
AWS
re:Invent

OPN308

Designing better ML systems: Learnings from Netflix

Savin Goyal
Software Engineer
Netflix

Rob Hilton
Principal Solutions Architect
AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



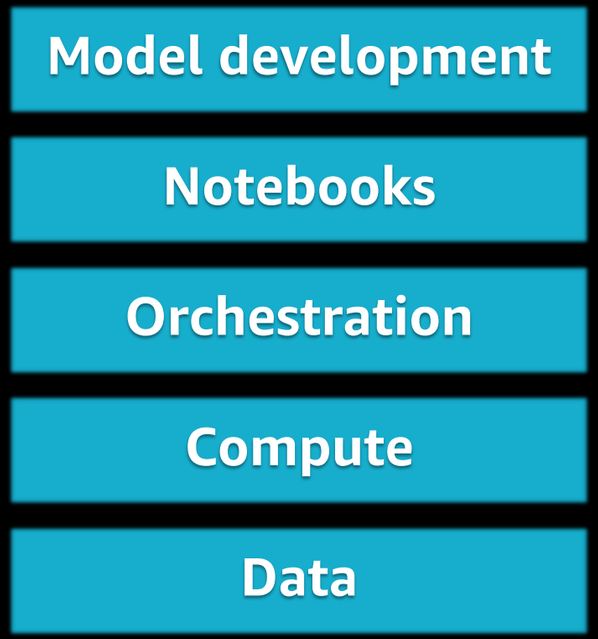
How do we improve our ML systems?

More data
Bigger models

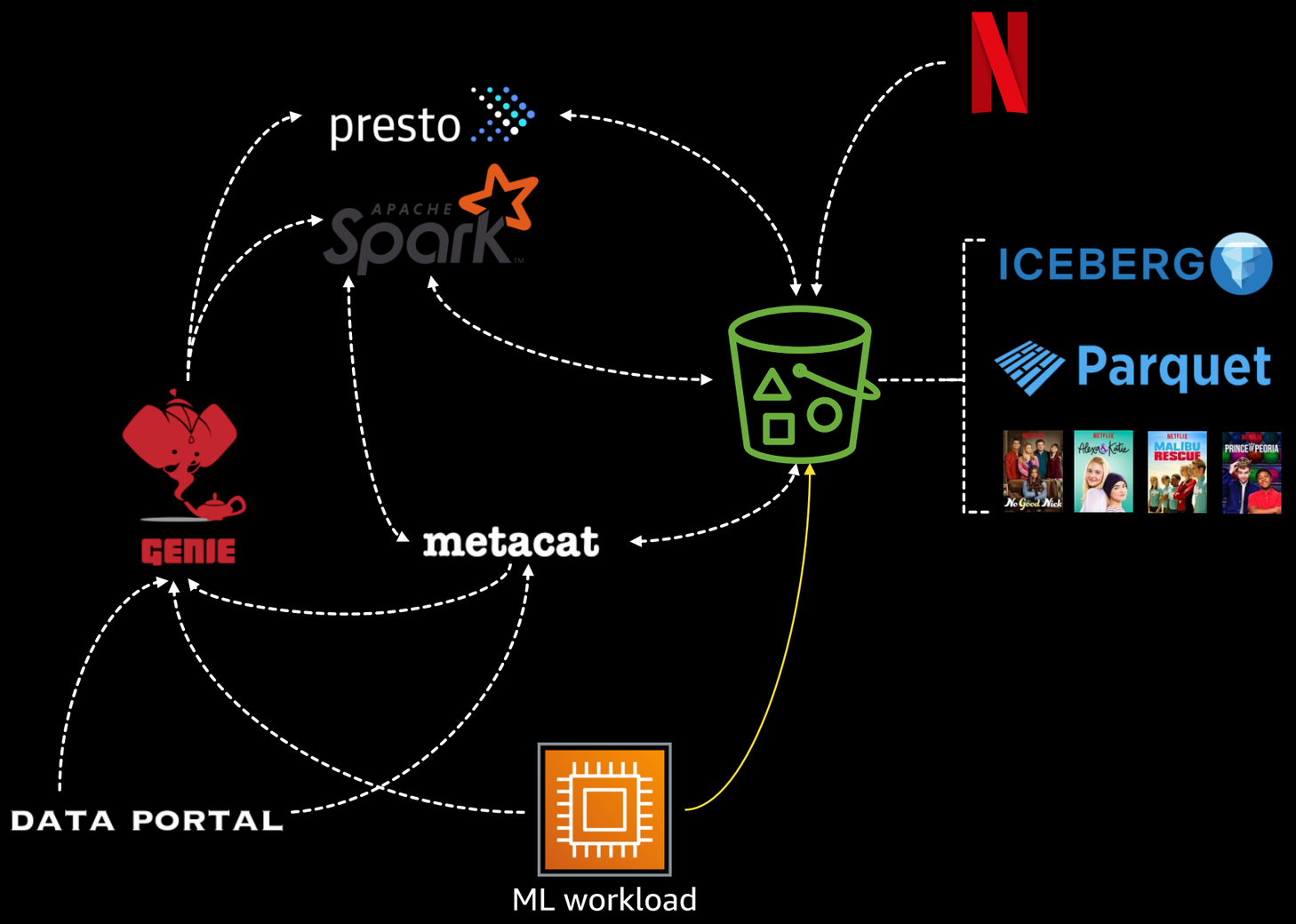
How do we improve our ML systems?

More data
Bigger models
Better infrastructure

Infrastructure stack for modern ML systems

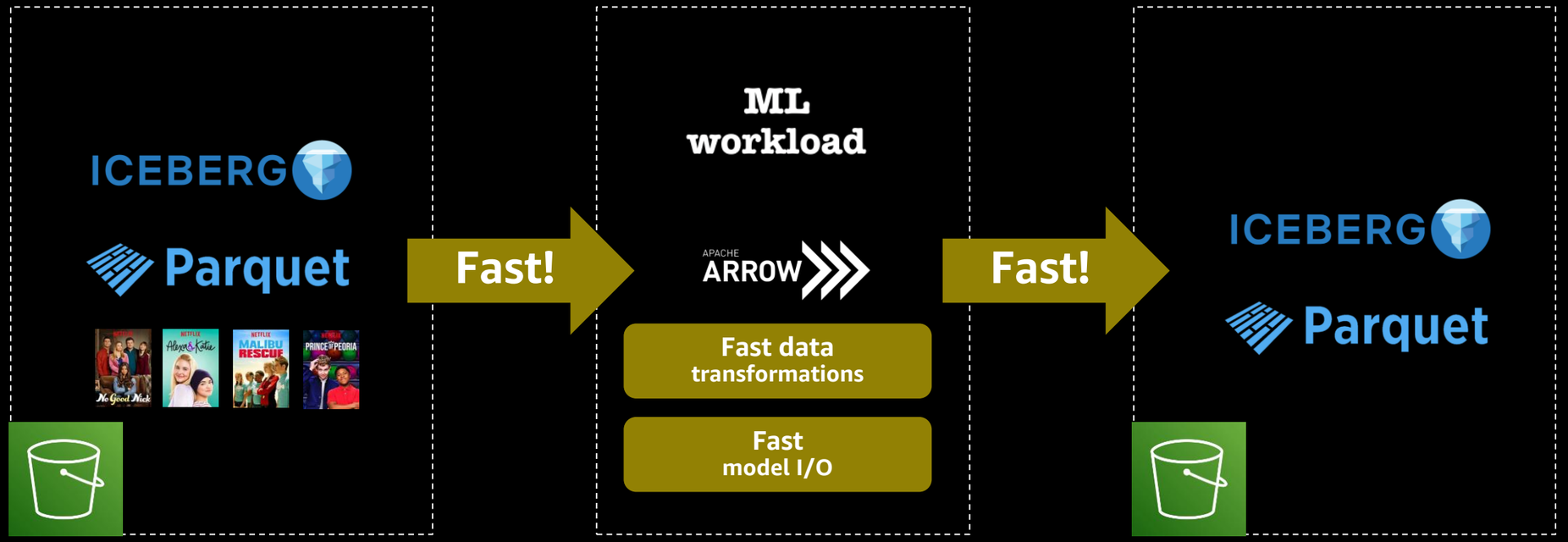


Amazon S3

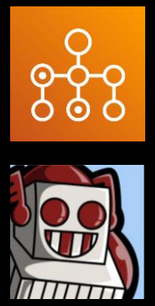

presto
APACHE Spark
ICEBERG
Parquet
GENIE
metacat
DATA PORTAL
ML workload


Data

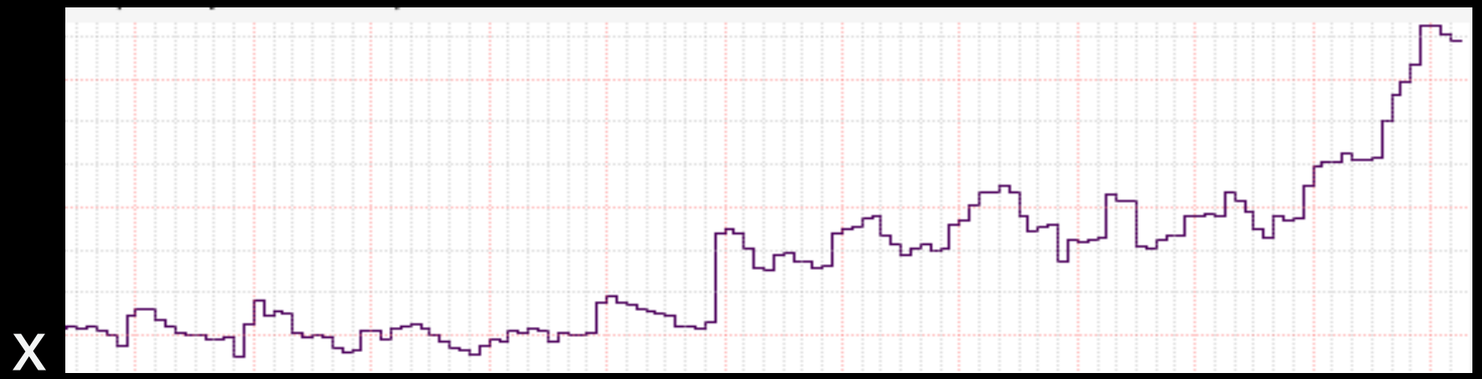
Fast data



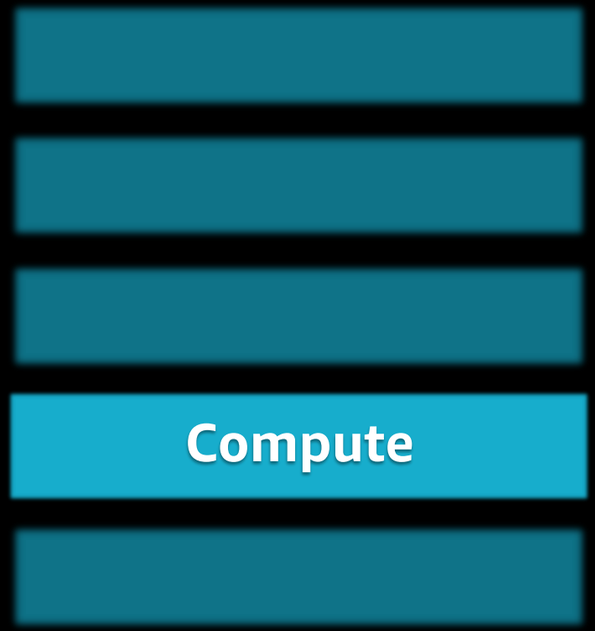
Quickly read / write / process TBs of data in Python!



AWS Batch
Netflix Titus

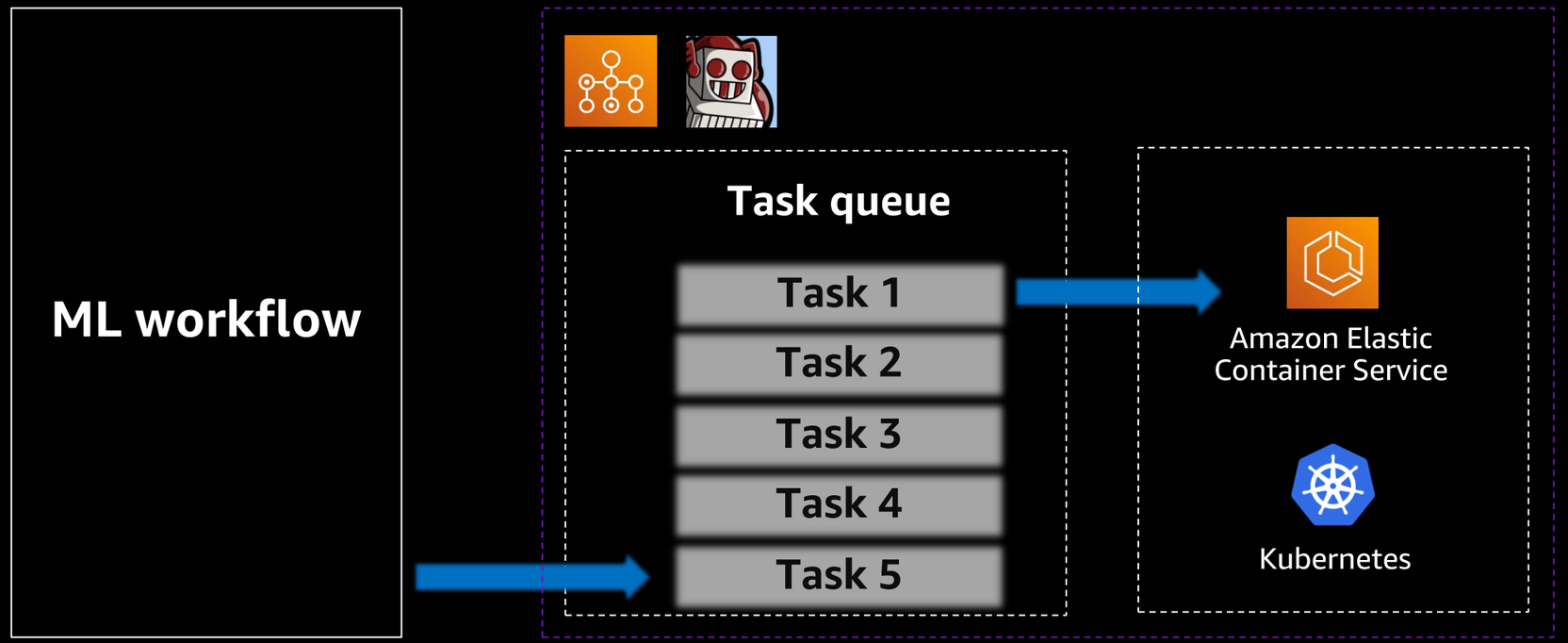


r4.16xlarge equivalents demand chart



ML workloads can be resource intensive

AWS Batch
Netflix Titus



Careful resource management

Cost efficiency is a key concern

Hard to estimate resource requirements – predictive bin packing!





AWS Step Functions
Netflix Conductor

ML workflows
- Can be long running
- Can quickly multiply (a/b tests, parameters …)

Decouple where (compute) from how (orchestration)



Amazon EventBridge
Netflix Conductor

ML workflows are not isolated islands

Use data signals to connect workflow to upstream and downstream ETLs



By Nikos Roussos. Farø bridges from above, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=17169933



Amazon SageMaker notebooks



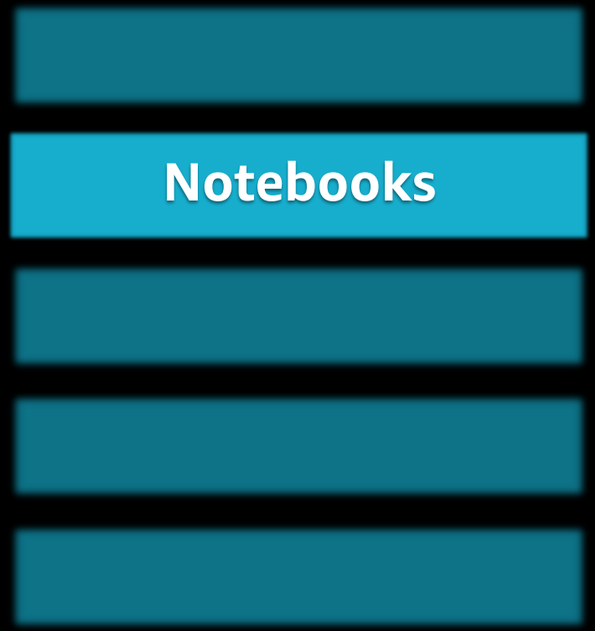
Jupyter Notebooks



Netflix Polynote

Great for exploring and prototyping ideas

Equally great for monitoring ML workflows

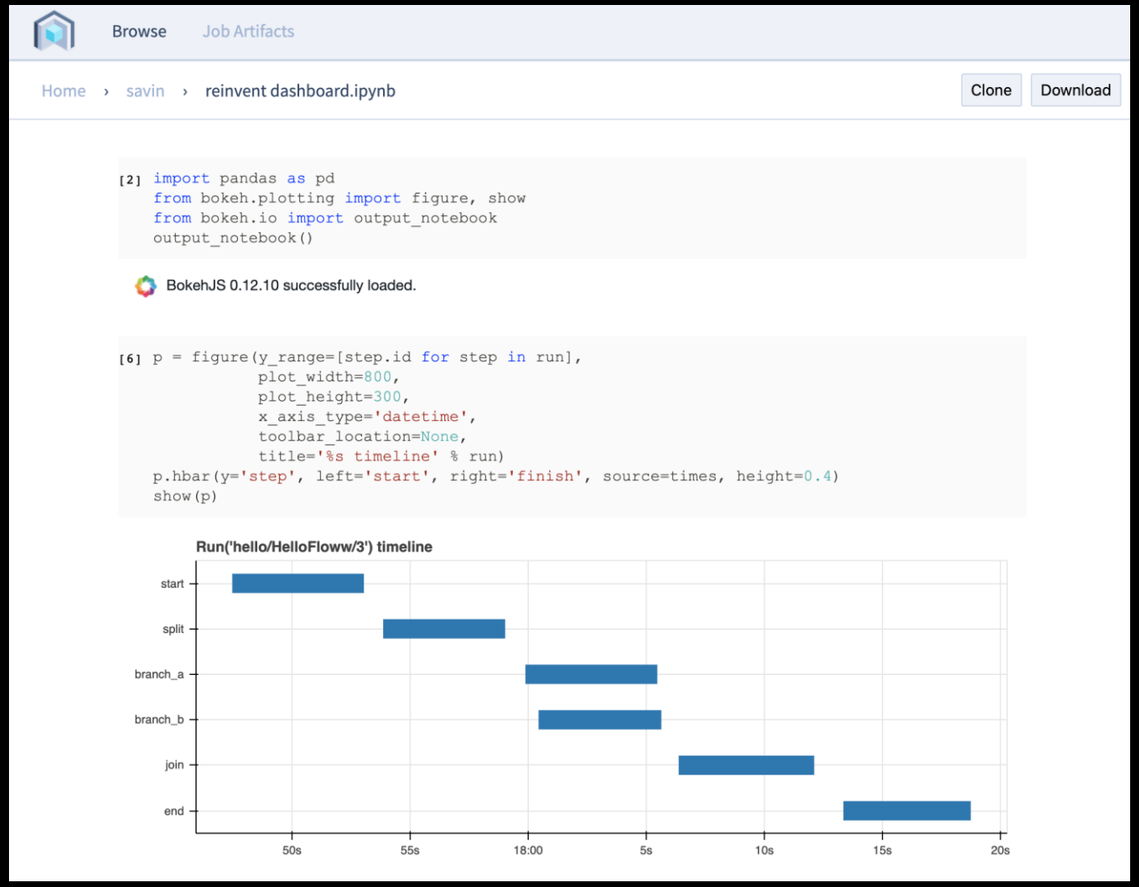




Papermill

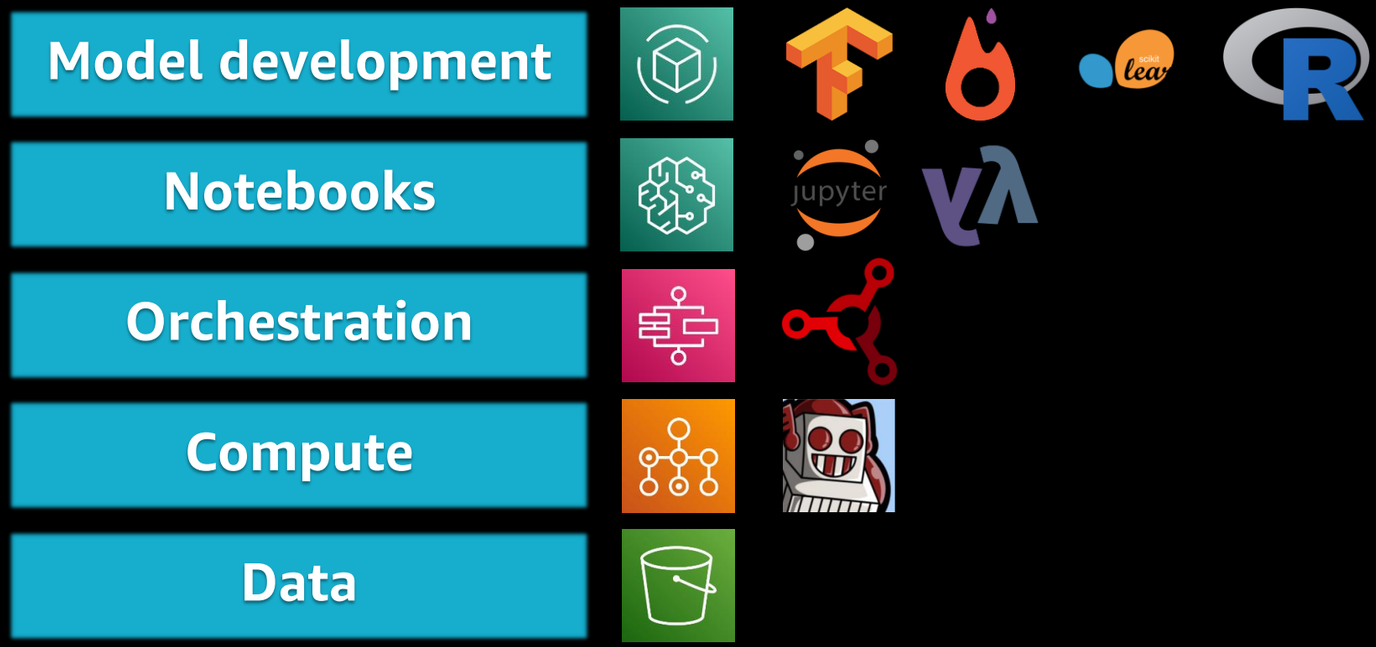


Commuter



Custom dashboards for every ML workflow execution

Infrastructure stack for modern ML systems



But how do I structure my ML project?



Model development

Notebooks

Orchestration

Compute

Data





Model development

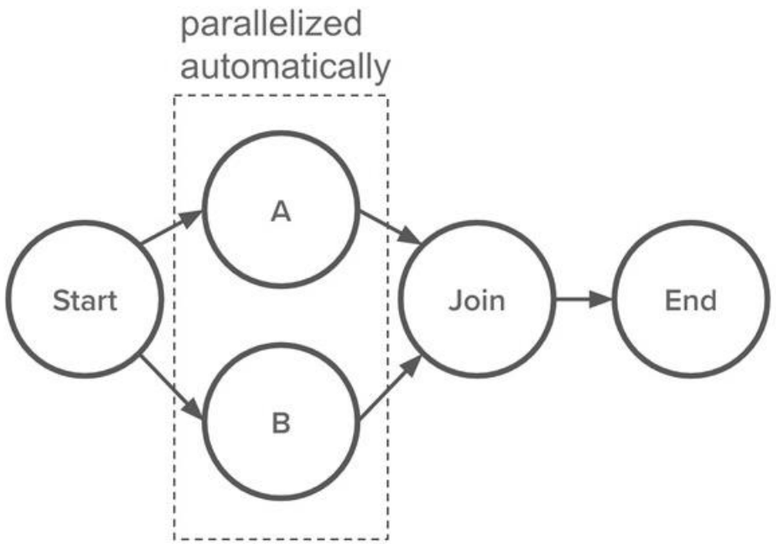
Notebooks

Orchestration

Compute

Data

1. Idiomatic Python code




```
class MyFlow(FlowSpec):

  @step
  def start(self):
    self.next(self.a, self.b)

  @step
  def a(self):
    self.next(self.join)

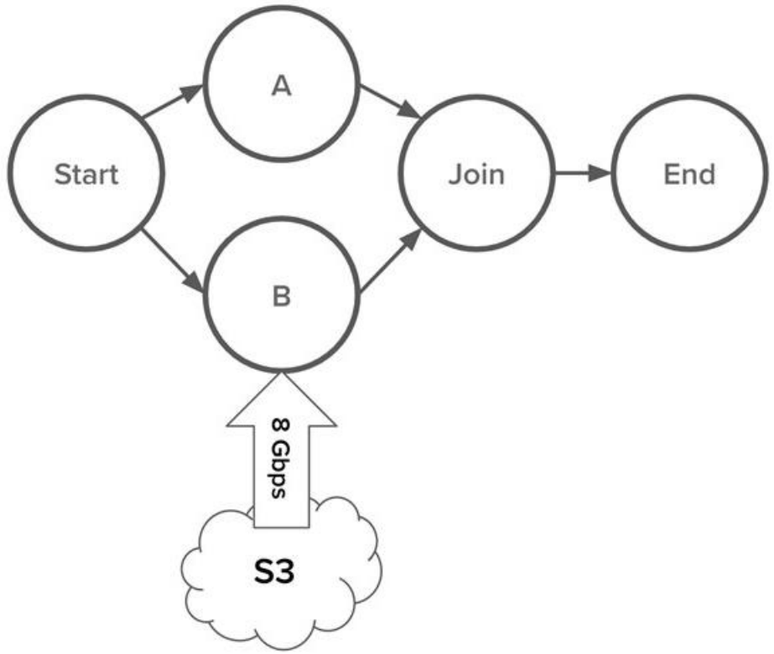
  @step
  def b(self):
    self.next(self.join)

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

1. Idiomatic Python code

```
class MyFlow(FlowSpec):
```

2. High-throughput data layer



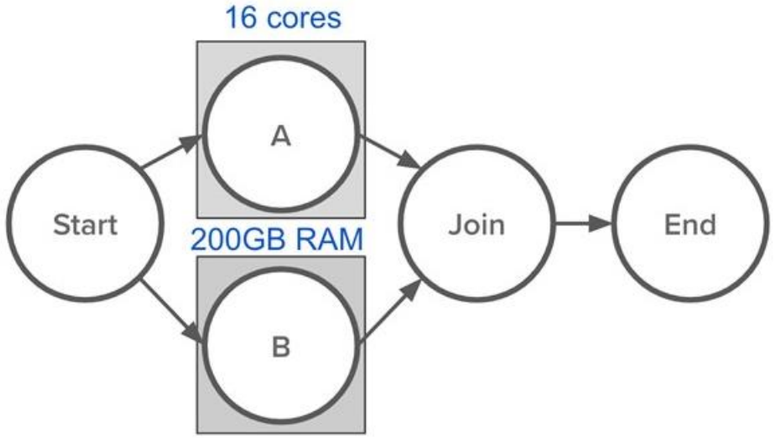

```
@step
def b(self):
    # - highly parallel S3 download
    # - PyArrow reader
    # - avoid conversion to Pandas
    df = MetaflowDataFrame\
         .from_parquet(parquet_files())
    self.next(self.join)
```

1. Idiomatic Python code

```
class MyFlow(FlowSpec):
```

2. High-throughput data layer

3. Easily scale out compute







```
@titus(cpu=16)
@step
def a(self):
  self.next(self.join)

@titus(memory=200000)
@step
def b(self):
  self.next(self.join)
```

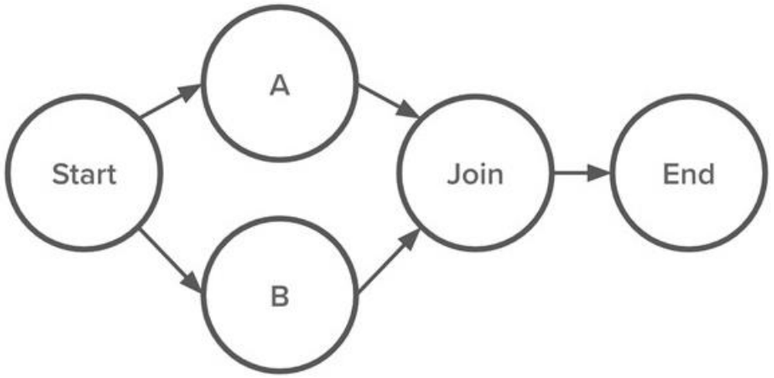
1. Idiomatic Python code

```
class MyFlow(FlowSpec):
```

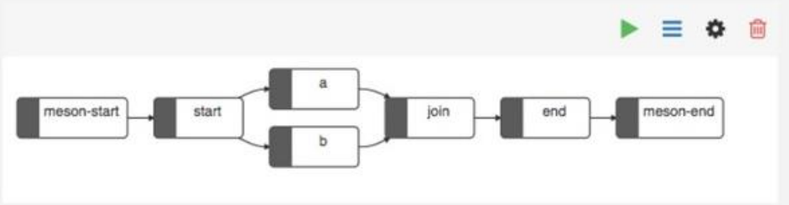
2. High-throughput data layer

3. Easily scale out compute

4. Production deployments



conductor



1. Idiomatic Python code

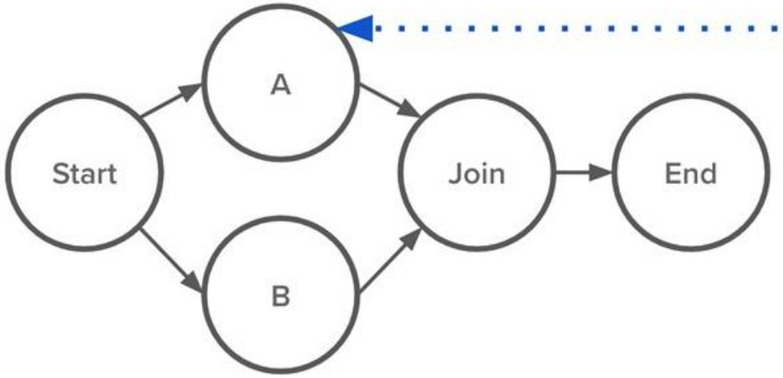
2. High-throughput data layer

3. Easily scale out compute

4. Production deployments

5. Metadata, monitoring & debugging

```
class MyFlow(FlowSpec):
```






```
run = Flow('MyFlow').latest_run

if run['A'].success:
  print(run['A'].data.x)
else:
  print(run['A'].stderr)

print(run['End'].data)
```

1. Idiomatic Python code

```
class MyFlow(FlowSpec):
```

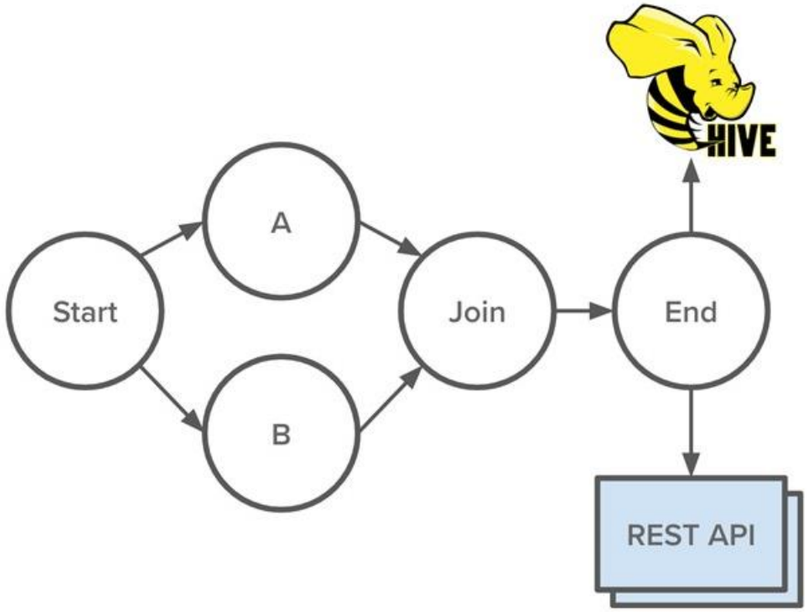
2. High-throughput data layer

3. Easily scale out compute

4. Production deployments

5. Metadata, monitoring & debugging

6. Output interfaces




```
class PredictionService(WebServiceSpec):

    @endpoint(name="predict")
    def predict(self, request_dict):
      data = self.artifacts.flow.best_fit
      model = data.get("estimator")
      ...
      return {'prediction': res}
```

1. Idiomatic Python code

```
class MyFlow(FlowSpec):
```

2. High-throughput data layer

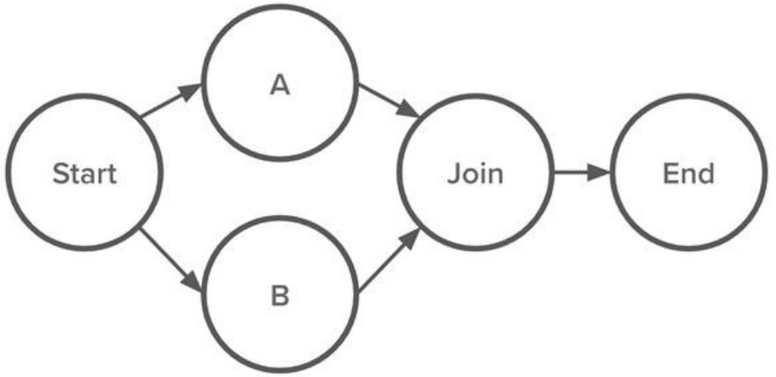
3. Easily scale out compute

4. Production deployments

5. Metadata, monitoring & debugging

6. Output interfaces

7. Polyglot: R




```
metaflow("MyFlow") %>%
  step(
    step = "start",
    next_step = c("a", "b")
  ) %>%
  step(
    step = "A",
    r_function = r_function(a_func),
    next_step = "join"
  ) %>%
  step(
    step = "B",
    r_function = r_function(b_func),
    next_step = "join"
  ) %>%
  step(
    step = "Join",
    r_function = r_function(join,
                            join_step = TRUE),
```





Model development

Notebooks

Orchestration

Compute

Data

Let’s build it!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s build it!

It’s easy – there are only a few small prerequisites to get an environment running that meets AWS best practices

- Create an S3 bucket
- Create an AWS Batch compute environment (in EC2 provisioning mode)
- Create an AWS Fargate Amazon ECS cluster
  (Don’t forget to create security groups allowing inbound access on port 80)
- Create an Amazon RDS PostgreSQL DB instance
  (Make sure your security groups allow inbound access from your Amazon ECS cluster)
- Create an … uh … NLB – for Amazon API Gateway
- Wait, did you create an Amazon VPC?
- You need some AWS Identity and Access Management (IAM) roles
  (How many? 11? Yes, 11)
- Create an Amazon DynamoDB table
  (Don’t forget to enable TTL)

Let's build it!

Create an Amazon ECS tas

It's easy – there are only

that meets AWS best p

- Create an S3 buc
- Create an Amaz
- Creat

- Yo
(Ho
- Crea
(Don't

n DynamoDB tab

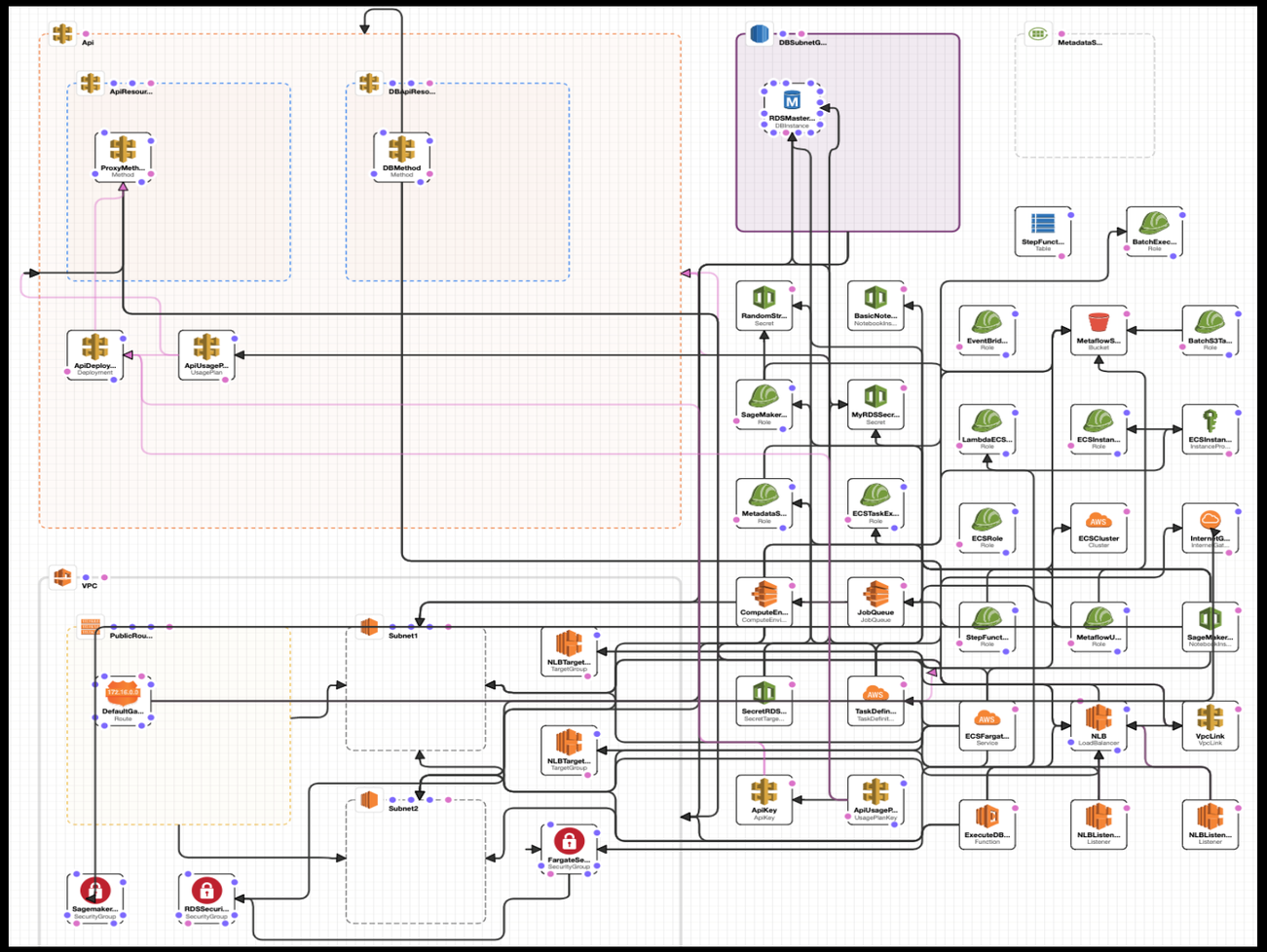
enable TTL)

e's a
mbda in
here
somewhere



Okay, so maybe it’s not quite that simple

Fortunately, AWS and Netflix have done the hard stuff for you

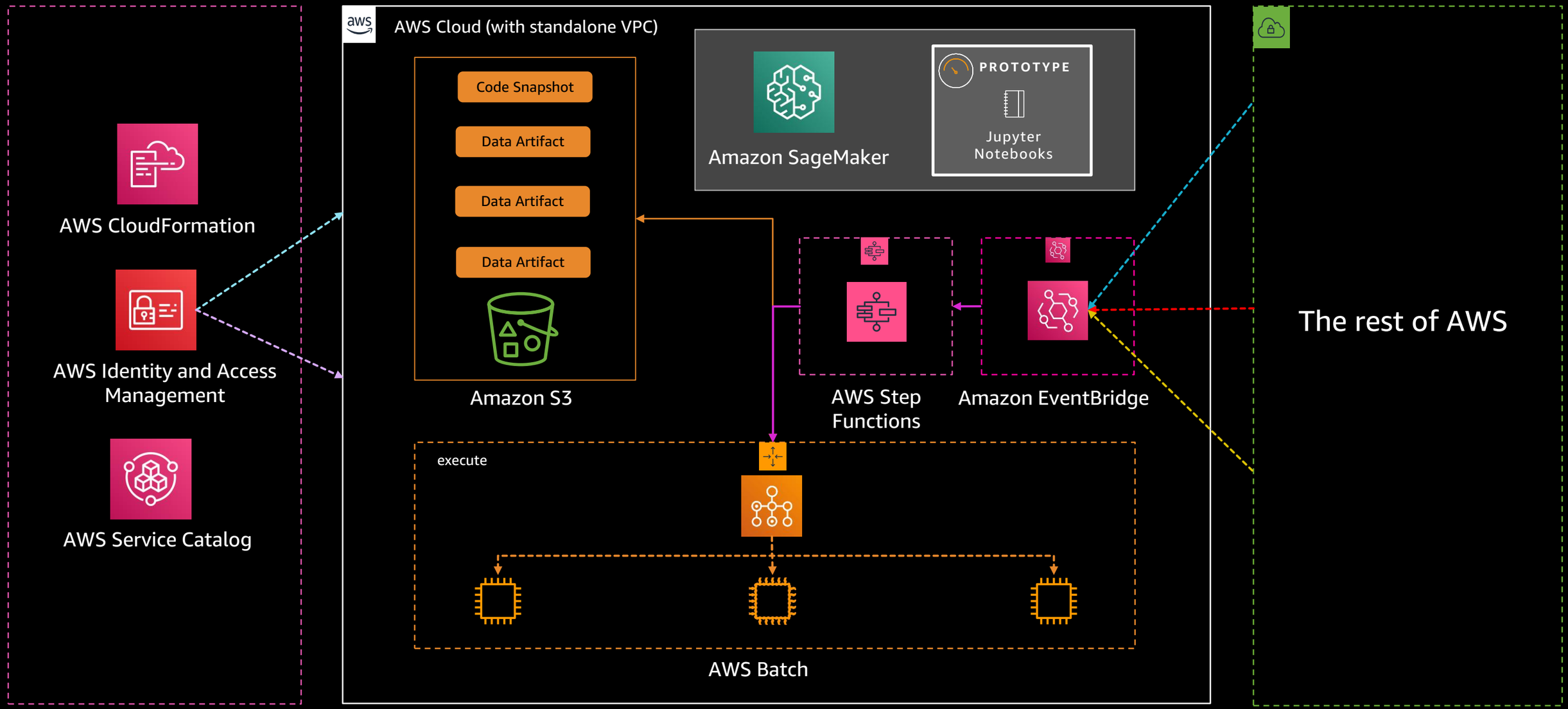


Open-source AWS CloudFormation template

- Flexible – covers multiple configs
- Secure – includes restrictive permissions
- Up-to-date – maintains feature parity

https://github.com/Netflix/metaflow-tools

Let’s dive deep



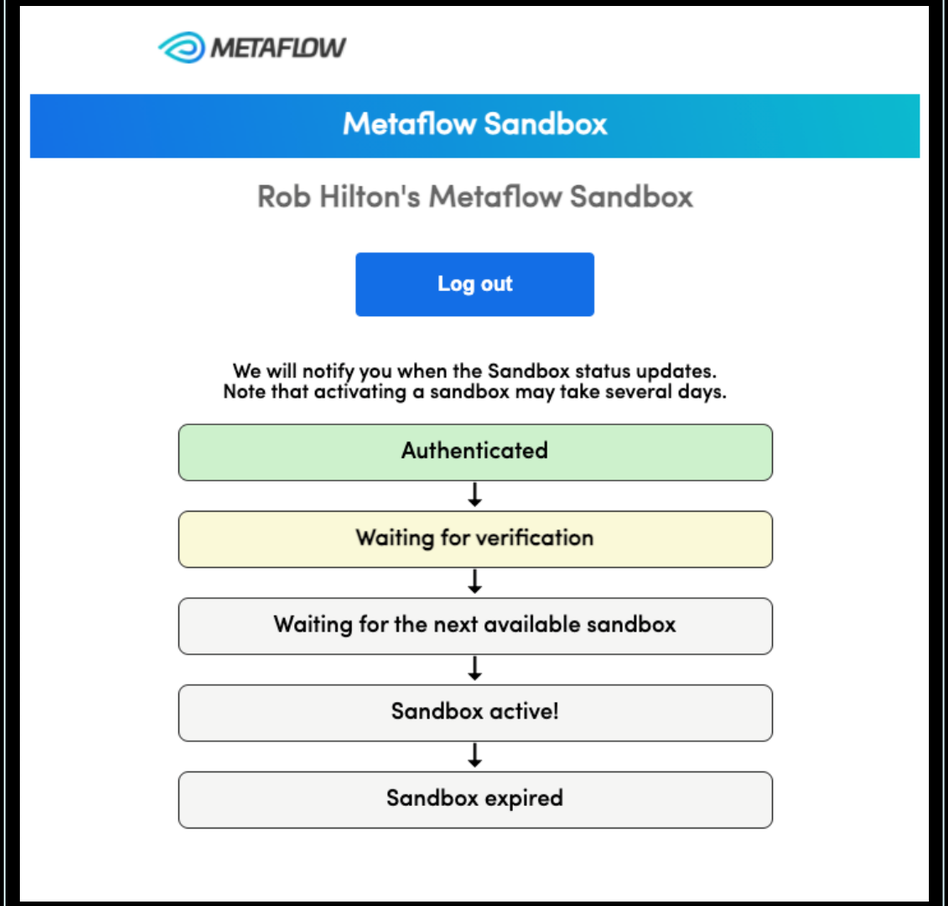
Bonus



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Evaluate on AWS using Metaflow Sandbox

metaflow.org/sandbox



Thank you!

Savin Goyal
savin@netflix.com

Rob Hilton
robhilt@amazon.com



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.