metabase Manual
Tutorial

ver.1.1
connectome.design Inc.

Revision history

2022.03	ver.1.0	First edition created.
2022.05	ver.1.1	Changed sample DEK from “boston” to “california”.

metabase tutorial [Basic operation procedure]

- Access metabase with a browser and log in.
- Automatically log out after a certain period of time without any operation.

Outline procedure for running the program

1. Join the lab and log in.
2. Create a project.
3. Get DEK from Marketplace.
4. Create a task by choosing the DEK and VM type.
5. Launch the task.
6. Open JupyterLab and run the program.
7. Stop the task.

   * The detailed procedure is shown below.

Creating a project

1. Global menu [Lab] → Select from the lab list.
   a. Only participating labs are displayed.
2. Create a new project from [+ Create Project].
   a. Project name: Free. A name that identifies you as having made it is desirable.
   b. Project description: Optional.
   c. Member: You can leave it as it is because you are the default administrator.
3. [Registration].

An example of getting the program DEK from the Marketplace, loading it into a task, and executing it

1. Getting the DEK.
    a. Go to [Marketplace] from the global menu (the whole menu at the top of the screen).
    b. Select one from the program DEK to display the details screen. (Example: "california_scikit-learn")
    c. [Obtain] → The source setting opens.
        i. Retailer Lab / AI Guild:*the lab for your project*Select
        ii. Retailer Project:*the project you created*Select
        iii. Saved DEK Name: Leave the default.
    d. [Obtain]-> [Go ToRetailer]-> You can confirm that you have obtained DEK in the project.

2. Creating a task.
    a. Go to the Tasks tab.
    b. Create a new task from [+ Create Task].
        i. Task name: Free. A name that describes the function of the task is desirable.
        ii. Task description: Optional.
        iii. Select DEK.
        (Example: "Program DEK: california_scikit-learn")
        iv. VM type: Select the one with [Recommended].
        v. Maximum uptime: Leave the default (3600).
    c. [Registration].

3. Invoking a task.
    a. Go to the Tasks tab.
    b. Select the task created earlier from the task name list, and select [Start Up] from the [Operation] button.
    c. The startup confirmation screen opens, so check the contents and click [Start Up].
    d. The status will be "Starting", so just wait.
    (Depending on the usage status of AWS, it may take about 5 minutes to start.) When the
    e. status becomes "Running", the startup is completed.
    f. Select [Open] from the [Operation] button.
    g. Jupyter Lab opens in a separate browser tab.

4. Program execution.
    a. When JupyterLab opens, you can see that the loaded DEK is located in the [source] folder.

b. Extract the zip file in the [source] folder.
    i. See "How to run the context menu for metabase in JupyterLab" below.
c. If "requirements.txt" exists in the extracted folder, install additional libraries as needed.
    i. See "How to run the context menu for metabase in JupyterLab" below.
d. Move the extracted folder to the [work] folder and access the [work] folder.
    i. See "How to run the context menu for metabase in JupyterLab" below.
    ii. If you do not need to save the execution result, you can execute it from the [source] folder without moving it.
e. Move to the program folder.
(Example: "california_scikit-learn")
f. Open the program file.
(Example: "california_scikit-learn.ipynb")
    i. When the Kernel selection screen is displayed, select "Python 3".
    ii. The source code opens on the Jupyter Notebook.
g. Run line by line with the Run button.
    i. If an import error (ModuleNotFoundError) occurs during execution, install additional libraries.
    ii. See "How to run the context menu for metabase in JupyterLab" below.
    iii. Rerun the steps once the additional libraries are installed.
h. If the execution result file is output, it will be saved in the [work] folder.

Procedure to stop the task

1. When you have finished executing, select [Close and Shutdown] from the File menu to close the source code.
    a. If you close it by any other operation, the process will remain running and the memory will not be released.
    b. Check the running process from the tab on the left side of the screen, and if unnecessary Kernel remains, Shut down.
    [Running Terminals and Kernels]

2. Return to the metabase task screen and select [Stop] from the [Operation] button.
    (You may stop the source code while it is open without performing Shutdown.)

3. The files under the [work] folder will be persisted even after the task is stopped, and will be restored the next time the task is started.
    (However, the empty folder is not saved.)

4. You can change the source code on the screen and save it in the [work] folder.

5. The initial value of task uptime is 3600 seconds.
    a. Task uptime can be extended by 1 hour with the [+1 hour] button on the upper right of Jupyter Lab.

6. Don't forget to stop after checking the operation.
    a. If it does not stop, a charge will be incurred for the set operation time.

Example of registering and running your own program as DEK

1. local PC, put together the set of programs you want to register in one folder.
    a. Create a folder for storing programs (example: "helloworld") and put the program files in it (example: "helloworld.ipynb").
    b. If there is a library that needs additional installation , create "requirements.txt" and place it at the top of the folder.
2. Zip the folder.
   (Example: “helloworld.zip”)
3. Log in to metabase and open your own project.
4. Open the [DEK] tab and select [+ Create DEK].
5. Set the necessary items such as the DEK name, upload the created zip file, and [Registration].
    a. For the default setting at the time of listing, select [Price: Free] or [Public setting: Public].
6. When you open the [DEK] tab, the created DEK will be displayed in the list.
7. Go to the [Tasks] tab and select [+ Create Task] to create a new task.
    a. Task name: Free. A "name that understands the function of the task" is desirable.
    b. Task description: Optional.
    c. DEK: Select the DEK registered earlier.
    d. VM type: Select the one with [Recommended].
    e. Maximum uptime: Leave the default (3600).
8. [Registration].
9. Subsequent operations are the same as the above-mentioned "Invoking a task".

Example of creating and running a program on metabase from scratch

1. Select or create a new project.
2. If there is data to be used, register it as data DEK.
    a. As with the program DEK, zip it into one folder and register it as a DEK.
    b. You can also get the data DEK from the Marketplace.

3. Create a task.
    a. If necessary, specify the data DEK.

4. Launch the task and open Jupyter Lab.
    a. The data DEK will be placed in the [source] folder, so unzip the zip file and move it to the [work] folder.

5. Create a new folder in the [work] folder, open Jupyter Notebook and create a program.
    a. Use the [work] folder so that the files created when the task is stopped will be persistent.
    b. Create one folder to save the program and place the program under it.
    c. If you want to register the created program as DEK, archive it, download it, and register it separately.
        i. See "How to run the context menu for metabase in JupyterLab" below.

How to execute the context menu for metabase in JupyterLab

- To facilitate the operation in JupyterLab, commands are prepared in the context menu (right-click menu).

1. **Unzip**: Zip.
2. **Install Dependencies**: Library installation from “requirements.txt”.
3. **Move To Work**: Move to the [work] folder.
4. **Archive For DEK**: Create a zip file for DEK from the program folder.

*Detailed usage is shown below.

1. **Unzip**: Unzip the zip file.
    a. Open the [source] folder from the folder list on the left side of the JupyterLab screen.
    b. DEK zip file is located, select the context menu [Unzip] and execute. (Example: "dek-51f82e15 ****. Zip")
    c. It can be confirmed that the DEK folder has been expanded in the same folder.

2. **Install Dependencies**: Read "requirements.txt" and install additional libraries.
    a. Search for “requirements.txt” in the file list on the left side of the JupyterLab screen.
    b. Select “requirements.txt” and execute [Install Dependencies] from the context menu.
    (* Just select the file. It will not open.)
    c. The library will be imported.
    * This command must be executed each time the task is started.

3. **Move To Work**: Move to the [work] folder.
    a. After Unzip,execute and select the expanded folder from the context menu
    MoveTo Work] [.
    (* Just select the folder. It will not open.)
    b. You can confirm that the specified folder has been moved to the [work] folder.

4. **Archive For DEK**: Create a zip file for DEK from the program folder.
    a. Select the folder where the created program is placed from the folder list on the left side of the JupyterLab screen,
    and execute [Archive For DEK] from the context menu.
    (* Just select the folder. Do not open)
    b. You can confirm that the archive zip file was created from the selected folder.
    c. The created zip file can be downloaded.

- When the command is executed, the Terminal screen opens, and when the process is completed, "Complete! Please press Enter to exit the terminal." Is displayed. Press the Enter key to close the Terminal screen.

- If a message (such as permission to overwrite an existing file) is displayed on the Terminal screen, check the contents and enter the instructions.

- "requirements.txt", do either of the following.
    - On the Jupyter Notebook, add the "! Pip install" step at the top and execute it.
    - Open Terminal and execute the "pip install" command.

- In either case, you can run it in either the [source / work] folder, but you have to run it every time you start the task.

- The downloaded archive zip file contains “requirements.txt” that has all the libraries included in the operating environment written out, so unzip the zip file in the local environment and select “requirements.txt” in the folder. Organize the contents of the above into "only the minimum required libraries added by yourself". The file that the folder is zipped again can be registered as a new DEK.