MLOps vs DevOps

Why Data Makes it Different

Hugo Bowne-Anderson

outerbounds

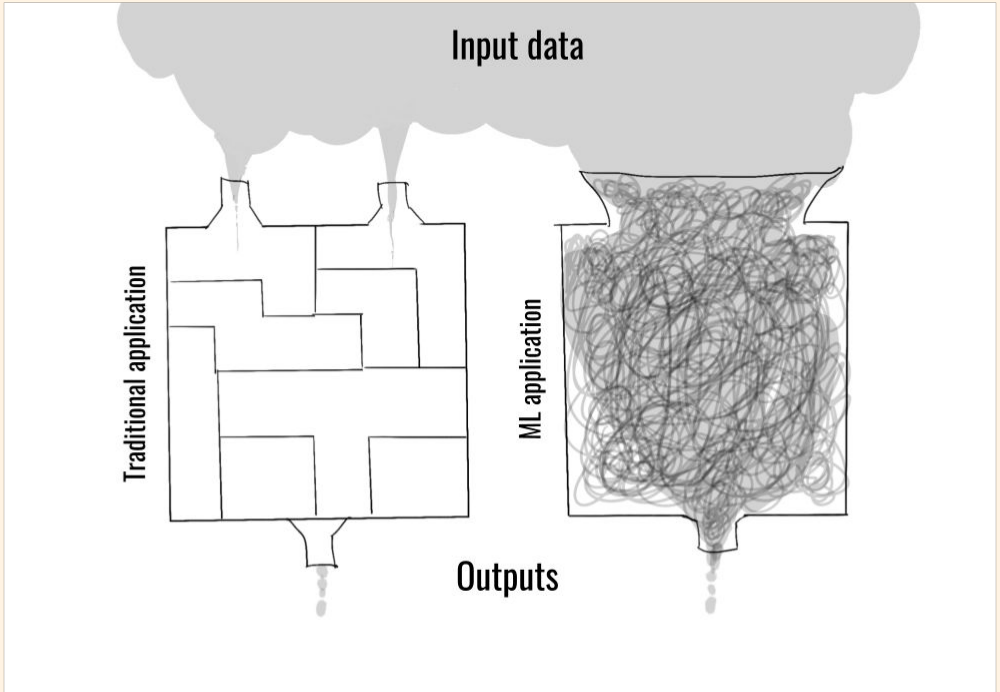
Is MLOps necessary?

1. **Why** does ML need special treatment in the first place? Can't we just fold it into existing DevOps best practices?
2. **What** does a modern technology stack for streamlined ML processes look like?
3. **How** can you start applying the stack in practice today?

Is MLOps necessary?

1. **Why** does ML need special treatment in the first place? Can't we just fold it into existing DevOps best practices?
2. **What** does a modern technology stack for streamlined ML processes look like?
3. **How** can you start applying the stack in practice today?

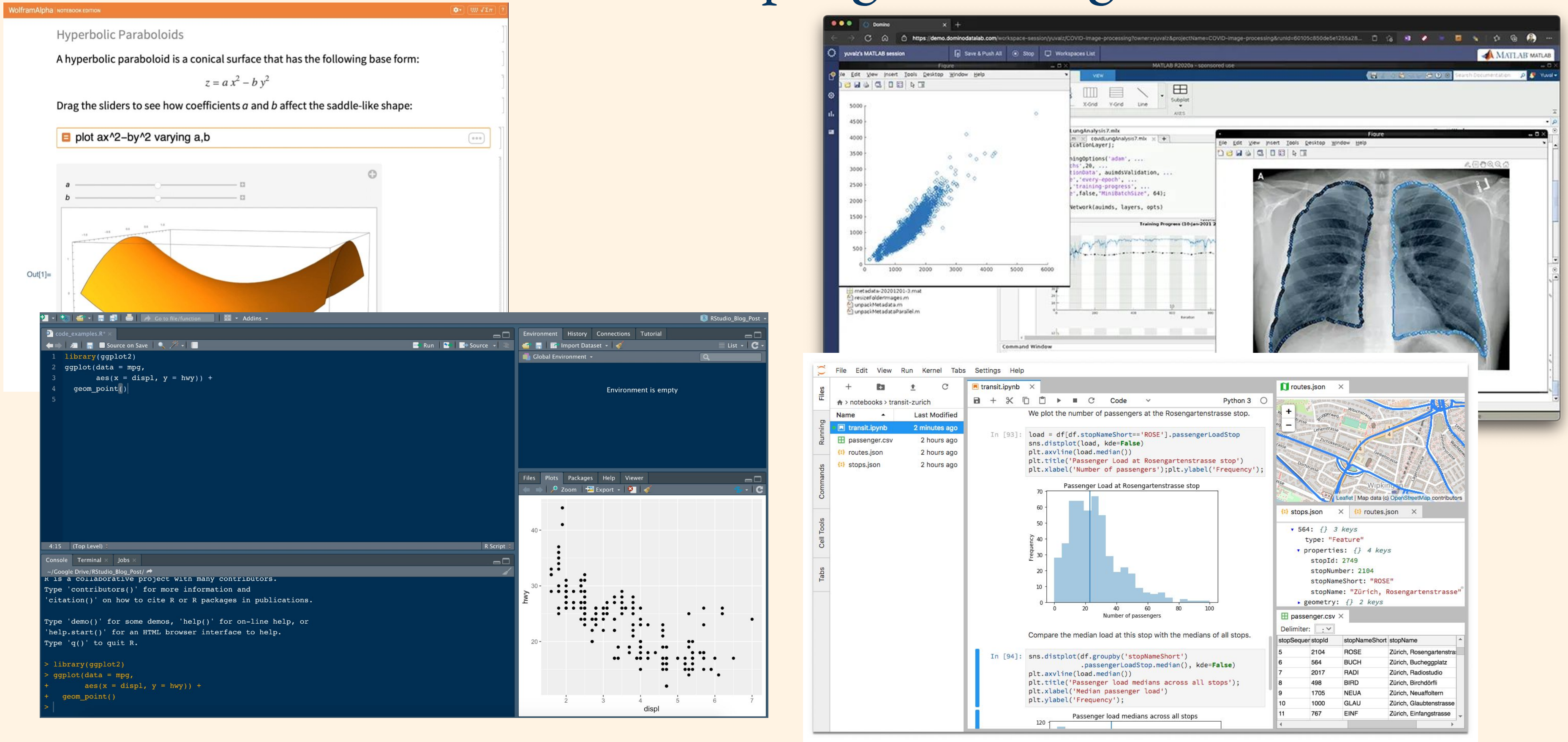
ML-powered applications



Data changes everything

1. ML applications are directly exposed to the constantly changing real world through data,
2. ML apps need to be developed through cycles of experimentation, and
3. The skillset and the background of people building the applications get realigned.

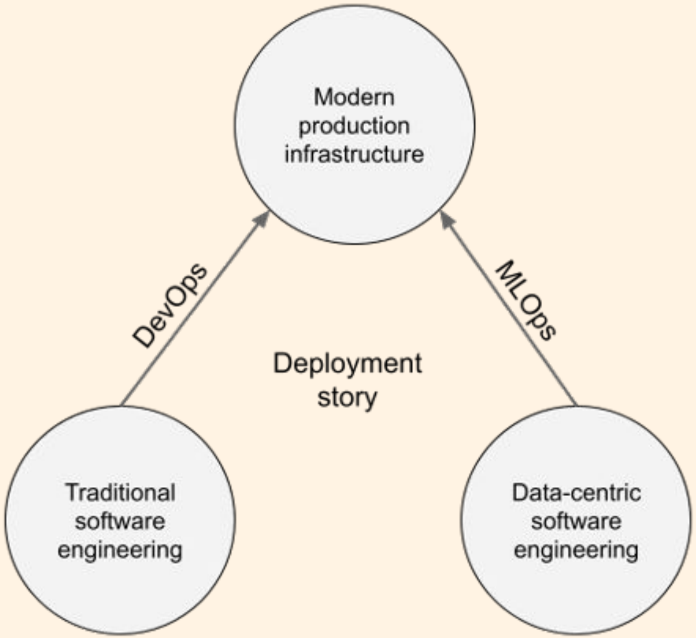
Data-centric programming!



Production-Ready ML Applications?

1. **The scale of operations:** often two orders of magnitude larger than in the earlier data-centric environments;
2. Modern ML applications need to be **carefully orchestrated**
3. **We need robust versioning** for data, models, code, and preferably even the internal state of applications
4. The applications must be **integrated to the surrounding business systems**

Production-Ready ML Applications?



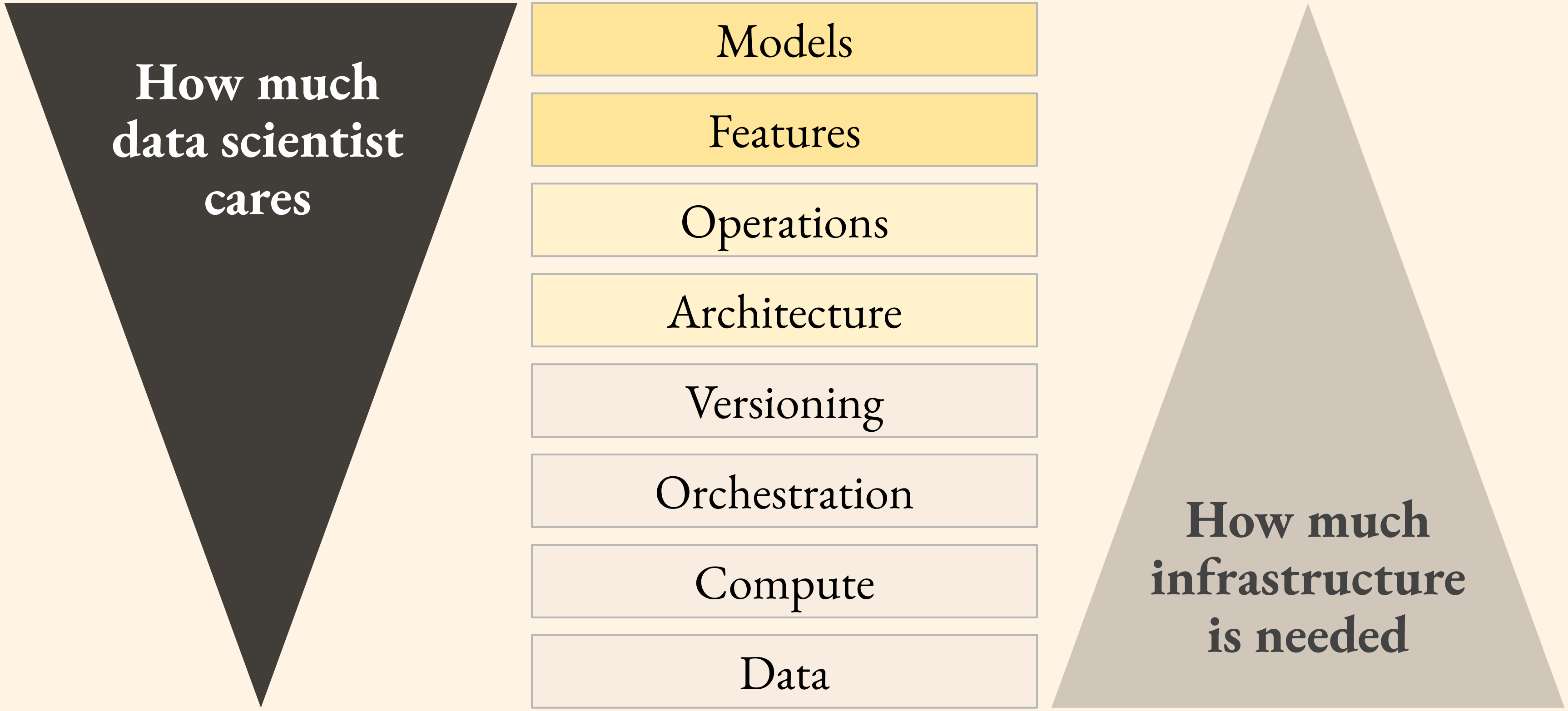
Is MLOps necessary?

1. **Why** does ML need special treatment in the first place? Can't we just fold it into existing DevOps best practices?
2. **What** does a modern technology stack for streamlined ML processes look like?
3. **How** can you start applying the stack in practice today?

The modern ML stack

Adapted from the book *Effective Data Science Infrastructure*

Scientists and the Modern ML stack

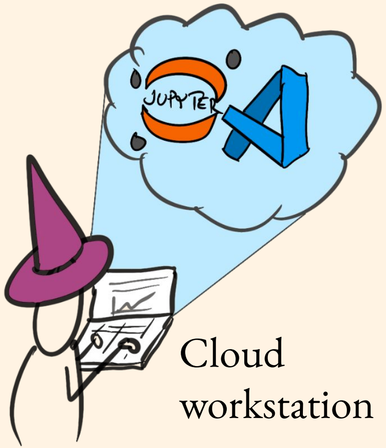


Let's see this at work:

a day in the life of a data scientist

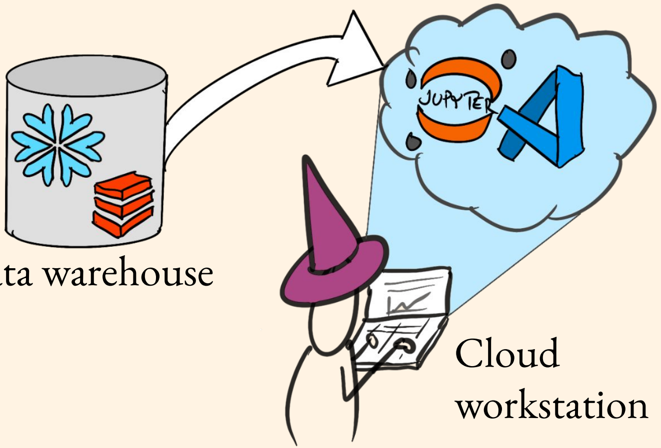
Here’s a data scientist

A modern data scientist uses a cloud workstation



Cloud
workstation

Data flows seamlessly from the data warehouse to the workstation

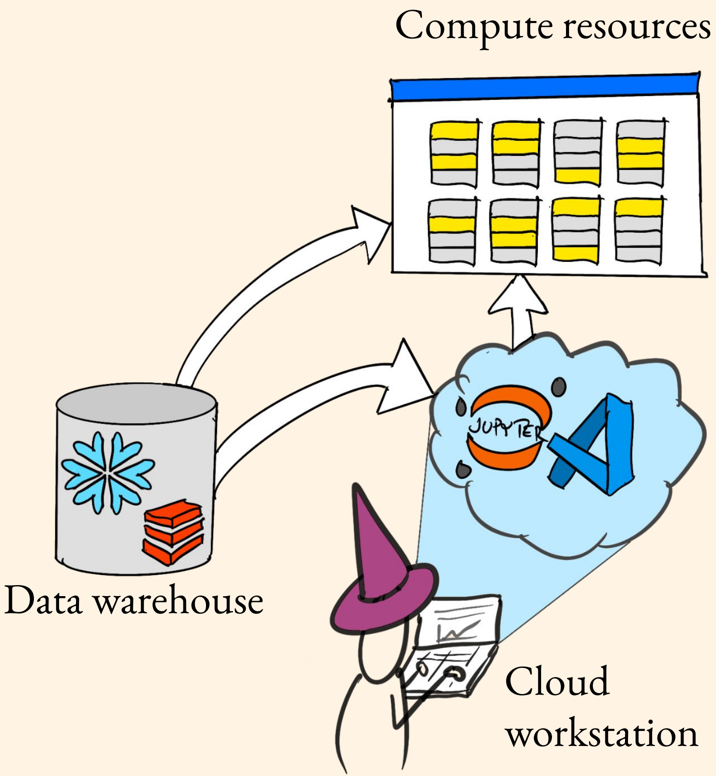


Data warehouse

Cloud workstation

Data

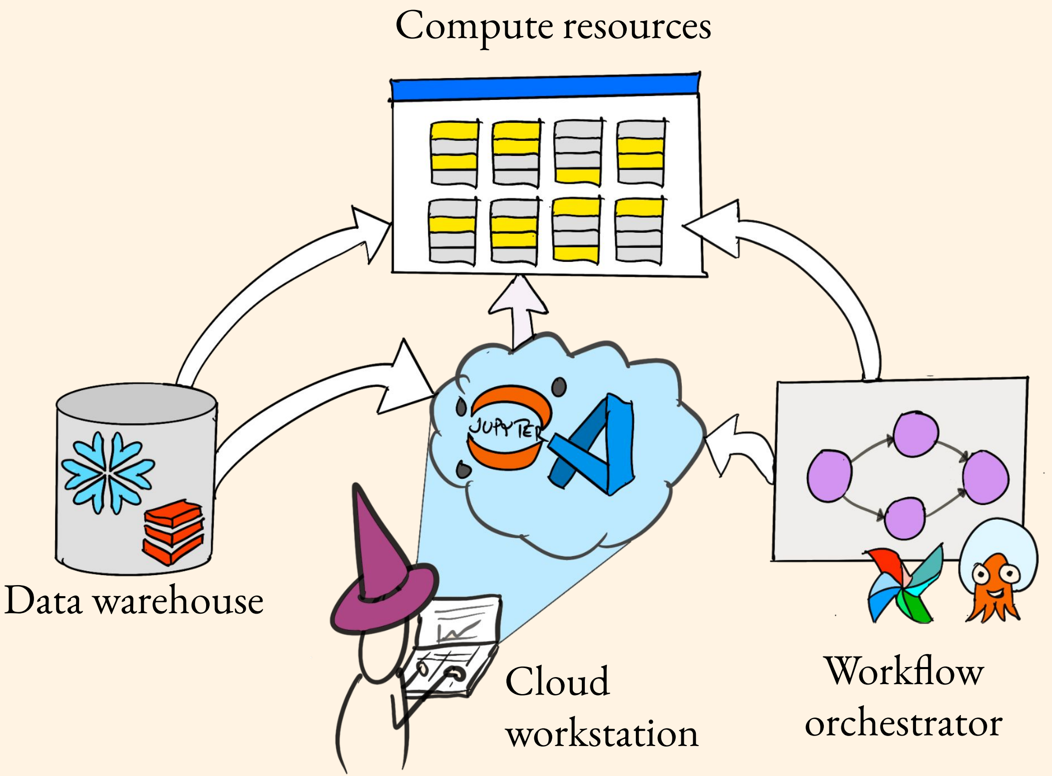
Experiments run at scale on a cloud-based compute cluster

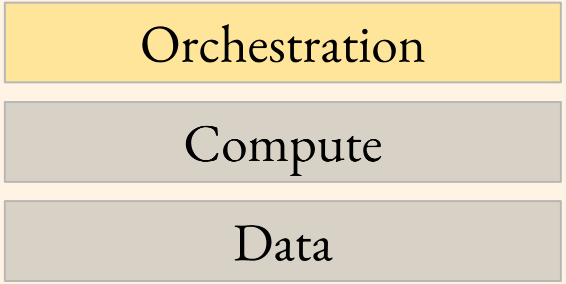


Compute

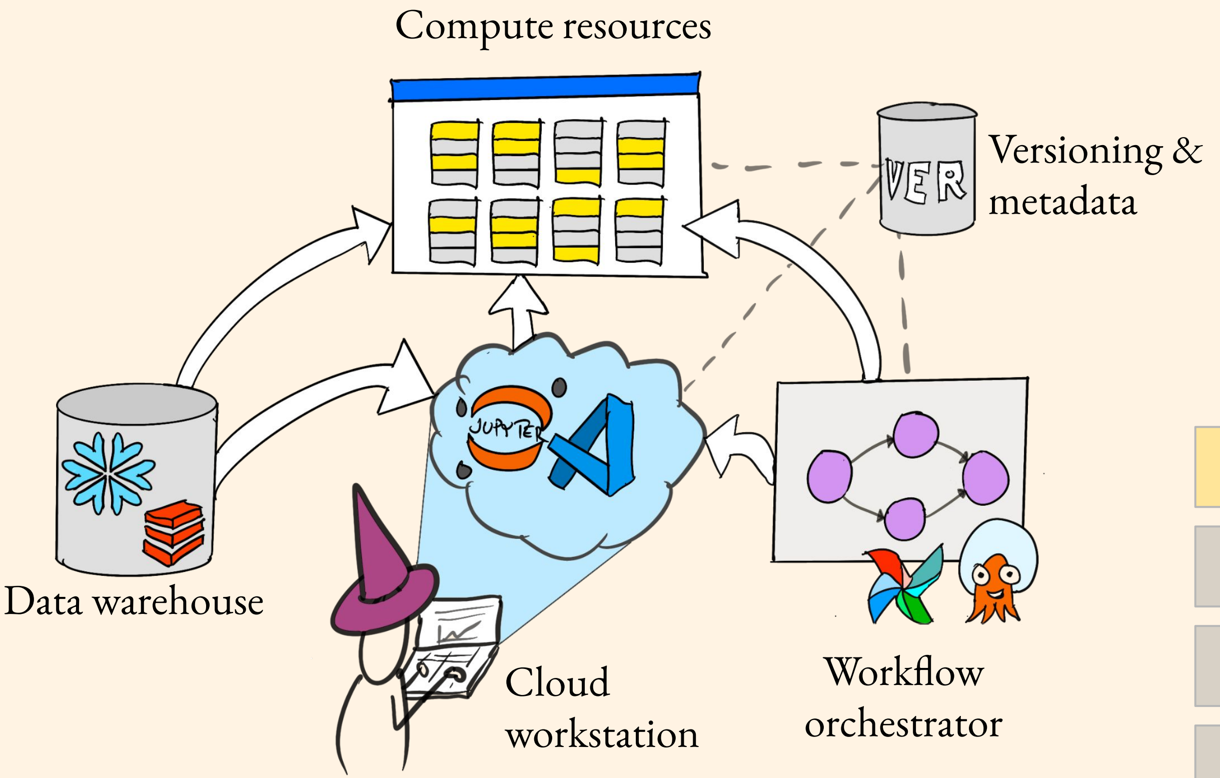
Data

Complete workflows are developed and tested locally



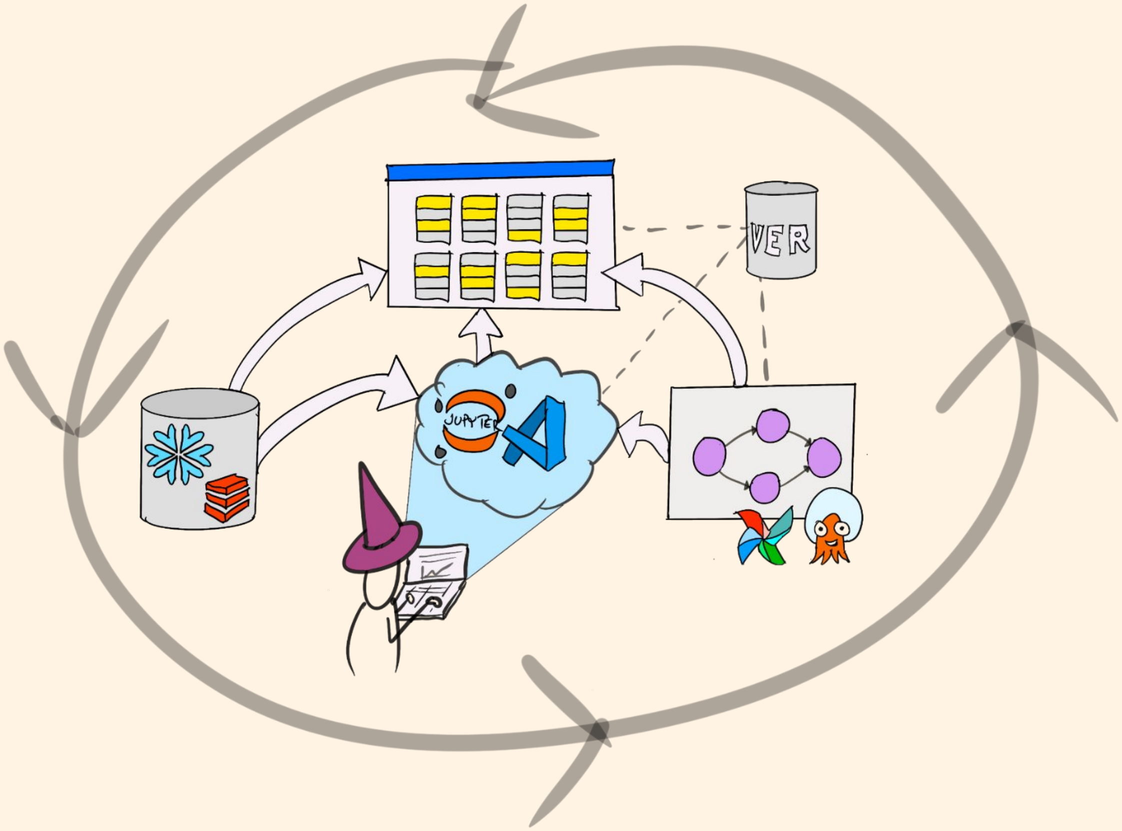


Code, models, logs, and metrics gets stored and versioned automatically



Versioning
Orchestration
Compute
Data

Data Scientist can develop, test, and iterate on projects rapidly



Is MLOps necessary?

1. **Why** does ML need special treatment in the first place? Can't we just fold it into existing DevOps best practices?
2. **What** does a modern technology stack for streamlined ML processes look like?
3. **How** can you start applying the stack in practice today?

Applying the ML Stack

1. Does the solution provide a delightful user experience for data scientists and ML engineers?
2. Does the solution provide first-class support for rapid iterative development and frictionless A/B testing?
3. Does the solution integrate with your existing infrastructure, in particular to the foundational data, compute, and orchestration layers?

Production-Grade Frameworks

METAFLOW

Vertex AI

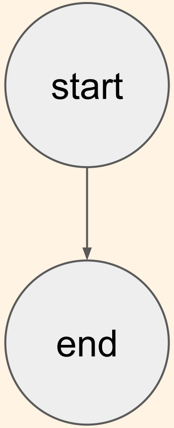
Amazon SageMaker
aws

Azure
Machine Learning
Studio

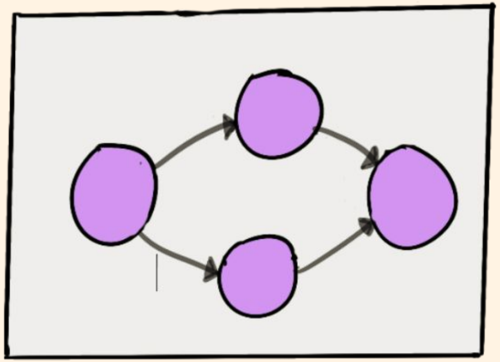
Example

METAFLOW

Define workflows with a human-friendly syntax

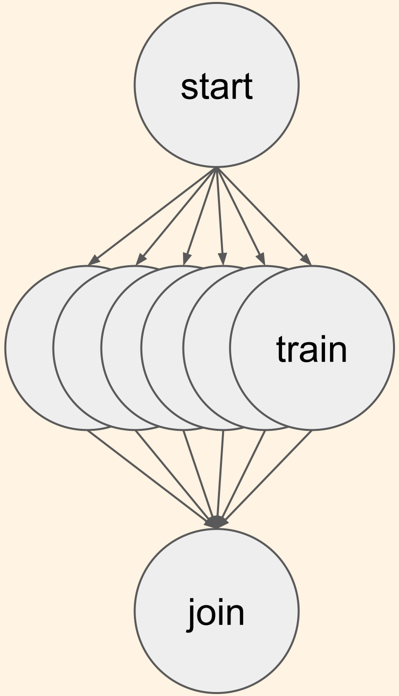



```
class MyFlow(FlowSpec):

    @step
    def start(self):
        import pandas as pd
        pd.DataFrame(big_one)
        self.next(self.end)

    @step
    def end(self):
        pass
```

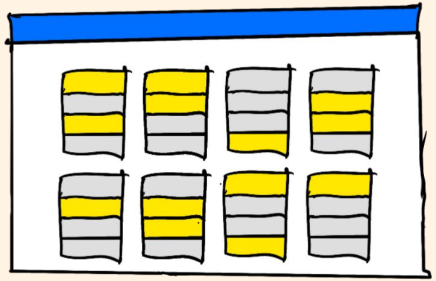


```
# python myflow.py run
```

Experiments run at scale on a cloud-based compute cluster

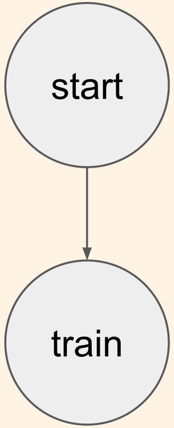



```python
@step
def start(self):
    self.params = list(range(100))
    self.next(self.train, foreach='params')

@resources(memory=128000)
@step
def train(self):
    self.model = train(...)
    self.next(self.join)

@step
def join(self, inputs):
    ...
```



```
# python myflow.py run —with kubernetes
```

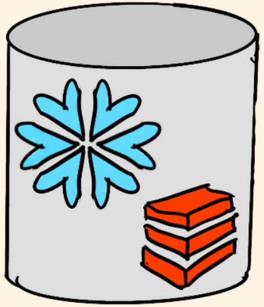
Everything gets versioned automatically



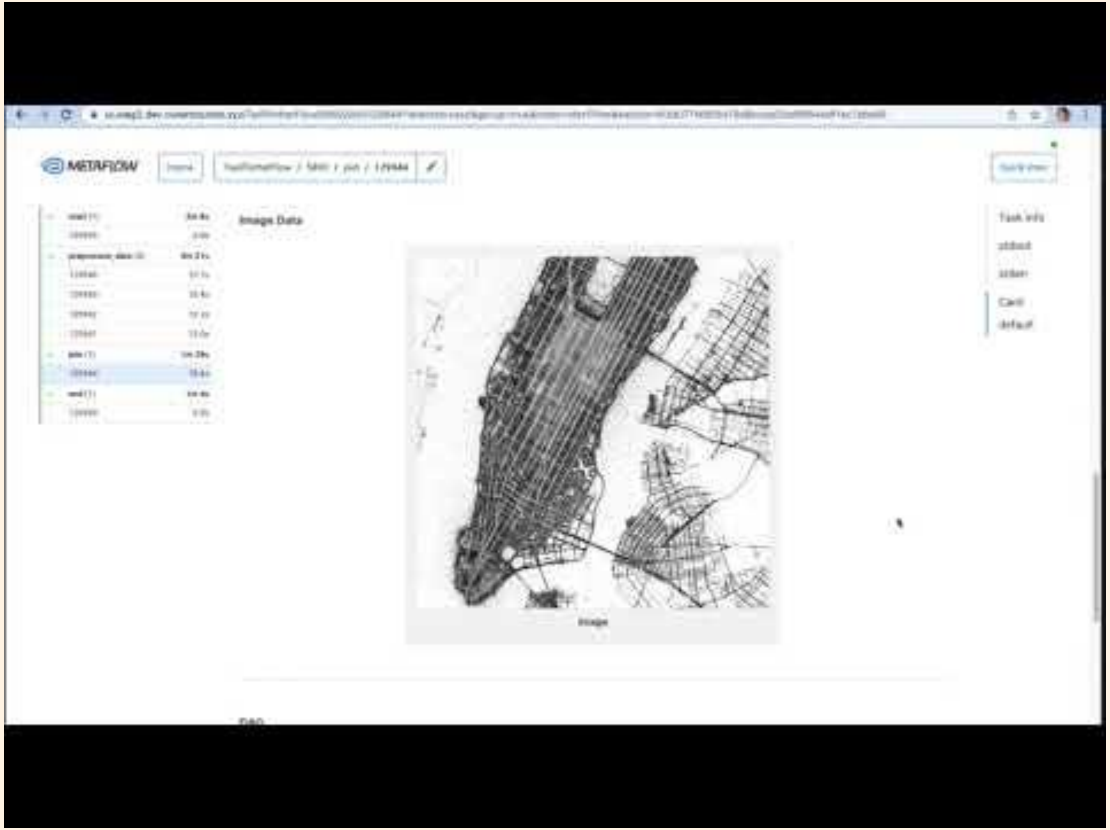

```
class MyFlow(FlowSpec):

    @step
    def start(self):
        self.alpha = 0.5
        self.next(self.train)

    @step
    def train(self):
        self.model = train_model(self.alpha)
```

Comes with tools for fast data access

```
class QueryFlow(FlowSpec):
    @step
    def query(self):
        self.ctas = "CREATE TABLE %s AS %s" % (self.table, self.sql)
        query = wr.athena.start_query_execution(self.ctas)
        output = wr.athena.wait_query(query)
        loc = output['ResultConfiguration']['OutputLocation']
        with metaflow.S3() as s3:
            results = [obj.url for obj in s3.list_recursive([loc])]
```



Data Scientist can develop, test, and iterate on projects rapidly



Thank you

Let’s chat and feel free to get in touch with me at

http://slack.outerbounds.co

outerbounds