Introduction to Elyra: AI-centric extensions to JupyterLab

Yiwen Li
Developer Advocate
IBM CODAIT
Center for Open Source Data and AI Technologies

yiwen.Li@ibm.com | Linkedin: https://www.linkedin.com/in/yiwen-li-47a019119

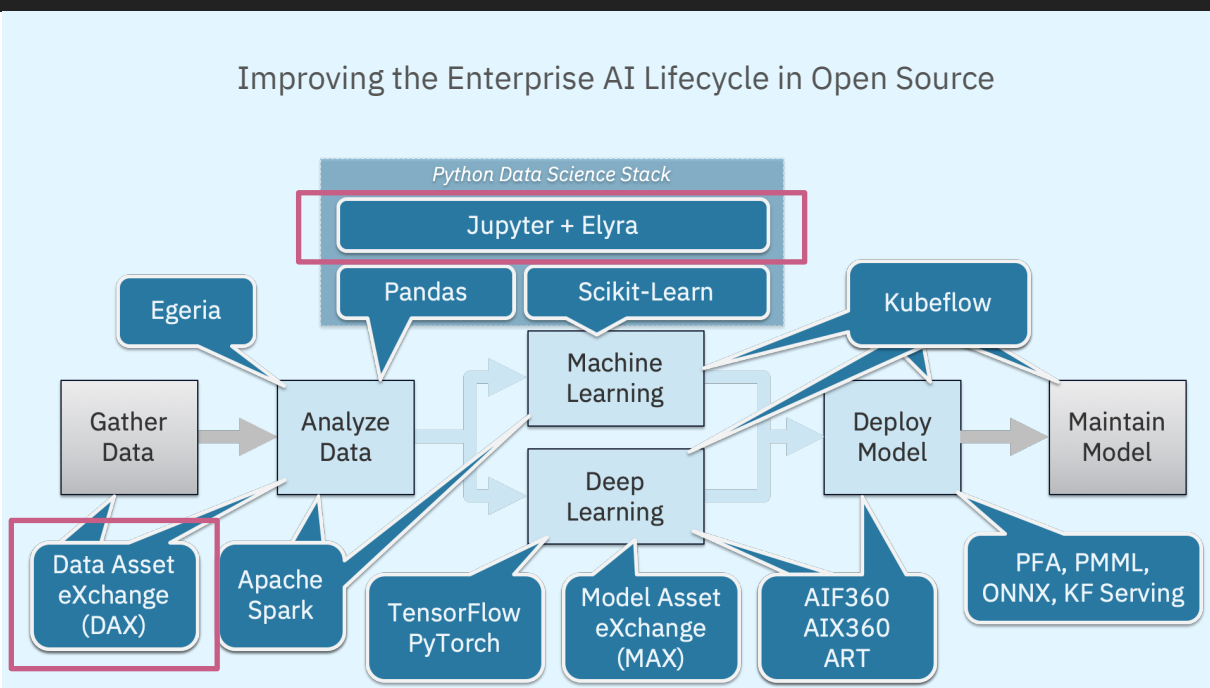
Center for Open Source Data & AI Technologies

CODAIT

Open Source @ IBM

codait.org

- CODAIT aims to make AI solutions dramatically easier to create, deploy, and manage in the enterprise.
- We contribute to and advocate for the open-source technologies that are foundational to IBM's AI offerings.
- 30+ open-source developers!



http://bit.ly/0901-ELYRA

What is Elyra?

And, why do we need Elyra?

Data Science Pipeline

Creating notebook pipelines using Elyra and KubeFlow pipelines

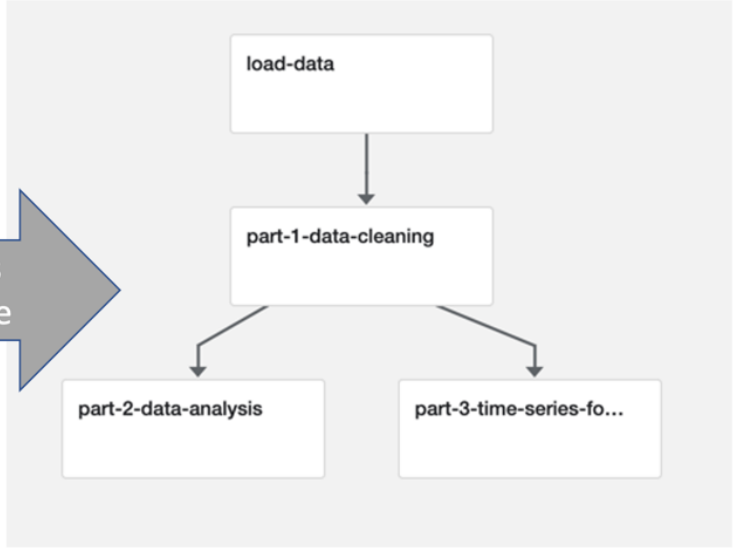
load_data.ipynb | Part 1 - Data Cleaning.ipynb | Part 2 - Data Analysis.ipynb | Part 3 - Time Ser

Markdown | git | Submit Notebook ... | Python 3

First, we select only the subset of data columns of interest and inspect the column types.

```
[ ]: # Choose what columns to import from raw data
column_subset = [
    'DATE',
    'HOURLYVISIBILITY',
    'HOURLYDRYBULBTEMPF',
    'HOURLYWETBULBTEMPF',
    'HOURLYDewPointTempF',
    'HOURLYRelativeHumidity',
    'HOURLYWindSpeed',
    'HOURLYWindDirection',
    'HOURLYStationPressure',
```

Run as pipeline



http://bit.ly/0901-ELYRA

Getting Started

What are the pre-requisites to run?

1. NodeJS 12+
2. Python 3.X
3. Anaconda (optional)
4. KubeFlow installation (optional)

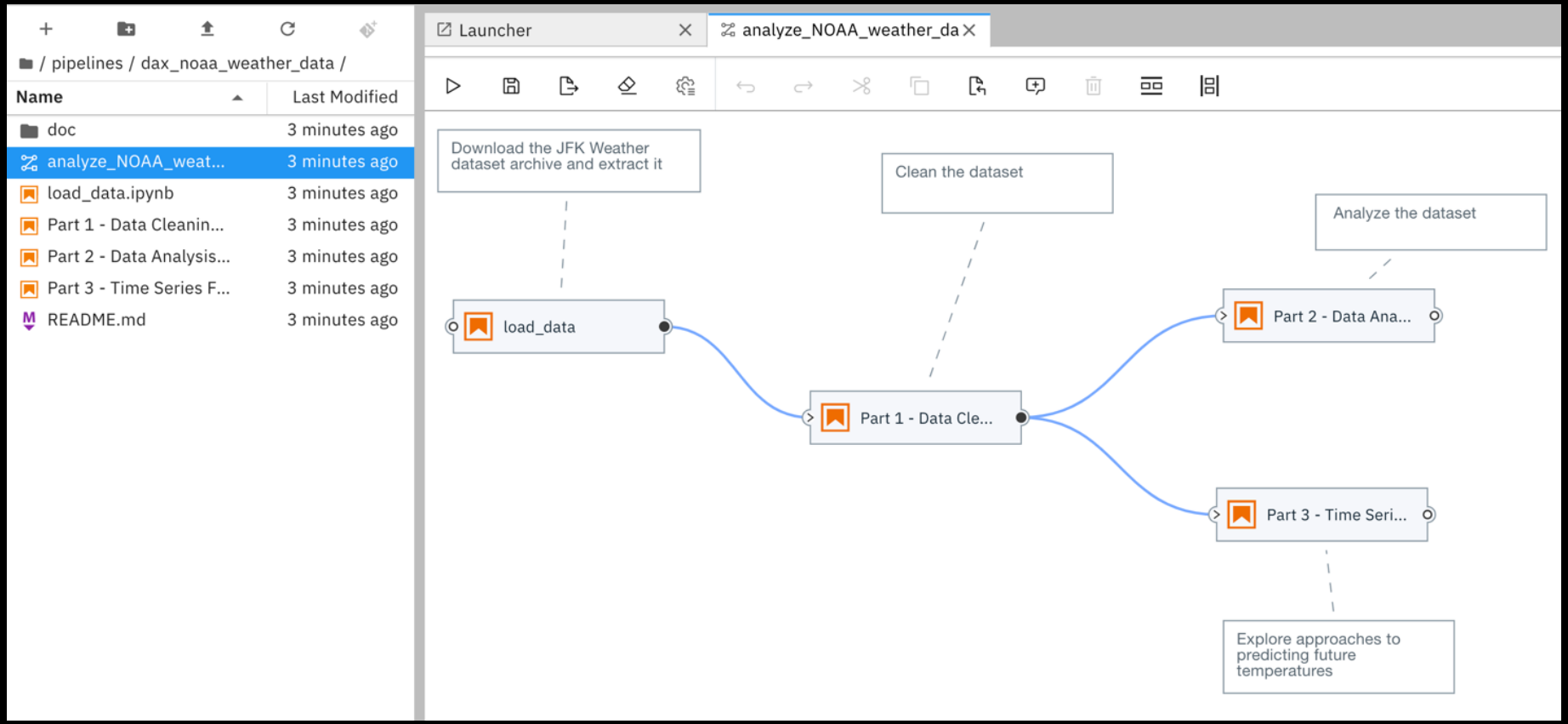
Install Elyra

To install Elyra:
$ pip install elyra==1.1.0 && jupyter lab build
Or:
$ pip install --upgrade elyra && jupyter lab build

To verify installation:
$ jupyter serverextension list
And
$ jupyter labextension list

Starting Elyra:
$ jupyter lab

Running notebooks as a pipeline

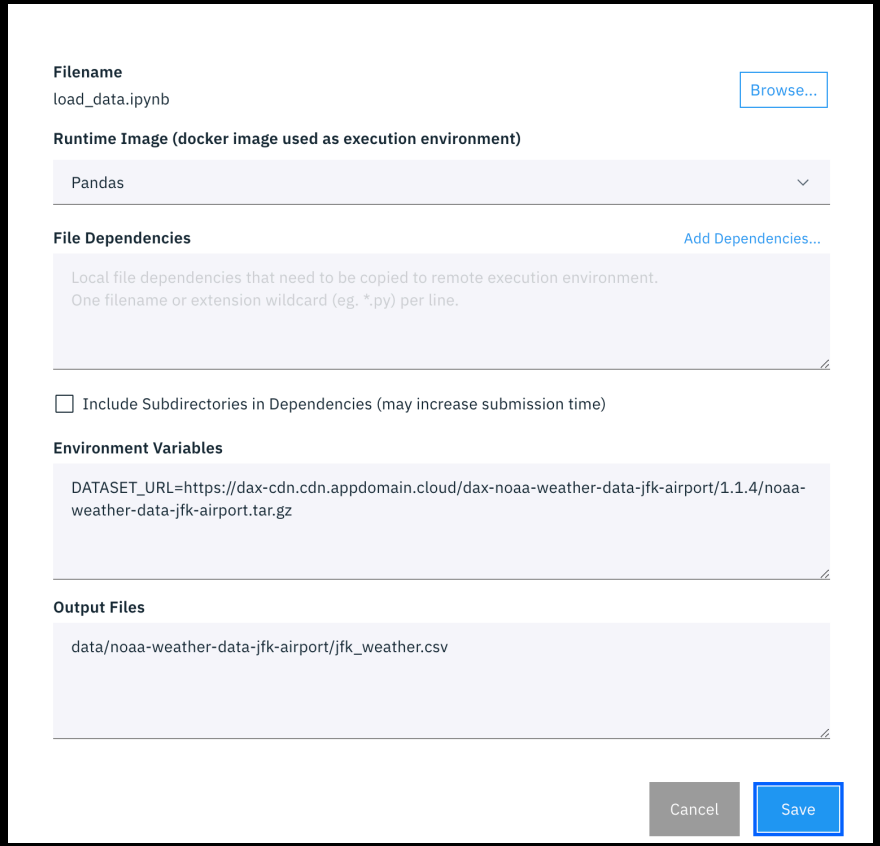


http://bit.ly/0901-ELYRA

Let’s run a pipeline locally and check out the notebooks!

http://bit.ly/0901-ELYRA

Define Run Variables

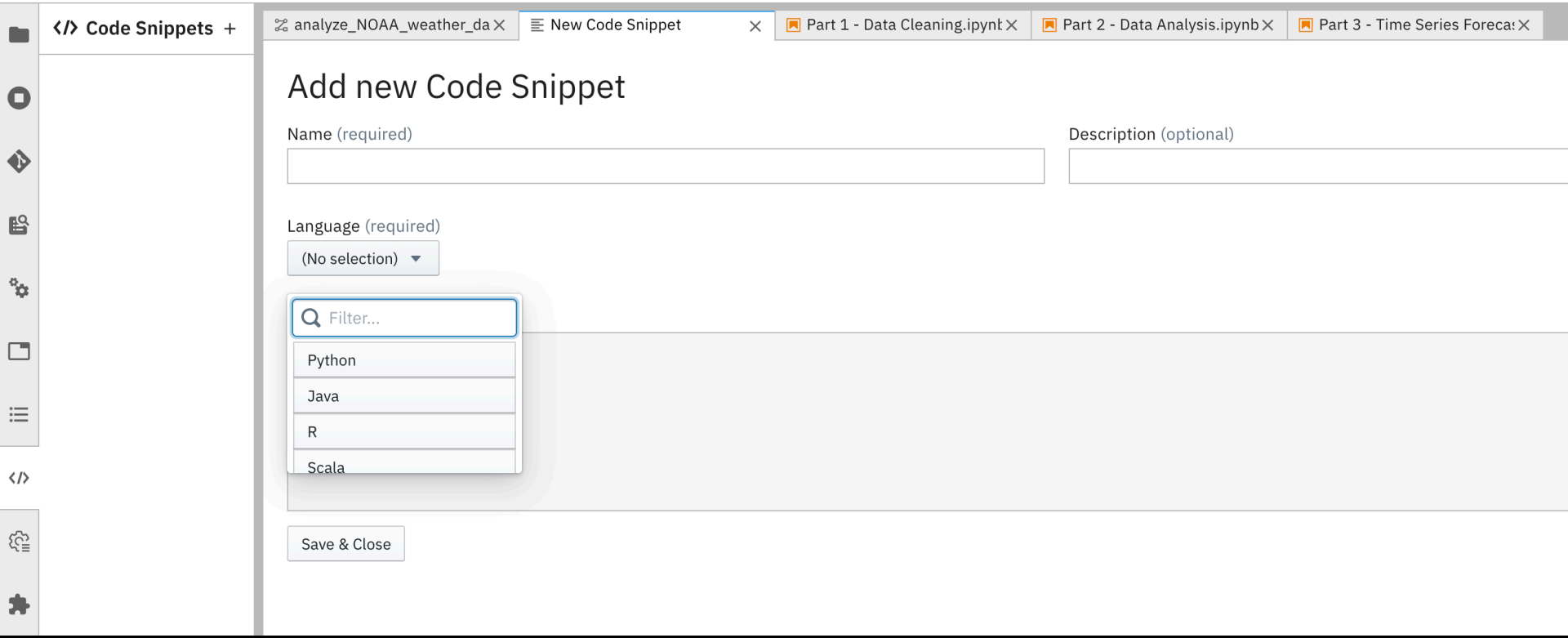


Several variables need to define:

- Docker runtime Image, such as Pandas image, TensorFlow image (w/ GPU support), Anaconda, or PyTorch (with CUDA-devel or with CUDA-runtime)
- File Dependencies
- Environment Variables
- Output files

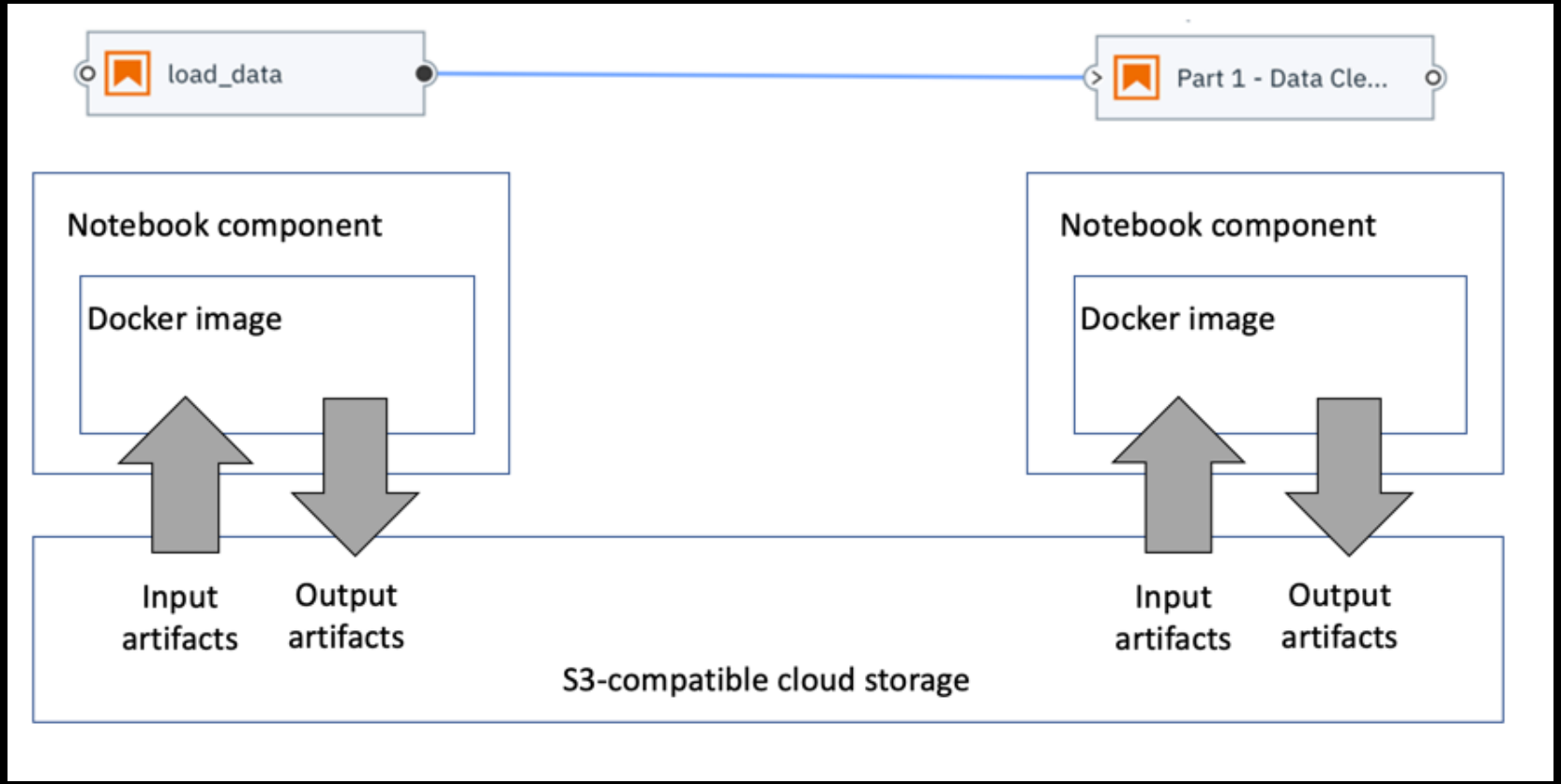
http://bit.ly/0901-ELYRA

Quickly add code snippet



http://bit.ly/0901-ELYRA

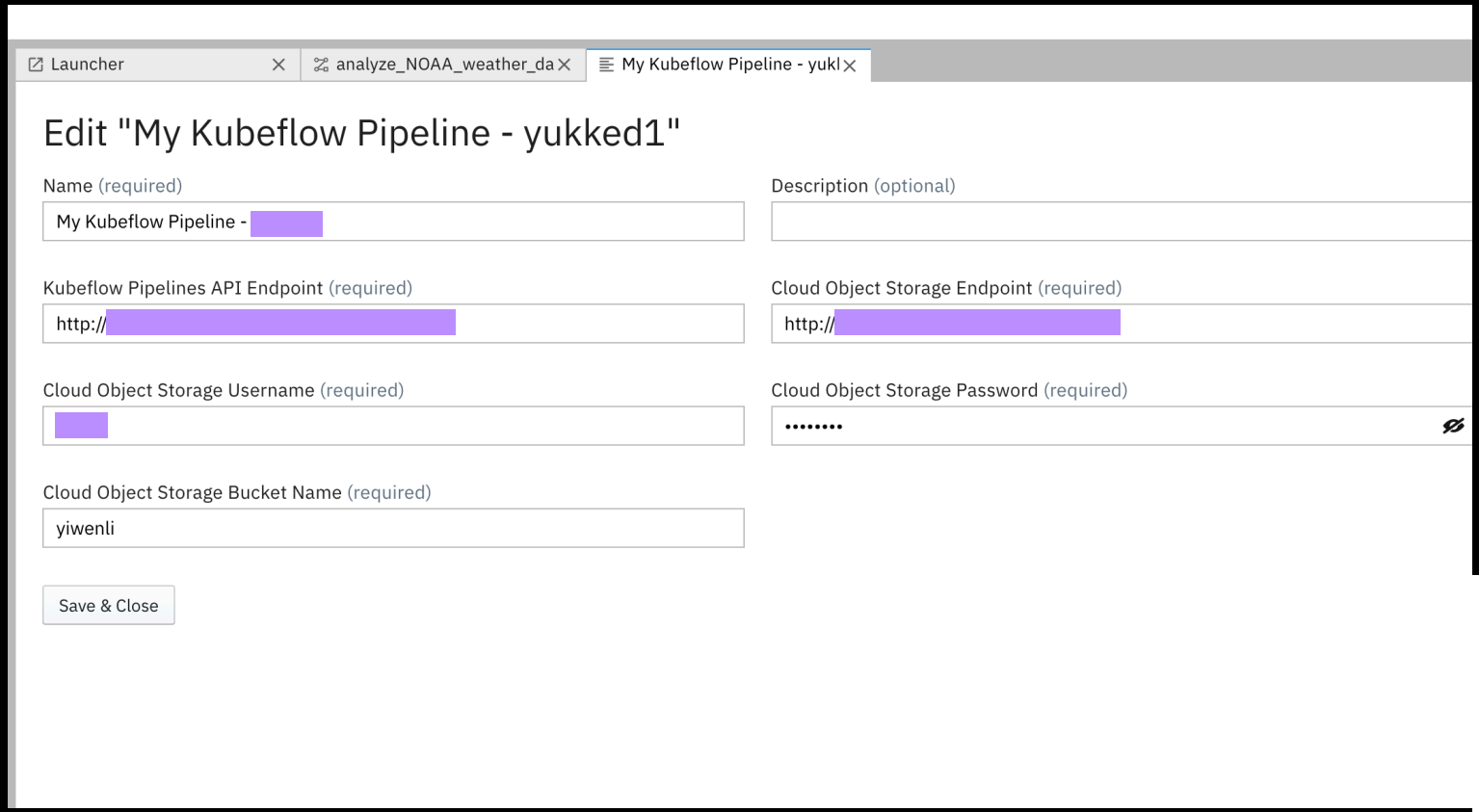
Configuring notebook nodes



http://bit.ly/0901-ELYRA

If you want to run your pipeline on KubeFlow…

Configure a Kubeflow Pipeline runtime



http://bit.ly/0901-ELYRA

Kubeflow

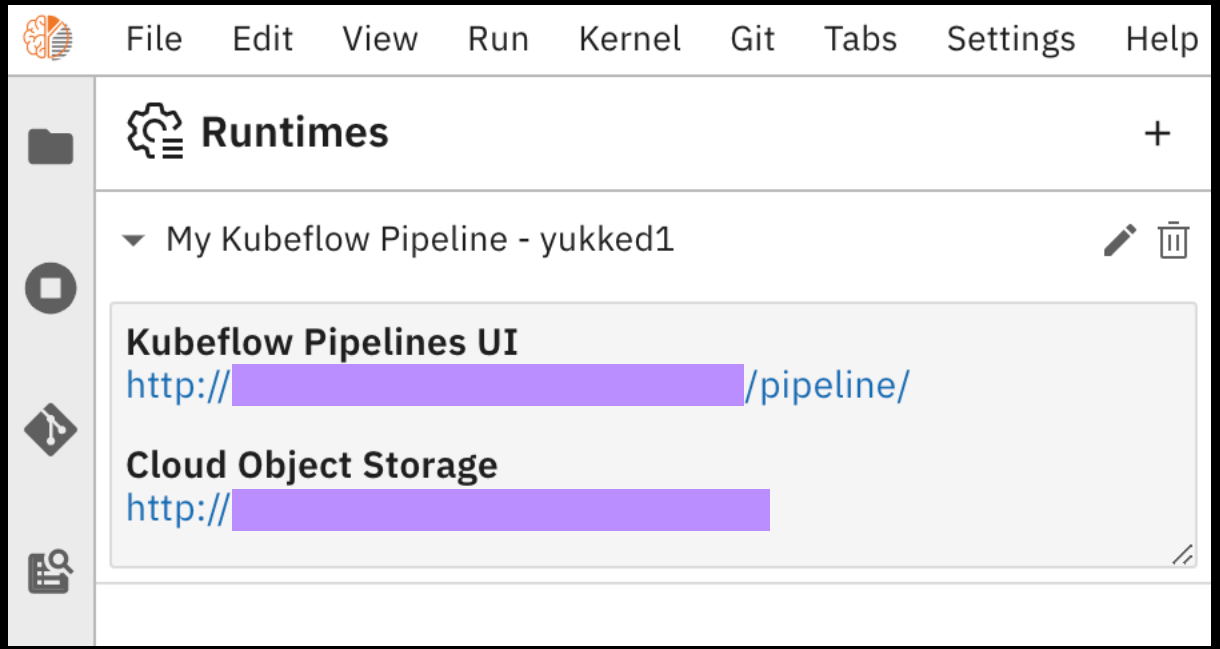
Manage runtime config using CLI

```
$ elyra-metadata install runtimes --schema_name=kfp  \
 --name=kfp_dev_instance \
 --display_name="KFP dev instance" \
 --api_endpoint=http://.../pipeline \
 --cos_endpoint=http://... \
 --cos_username=... \
 --cos_password=... \
 —cos_bucket=...

$ elyra-metadata list runtimes
Available metadata instances for runtimes (includes invalid):
Schema   Instance       Resource
------   --------       --------
kfp      kfp_dev_instance /Users/.../kfp_dev_instance.json
```

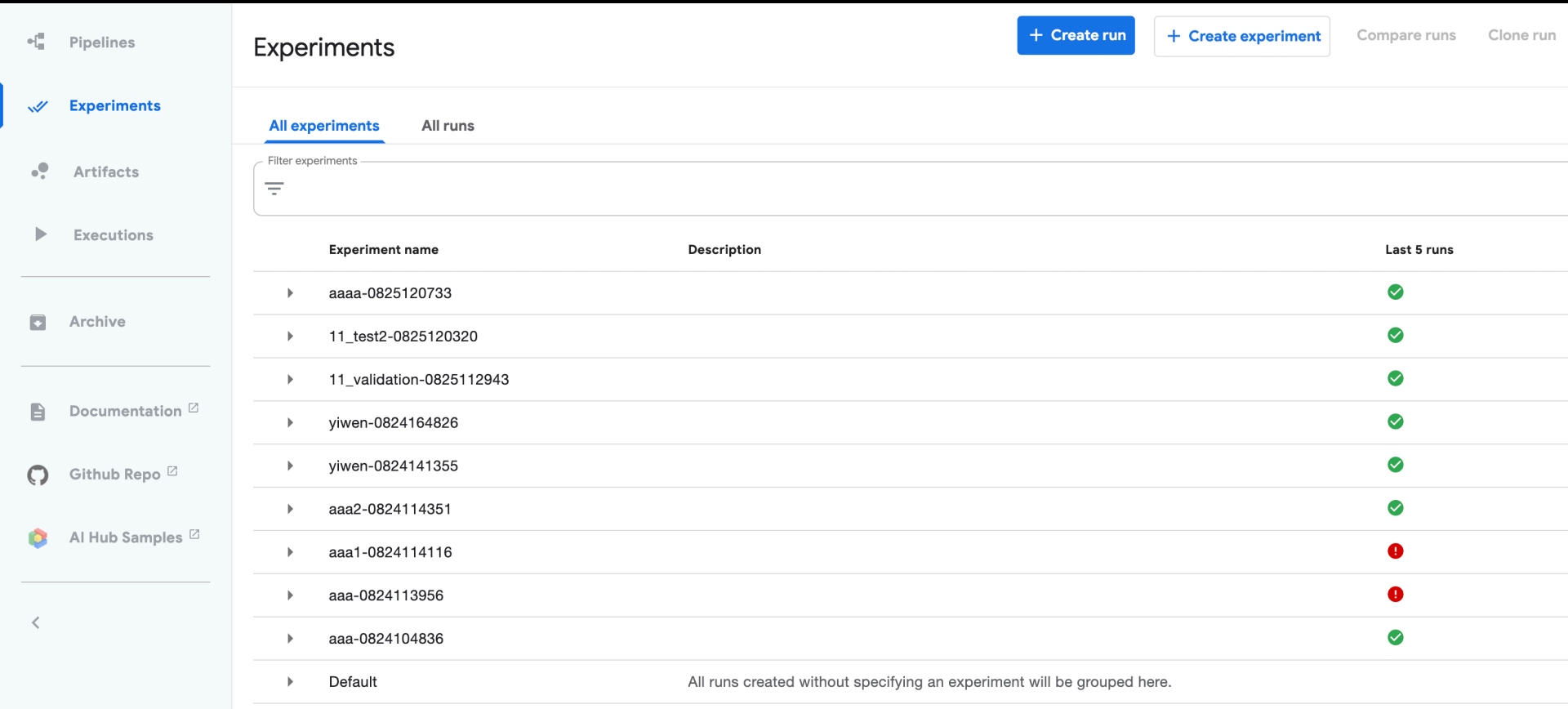
```
$ elyra-metadata list runtimes --json
[
 {
  "name": "kfp_dev_instance",
  "display_name": "KFP dev instance",
  "metadata": {
   "api_endpoint": "http://.../pipeline",
   "cos_endpoint": "http://...",
   "description": "...",
   "cos_username": "...",
   "cos_password": "...",
   "cos_bucket": "..."
  },
  "schema_name": "kfp",
  "resource": "/Users/.../kfp_dev_instance.json"
 }
]
$ elyra-metadata remove runtimes --name=kfp_dev_instance
 Metadata instance '...' removed from namespace 'runtimes'.
```

Monitor a notebook run

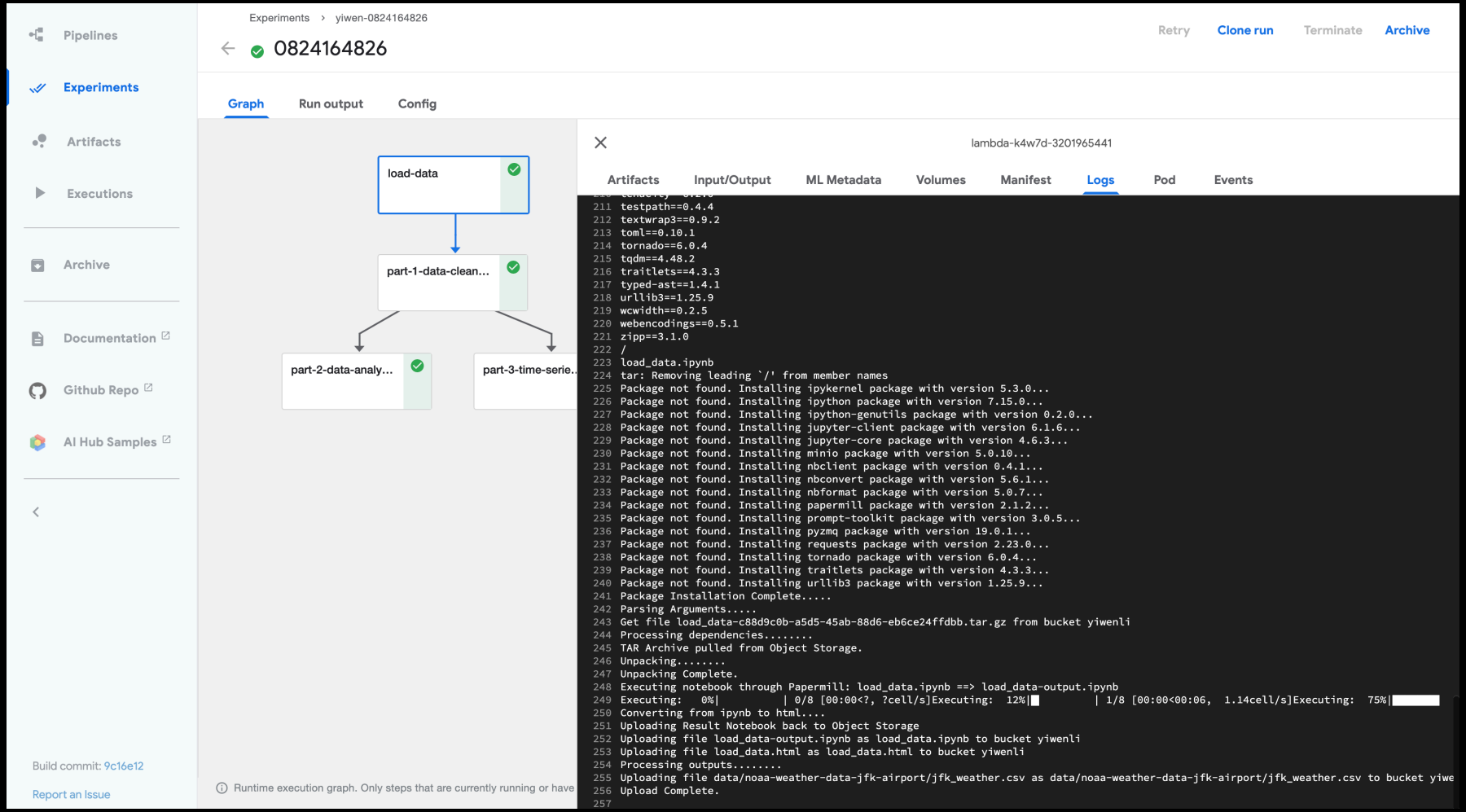


Let’s view the results on KubeFlow!

http://bit.ly/0901-ELYRA

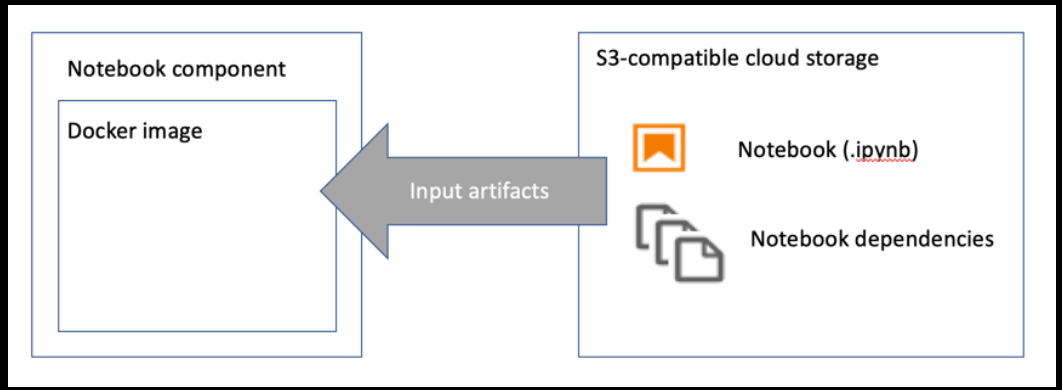


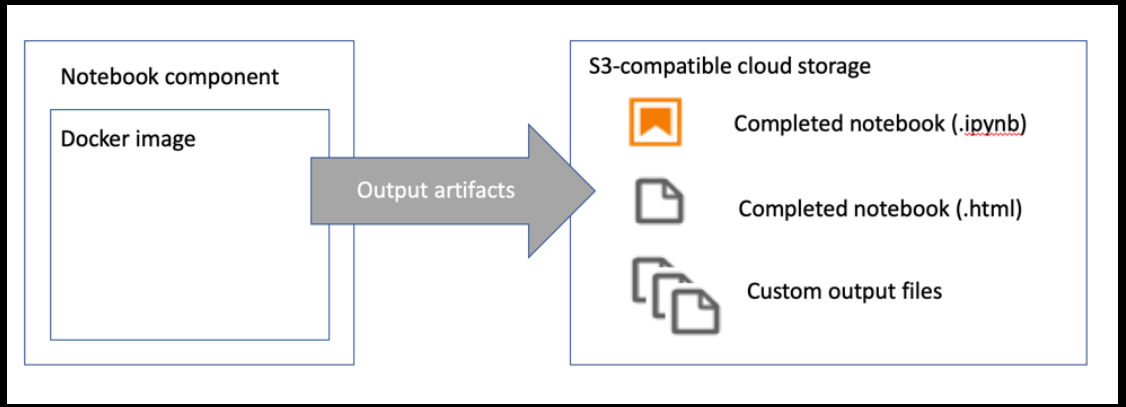
http://bit.ly/0901-ELYRA



http://bit.ly/0901-ELYRA

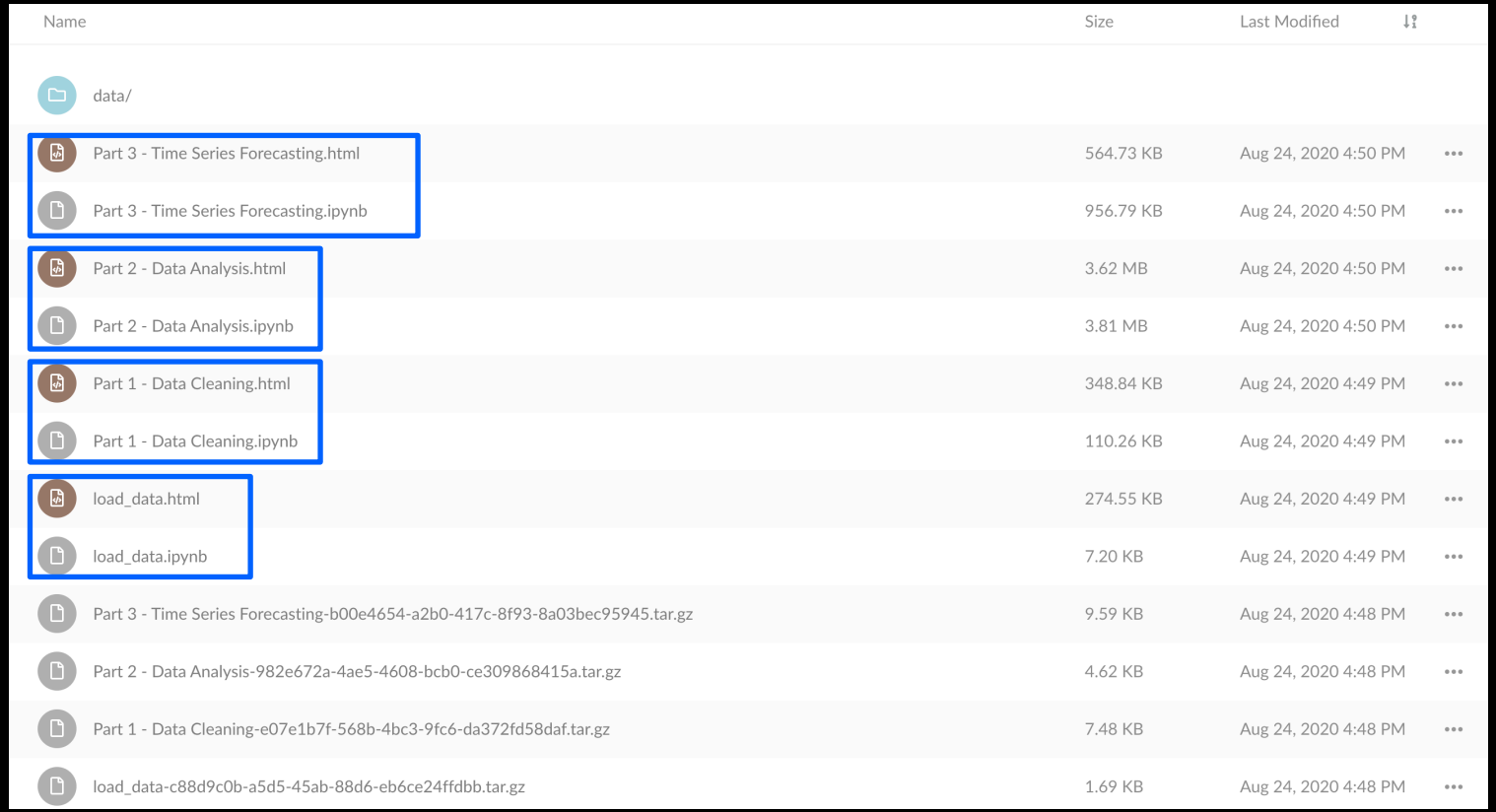
How KubeFlow pipeline works?





http://bit.ly/0901-ELYRA

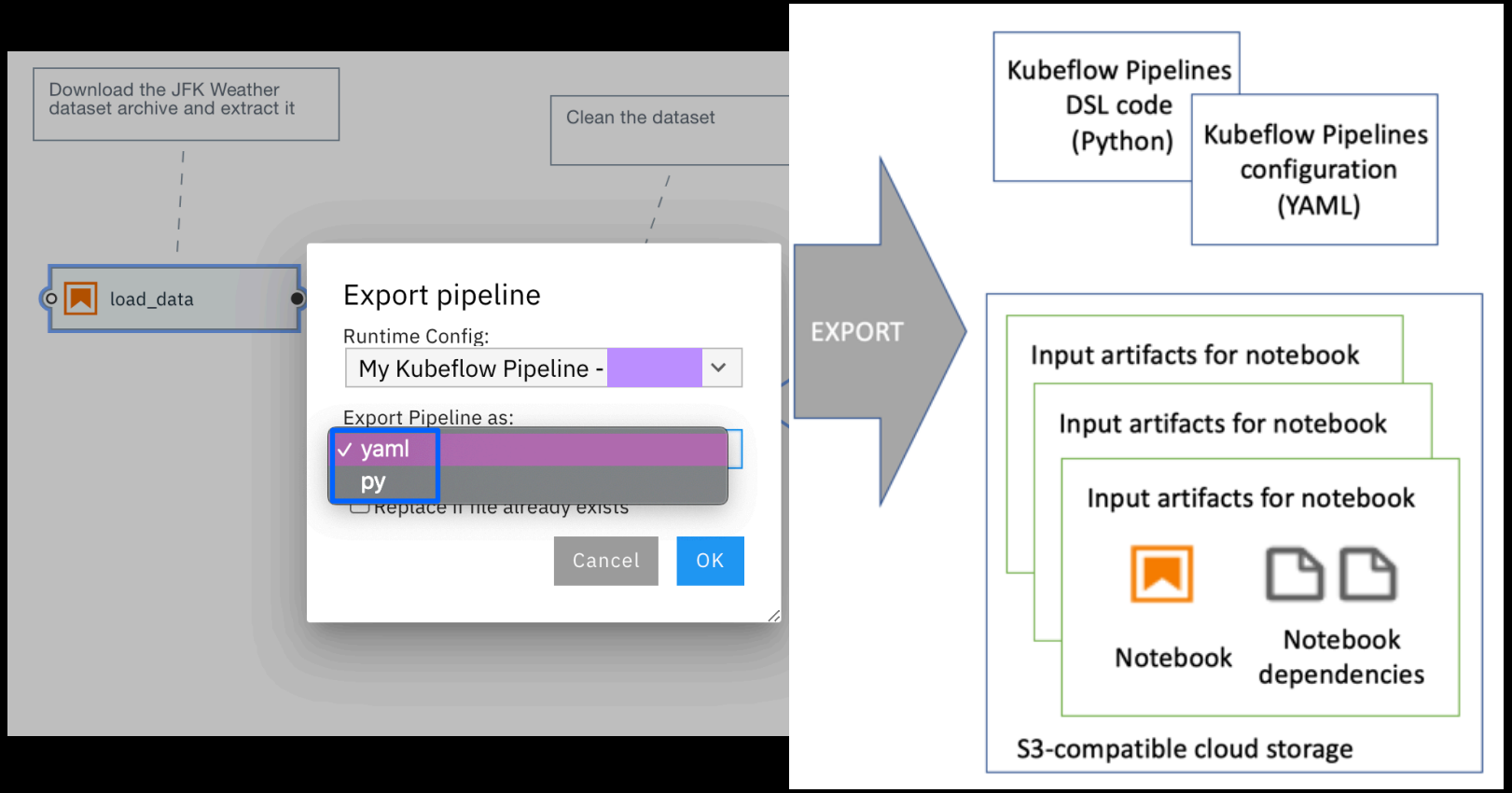
Pipeline outputs



Name	Size	Last Modified
data/		
Part 3 - Time Series Forecasting.html	564.73 KB	Aug 24, 2020 4:50 PM
Part 3 - Time Series Forecasting.ipynb	956.79 KB	Aug 24, 2020 4:50 PM
Part 2 - Data Analysis.html	3.62 MB	Aug 24, 2020 4:50 PM
Part 2 - Data Analysis.ipynb	3.81 MB	Aug 24, 2020 4:50 PM
Part 1 - Data Cleaning.html	348.84 KB	Aug 24, 2020 4:49 PM
Part 1 - Data Cleaning.ipynb	110.26 KB	Aug 24, 2020 4:49 PM
load_data.html	274.55 KB	Aug 24, 2020 4:49 PM
load_data.ipynb	7.20 KB	Aug 24, 2020 4:49 PM
Part 3 - Time Series Forecasting-b00e4654-a2b0-417c-8f93-8a03bec95945.tar.gz	9.59 KB	Aug 24, 2020 4:48 PM
Part 2 - Data Analysis-982e672a-4ae5-4608-bcb0-ce309868415a.tar.gz	4.62 KB	Aug 24, 2020 4:48 PM
Part 1 - Data Cleaning-e07e1b7f-568b-4bc3-9fc6-da372fd58daf.tar.gz	7.48 KB	Aug 24, 2020 4:48 PM
load_data-c88d9c0b-a5d5-45ab-88d6-eb6ce24ffdbb.tar.gz	1.69 KB	Aug 24, 2020 4:48 PM

http://bit.ly/0901-ELYRA

Export a pipeline



http://bit.ly/0901-ELYRA

Ways to run pipelines

Try JupyterLab and Elyra on pre-built docker image:

```
$ docker run -it -p 8888:8888 elyra/elyra:1.0.0 jupyter lab --debug
```

Or you can use Binder:

https://mybinder.org/v2/gh/elyra-ai/elyra/v1.0.0?urlpath=lab/tree/binder-demo

Or you can install Elyra:

https://github.com/elyra-ai/elyra#installation

Get involved with Elyra!

- Open an enhancement request

- Open an issue

- Contributing to Elyra!

http://bit.ly/0901-ELYRA

Curious about the notebooks in the demo?

More is available on Data Asset eXchange!

http://bit.ly/0901-ELYRA