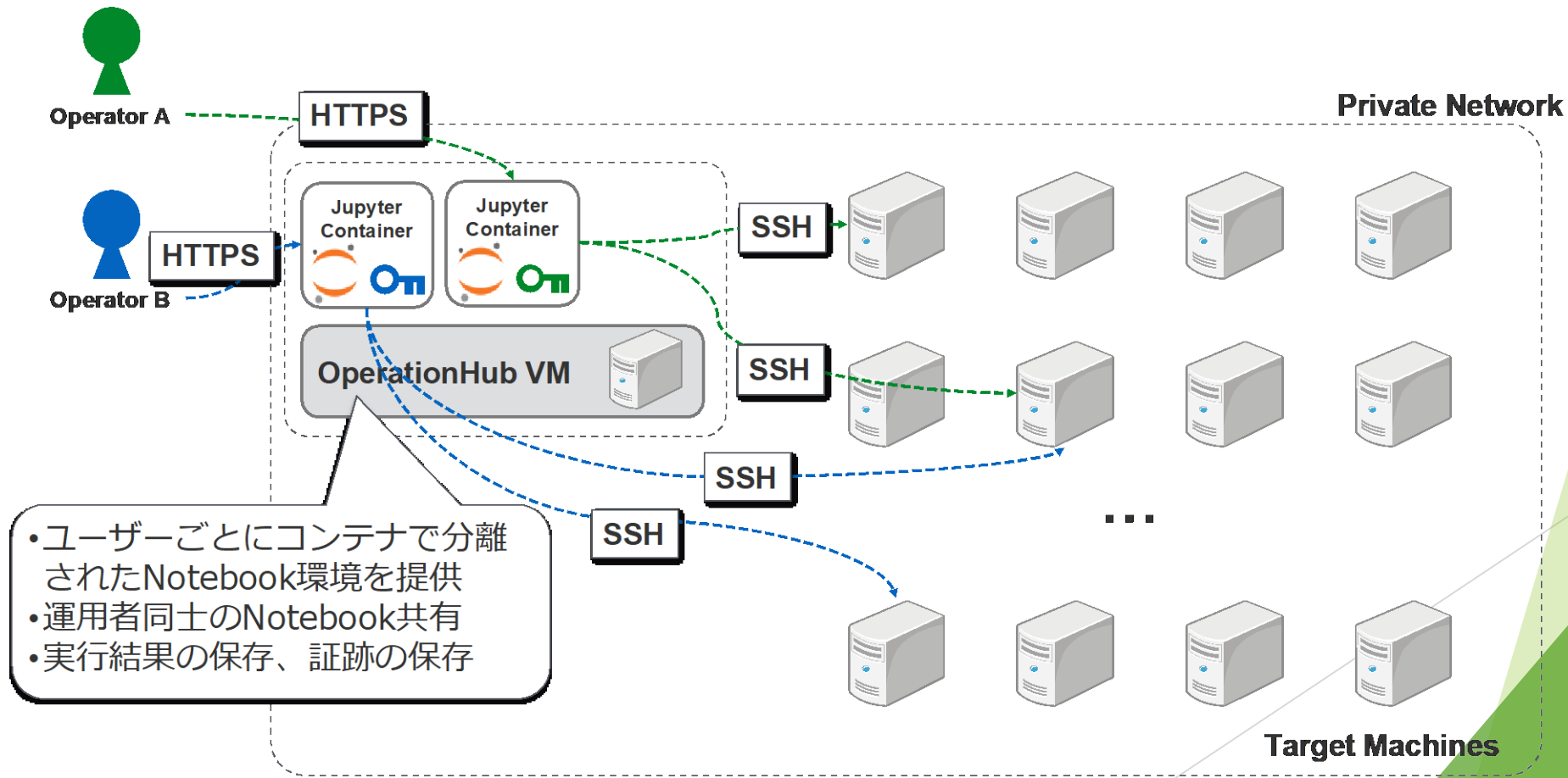
# Jupyter Notebookによる情報システムの構築・運用のための環境: OperationHub

- ▶ OperationHub = JupyterHub + 情報システム構築・運用のための便利機能群
- ▶ JupyterHub … 複数のユーザーを収容可能なJupyter Notebook環境
  - ▶ 午前のGakuNin RDMデータ解析機能も、実体はJupyterHub



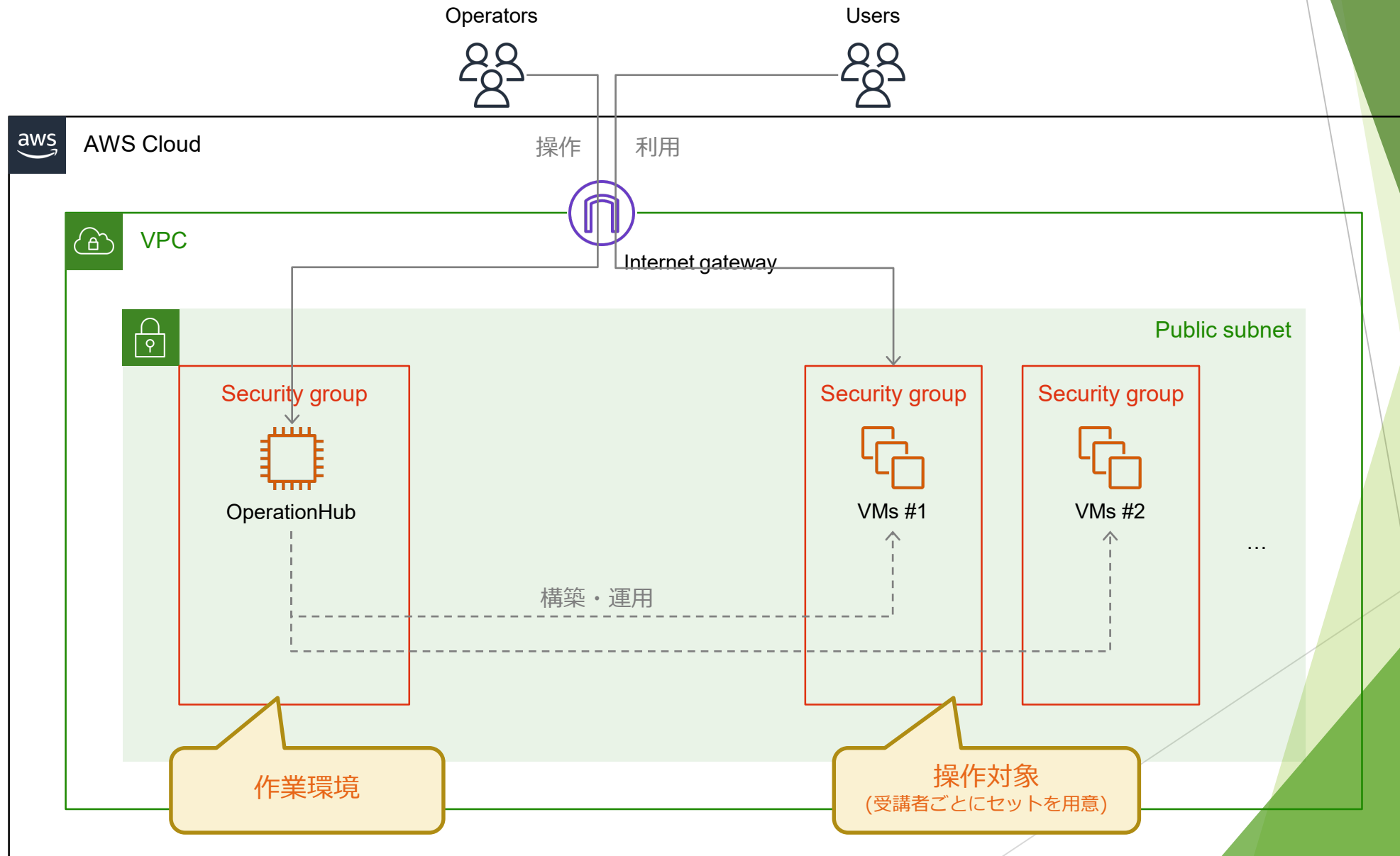
# Jupyter Notebookによる情報システムの構築・運用のための環境: OperationHub

- ▶ JupyterHubベースの「運用管理サーバ」
- ▶ <https://github.com/NII-cloud-operation/OperationHub>



# 本セミナーにおける環境構成

- ▶ OperationHub経由で、仮想マシン(VM)を操作します



# (演習)OperationHubへのログイン

- ▶ OperationHubにログインし、構築前の準備をしましょう (15min)
- ▶ 1. CA証明書のインストール
  - ▶ 事前に配布した **seminar-20230829-ca.crt** をご自身の環境にインストールしてください
  - ▶ 手順は次ページに記載します
- ▶ 2. OperationHubへのログイン
  - ▶ [https://\(seminar用OperationHubのURL\)](https://(seminar用OperationHubのURL)) を開き、以下でログインしてみましょう
    - ▶ ユーザー名 **c{受講者番号:01~10}operator**
    - ▶ パスワードはユーザーに対応するパスワード(2つ)のうち、左のもの
  - ▶ 証明書関連のエラーが表示される場合は、1.が実施できていない可能性があります
    - ▶ Firefoxだとエラーになりますので、Firefox以外でお願いします



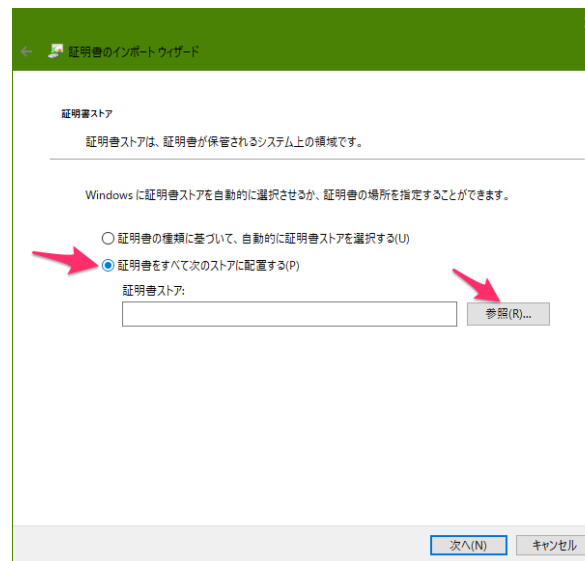
# CA証明書のインストール (Windows) ①

- ▶ 1. CA証明書のファイルseminar-20230829-ca.crtをダブルクリックします
- ▶ 2. セキュリティの警告ダイアログが表示されます。「開く」をクリックして、証明書ファイルを開きます
- ▶ 3. 「証明書のインストール」をクリックします。
- ▶ 4. 保存先が「現在のユーザー」になっていることを確認し、「次へ」をクリックします
- ▶ 5. 「証明書をすべて次のストアに配置する」を選択します
- ▶ 6. 「参照」をクリックし「信頼されたルート証明機関」を選んで「OK」をクリックします

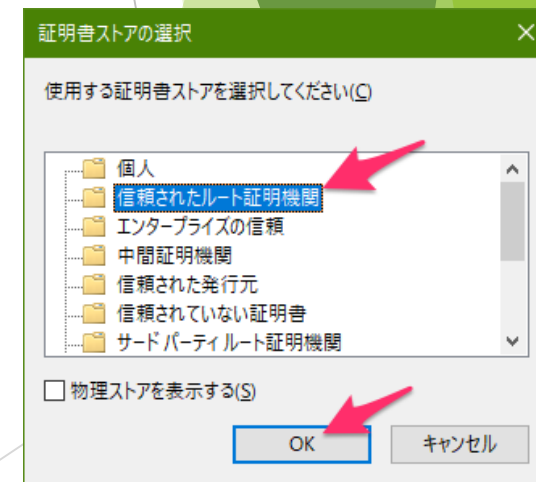
3.



5.



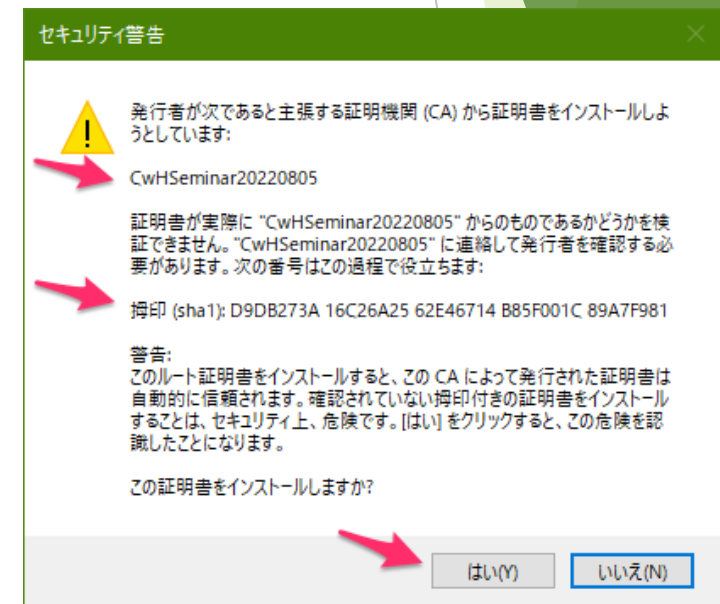
6.



# CA証明書のインストール (Windows) ②

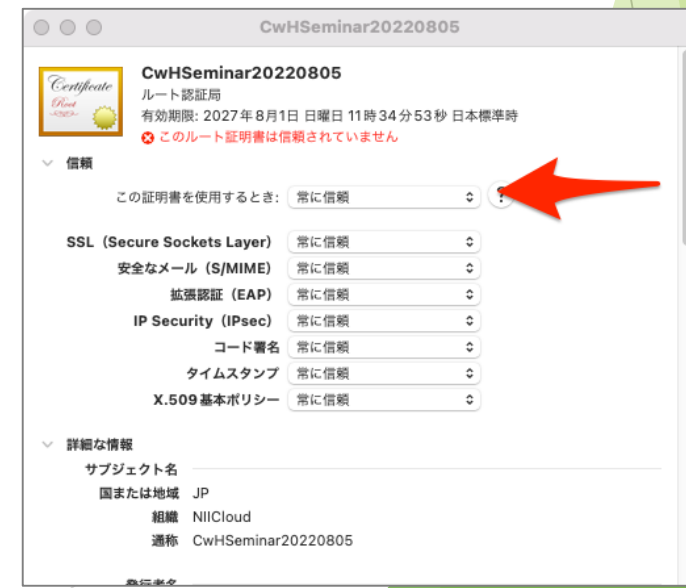
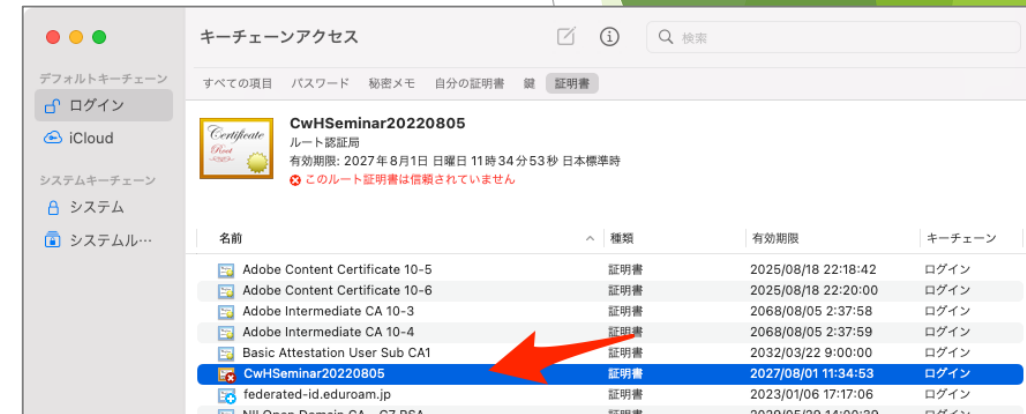
- ▶ 7. 証明書ストアに「信頼されたルート証明機関」が選択されていることを確認し、「次へ」をクリックします
- ▶ 8. インポート先などの確認を求められるので、「完了」をクリックします
- ▶ 9. 最後に再びインストールを確認されるので、「はい」をクリックしてインストールします
- ▶ 10. 「正しくインポートされました。」と表示されればインポート完了です

9.



# CA証明書のインストール(Mac)

- ▶ 1. キーチェーンアクセスを起動します
- ▶ 2. リスト上部の「証明書」をクリックします
- ▶ 3. キーチェーンアクセスに **seminar-20230829-ca.crt** をドラッグ&ドロップします
  - ▶ CwHSeminar20230829 という証明書がインポートされます
  - ▶ この時点では赤色のアイコンがついています
- ▶ 4. CwHSeminar20230829 をダブルクリックします
- ▶ 5. ダイアログが開くので、「信頼」をクリックし、この証明書を使用するとき: 「常に信頼」を選択します
- ▶ 6. ダイアログを閉じると設定が反映されます
- ▶ キーチェーンアクセスの証明書一覧画面で、CwHSeminar20230829 が青色のアイコンが表示されていればCA証明書が信頼された状態です



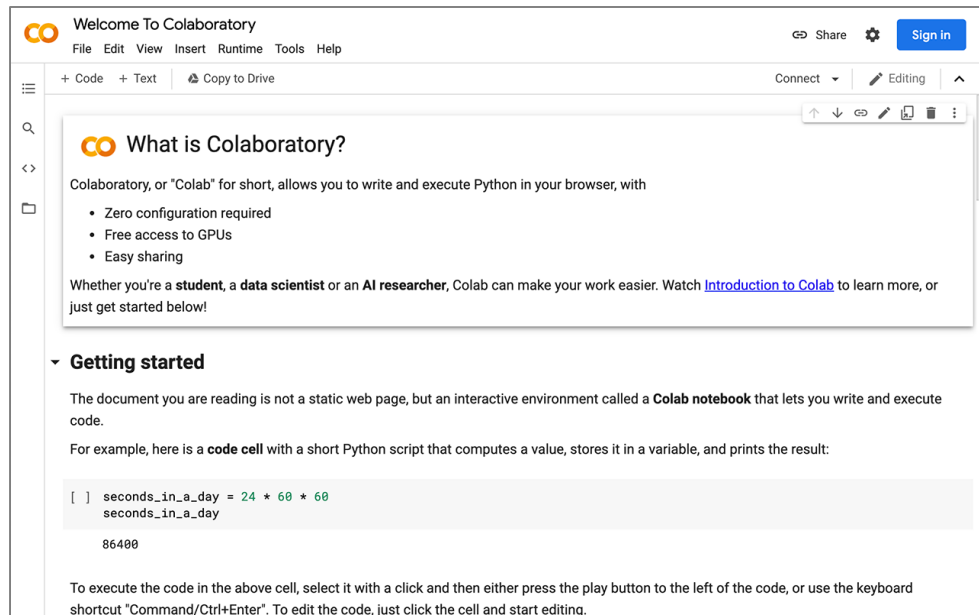
# Jupyter Notebookとは

- ▶ Jupyter Notebook ご存知ですか？/使ったことありますか？
- ▶ ジュピター あるいは ジュパイター(**J**ulia + **P**ython + **R**) ノートブック
- ▶ Webベースのインタラクティブなユーザインタフェース + スクリプト実行環境
- ▶ 多種のスクリプトを利用可能
  - ▶ Python
  - ▶ R
  - ▶ Julia
  - ▶ Perl
  - ▶ PHP
  - ▶ 他にも: <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>



# オープンソースソフトウェア

- ▶ 多くの開発者により活発に開発が進められている
- ▶ Jupyter Notebookを利用したサービスも増えてきている
  - ▶ Google Colab(Google Colaboratory)
  - ▶ Microsoft Azure Machine Learning



Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive

## What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

### Getting started

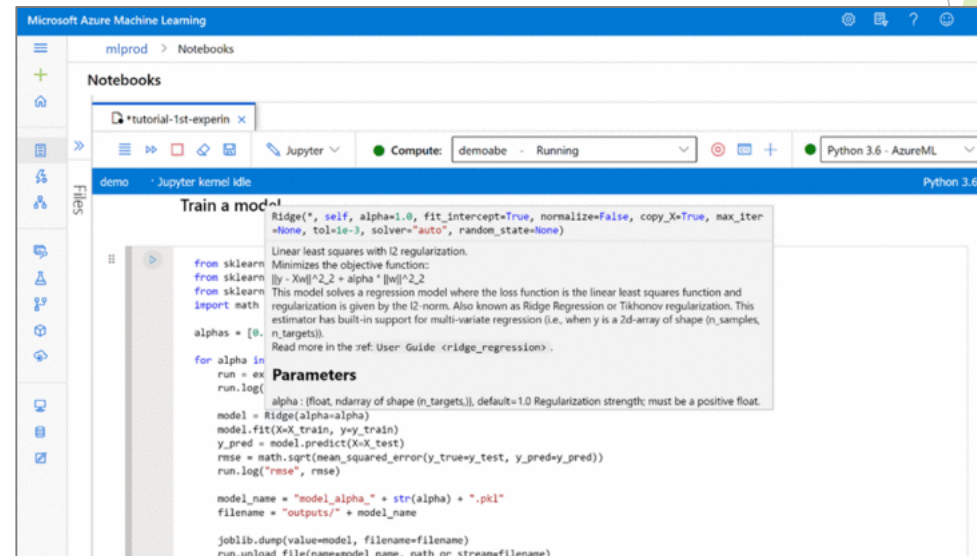
The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day

86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.



Microsoft Azure Machine Learning

miproduct > Notebooks

## Notebooks

\*tutorial-1st-experin x

Jupyter Compute: demoabe - Running Python 3.6 - AzureML

demo - Jupyter kernel idle Python 3.6.9

### Train a model

```
Ridge(*, self, alpha=1.0, fit_intercept=True, normalize=False, copy_X=True, max_iter=None, tol=1e-3, solver="auto", random_state=None)
Linear least squares with l2 regularization.
Minimizes the objective function:
|y - Xw|^2_2 + alpha * ||w||^2_2
This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. Also known as Ridge Regression or Tikhonov regularization. This estimator has built-in support for multi-variate regression (i.e. when y is a 2d-array of shape (n_samples, n_targets)).
Read more in the ref: User Guide <ridge_regression>..

Parameters
alpha : float, ndarray of shape (n_targets,), default=1.0 Regularization strength; must be a positive float.

joblib.dump(value=model, filename=filename)
run.unload(filename=model_name, nath or <streamfilename>
```

# Jupyter Notebookの ユーザインタフェース

- ▶ ブラウザウィンドウを表示し、ログインしたOperationHub画面を見てみましょう
- ▶ ファイル一覧画面 - 利用者がファイルを開いたり、ファイルを作成したりするために利用します

Literate Computing  
for Reproducible Infrastructure

Logout Control Panel

Files Running Clusters Nbextensions

Select items to perform actions on them. Upload New ↕

<input type="checkbox"/>	0	▼	📁 /	Name ↓	Last Modified	File size
<input type="checkbox"/>	📁		info		16日前	
<input type="checkbox"/>	📁		textbook		3時間前	
<input type="checkbox"/>	📁		tools		16日前	
<input type="checkbox"/>	📄		ensyu02a.ipynb		1日前	159 kB
<input type="checkbox"/>	📄		ensyu03a.ipynb		1日前	125 kB
<input type="checkbox"/>	📄		ensyu04a.ipynb		1日前	105 kB
<input type="checkbox"/>	📄		ensyu05a.ipynb		1日前	395 kB
<input type="checkbox"/>	📄		ensyu06a.ipynb		1日前	159 kB

フォルダ

ファイル

# Jupyter Notebookファイルを開く

- ▶ 拡張子 `.ipynb` を持つファイル = Jupyter NotebookあるいはNotebookファイル



- ▶ 01\_ensyu02a.ipynb のファイル名部分(リンク)をクリックしてみましょう
- ▶ Notebook画面が表示されれば成功



# Notebook画面

- ▶ Notebook画面
  - ▶ Notebookはセル(Cell)の集まり
  - ▶ セルには種類がある
    - ▶ コード(Code)セル
    - ▶ マークダウン(Markdown)セル
    - ▶ …他にも
- ▶ コードセル
  - ▶ 実行可能なスクリプトを記述可能
  - ▶ 今回のセミナーでは Python を使用
- ▶ マークダウンセル
  - ▶ 説明文を記述可能

Literate Computing  
for Reproducible Infrastructure

ensyu02a (unsaved changes)

Logout Control Panel

File Edit View Insert Cell Kernel Navigate Widgets Help **メニュー** Trusted Python 3 O

📁 + 🔍 📄 ⬆️ ⬆️ ▶ Run ■ ↺ ▶ Markdown 🗨️ ☰ **ショートカット**

▼ **プログラミング入門**

**第2回 計算してみよう** } セル

次のセルに、自分の名前と学籍番号を記入して下さい。 } セル

学籍番号： } セル  
氏名：

【進め方の注意】 } セル

- 文書を読んで、手順に沿って問いに答えながら進めて下さい。
- 最後の確認問題を解いて、ファイルを提出して下さい。
- 途中わからないことがあれば、教員またはTAに質問して下さい。

▼ **1 計算をしてみる** } セル

○演習：次の式を評価して計算をしない。  
(セルの評価を行うには、SHIFT + ENTERを押します)

In []: # 計算例 } セル  
1+2

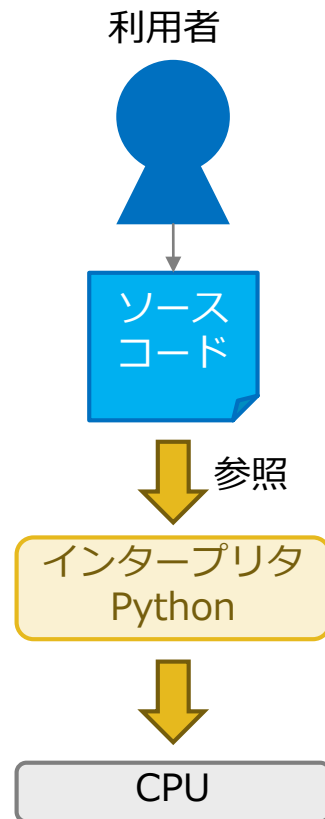
○演習：次のセルで、5 + 4 の計算をしない。



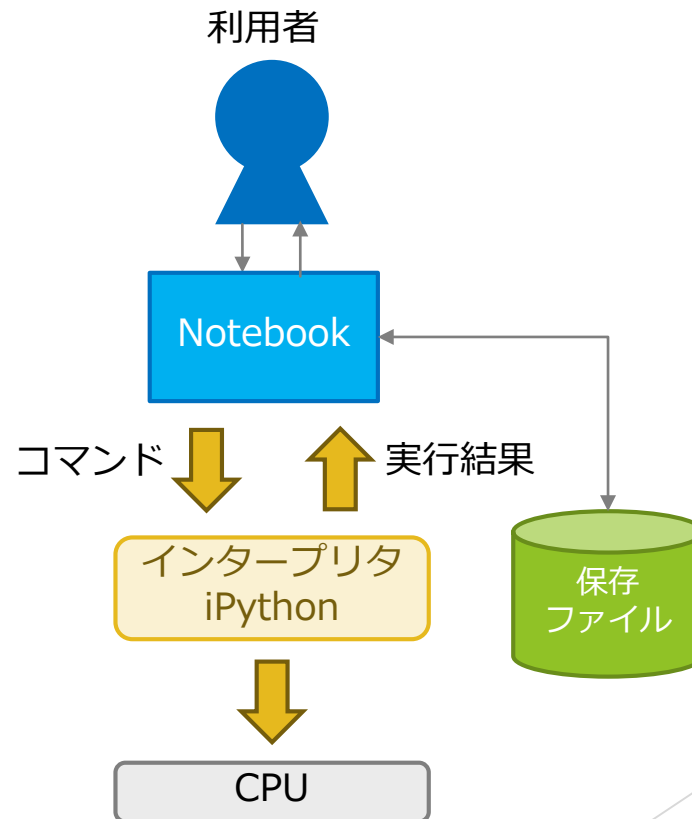
# スクリプトの実行モデル

- ▶ Jupyter Notebookに置いてスクリプトがどのように実行されるか？
- ▶ Jupyter Notebookにおける逐次実行
  - ▶ インタープリタのことを Jupyter Notebookでは カーネル(Kernel) と呼ぶ

## スクリプトの一括実行



## Jupyter Notebookの逐次実行



# スクリプトの実行

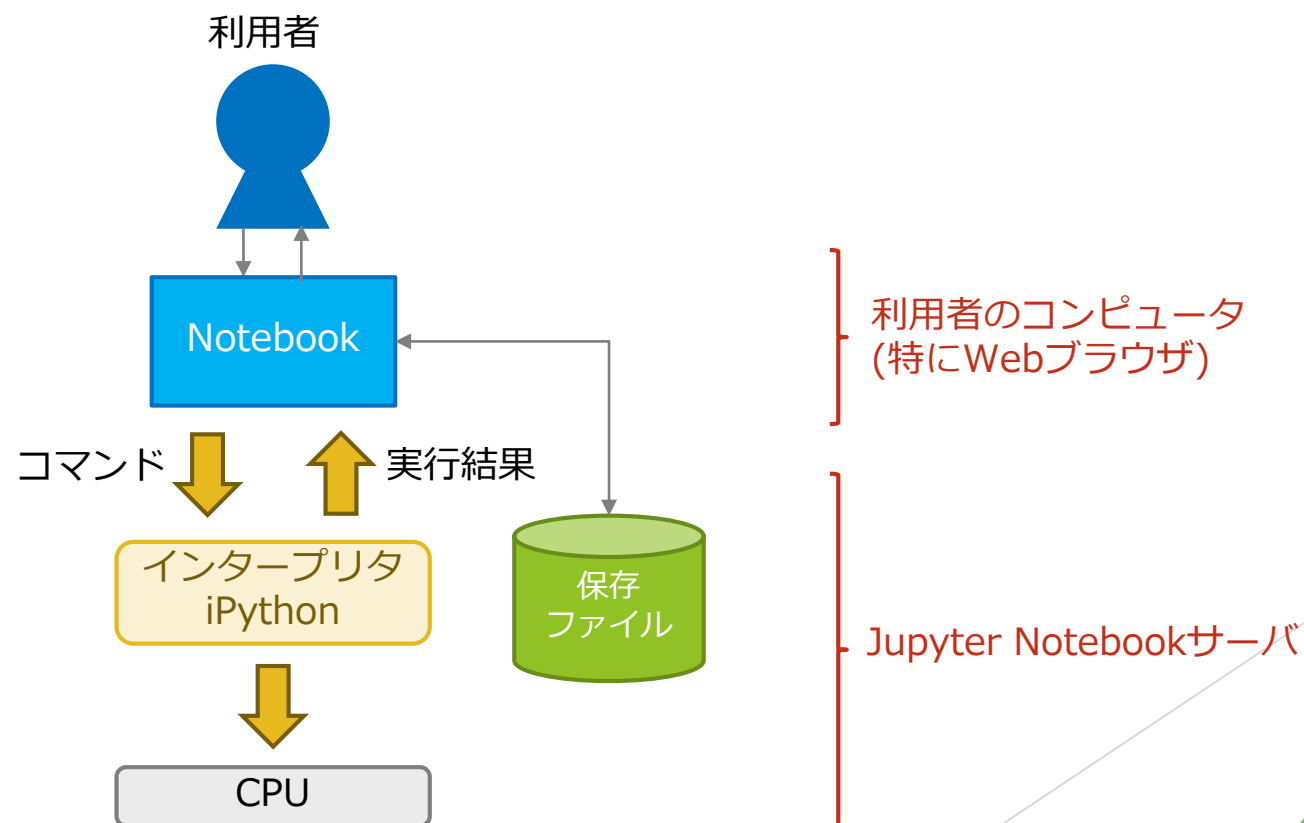
- ▶ ブラウザで開いた 01\_ensyu02a.ipynb の最初のコードセルを実行してみましょう

The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** Literate Computing for Reproducible Infrastructure, ensyu02a (unsaved changes), Logout, Control Panel.
- Menu:** File, Edit, View, Insert, Cell, Kernel, Navigate, Widgets, Help.
- Toolbar:** File, Add, Undo, Redo, Up, Down, Run (highlighted with a red arrow and text "2. Run ボタンを押す"), Stop, Refresh, Next, Code dropdown, Keyboard shortcuts, and a hamburger menu.
- Section 1:** 1 計算を試みる (1 Try calculating).
- Text:** ○演習：次の式を評価して計算をこなさい。(セルの評価を行うには、SHIFT + ENTERを押します) (Exercise: Evaluate the following expression and complete the calculation. (To evaluate the cell, press SHIFT + ENTER))
- Code Cell:** In [1]: # 計算例  
1+2
- Output:** Out[1]: 3 (3. 計算結果が出力される) (3. Calculation result is output)
- Text:** ○演習：次のセルで、 $5+4$ の計算をこなさい (Exercise: Complete the calculation of  $5+4$  in the next cell)

# スクリプトの実行 - 何が起きたのか？

- ▶ 利用者のコンピュータ と Jupyter Notebookサーバ それぞれの関係
- ▶ Jupyter Notebookのメリット
  - ▶ 利用者はWebブラウザだけあればOK = 演習における環境設定にまつわる問題を回避



# Notebook画面での操作

## ▶ セルの編集

次のセルに、自分の名前と学籍番号を記入して下さい。

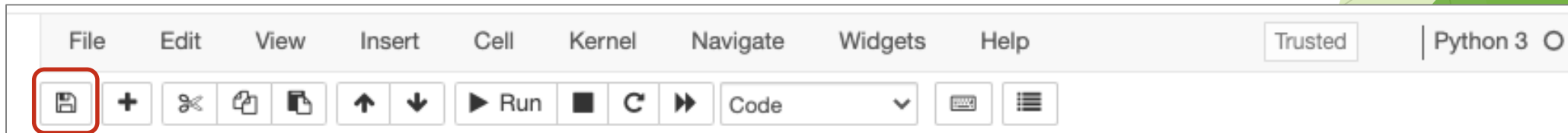
1. 編集したいセルをクリックし、セルを選択状態にする

学籍番号： DUMMY |      2. キーボードを使ってセルの内容を編集する  
氏名：

【進め方の注意】


## ▶ 編集・実行経過の保存

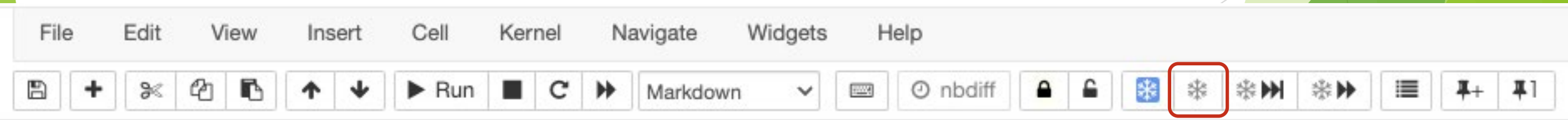
- ▶ 最新の編集・実行経過はWebブラウザで保持され、適切なタイミングでJupyter Notebookサーバに対して変更が送信される
- ▶ 予期せぬWebブラウザのクラッシュ等で最新の編集内容が失われることがあるので注意



フロッピーアイコンまたはFileメニューのSave and Checkpointをクリックする  
(直前に保存したバージョンへは、FileメニューのRevert to Checkpointで戻ることができる)

# NIIクラウド独自拡張

- ▶ Jupyter Notebookで環境構築をするにあたり、必要な拡張をNIIクラウド担当で行なっています。
  - ▶ Freeze機能
  - ▶ まとめ実行機能
  - ▶ 出力ピンどめ(出力退避)機能
- ▶ セルを実行すると、それが成功した場合はFreeze状態  となります
- ▶ Freeze状態のセルは実行操作をしても再実行されません - 考え方については後ほど…
- ▶ Freeze状態のセルを再実行したい場合は、セルをクリック(選択)した状態で以下のUnfreezeボタンをクリックしてください



# (演習) Jupyter Notebook操作の基本操作 を体験-Notebookを実行

- ▶ **OperationHubにログインした状態**で、01\_ensyu02a, 02\_ensyu03a Notebookをできるだけ多く実行してみましょう (15分)
- ▶ 説明を読みながら進めてみてください
- ▶ 余裕があれば、03\_ensyu04, 04\_ensyu09ex\_sample\_COVID-19 も試してみてください

# リッチなコンテンツ: matplotlibとpandas

- ▶ Jupyter Notebookではテキストだけでなく、図や表も扱える
  - ▶ CoursewareHub自体も図や表を活用… 簡単に紹介
- ▶ Matplotlib
  - ▶ グラフ描画
- ▶ Pandas
  - ▶ 表操作・表の画面表示

# Matplotlibとは

- ▶ Pythonで利用可能な可視化のためのライブラリ
- ▶ データ分析のデファクトスタンダード
  - ▶ 数値計算ライブラリ numpy と組み合わせさまざまなデータ可視化が可能
- ▶ 静的な出力とインタラクティブな出力
  - ▶ ex.) 静的 … 2次元のグラフ画像を出力する, 動的 … 3次元のグラフ画像をマウスで視点操作し様々な角度から眺める
  - ▶ 今回は静的な出力について紹介



# Matplotlibの使い方

- ▶ 基本的な手順
  - ▶ ライブラリのインポート
  - ▶ Figureの生成
  - ▶ Figure内にPlotを作っていく
  - ▶ 詳しくは演習Notebookで…
- ▶ より深く知りたい方は… Usage Guide  
<https://matplotlib.org/stable/tutorials/introductory/usage.html>
  - ▶ →Usage Guide自体がJupyter Notebook形式で書かれている！
- ▶ Notebook 04\_ensyu09ex\_sample\_COVID-19.ipynb も参考に…

```
# 新規陽性者数について同様に行う
```

```
df_newly_confirmed_cases_daily = pd.read_csv('newly_confirmed_cases')  
df_tokyo_confirmed_cases = df_newly_confirmed_cases_daily[['Date',
```

```
df_tokyo_confirmed_cases_copy = df_tokyo_confirmed_cases.copy()  
df_tokyo_confirmed_cases_copy['Date'] = pd.to_datetime(df_tokyo_c  
df_tokyo_confirmed_cases_copy = df_tokyo_confirmed_cases_copy.se  
df_tokyo_confirmed_cases_copy.tail(1)
```

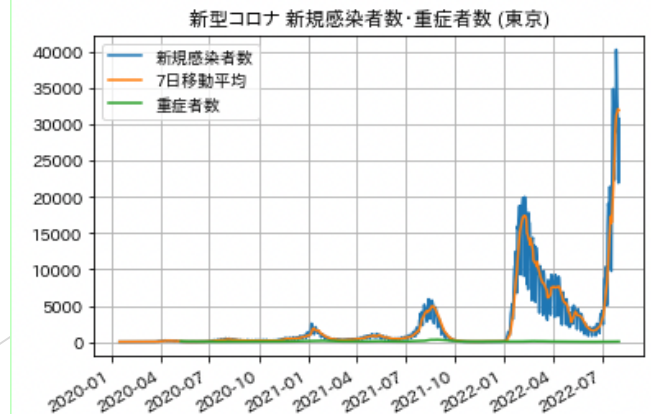
Tokyo	
Date	
2022-08-02	30842

```
plt.figure(facecolor='w')  
plt.plot(df_tokyo_confirmed_cases_copy, label="新規感染者数")  
plt.plot(df_tokyo_confirmed_cases_copy.rolling(window=7, min_periods=
```

```
plt.plot(df_tokyo_severe_cases_copy, label="重症者数")
```

```
plt.title("新型コロナ 新規感染者数・重症者数 (東京)")  
plt.gcf().autofmt_xdate() # 日付ラベル重なり解消  
plt.grid()  
plt.legend()
```

```
plt.savefig('df_tokyo_severe_cases.jpg')  
plt.show()
```



# Pandasとは

- ▶ Pythonで利用可能なデータ分析・操作ライブラリ
  - ▶ 特に、数表・時系列データ操作のためのデータ構造と演算機能
- ▶ 表形式で出力・確認を行う場面のために簡単に紹介

# Pandasの使い方

- ▶ 基本的な手順
  - ▶ ライブラリのインポート
  - ▶ 表の作成
  - ▶ 表の操作
  - ▶ 表の出力 … 画面表示・ファイル出力
- ▶ より深く知りたい方は… User Guide  
[https://pandas.pydata.org/docs/user\\_guide/10min.html](https://pandas.pydata.org/docs/user_guide/10min.html)
  - ▶ こちらはiPython形式！

The screenshot shows the pandas documentation website. The top navigation bar includes 'Getting started', 'User Guide', 'API reference', 'Development', and 'Release notes'. A search bar is present on the left. The main content area is titled '10 minutes to pandas' and contains an introduction, import instructions, and code examples for creating a Series and a DataFrame.

**10 minutes to pandas**

This is a short introduction to pandas, geared mainly for new users. You can see more complex recipes in the [Cookbook](#).

Customarily, we import as follows:

```
In [1]: import numpy as np
In [2]: import pandas as pd
```

**Object creation**

See the [Data Structure Intro](#) section.

Creating a **Series** by passing a list of values, letting pandas create a default integer index:

```
In [3]: s = pd.Series([1, 3, 5, np.nan, 6, 8])
In [4]: s
Out[4]:
0    1.0
1    3.0
2    5.0
3    NaN
4    6.0
5    8.0
dtype: float64
```

Creating a **DataFrame** by passing a NumPy array, with a datetime index and labeled columns:

```
In [5]: dates = pd.date_range("20130101", periods=6)
In [6]: dates
Out[6]:
DatetimeIndex(['2013-01-01', '2013-01-02', '2013-01-03', '2013-01-04',
              '2013-01-05', '2013-01-06'],
              dtype='datetime64[ns]', freq='D')
In [7]: df = pd.DataFrame(np.random.randn(6, 4), index=dates, columns=list("ABCD"))
In [8]: df
Out[8]:
                A         B         C         D
2013-01-01  0.469112 -0.282863 -1.509059 -1.135632
2013-01-02  1.212112 -0.173215  0.119789 -1.044236
```

# ドキュメントとしてのJupyter Notebook

- ▶ Jupyter Notebookは 説明 と コード 両方が記述できる
- ▶ 説明 の記述 … Markdownを利用
- ▶ Markdown
  - ▶ プレーンテキスト(Wordのような色・形状・図のような書式情報を含まない)形式により、書式付きの文書を記述できる
  - ▶ ex.) **\*\*太字\*\*** → **太字**
    - ▶ アスタリスク(\*)2つのセットで文字列を囲む→その文字列を太字にすると解釈される
- ▶ Jupyter NotebookにおけるMarkdown
  - ▶ Markdownには多くの方言
  - ▶ Jupyter Notebookが準拠しているのは:  
<https://daringfireball.net/projects/markdown/>

# Markdownの例

## ▶ 01\_ensyu02a.ipynb 4 変数 の例

### 画面出力

ダブルクリックで  
Markdownテキスト編集モードに

### Markdown

#### ▼ 4.1 変数とその命名規則

計算に利用する数字や文字列などを格納する記憶場所を「変数」と呼びます。

変数は自由に名前をつけることができますが、以下の規則があります。

1. アルファベット、数字、下線（アンダースコア）を利用する
2. 最初の一文字はかならずアルファベット
3. アルファベットの大文字と小文字は区別される
4. and, not, if, forなどのpythonの予約後は使えない

正しい例：

- abc
- no\_2
- x2\_\_\_\_xyz

誤りの例：

- 2a
- no.2
- for

注意：

Pythonでは日本語の変数名も使えるが、本授業では混乱をさけるため使わないこと。

#### ### 変数とその命名規則

計算に利用する数字や文字列などを格納する記憶場所を「変数」と呼びます。

変数は自由に名前をつけることができますが、以下の規則があります。

1. アルファベット、数字、下線（アンダースコア）を利用する
1. 最初の一文字はかならずアルファベット
1. アルファベットの大文字と小文字は区別される
1. and, not, if, forなどのpythonの予約後は使えない

正しい例：

- `abc`
- `no\_2`
- `x2\_\_\_\_xyz`

誤りの例：

- `2a`
- `no.2`
- `for`

注意：

Pythonでは日本語の変数名も使えるが、本授業では混乱をさけるため使わないこと。

Runでレンダリング

# Markdownの主な記法

機能	記述例	表示例
ヘッダ1(大見出し)	# はじめに	はじめに
ヘッダ2(中見出し)	## 式の評価	式の評価
ヘッダ3(小見出し)	### 関数の値	関数の値
ボールド	**注意点**	注意点
イタリック	*Italic*	<i>Italic</i>
取り消し	~~取り消し~~	取り消し
リンク	[HP](www.nii.ac.jp)	<u>HP</u>
リスト	- 項目1 - 項目2	• 項目1 • 項目2
番号付きリスト	1. 項目1 1. 項目2	1. 項目1 2. 項目2
引用	> 引用文 > 引用文	引用文 引用文

# 画像の貼りこみ

- ▶ 2つの方法
  - ▶ Notebookファイル**外**の画像ファイルを参照する
  - ▶ Notebookファイル**内**に画像ファイルを埋め込む

# Notebook外の画像ファイルを参照する

- ▶ 画像ファイルをアップロードする



- ▶ (\*)ファイルのアップロードは、一覧部分にファイルをドラッグ&ドロップするか、Uploadボタンで実行可能
- ▶ マークダウンセルから画像ファイルを参照する
  - ▶ 画像ファイルを参照するには、 ![画像タイトル](画像ファイルへの相対パス)

最初はリストfirstにジョーカー"J"が含まれているか、次の線形探索アルゴリズムを使用して調べます。

![フローチャート](fig\_09\_032.png)



# Jupyter Notebook内の画像の登録・参照

- ▶ マークダウンセルに画像ファイルをドラッグ&ドロップする

以下に画像を貼りつけます。

Markdownの記法.png

画像ファイルを エクスプローラー(Windows)/Finder(Mac) 等からドラッグ&ドロップ

- ▶ 画像ファイルはマークダウンセルに添付される
  - ▶ 画像データは .ipynb ファイル内に保持される
  - ▶ Cell Toolbar > Attachments メニューで内容確認可能

The screenshot shows the Jupyter Notebook interface with the 'Attachments' menu open. The menu options are: None, Edit Metadata, Raw Cell Format, Slideshow, Attachments, and Tags. The 'Attachments' option is highlighted. Below the menu, the text '以下に画像を貼りつけます。' is visible, followed by a link to the image file: `![[Markdown%E3%81%AE%E8%A8%98%E6%B3%95.png]]` and its attachment path: `(attachment:Markdown%E3%81%AE%E8%A8%98%E6%B3%95.png)`.



以下に画像を貼りつけます。

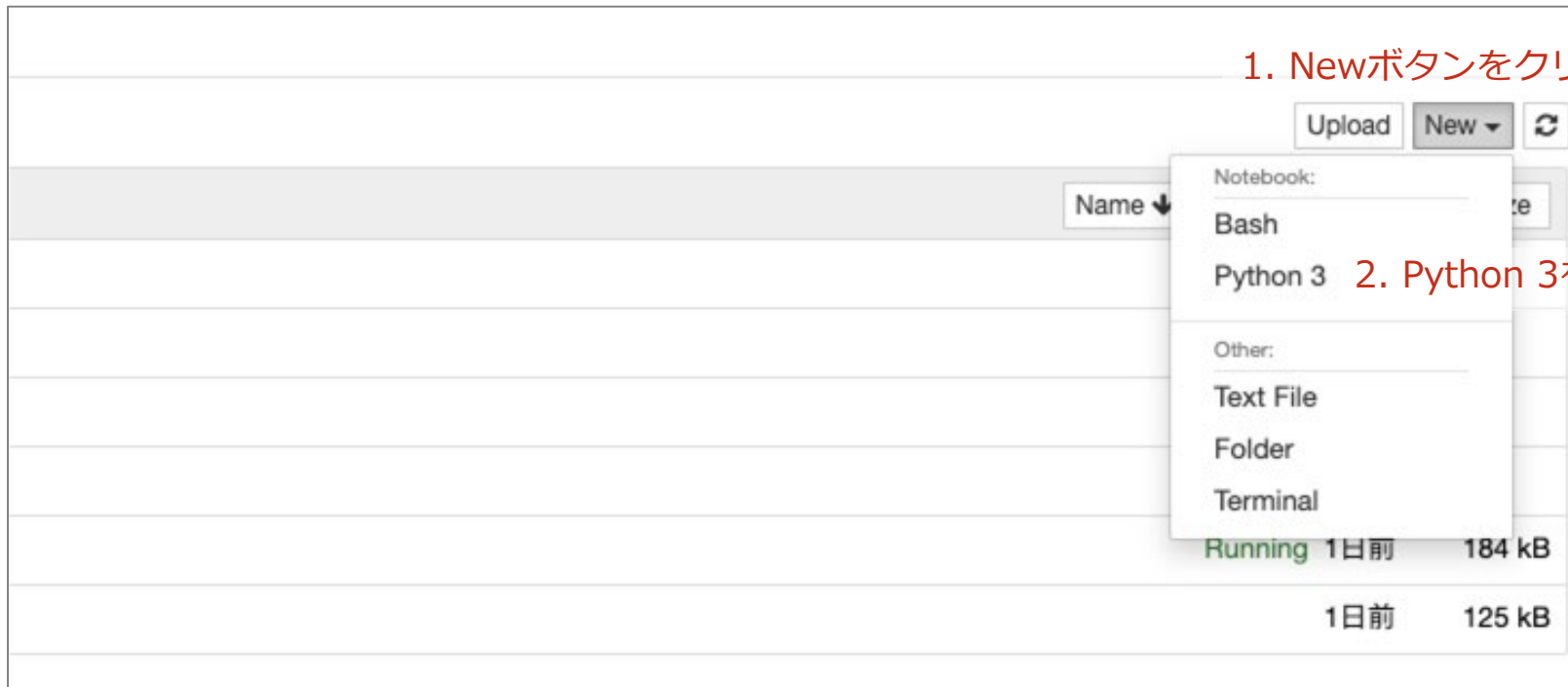
機能	記述例	表示例
ヘッダ1(大見出し)	# はじめに	<b>はじめに</b>
ヘッダ2(中見出し)	## 式の評価	<b>式の評価</b>
ヘッダ3(小見出し)	### 関数の値	<b>関数の値</b>
ボールド	**注意点**	<b>注意点</b>
イタリック	*Italic*	<i>Italic</i>
取り消し	--取り消し--	取り消し
リンク	[HP]( <a href="http://www.nii.ac.jp">www.nii.ac.jp</a> )	<a href="http://www.nii.ac.jp">HP</a>
リスト	- 項目1 - 項目2	• 項目1 • 項目2
番号付きリスト	1. 項目1 1. 項目2	1. 項目1 2. 項目2
引用	> 引用文 > 引用文	引用文 引用文

# 手順例

- ▶ Jupyter Notebookを新規作成
- ▶ Markdownセルを作成
- ▶ 見出し、本文の記述

# Jupyter Notebookを新規作成

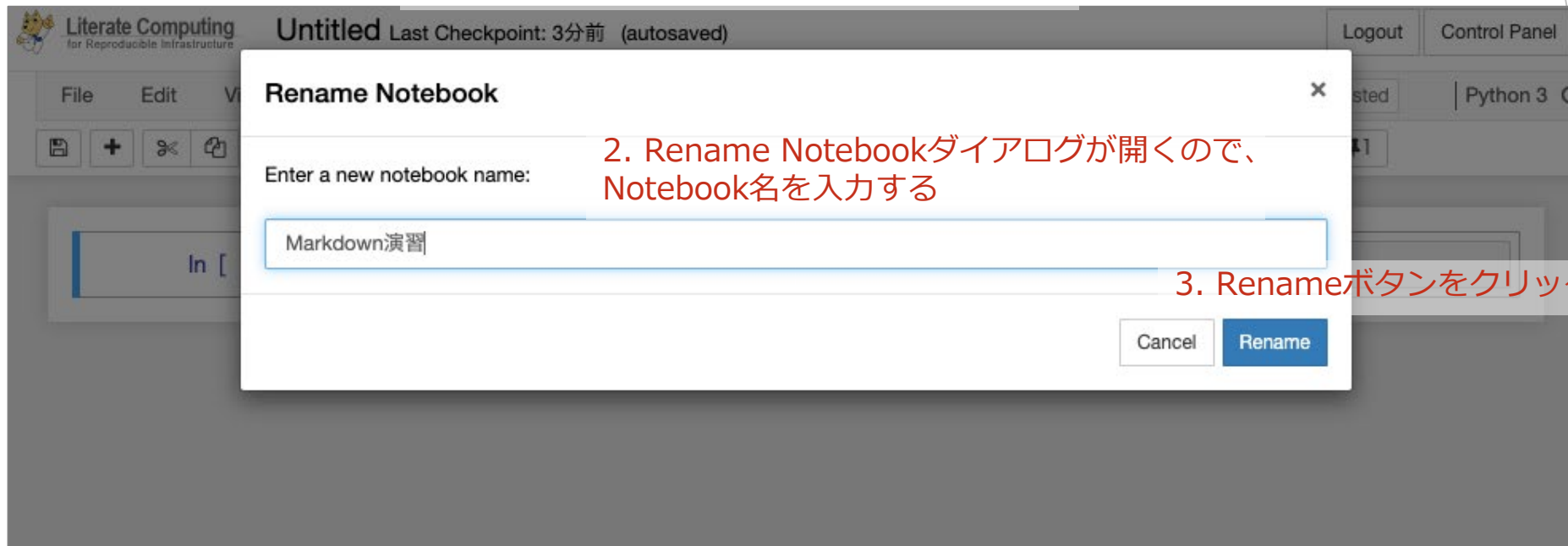
- ▶ 新規にNotebookを作成し、Markdownセルを作成してみましょう
- ▶ ファイル一覧の Newボタン > Python 3 を選択することで、新規Notebookを作成できます



# Jupyter Notebookを新規作成

- ▶ Webブラウザの別タブで *Untitled* という名前のNotebookが開きますので、Notebook名を変更しましょう
- ▶ Notebook名は「Markdown演習」としてください

1. Notebook名(Untitled) をクリックする

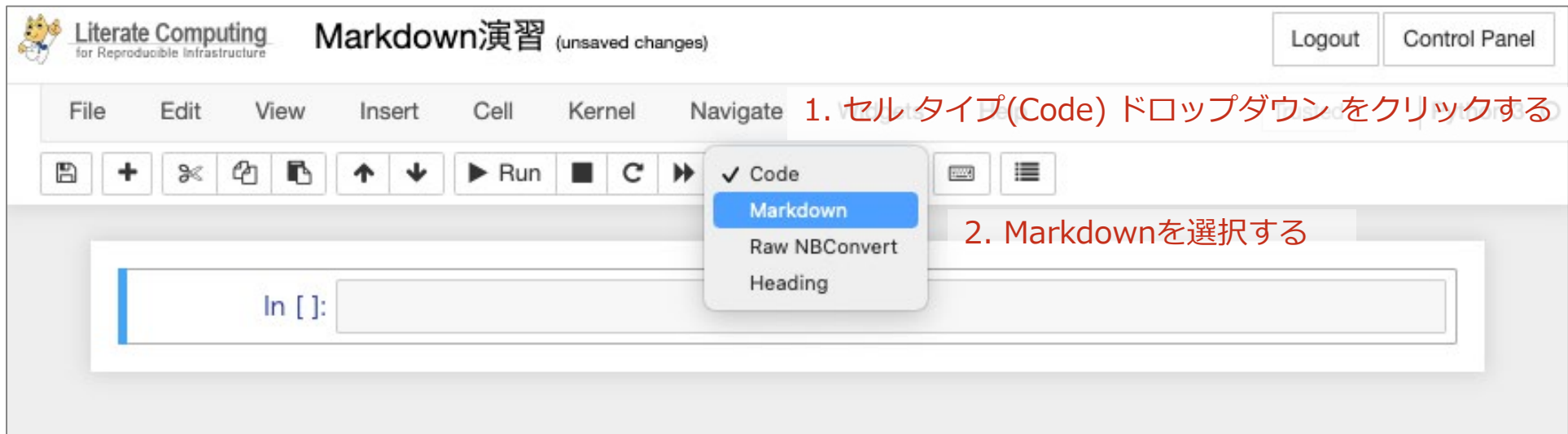


2. Rename Notebookダイアログが開くので、  
Notebook名を入力する

3. Renameボタンをクリックする

# マークダウンセルを作成

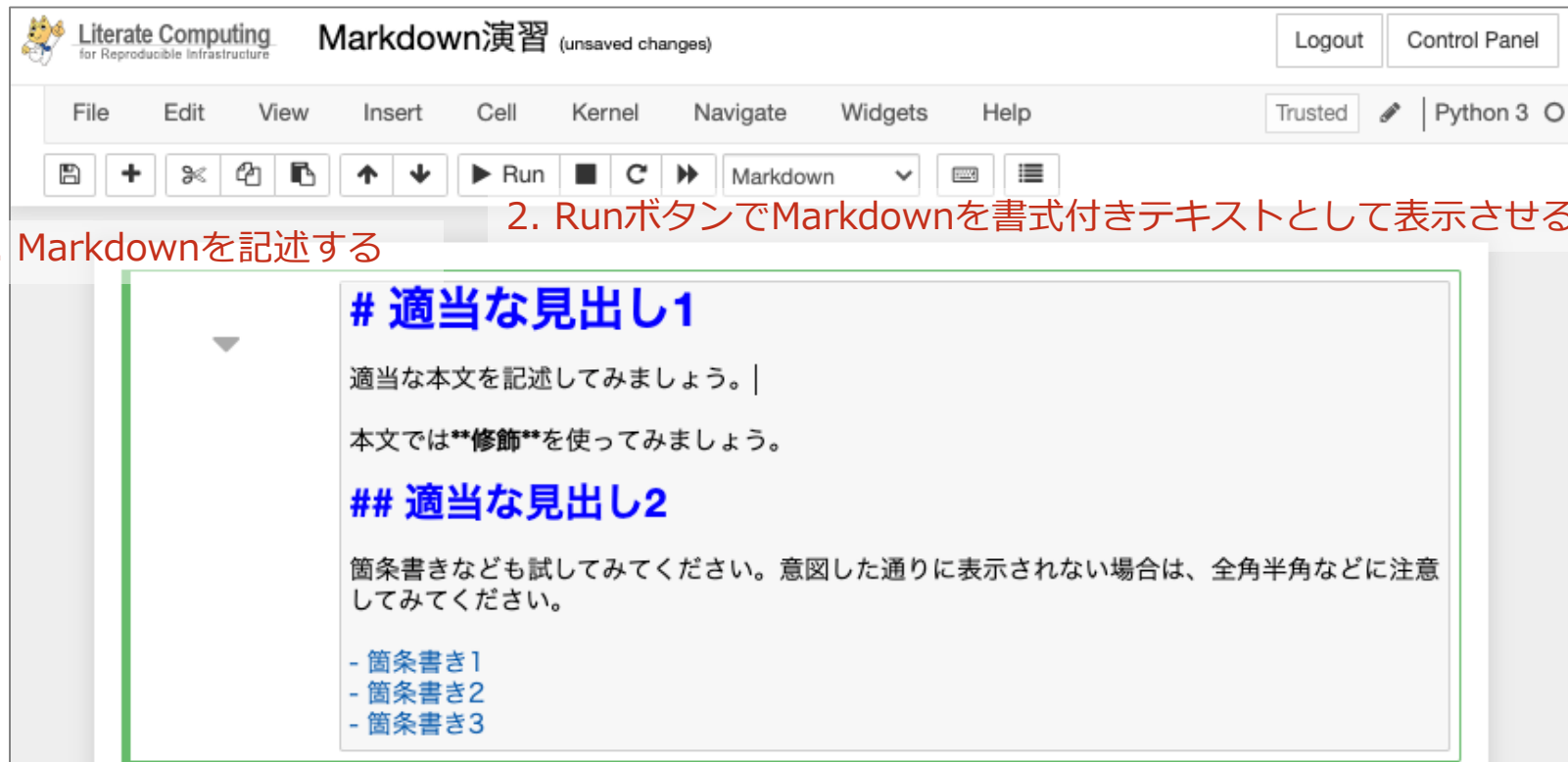
- ▶ 初期状態ではコードセルが1つだけあるので、これをマークダウンセルに変更してみましょう



The screenshot shows the JupyterLab interface for a notebook titled "Markdown演習 (unsaved changes)". The top navigation bar includes "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Navigate". A dropdown menu is open over the "Cell" menu, showing options: "Code" (selected with a checkmark), "Markdown", "Raw NBConvert", and "Heading". The main workspace contains a single code cell labeled "In []:". Red annotations are present: "1. セルタイプ(Code) ドロップダウンをクリックする" points to the dropdown menu, and "2. Markdownを選択する" points to the "Markdown" option.

# 見出し・本文の記述

- ▶ 自由にMarkdownを記述し、書式がどのように表示されるかを見てみましょう
- ▶ Runで書式付きテキストとして表示した後は、ダブルクリックで再編集できます



The screenshot shows the Literate Computing interface for a Jupyter Notebook. The title bar reads "Literate Computing for Reproducible Infrastructure Markdown演習 (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Navigate, Widgets, and Help. The toolbar contains icons for file operations, a Run button, and a dropdown menu currently set to "Markdown".

1. Markdownを記述する

2. RunボタンでMarkdownを書式付きテキストとして表示させる

The preview pane displays the rendered Markdown content:

```
# 適当な見出し1
適当な本文を記述してみましょう。|
本文では**修飾**を使ってみましょう。
## 適当な見出し2
箇条書きなども試してみてください。意図した通りに表示されない場合は、全角半角などに注意
してみてください。
- 箇条書き1
- 箇条書き2
- 箇条書き3
```

# (演習)Jupyter Notebook操作の基本操作 を体験-Notebookを編集

- ▶ 新規にNotebookを作成し、「Markdown演習」というタイトルを持つNotebookを作ってみましょう (10分)
- ▶ コンテンツは以下の条件を満たすように、自由に記述してください
  - ▶ 見出しを3つ以上
  - ▶ それぞれの見出しの配下に1段落以上の本文
  - ▶ 太字・斜体の文字を入れる

# Jupyter Notebookの運用への適用 - LC4RIについて



# ここまでで…

- ▶ インフラ構築・運用固有の操作に入る前に、Jupyter Notebookの操作を体験いただきました
- ▶ Jupyter Notebookをどのようにインフラ構築・運用に使うのかを紹介していきます

# 背景: NIIにおけるクラウド運用

- ▶ NIIでは 所内研究者向けに**ベアメタルプライベートクラウド**を運用
  - ▶ OpenStackを改造して使用 / 現在4世代目の環境の運用試行中
- ▶ ベアメタルクラウド上でのアプリケーション構築支援も
  - ▶ 例えば…
  - ▶ Hadoop
  - ▶ Kubernetes
  - ▶ CoursewareHub(JupyterHubベースの講義環境)
  - ▶ …
- ▶ **OSSベース: サポート窓口などない独力での運用**
  - ▶ 少ないメンバーでいかに効率的にインフラ運用をするか?
  - ▶ 開始時4名(現在5名), コードの開発・保守行為から逃げられない
  - ▶ メンバーのスキルセットの違い … できるだけ属人化を防ぎたい

# 背景: インフラ運用における高度化の例

- ▶ DevOps
  - ▶ 開発者と運用者の連携を緊密にすることで、開発からサービスデリバリーまでを迅速に行う
  - ▶ **人間同士の協働 x 自動化** … ではあるものの、開発されているツールは自動化に偏っており、**人間同士の協働を重視したツール**は少ない(\*)
  - ▶ (関連アプローチに関して)Infrastructure as Code(IaC) … 多くが学術論文ではなくインターネット記事等の**グレー文献(\*\*)**
- ▶ 人間同士の協働をしながら自動化の恩恵も受けるためには？

\* Leite, L., Rocha, C., Kon, F., Milojcic, D. and Meirelles, P.: A Survey of DevOps Concepts and Challenges, ACM Comput. Surv. 52, 6, Article 127, pp. 35 (2019).

\*\* Kumara, I., Garriga, M., Romeu, A.U., Nucci, D.D., Palomba, F., Tamburri, D.A., and van den Heuvel, W.-J.: The do's and don'ts of infrastructure code: A systematic gray literature review, Information and Software Technology, Volume 137, 106593 (2021).

# LC4RI: Literate Computing for Reproducible Infrastructure

- ▶ 基本アイデア

… Jupyter Notebook + Ansible



- ▶ 運用に関する**すべての作業**における

- ▶ 実行コード
- ▶ 実行結果
- ▶ 説明

- ▶ をNotebookとして、ひとまとめに

3.4.3.2 DataNodeの起動

```
[30]: !ansible-playbook -i {target_group} {playbook_dir}/start_datanode.yml --vault-password-file {work_dir}/.vault
```

PLAY [hadoop\_slavenode] \*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*

ok: [10.23.249.16]

TASK [start\_hadoop-hdfs-datanode] \*\*\*\*\*

changed: [10.23.249.16]

PLAY RECAP \*\*\*\*\*

ok=2 changed=1 unreachable=0 failed=0 skipped=0 ignored=0

コードセルによる  
コマンド実行指示

正常にSlave Nodeが起動できているか、dfsadmin reportを確認しておく。

```
[31]: !ansible -b --become-user hdfs -m shell -a 'kinit -k -t /etc/hadoop/conf/hdfs.keytab hdfs/${hostname} && \
hdfs dfsadmin -report' -i {target_cluster} hadoop_client --vault-password-file {work_dir}/.vault
```

```
10.23.249.31 | CHANGED | rc=0 >>
Configured Capacity: 9004849971200 (8.19 TB)
Present Capacity: 8584014228428 (7.81 TB)
DFS Remaining: 3520710734717 (3.20 TB)
DFS Used: 5063303493711 (4.61 TB)
DFS Used%: 58.99%
Replicated Blocks:
  Under replicated blocks: 4
  Blocks with corrupt replicas: 0
  Missing blocks: 0
  Missing blocks (with replication factor 1): 0
  Low redundancy blocks with highest priority to recover: 0
  Pending deletion blocks: 0
Erasure Coded Block Groups:
  Low redundancy block groups: 0
  Block groups with corrupt internal blocks: 0
  Missing block groups: 0
  Low redundancy blocks with highest priority to recover: 0
  Pending deletion blocks: 0
```

マークダウン  
セルによる  
見出し・説明

コードセルの実行結果

# LC4RIプラクティス(1): 試行錯誤の奨励

- ▶ 前提
  - ▶ 情報インフラは**常に変化する対象**である
- ▶ 基本的な考え方
  - ▶ 完全な手順 よりも **試行錯誤**: Human-in-the-loopを重視
  - ▶ 運用者のシステムへの理解を深め、効率的・柔軟な対応を実現
- ▶ 試行錯誤を促進するために
  - ▶ 同型のハードウェアを揃えることで、**失敗可能な環境を用意する**

# LC4RIプラクティス(2): 証跡の保存と運用者間の共有

- ▶ 前提
  - ▶ 情報インフラに対する個々の場面での対応は**それぞれ微妙に異なる**
- ▶ 基本的な考え方
  - ▶ 全ての場面でのJupyter Notebookの使用を奨励 … **過去の証跡を(できる限り)全て Notebookとして残す**
  - ▶ GUI等で操作する場合も**Jupyter Notebookにメモ・スクリーンキャプチャだけでも**保存する
  - ▶ 運用メンバーのJupyter Notebookは全て相互に共有
- ▶ 行為と(出力)結果がセットになった証跡を逐一蓄積することで…
  - ▶ **実際に行った行為と実際の結果ベースでのノウハウ共有**が可能になる
  - ▶ 過去の運用作業のセルを再利用しながら作業を進める

# 運用Notebookの例

- ▶ Hadoop DataNodeのディスク故障対応
- ▶ クラスタ全体の状態確認・操作
- ▶ ノードの状況・故障の内容によって(理想的な)対応は異なる
- ▶ クラスタの状況を理解するプロセスも書き下されることになる

Literate Computing for Reproducible Infrastructure 20200421\_02\_DataNode の修理対応 Last Checkpoint: 2020/04/21 (read only) Logout Control Panel

File Edit View Insert Cell Kernel Navigate Widgets Help Not Trusted Python 3.0

Contents

- 1 About: DataNodeの修理対応
  - 1.1 Operation Note
  - 1.2 Target Cluster
- 2 現状の確認
  - 2.1 HDFSクラスタ - 現状
  - 2.2 ディスクの状態確認
- 3 交換のためマシンをDecommission
  - 3.1 デコミッションしたいノード情報をg
  - 3.2 Hadoopのコンフィグファイルを更新
  - 3.3 デコミッション設定を反映させる
  - 3.4 デコミッションが行われたことを確認
- 4 後始末

## 2 現状の確認

### 2.1 HDFSクラスタ - 現状

念のため、dfsadminが提供するレポートを確認する。

```
!ansible --become --become-user hdfs -a 'hdfs dfsadmin -report' hadoop_namenode_primary -l {target_group}
```

```
Configured Capacity: 374028785090560 (340.18 TB)
Present Capacity: 355014096361215 (322.88 TB)
DFS Remaining: 295481996447160 (268.74 TB)
DFS Used: 59532099914055 (54.14 TB)
DFS Used%: 16.77%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-----
Live datanodes (12):
Name: [redacted]:1004 ([redacted])
Hostname: [redacted]
Decommission Status : Normal
Configured Capacity: 29528238325760 (26.86 TB)
DFS Used: 7937922090629 (7.22 TB)
Non DFS Used: 767453563 (731.90 MB)
```

死んではない。

以下、操作用にIPアドレスとホスト名を残しておく。

```
target_hostname = '[redacted]'
target_hostname
```

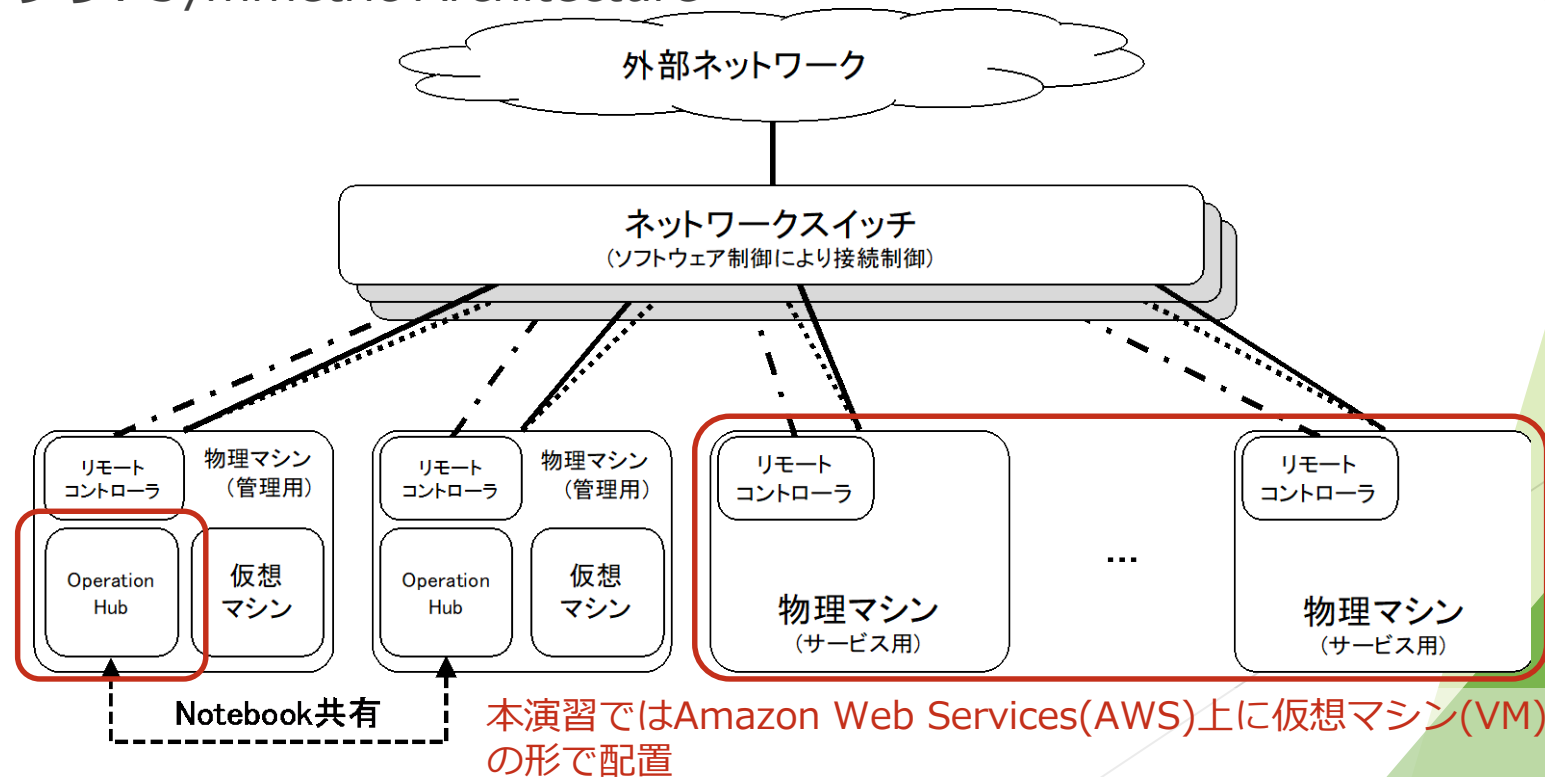
```
Out[5]: 'sn02031601'
```

```
%run scripts/loader.py
header, machines = read_machines("hosts.csv")
pd.DataFrame(map(lambda m: get_row(header, m), machines), columns=header)
```

Cluster	sn	ip	port	disk	dfs	non-dfs	total	used	used%	is-decommissioned	is-replicated	is-correct	is-live
4	Cluster1	sn0203	10.15.4.24	16	65536	10.0	15.0	64512.0	...	False	False	False	False
5	Cluster1	sn0203	10.15.4.20	16	65536	10.0	15.0	64512.0	...	False	False	False	False
6	Cluster1	sn0203	10.15.4.16	16	65536	10.0	15.0	64512.0	...	False	False	False	False
7	Cluster1	sn0203	10.15.4.14	16	65536	10.0	15.0	64512.0	...	False	False	False	False

# 実践環境

- ▶ 操作環境: OperationHub
- ▶ 操作対象インフラ: Symmetric Architecture





# Jupyter NotebookとLinux操作

- ▶ Jupyter Notebookから構築・運用操作をする = Linuxに対する操作 (Command Line呼び出し)をJupyter Notebookからやる
- ▶ 対象は Linux(Rocky Linux 8) ベースの環境 - この操作を体験

# Jupyter Notebookから Linuxコマンドを実行

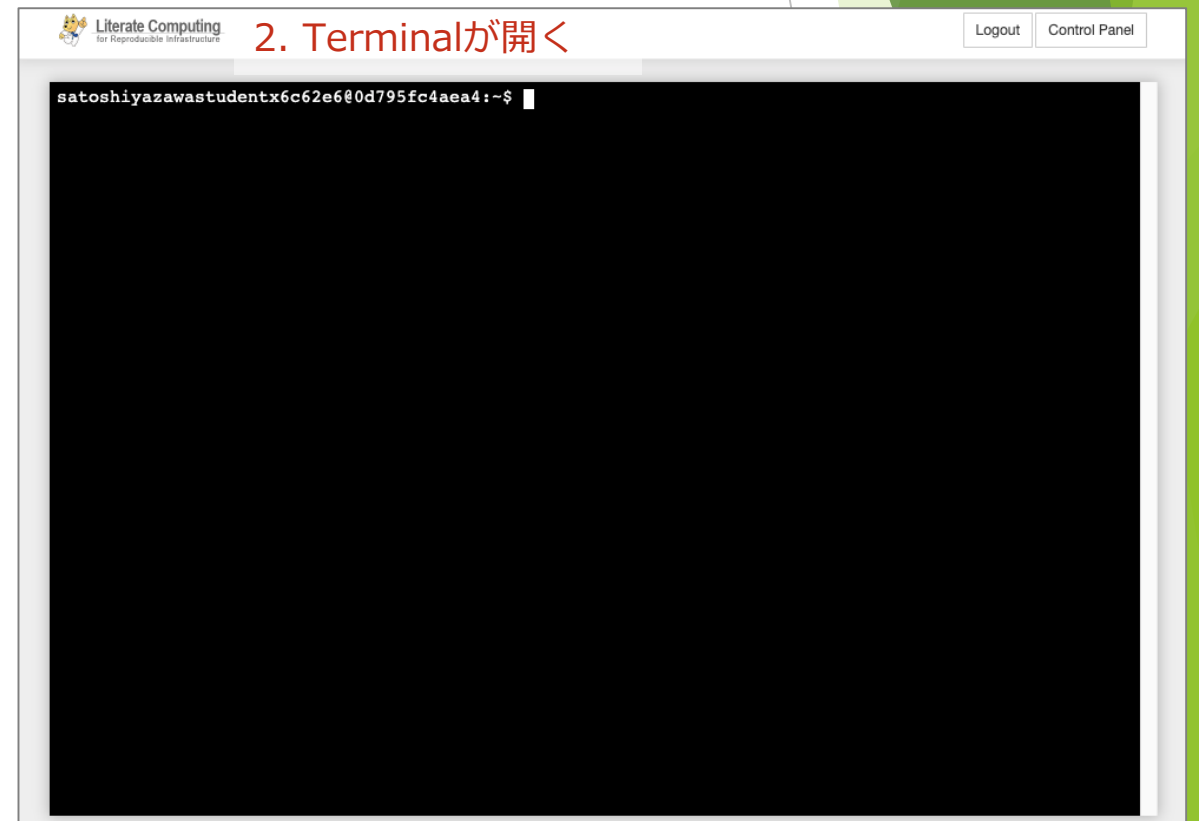
- ▶ Jupyter Notebookでは、コードセルの先頭に！を記載するとコマンド文字列を実行することができる
- ▶ {変数名} で変数展開される - iPythonのルール

```
In [8]: exercise_path = '/home/jupyter/textbook'  
!ls -la {exercise_path}
```

```
total 3180  
drwxrwxrwx. 2 root          root    4096 Sep 28 23:00 .  
drwxr-xr-x. 5 root          root     51 Sep 29 01:35 ..  
-r--r--r--. 1 satoشييازawateacherx20d122 users 159179 Sep 28 03:16 ensyu02a.ipynb  
-r--r--r--. 1 satoشييازawateacherx20d122 users 124724 Sep 28 03:16 ensyu03a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users 104955 Sep 28 03:16 ensyu04a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users 394781 Sep 28 03:16 ensyu05a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users 158767 Sep 28 03:16 ensyu06a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users 200027 Sep 28 03:16 ensyu07a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users 156068 Sep 28 03:16 ensyu08a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users  78998 Sep 28 03:16 ensyu09a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users 180661 Sep 28 03:16 ensyu10a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users  67104 Sep 28 03:16 ensyu11a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users  85770 Sep 28 03:16 ensyu12a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users  28803 Sep 28 03:16 ensyu12a.png  
-rw-r--r--. 1 satoشييازawateacherx20d122 users   7353 Sep 28 03:16 ensyu12b.png  
-rw-r--r--. 1 satoشييازawateacherx20d122 users  98450 Sep 28 03:16 ensyu13a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users 144020 Sep 28 03:17 ensyu14a.ipynb  
-rw-r--r--. 1 satoشييازawateacherx20d122 users  83070 Sep 28 03:17 ensyu15a.ipynb
```

# Jupyter Notebookが提供するTerminal

- ▶ Jupyter NotebookからTerminalを開き、コマンド操作をすることができる



# (演習) OperationHubから仮想マシンに接続してみよう

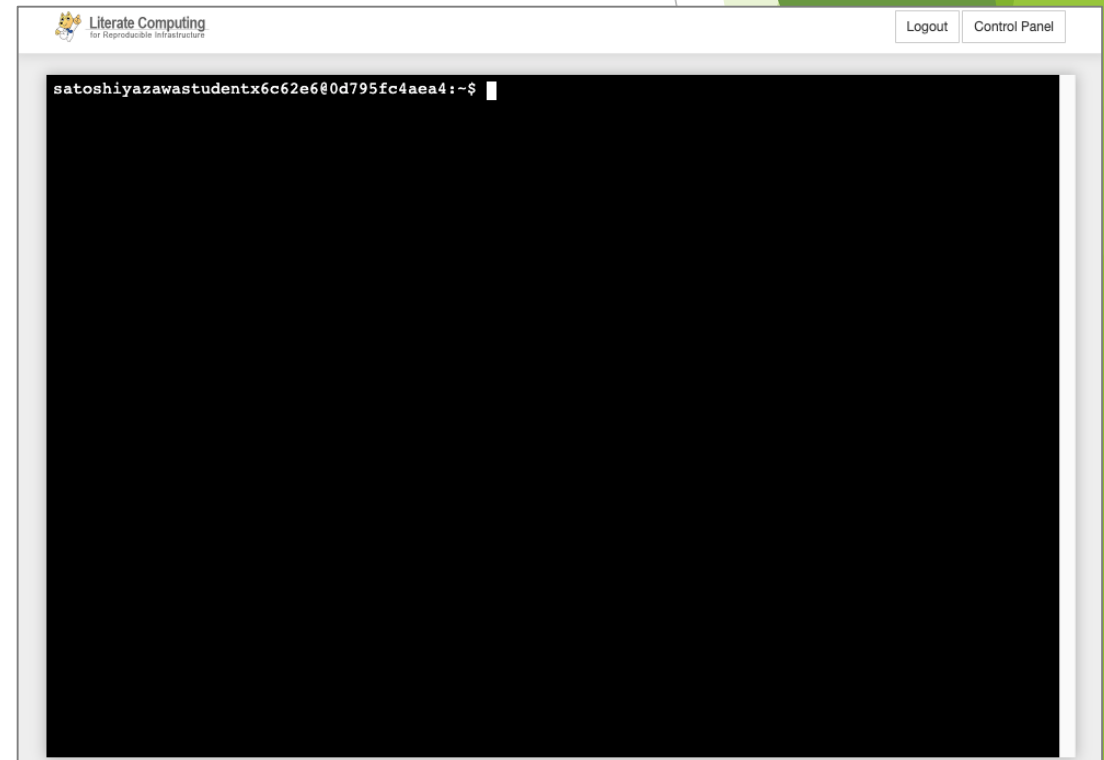
- ▶ Terminalを開き、受講者用のファイルを取得しましょう
- ▶ うまくいかない場合は、Terminalを開き直してみてください

```
# Notebookを格納したディレクトリに移動する
$ cd notebooks

# 自分用の教材ファイルをコピーする
$ cp users/ohadmin/resources/$(whoami)/* .

# ファイルの一覧を確認する
$ ls -la
```

- ▶ ファイルの一覧には、以下のファイルが含まれていることを確認してください
  - ▶ hosts
  - ▶ D00\_LC4RIをはじめの.ipynb 他4つのNotebook

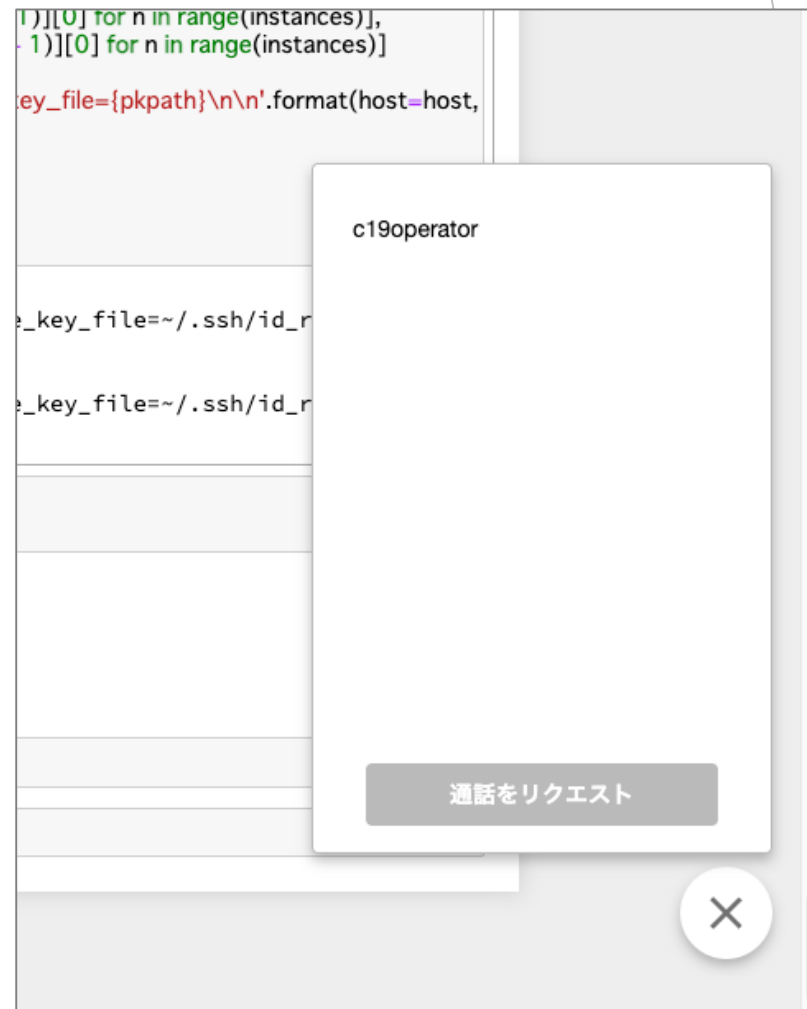
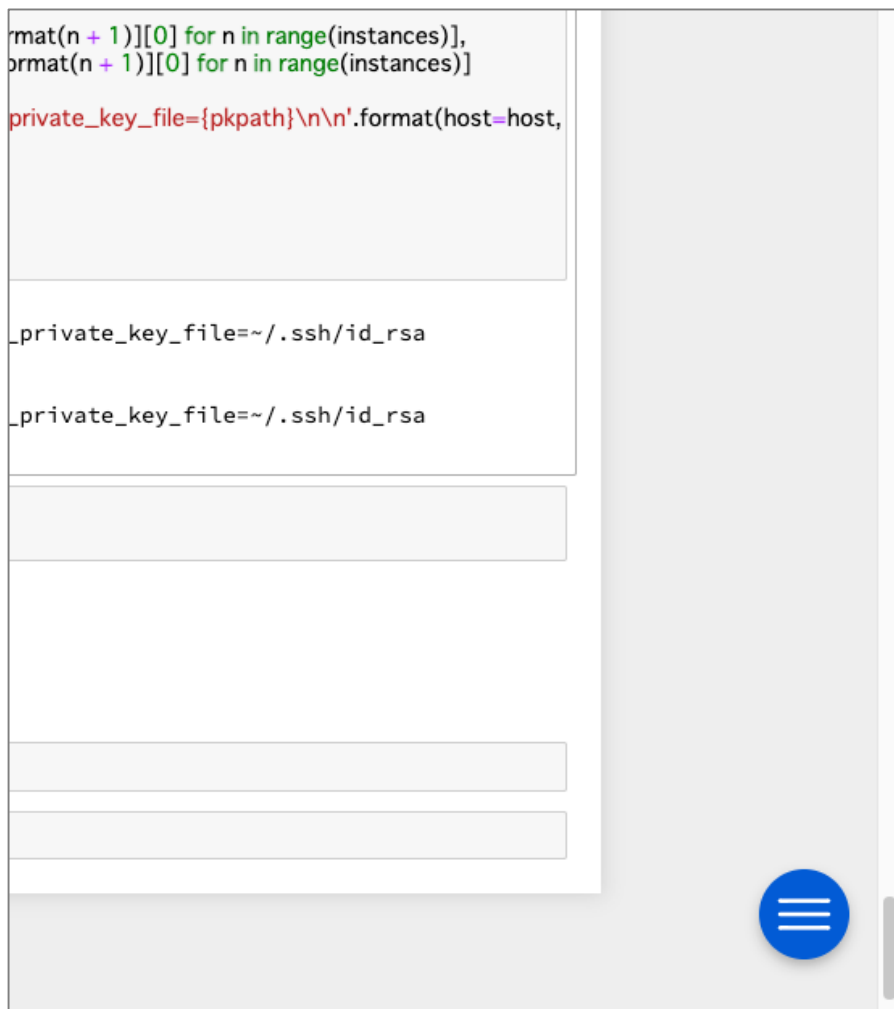


# (演習) OperationHubから仮想マシンに接続してみよう

- ▶ **D00\_LC4RIをはじめる** を実行してみましよう (20min)
- ▶ 各受講者向けに別々の独立した仮想マシン(VM)を用意してあります。
  - ▶ SSH秘密鍵のパスフレーズは、ユーザーに対応するパスワード(2つ)のうち、右のものをお使いください
- ▶ 以下の内容を含んでいます。
  - ▶ TerminalからSSH, SCP接続
  - ▶ NotebookからAnsibleコマンドで接続
  - ▶ LC4RI拡張の基本操作
  - ▶ パスワード入力
- ▶ 演習がありますので、適宜セルを変更しながら試してください

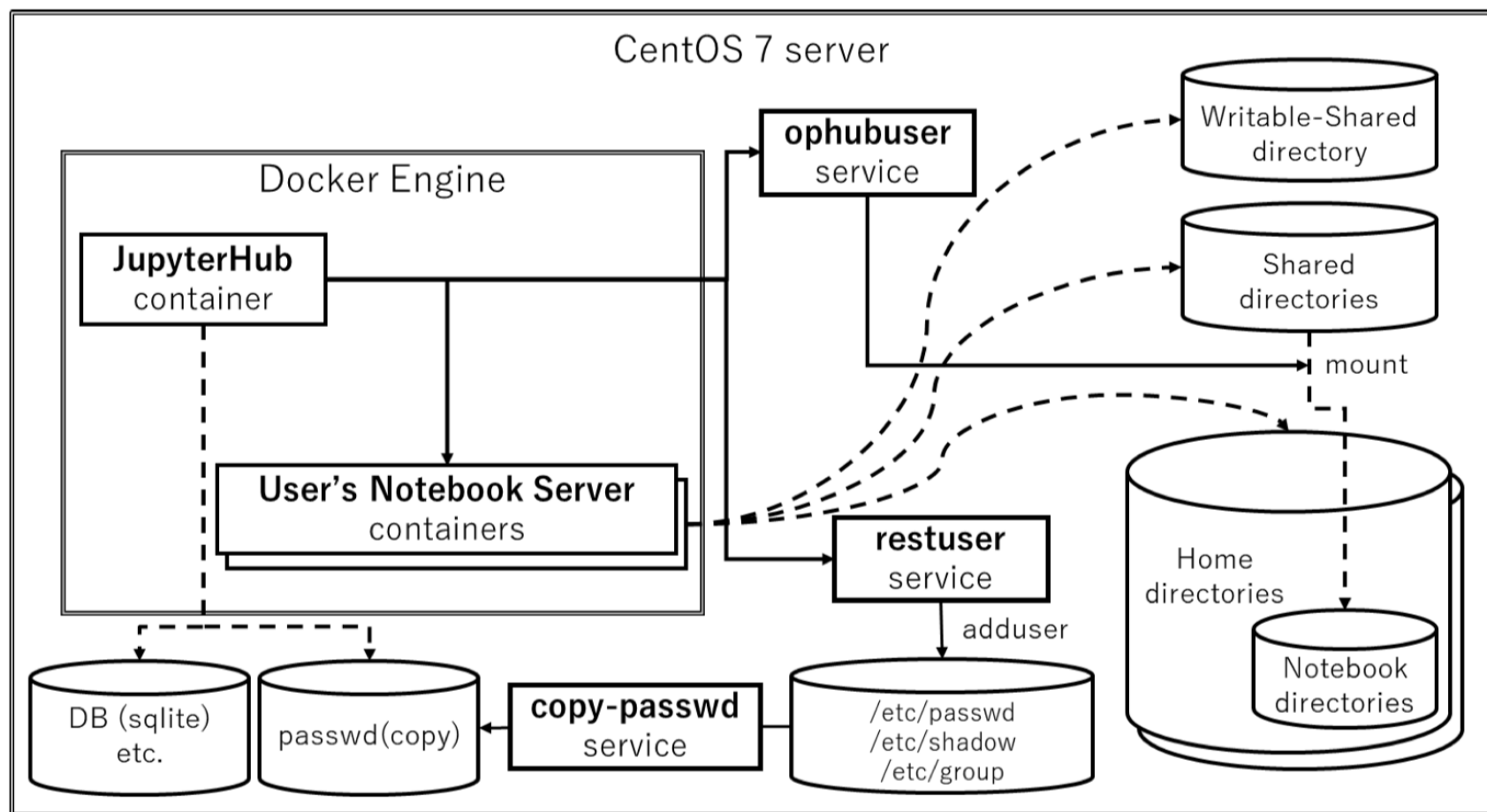
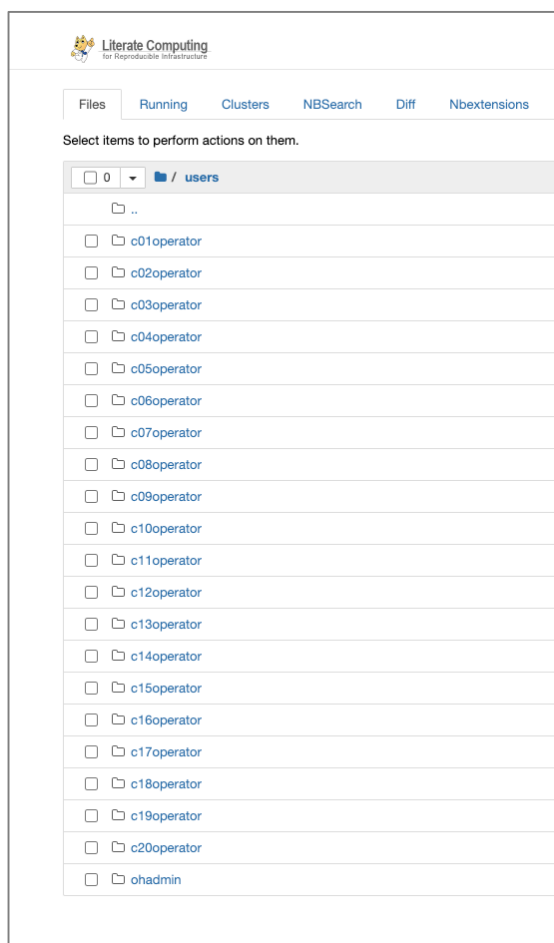
# 不明時の問い合わせ: NBWhisper

- ▶ 不明点がある場合は、個別通話もできます。c15operator- 以降が講師です
- ▶ 不明なときはぜひ通話リクエストをお願いします



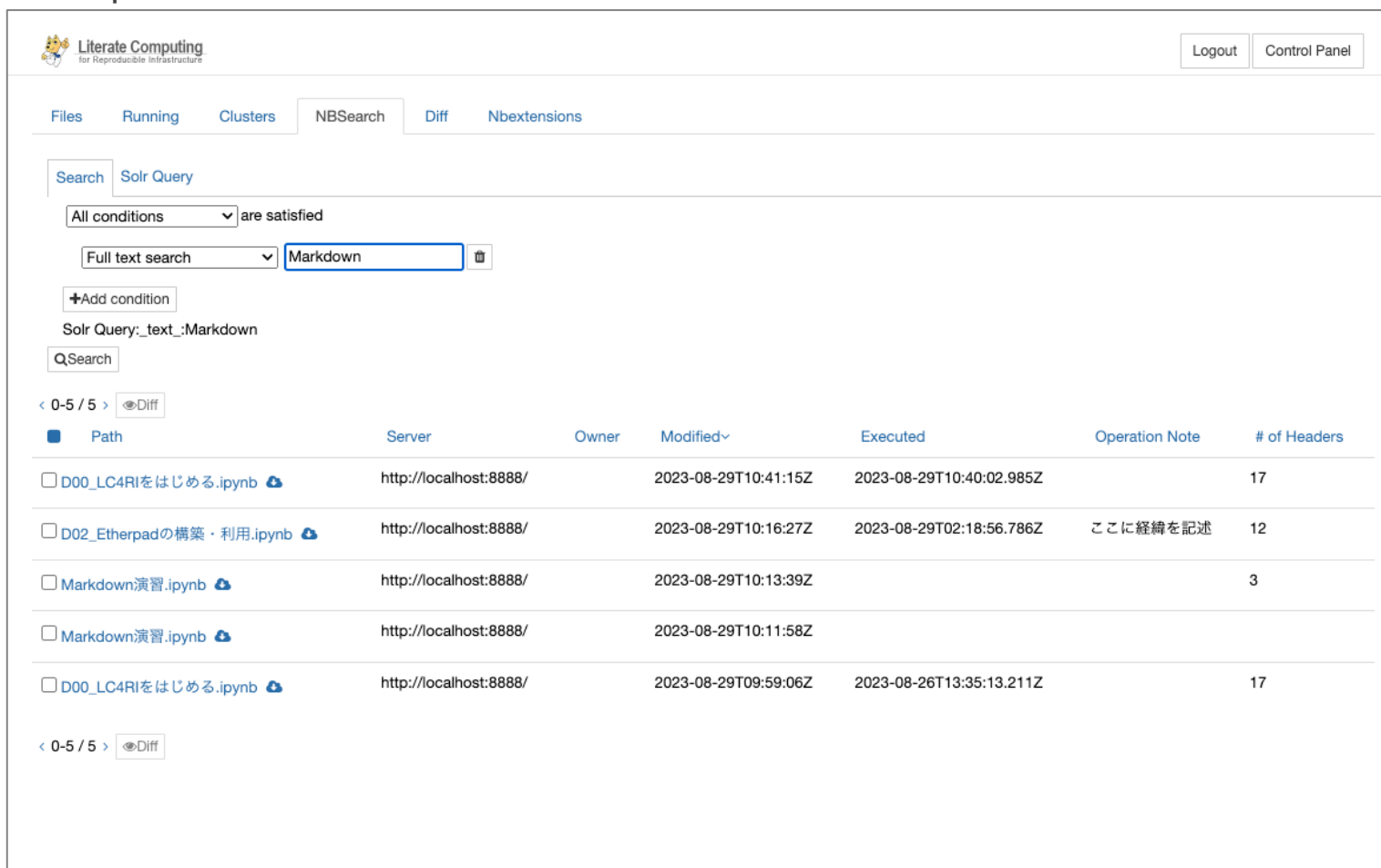
# 運用者間のデータ共有

- ▶ usersディレクトリ
  - ▶ 他のユーザーが作成したNotebookが全て見える: users/ohadminの中を覗いてみると…?



# 運用者間のデータ共有

- ▶ nbsearch
  - ▶ 同一OperationHub内のNotebookの検索エンジン



The screenshot displays the nbsearch web interface. At the top, it features the 'Literate Computing for Reproducible Infrastructure' logo and navigation links for 'Files', 'Running', 'Clusters', 'NBSearch', 'Diff', and 'Nbextensions'. The 'NBSearch' tab is active. Below the navigation, there is a search bar with a 'Search' button and a 'Solr Query' input field. The search criteria are set to 'All conditions are satisfied' and 'Full text search' for 'Markdown'. A '+Add condition' button is visible. The search results are displayed in a table with columns: Path, Server, Owner, Modified, Executed, Operation Note, and # of Headers. The table shows five entries, with the second entry having an 'Operation Note' that says 'ここに経緯を記述'.

Path	Server	Owner	Modified	Executed	Operation Note	# of Headers
<input type="checkbox"/> D00_LC4RIをはじめ.ipynb	http://localhost:8888/		2023-08-29T10:41:15Z	2023-08-29T10:40:02.985Z		17
<input type="checkbox"/> D02_Etherpadの構築・利用.ipynb	http://localhost:8888/		2023-08-29T10:16:27Z	2023-08-29T02:18:56.786Z	ここに経緯を記述	12
<input type="checkbox"/> Markdown演習.ipynb	http://localhost:8888/		2023-08-29T10:13:39Z			3
<input type="checkbox"/> Markdown演習.ipynb	http://localhost:8888/		2023-08-29T10:11:58Z			
<input type="checkbox"/> D00_LC4RIをはじめ.ipynb	http://localhost:8888/		2023-08-29T09:59:06Z	2023-08-26T13:35:13.211Z		17



# OperationHubの管理

- ▶ JupyterHub管理権限 – 受講者全員付与しています … 誤操作注意
  - ▶ 右上Control Panel → Admin
  - ▶ 全メンバーのサーバーを管理可能

The screenshot shows the JupyterHub Admin interface. At the top left is the logo for 'Literate Computing for Reproducible Infrastructure'. The navigation bar includes 'Home', 'Token', 'Admin', and 'Services'. The user 'ohadmin' is logged in, with a 'Logout' button. A search bar for users is present, along with a link to '> Manage Groups'. Below this is a table with columns: 'User', 'Admin', 'Server', 'Last Activity', 'Running', and 'Actions'. The table lists several users, including 'ohadmin' and 'c01operator' through 'c08operator'. Each user row has a dropdown arrow on the left and specific action buttons in the 'Running' and 'Actions' columns.

User	Admin	Server	Last Activity	Running	Actions
Add Users			Start All	Stop All	Shutdown Hub
ohadmin	admin	[MAIN]	3 minutes ago	Stop Server Access Server	Edit User
c01operator	admin	[MAIN]	Never	Start Server Spawn Page	Edit User
c02operator	admin	[MAIN]	Never	Start Server Spawn Page	Edit User
c03operator	admin	[MAIN]	Never	Start Server Spawn Page	Edit User
c04operator	admin	[MAIN]	Never	Start Server Spawn Page	Edit User
c05operator	admin	[MAIN]	Never	Start Server Spawn Page	Edit User
c06operator	admin	[MAIN]	Never	Start Server Spawn Page	Edit User
c07operator	admin	[MAIN]	Never	Start Server Spawn Page	Edit User
c08operator	admin	[MAIN]	Never	Start Server Spawn Page	Edit User

# LC4RIを実践してみようー MinIOを 構築・運用

# MinIOとは

- ▶ 高性能・軽量なオブジェクトストレージ
- ▶ Amazon S3(Simple Storage Service)と互換性のあるAPIを持つ
- ▶ 単一ノードでも、複数ノードでも配備可能
- ▶ <https://min.io/docs/minio/linux/operations/concepts/architecture.html>

# (演習) vm1仮想マシンにMinIOを構築し、 利用してみよう

- ▶ **D01\_MinIOの構築・利用** を実行してみましよう (25min)
- ▶ 受講者向けに用意したVM 2台のうち、**vm1** に構築してください
  - ▶ 誤ってvm2に構築してしまった場合は教えてください - vm1, vm2の内容をリセットします
- ▶ 以下の内容を含んでいます。
  - ▶ MinIOのインストール ... パッケージのダウンロード・インストール・サービスの起動
  - ▶ MinIOの利用 ... MinIO Consoleの利用・MinIO Clientの利用
- ▶ 演習がありますので、適宜セルを変更しながら試してください

# MinIO Console

- ▶ インストール手順を実施すると、MinIO Consoleを利用できます
- ▶ ここにファイルをアップロードしたり、ダウンロードしたりして動作を確認してみてください

LC4RIで実際に環境を作ってみよう  
– Etherpadを構築・運用

# Etherpadとは

- ▶ リアルタイムオンラインテキストエディタ
  - ▶ <https://etherpad.org/>
- ▶ インストールの容易さ - シンプルなシステム構成
- ▶ 利用の手軽さ - シンプルな機能

# (演習) vm2仮想マシンにEtherpadを構築し、利用してみよう

- ▶ **D02\_Etherpadの構築・利用** を実行してみましよう (50min)
- ▶ 受講者向けに用意したVM 2台のうち、**vm2** に構築してください
  - ▶ 誤ってvm1に構築してしまった場合は教えてください - vm1, vm2の内容をリセットします
- ▶ 以下の内容を含んでいます。
  - ▶ Docker Engineのインストール
  - ▶ Etherpadのインストール … インストールしたら、Jupyter Notebookの記載に従ってURLを送付ください
  - ▶ Etherpadの設定変更 (余裕がある場合/任意)
- ▶ 演習がありますので、適宜セルを変更しながら試してください



# LC4RIの実践 のまとめ

- ▶ OperationHubによる Jupyter Notebook を使ったシステム構築・運用 (LC4RI)を実際に体験いただきました
- ▶ VMの準備、SSHキーを準備するところは講師側で行いましたが、SSH接続可能なマシンができたとして、LC4RIではソフトウェアの構築・利用をどのように行うべきか、その一端をご理解いただけたのではないかと思います

# お疲れさまでした

- ▶ Jupyter Notebookでの構築はいかがでしたでしょうか
- ▶ 実は、今回の演習環境の構築もLC4RIで行っています … そのご紹介
  - ▶ 解析基盤での起動
  - ▶ パブリッククラウドサービスとの連携
- ▶ 1年に1回のオペレーションでも、確実に実施できた(はず)
- ▶ OperationHubの導入は <https://github.com/NII-cloud-operation/OperationHub> をご参照ください！
  - ▶ ぜひご興味あればお問い合わせください！構築サポートもご相談可能です
  - ▶ スライド最後に付録として起動方法動画を掲載しています