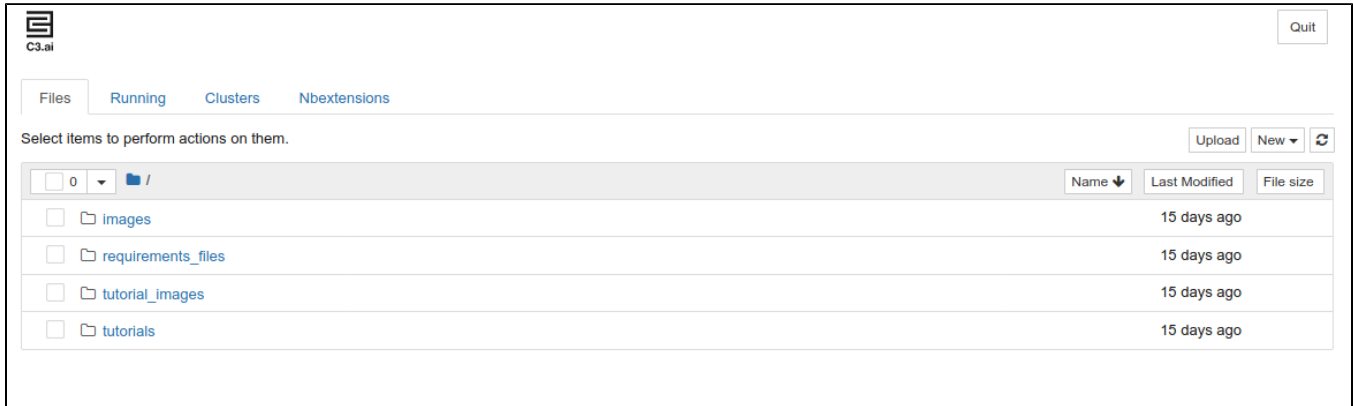


C3 AI Integrated Jupyter Notebook

For scientists wanting to jump right into using C3 AI Suite through Python or R, the integrated Jupyter notebook is a great option. Once you get your C3 AI environment tenant/tag (and Vanity Url), you can easily access the integrated Jupyter Notebook by navigating to `<vanity_url>/jupyter`.

When you access the Jupyter path, C3 AI will spin up a new container containing a Jupyter instance. The first time you do this, it will take a few minutes for this provisioning to be completed behind the scenes. Once completed, you'll arrive at a Jupyter directory!

Here is some additional guidance for setting up the Python Runtime Action: [DTI Guide to Python Action Runtimes in the C3 AI Suite](#).



There are a couple of options to interact with the Jupyter notebooks:

- Create a new notebook: Select 'New' in the upper right and create a new notebook. For now you can only create a new Python3 notebook.
- Open an existing tutorial notebook: Navigate to the 'tutorials' directory in which many C3 AI Data Science-centric tutorials are available.
- Upload an existing notebook: Press the 'Upload' button and select either a notebook of your own or one mentioned on the DTI training pages. (Be aware that the ``c3`` object will already be available in this case so you will have to comment out cells which establish this object.)

Creating new notebooks and review a tutorial

This notebook comes already connected to the package currently provisioned on your tenant/tag and so, for example, if you have the Covid-19 Data Lake provisioned you'll be able to do the following with no preparation:

Simply navigate to the ``tutorials`` directory and select the tutorial you want to use. These tutorial notebooks should work out of the box. **The C3.ai DTI highly recommends going through the ``TutorialIntroJupyterNotebook.ipynb`` notebook.** This notebook provides a great foundation for working with C3 AI Suite through Jupyter.

C3.ai

TutorialIntroJupyterNotebook (read only)

FileEditViewInsertCellKernelWidgetsHelp

Not TrustedPython 3

RAM Usage: 646.1 MB / 4.0 GBLoad Avg: 0.14 0.14 0.10

▼

Tutorial: Jupyter Notebook using C3 Python SDK

This tutorial discusses the basics of using C3 Notebook.

You will learn how to:

1. Utilize **C3 Python SDK** in conjunction with **C3 Notebook** to develop software on an externally persisted Jupyter Notebook.
2. Use the **help** function to learn more about C3 types and methods.
3. Manage your notebook **kernel**.
4. Utilize **file operations** to upload, download, and create files that will persist externally.

Some additional advanced topics at the end:

1. View, create, and use **action runtimes**.
2. Use advanced legacy **file operations**.

▼

Setup

This section will walk you through the basic setup and verification steps.

First, let us verify that we are working with Python 3. To take full advantage of C3 Type system, we strongly recommend that you use Python 3 or later versions.

In [1]:

```
# NotVerify: result
import sys
sys.version
```

executed in 31ms, finished 20:33:50 2020-02-18

Out[1]: '3.7.3 (default, Mar 27 2019, 22:11:17) \n[GCC 7.3.0]'

The Notebook startup hook will automatically connect to your C3 environment, download the Python SDK, and instantiate a `c3` variable that contains the connection to your environment.

Let us verify that there exists a `c3` variable and that it represents the type system:

In [2]:

```
# NotVerify: result
c3
```

executed in 194ms, finished 20:33:50 2020-02-18

Out[2]: c3.TypeSystemBase(connection=ServerConnection(url='http://c3server:8080', auth='XXXX', tenant='notebook', tag='prod'))

Now that we have done a few sanity checks on the system setup, let us move on to the basics!

Upload a new notebook

Finally, you can upload a new notebook from your local device by using the 'Upload' button.