Creating Data Pipelines with Elyra, a visual DAG composer and Apache Airflow

Alan Chin
IBM - CODAIT
About me – Alan Chin

Sr. Software Engineer – Build and Infrastructure – CODAICT
• Over 4 years working with Open Source Projects
• Currently Contributing to the Elyra and Jupyter Enterprise Gateway Projects

akchin@us.ibm.com
@AlanChin11
https://github.com/akchinSTC
https://www.linkedin.com/in/alankchin/
Overview

• Elyra, what is it?
• Creating a Notebook Based Pipeline
• Elyra Pipeline Editor
  • Airflow Pipeline Processor
  • Example: How Elyra uses Apache Airflow
• Short Demo
• New upcoming changes in 3.0
  • Operator/component support
Elyra, what is it?

Elyra at its core, is a curated collection of JupyterLab UI and server extensions, designed to compliment each other and is completely Open Source.
Elyra, what is it?

- Language Server Protocol Integration
- Notebook and Python Script Table of Contents Navigation
- Git version control Integration
- Code Snippets
- Hybrid runtime support with Jupyter Enterprise Gateway
- Pipeline Visual Editor
Elyra, what is it?

• Provides users with a low-code/no-code solution to creating data pipelines in Apache Airflow and Kubeflow Pipelines on Kubernetes.

• Designed around concepts and patterns common in pipeline construction and put into a familiar, easy-to-navigate interface for Data Scientists and Engineers.
Let’s make a Pipeline!
Classic Data Pipeline

Read in Data → Do something with data → Output and store
Classic Data Pipeline

Read in Data → Do something with data → Output and store

Read in Data
The Things You Don’t See

- Python Libraries
- Learning Frameworks
- Misc. Environmental Configs
- Hardware Requirements

Read in Data

Do something with data

Output and store

• Intermediate artifacts

• Intermediate artifacts
Let’s Use Containers

- Common pattern
- Prebuilt means time savings
- Consistent and reproducible
- Isolation

• Python Libraries
• Learning Frameworks
• Misc. Environmental Configs
• Hardware Requirements
Hardware

- CPU and Memory
- Architecture
- GPUs and TPUs

- Python Libraries
- Learning Frameworks
- Misc. Environmental Configs
- Hardware Requirements
Let’s use a Container Orchestrator

- OpenShift and Kubernetes
  - Workers can have different hardware configurations to accommodate various workloads
  - Cluster can scale up or down depending on your resource needs

- Python Libraries
- Learning Frameworks
- Misc. Environmental Configs
- Hardware Requirements
Containerized Data Pipeline

1. Read in Data
2. Do something with data
3. Output and store
Expressing the pipeline as code:

- Scripts in Bash, Python, DSL
- Popular Pipeline Projects
  - Apache Airflow
  - KubeFlow Pipelines (DSL and SDK Compiler)
- Users would need to spend time learning how to compose the pipeline using the DSL / libraries
Using Processors for Runtimes

- Processor Base Class allows users to extend Elyra’s pipeline processor to use different workflow orchestrators
- Currently supports Apache Airflow, Local and Kubeflow Pipelines
Example: Apache Airflow Processor

Elyra Airflow Processor

- process()
- Constructs graph logic of pipeline
- Uploads any required files to S3
- Builds the DAG python file
- Pushes DAG to Github

S3

kubernetes

Apache Airflow

IBM Developer / CODAIT / © 2020 IBM Corporation
Example: Apache Airflow Processor

Do something with data

Papermill executes notebook and creates an output notebook / additional artifacts

Artifacts are then pushed to S3
Elyra Pipeline Editor

Demo Time!
Node Properties

Container parameters are taken in properties config menu
- Image to be used
- Local file dependencies to be uploaded to S3
- Hardware resource requirements
- Environmental variables to be set
- Intermediate Outputs for downstream nodes

Passed to processors where they are marshalled into the correct structure for pipeline construction
Upcoming changes and new in 3.0

- **Airflow Operator Support**
  - Will allow users to specify and import core or contributor operators for use in Elyra’s visual pipeline editor
  - Currently in pre-release/experimental phase
  - Short Demo
Wrap Up

• Deconstruct / Modularize existing pipeline(s) into Notebooks
• Determine what your pipeline resource/ environmental requirements are
• Build and run your notebook/script-based pipeline with Elyra’s pipeline editor
• If you need assistance, please don’t hesitate to open an issue on our Github page OR just ask us on gitter!

https://gitter.im/elyra-ai/community
Check us out!

Getting started with Elyra

Elyra’s Github
https://github.com/elyra-ai/elyra

COVID notebooks Github
https://github.com/CODAIT/covid-notebooks

Contributing to these projects
• Just star and fork!
• Bug reports
• Propose improvements
• Code reviews
• Community meetings
Questions?
Thank You for Joining!
Stay Safe and Stay Healthy!