

Flyte: a Robust and end-to-end ML and Data Processing Platform



Eduardo Apolinario
04/26/2023



whoami

- OSS Team lead @ Union.ai
- 10+ years of experience straddling the border between Infrastructure and Product
- ❤️ OSS



@curupa / eapolinario



su **Union** whoami

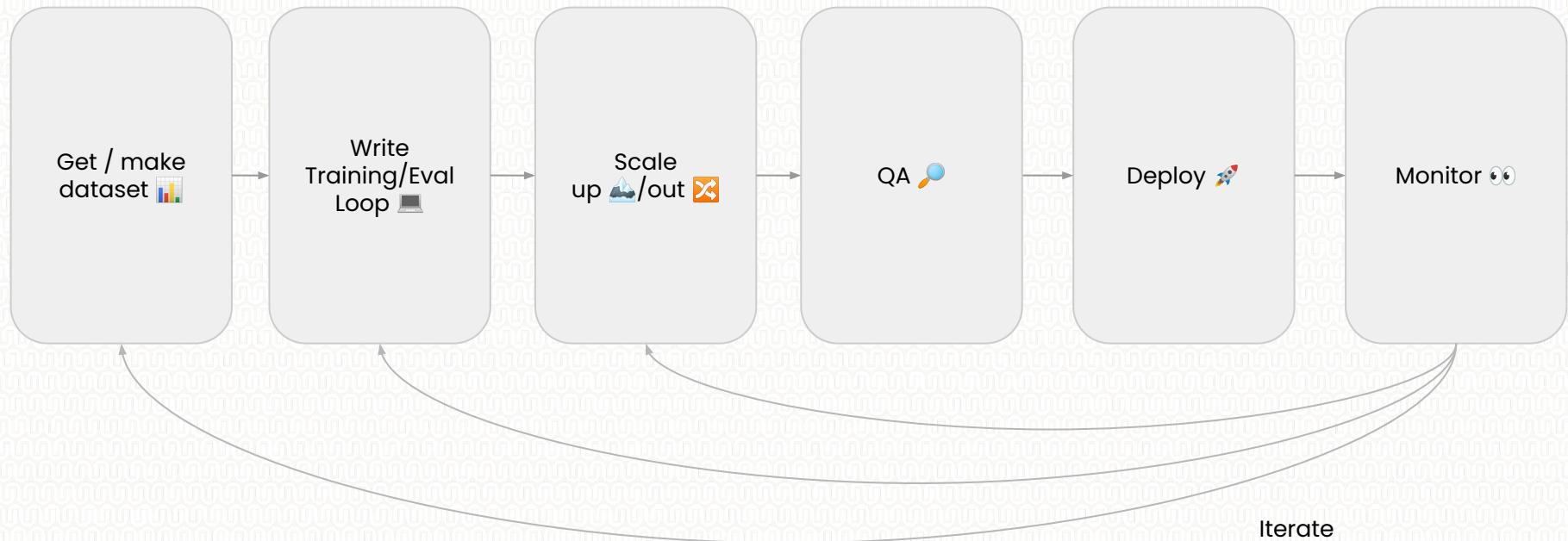
- Harness the power of Flyte without the overhead
- Free Data and ML teams from infra constraints
- <https://union.ai/>

Motivation



The Life of an ML Engineer

“My umpteenth ML project”



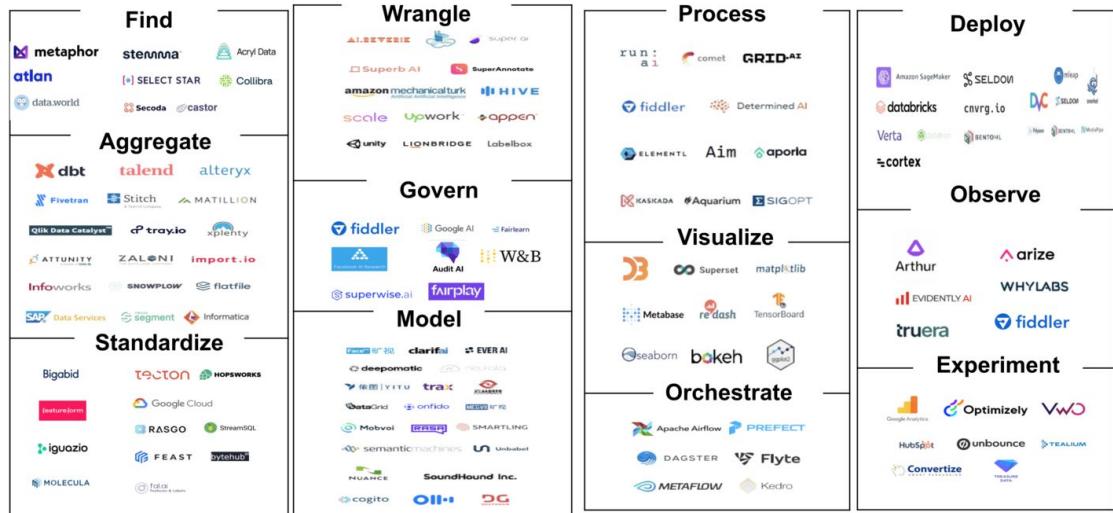
Insights

- ML pipelines are  data pipelines
- Software is stateless , data is stateful 
- If data shifts , models deteriorate 

Why we built Flyte

Challenge 1

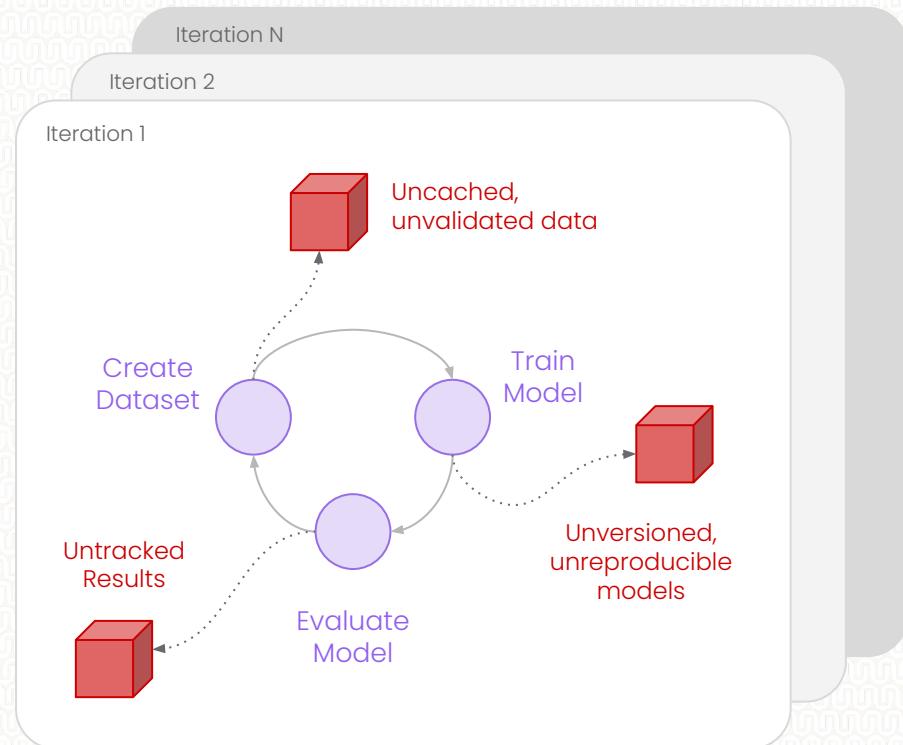
The ecosystem of tools is constantly and rapidly evolving



Credit: Sandeep Uttamchandani

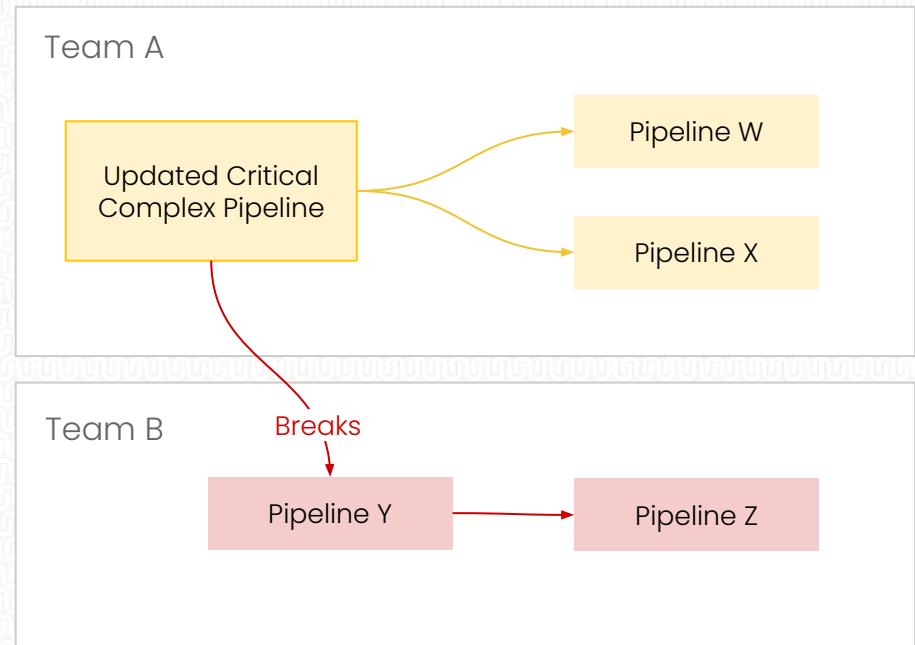
Challenge 2

Developing datasets & models can be wasteful and inefficient.



Challenge 3

Data/ML infrastructure
doesn't scale well across
teams/orgs



Challenge 4

Complex ML workflows
require a dedicated
infrastructure team

Provisioning
CPU/GPU/Memory

Framework/Library
Independence

Multi-tenancy

Auto-scaling

Efficiency: Caching,
Model Checkpoints

Cost Controls: Spot
Machines

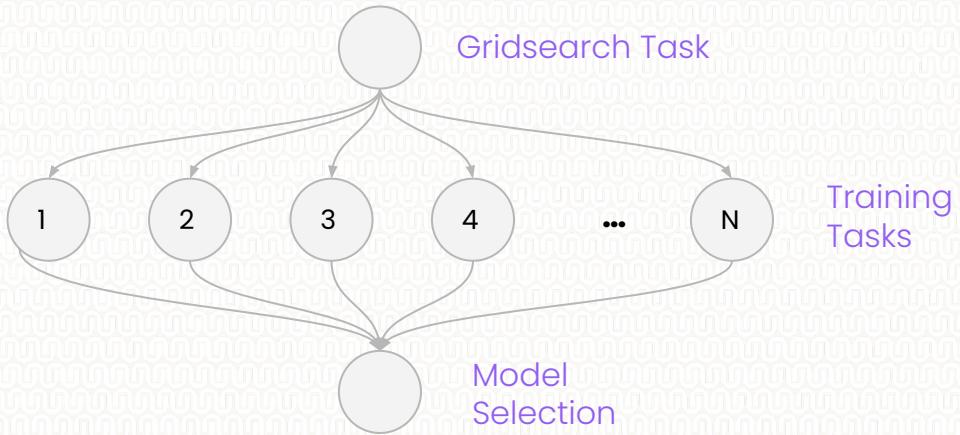
Data Quality
Assurance

Model Monitoring

Challenge 5

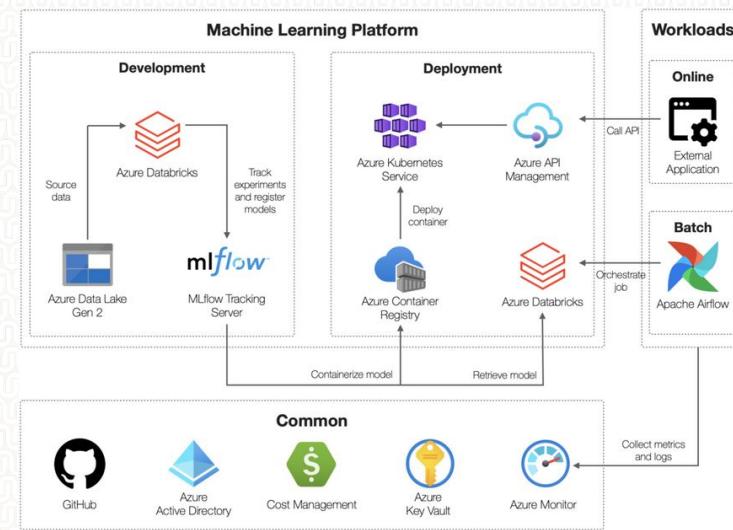
ML pipelines require dynamism, i.e. the execution graph depends on the inputs

Input: N hyperparameter configurations



Orchestration to the rescue! ✨

Orchestrators coordinate the logical flow of computations needed to get data from its *raw state*  into a *desired state* 



What would I want out of an orchestrator?

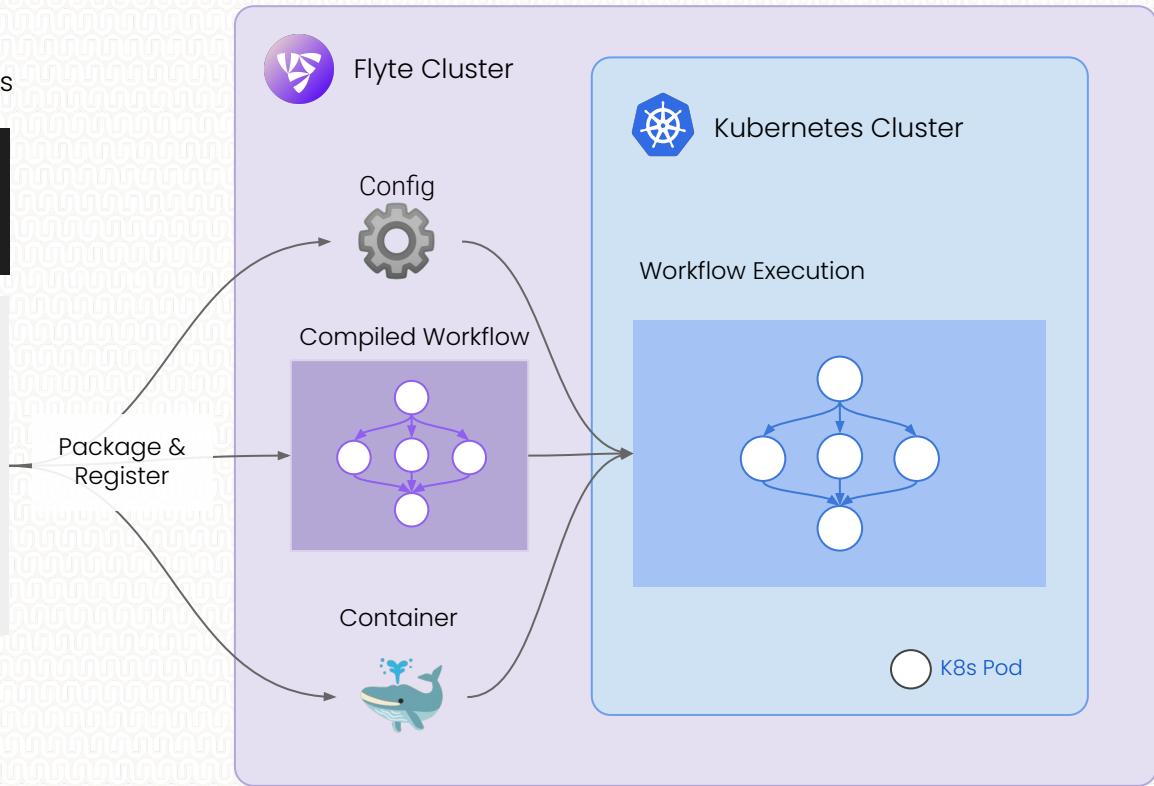
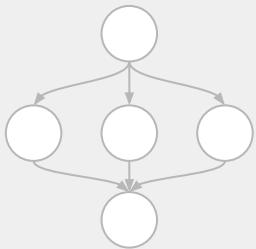
Challenges	Requirements
 Rapidly evolving ecosystem	A future-proof system that's language- and framework- agnostic.
 Data/model development inefficiency	Out-of-the-box support for data lineage tracking, caching, immutability, and versioning.
 Poor scaling across teams/orgs	Isolated units of compute that can be arbitrarily composed together and reused across many different pipelines.
 Need for infrastructure expertise	Declarative provisioning of compute, memory, disk requirements.
 Dynamic execution graphs	Support for DAGs whose structure can be determined at runtime

How does Flyte address these challenges?

Flyte Overview

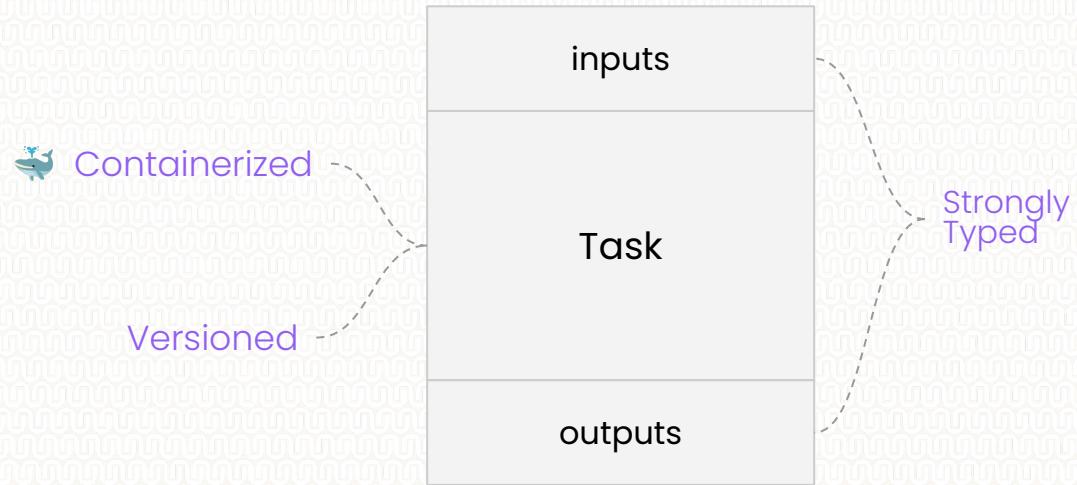
Create Tasks and Workflows

```
# workflows.py
from flytekit import task, workflow
```



Tasks

The smallest unit of work in Flyte.

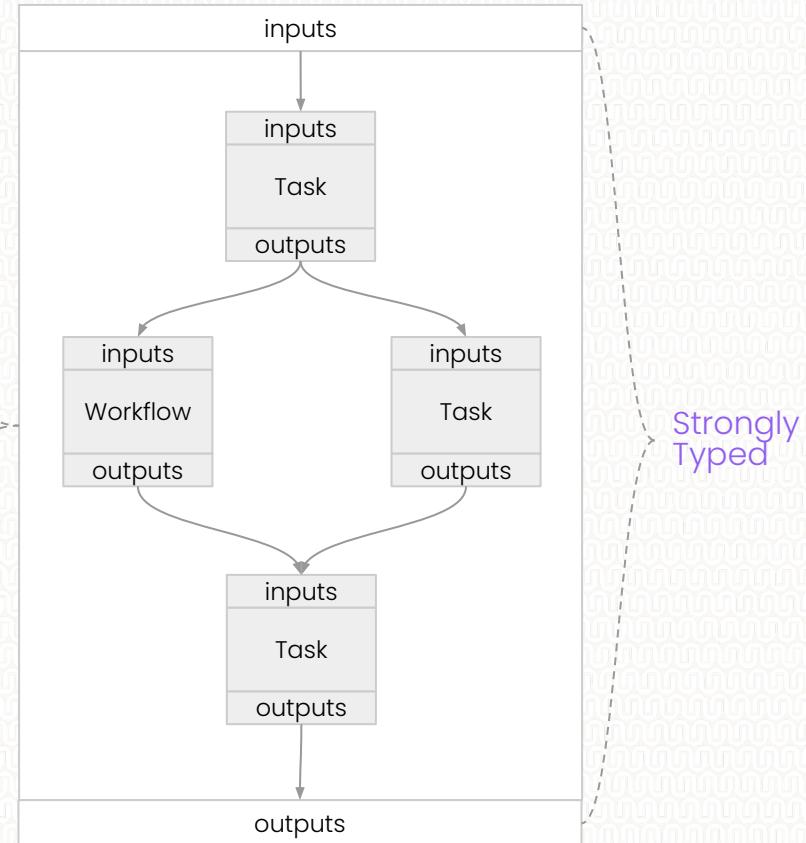


Workflows

Compositions of Tasks
to achieve complex
computations

Data Flow is
1st Class
Citizen

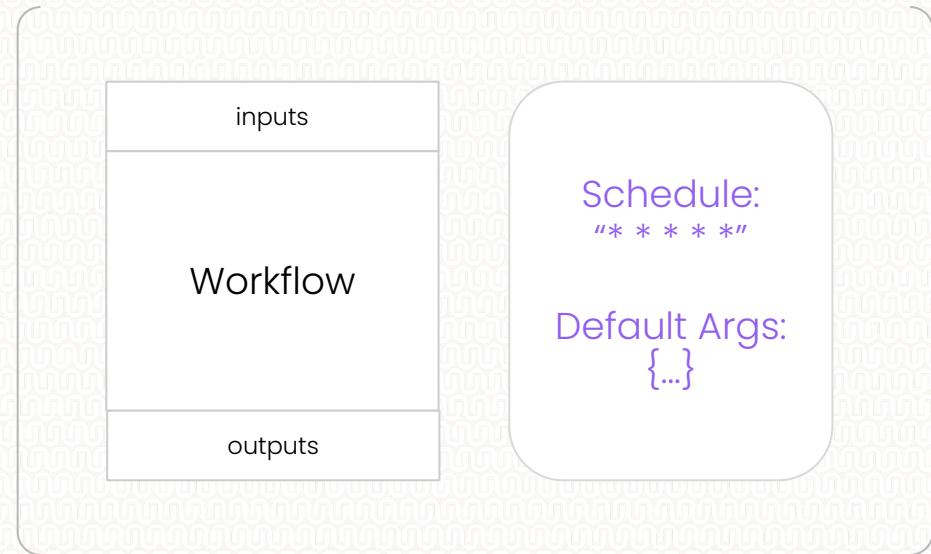
Versioned



Launch Plans

Customizing and scheduling the invocation behavior of workflows

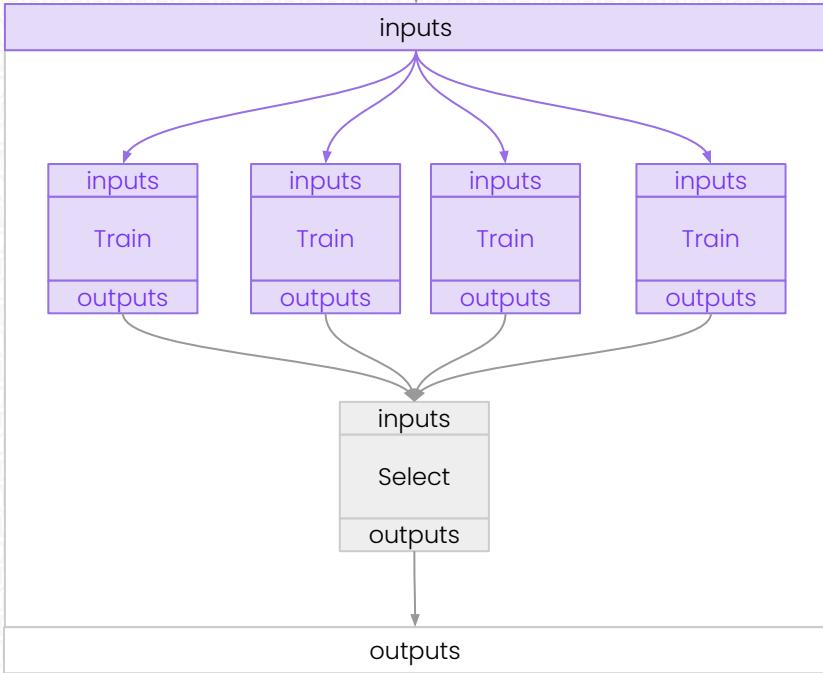
Launch Plan



Dynamic Workflows

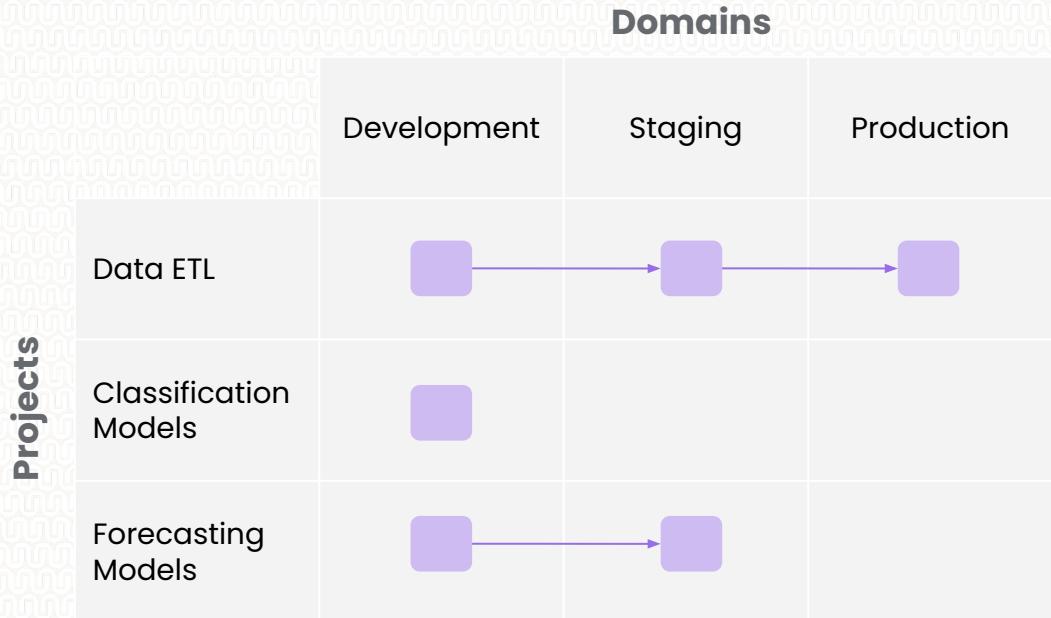
Compositions of Tasks to
achieve complex
computations

```
{"learning_rate": [0.1, 0.01, 0.001, 0.0001]}
```



Projects and Domains

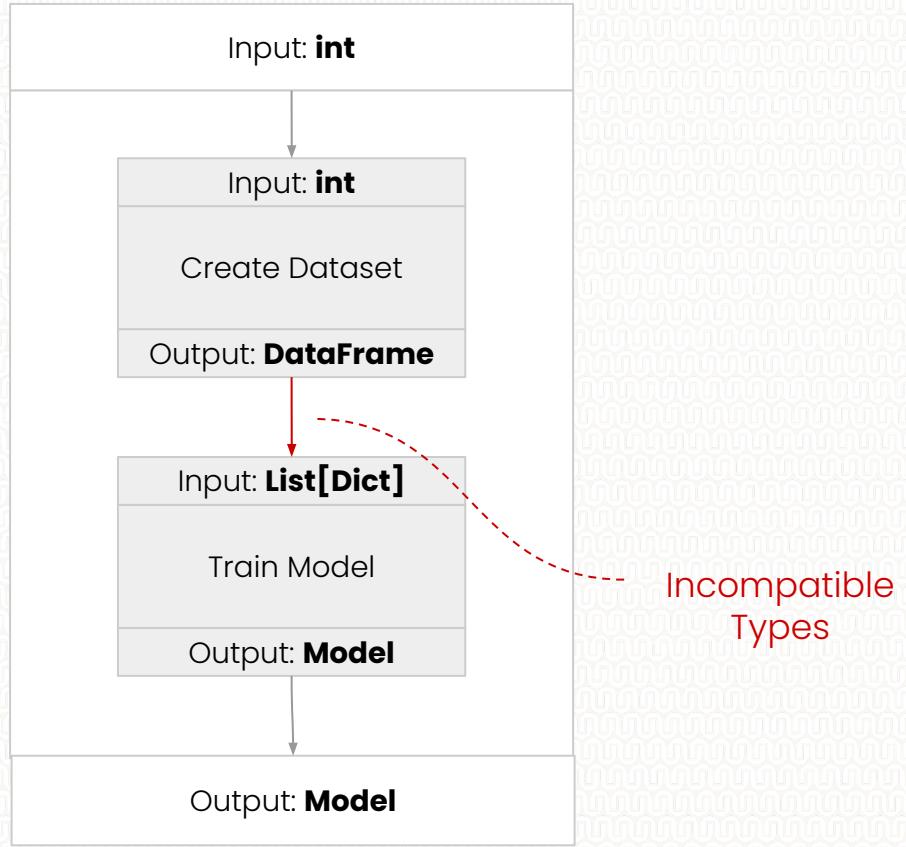
Logical groupings of tasks and workflows for built-in multi-tenancy and isolation.



What's unique about Flyte?

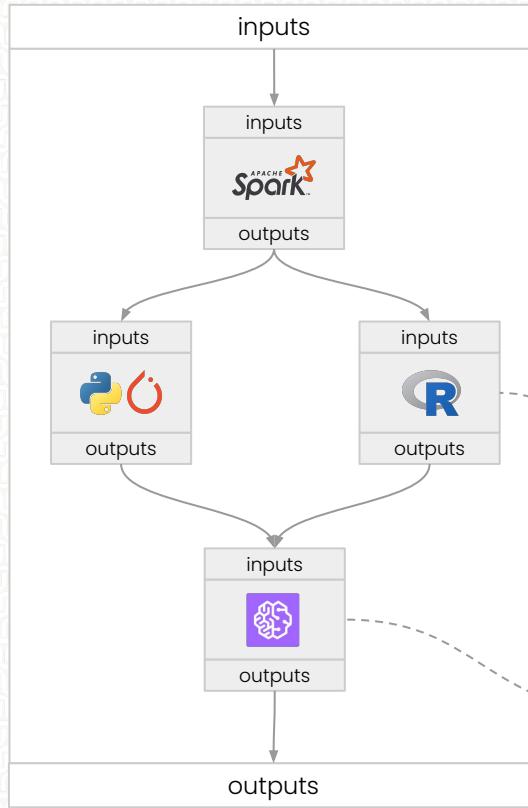
Type Safety

Get errors about your execution graph at compile-time, even before executing your code



Language-independence

Create Workflows in Python, Java, and Scala.

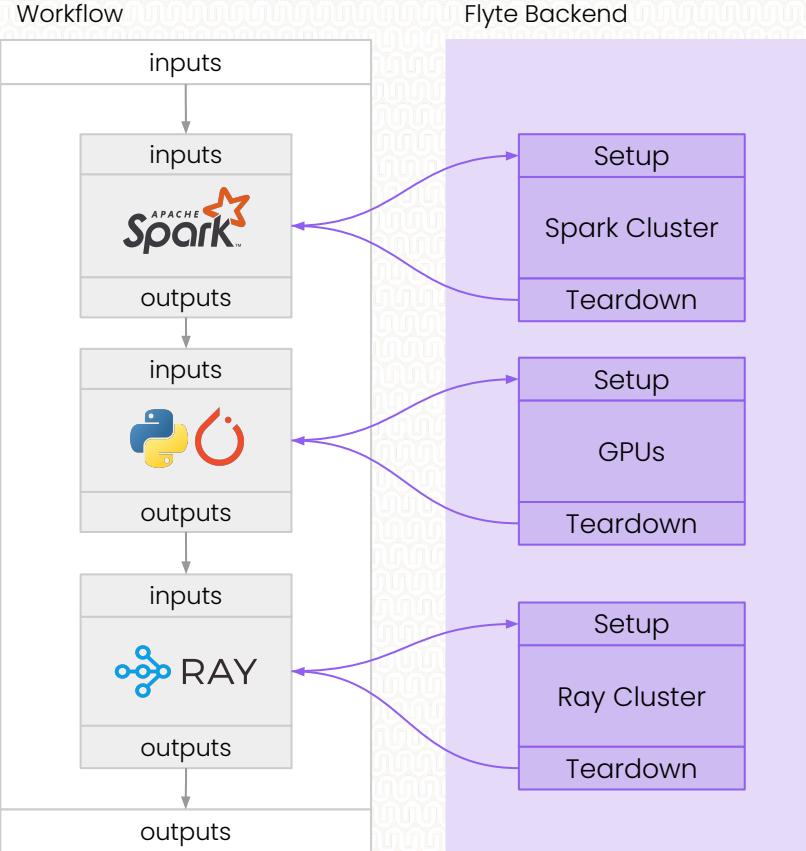


Execute any language or framework

Integrate external services

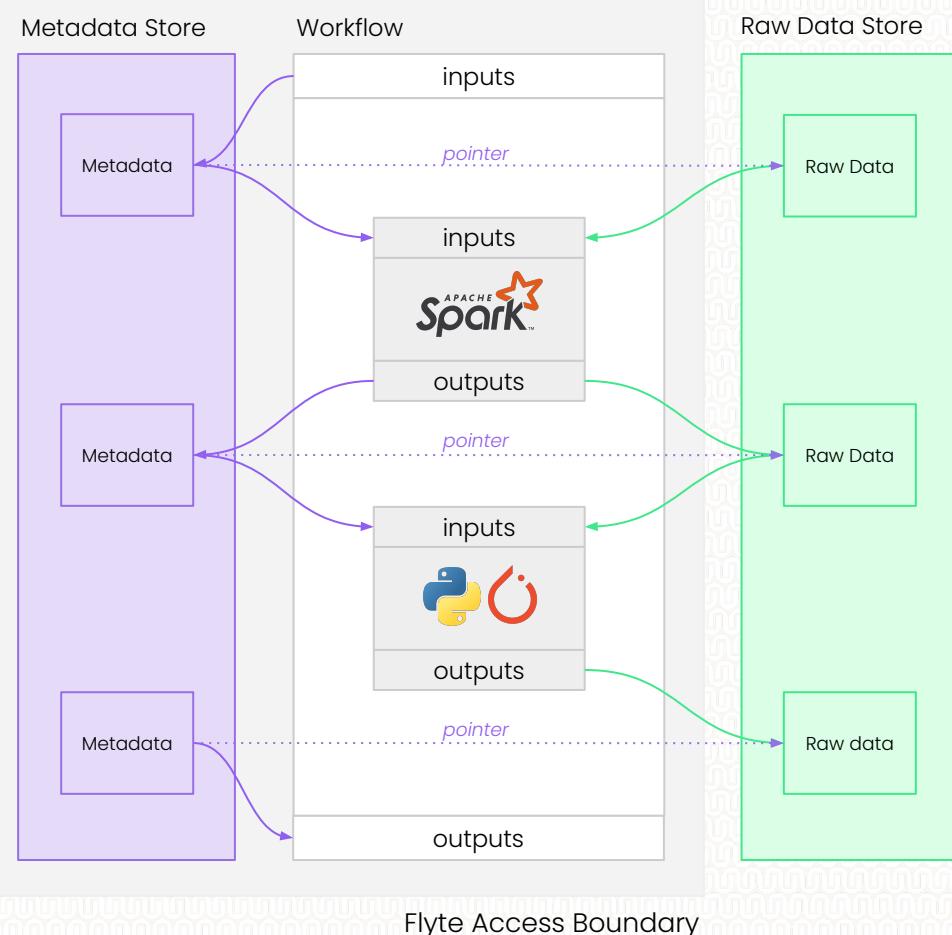
Declarative Infrastructure

Declaratively provisions ephemeral cluster, CPU/GPU, and memory resources.



Abstracted Data Persistence

Don't worry about how data is serialized/deserialized as your execution graph runs



How are People Using Flyte?



Spotify®

Use Case:

End-to-end profit & loss forecasting

Flyte's Impact:

Increases velocity to deliver models

Breaks down silos between teams

freenome

Use Case:

Early detection of cancer via ML

Flyte's Impact:

Powers full data and ML stack

Accelerates clinical research

**Use Case:**

Digital 3D Mapping of the World

Flyte's Impact:

Enables processing ~2.5 petabytes of data

Unlocks multi-cloud provider capabilities



gojek

Use Case:

Price optimization, rider matching, etc.

Flyte's Impact:

Easily roll back critical pipeline bugs

Unlocks scale and reliability of pipelines

Summary

-  Flyte orchestrates compute, data, and infrastructure
-  Type safety means you can catch compile-time bugs early
-  Container-native tasks ensures reproducibility
-  Scales your production workflows seamlessly
-  Supports the canonical data science and ML tech stack
-  Easily customizable and extendable
-  Breaks the data and model silos between teams

Getting Started with Flyte

 Follow our Getting Started Guide:

https://docs.flyte.org/en/latest/getting_started/index.html

 Join us on Slack:

<https://slack.flyte.org/>

 See the Code:

<https://github.com/flyteorg/flyte>



Harnessing the Power of Flyte™ without the Overhead

Flyte™ is helping organizations like Spotify and Lyft build a new generation of products that make elegant use of complex data and machine learning. Now Union AI, the team behind Flyte™, has created a managed version of the workflow orchestrator, freeing data and ML teams from infrastructure constraints and setup.

[Try Union Cloud](#)

We're Hiring!

<https://www.union.ai/careers>



PyData Flyte Happy Hour

4/27 @ 7-9pm Pacific

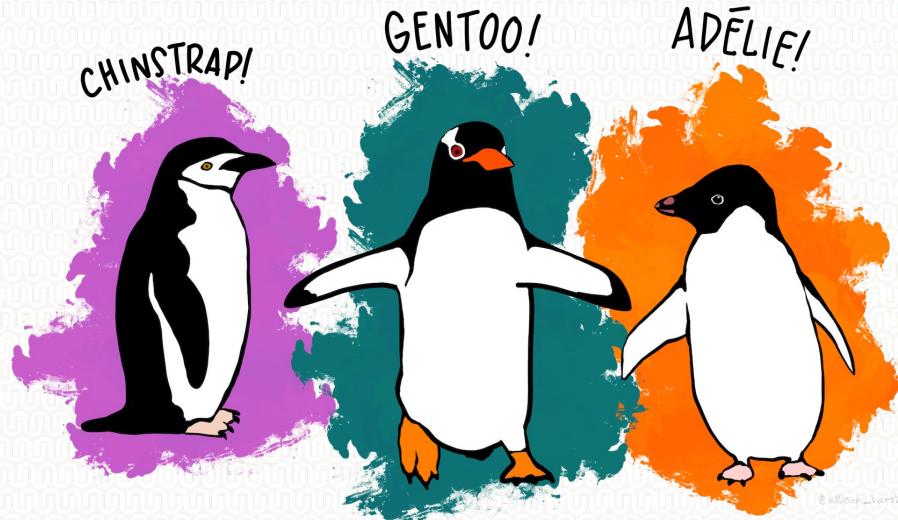
Location:

Locust Cider Redmond
7425 166th Ave NE Suite C110
Redmond, WA 98052

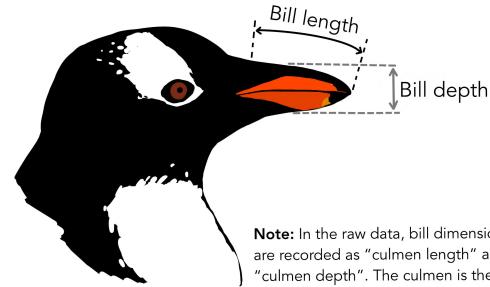
➡ <https://go.union.ai/lFcSgnm>

Building a Model Training Pipeline with Flyte

Training a penguin species classification model



<https://allisonhorst.github.io/palmerpenguins/>

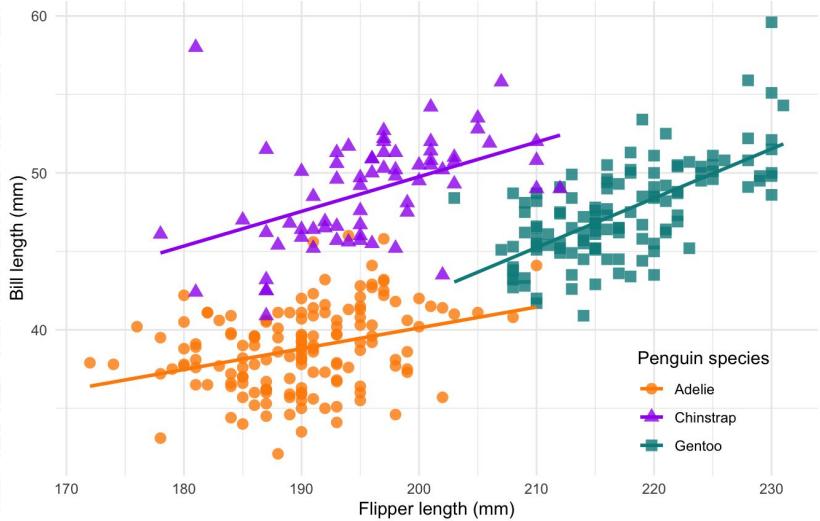


Note: In the raw data, bill dimensions are recorded as "culmen length" and "culmen depth". The culmen is the dorsal ridge atop the bill.

The data:

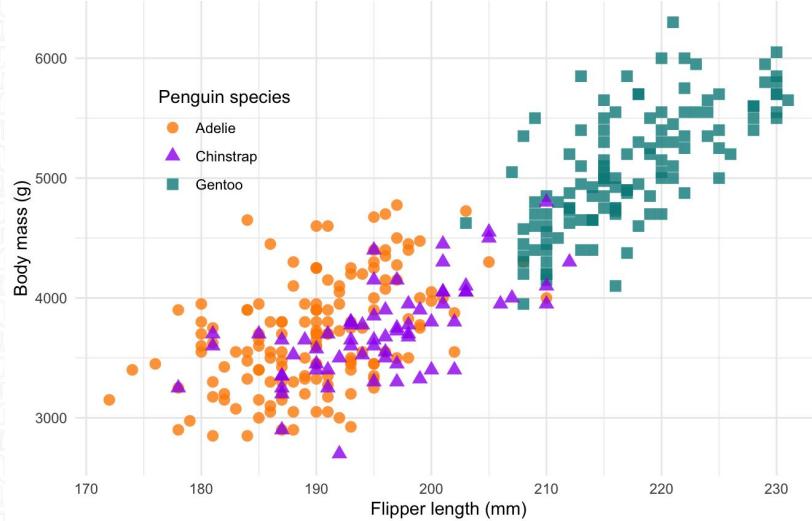
Flipper and bill length

Dimensions for Adelie, Chinstrap and Gentoo Penguins at Palmer Station LTER



Penguin size, Palmer Station LTER

Flipper length and body mass for Adelie, Chinstrap, and Gentoo Penguins



<https://allisonhorst.github.io/palmerpenguins/>

It's showtime!

<https://tinyurl.com/pydata-seattle-2023-tutorial>