intuit turbotax quickbooks mint

OpML '20

SpoK



(Running Big Data Applications @ Scale on K8s)

Mike Arov, Nagaraj Janardhana

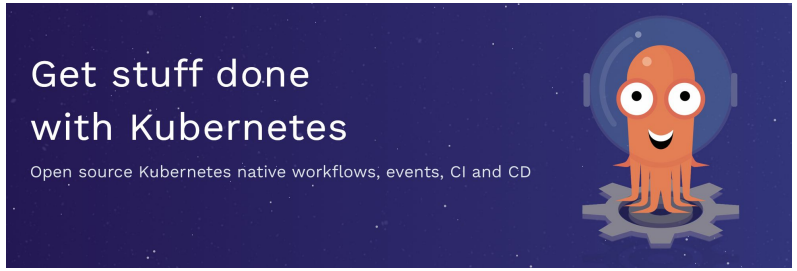
Why Spark on Kubernetes (SpoK)?

Intuit manages Data Engineering on AWS Cloud (S3, EMR)

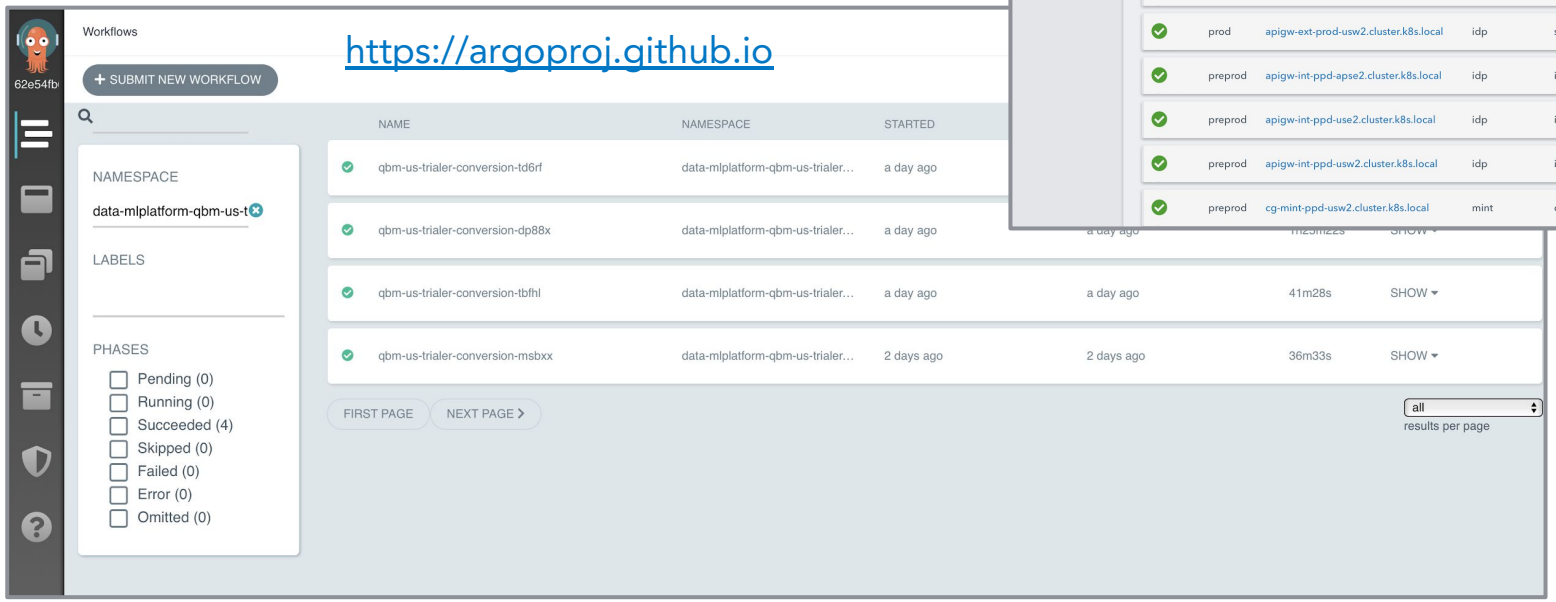
But we also wanted:

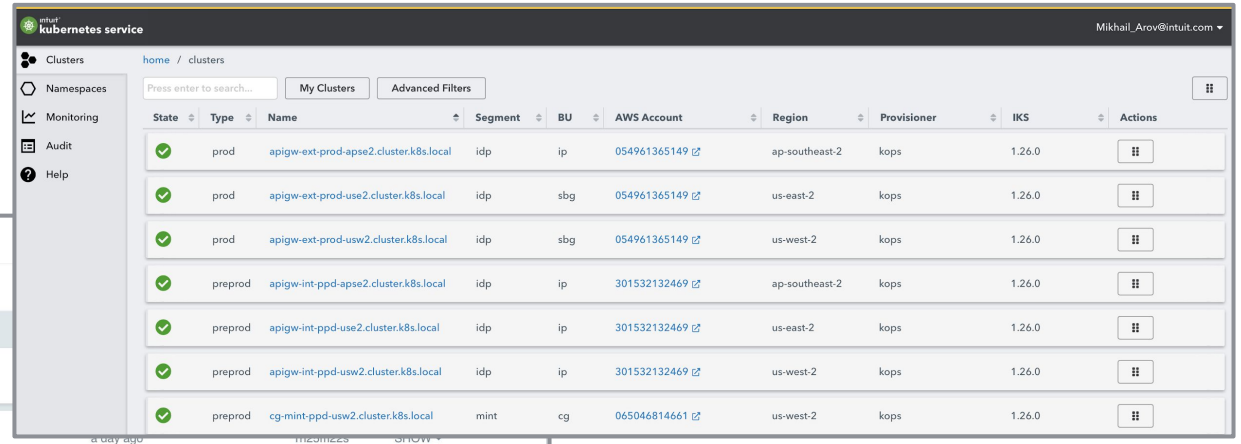
- Integration into the company K8s Infrastructure.
- Containers over Bootstraps.
- Kubernetes over YARN.
- CD for deployment for Data Processing Jobs.

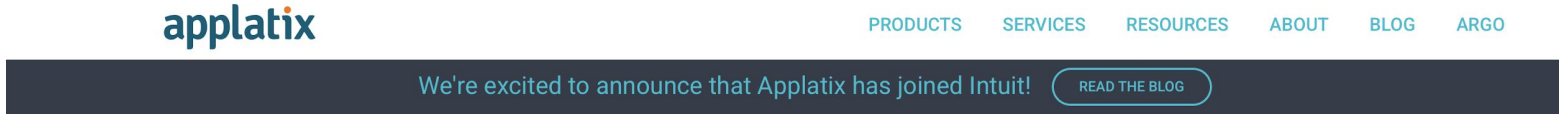
Existing K8s Infrastructure



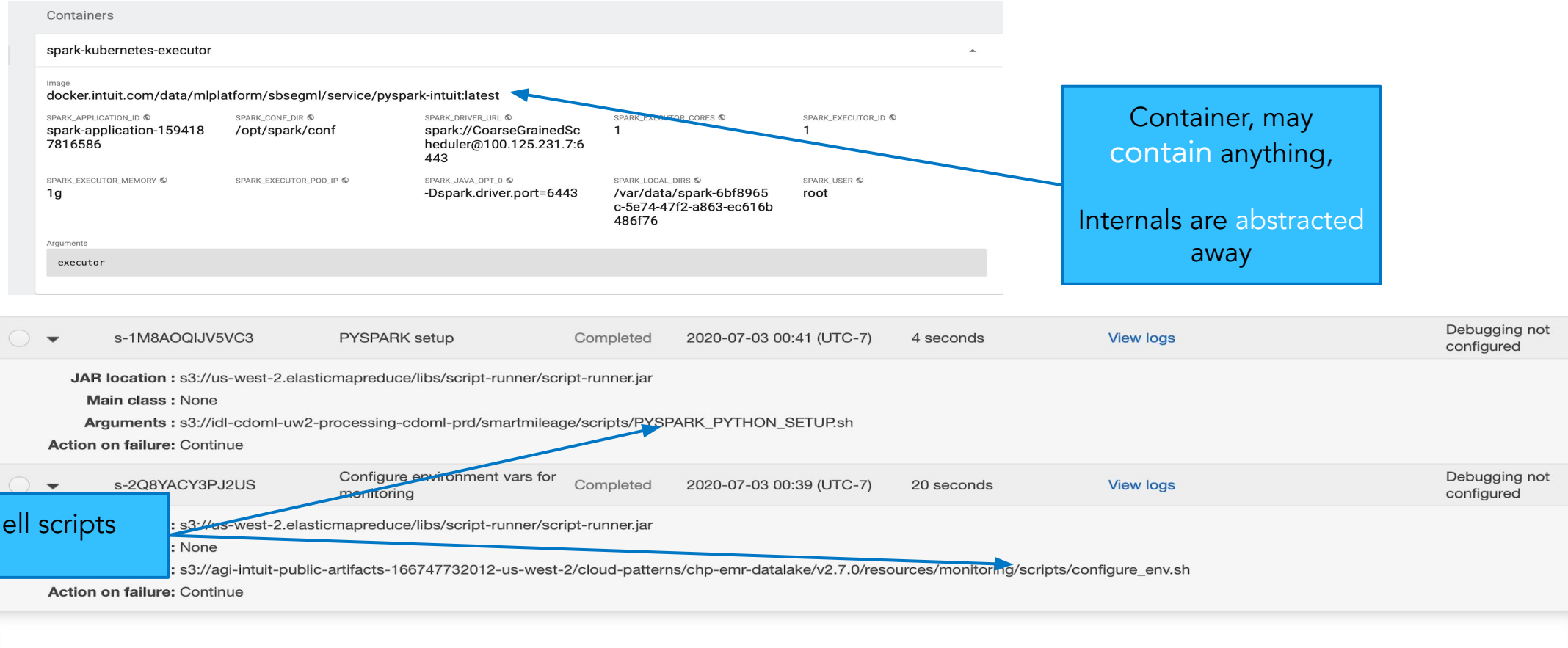
https://argoproj.github.io





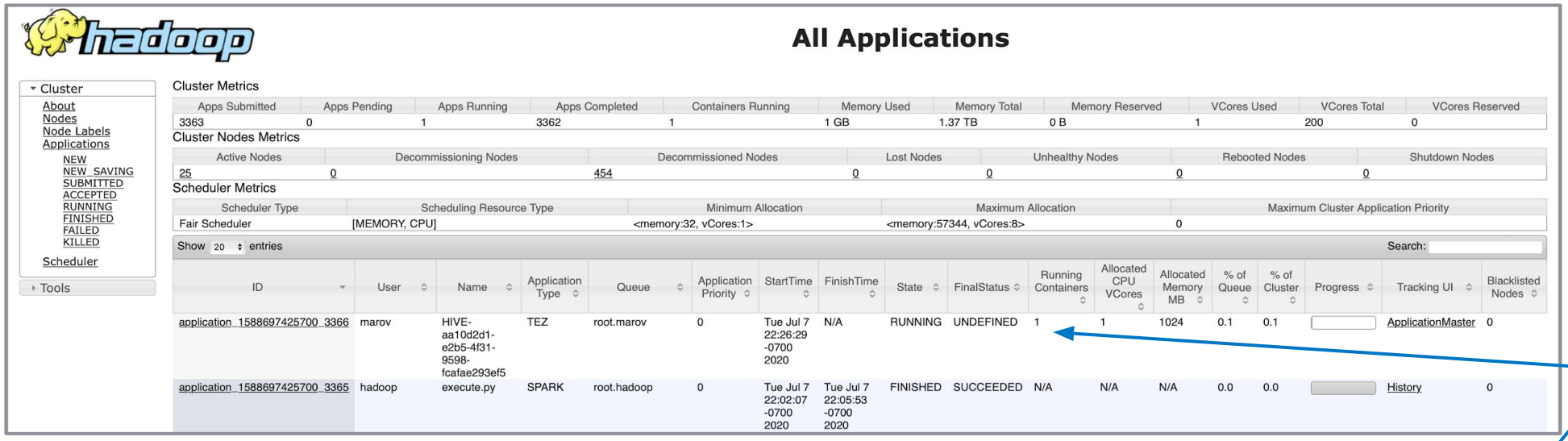


Containers vs Bootstraps

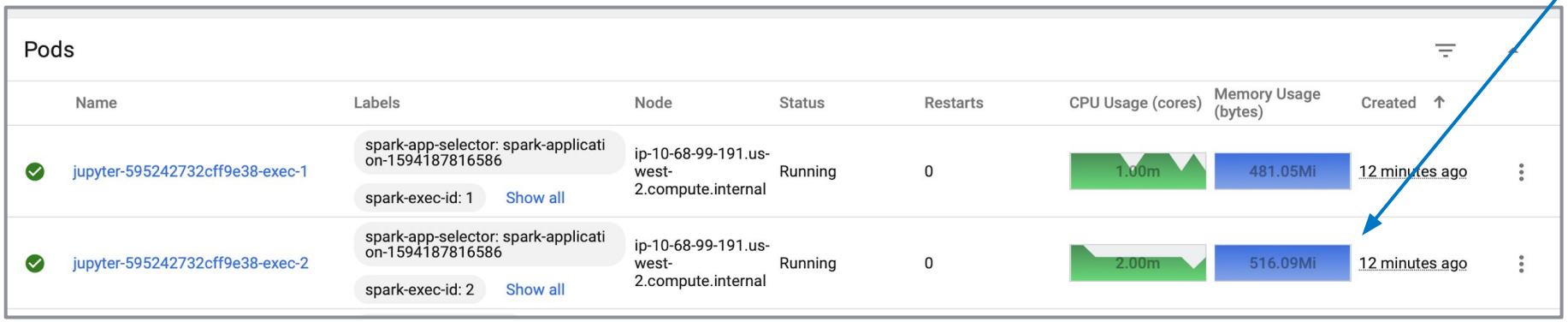


Especially powerful for ML with complex internal configs!

YARN vs Kubernetes

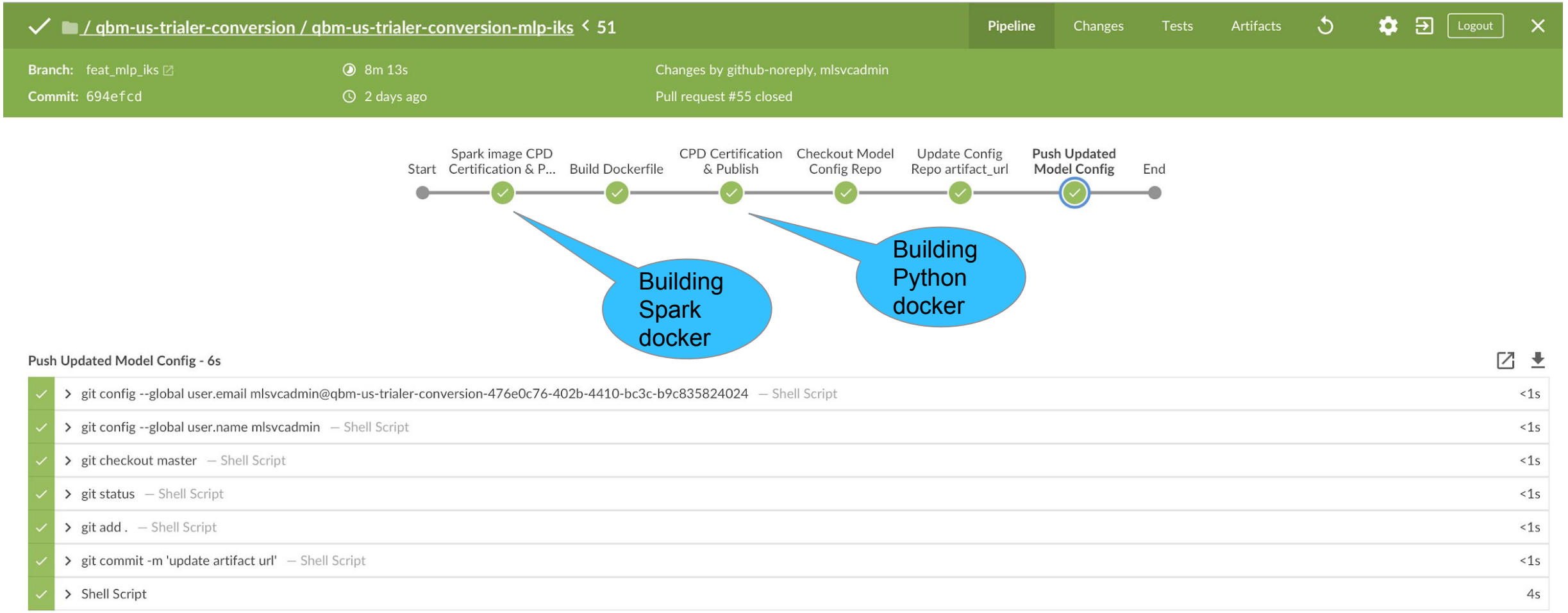


Spark executors



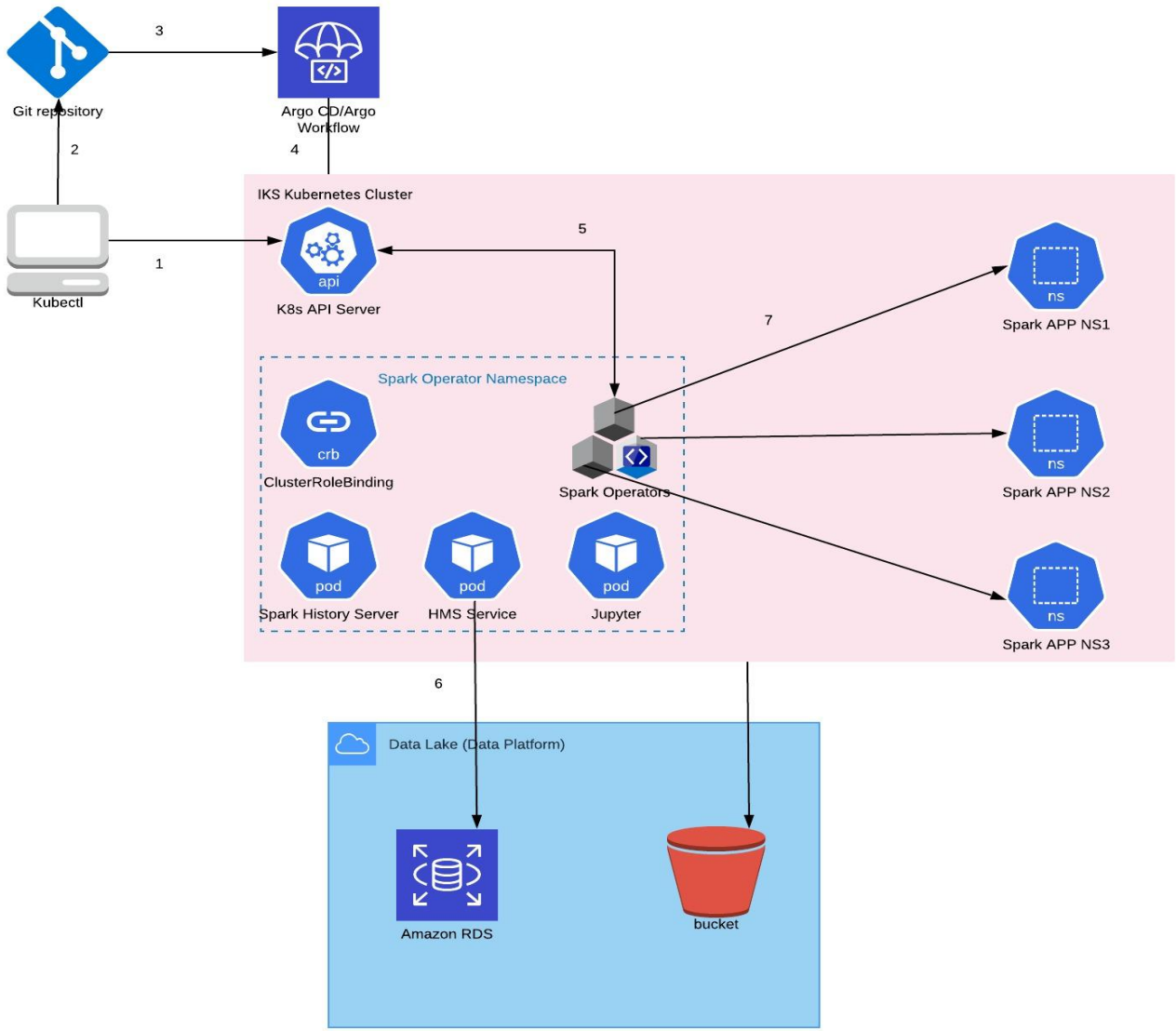
Kubernetes is more modern: easier to config, declarative, better executor visibility

CD for deployment

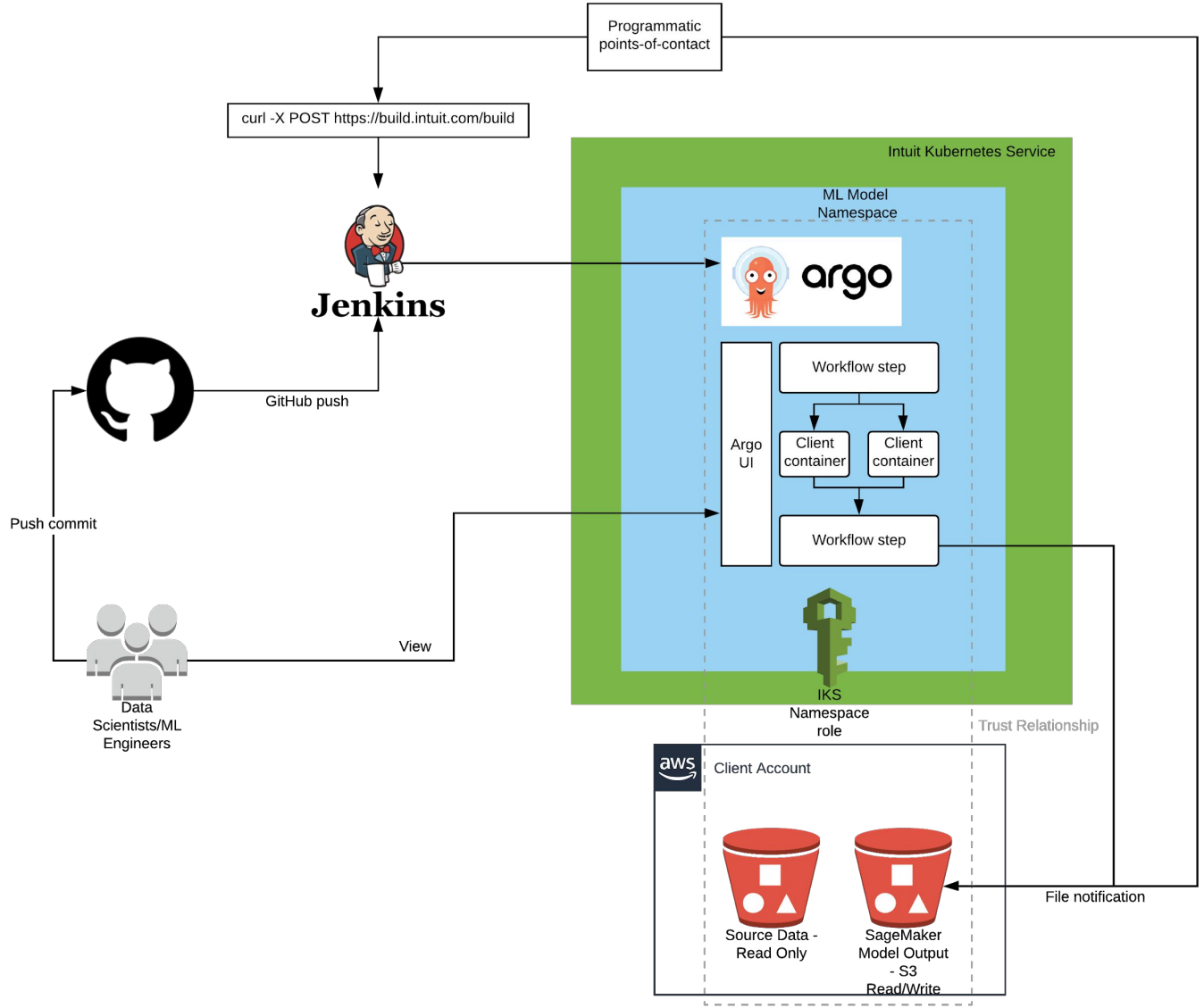


K8s Native Workflow

- Exploration in Jupyter -> Commit to Git
- Gitops deployment using ArgoCD
- ArgoCD submits Spark jobs
- Hive Meta Store via HMS service
- Data accessed from the data lake

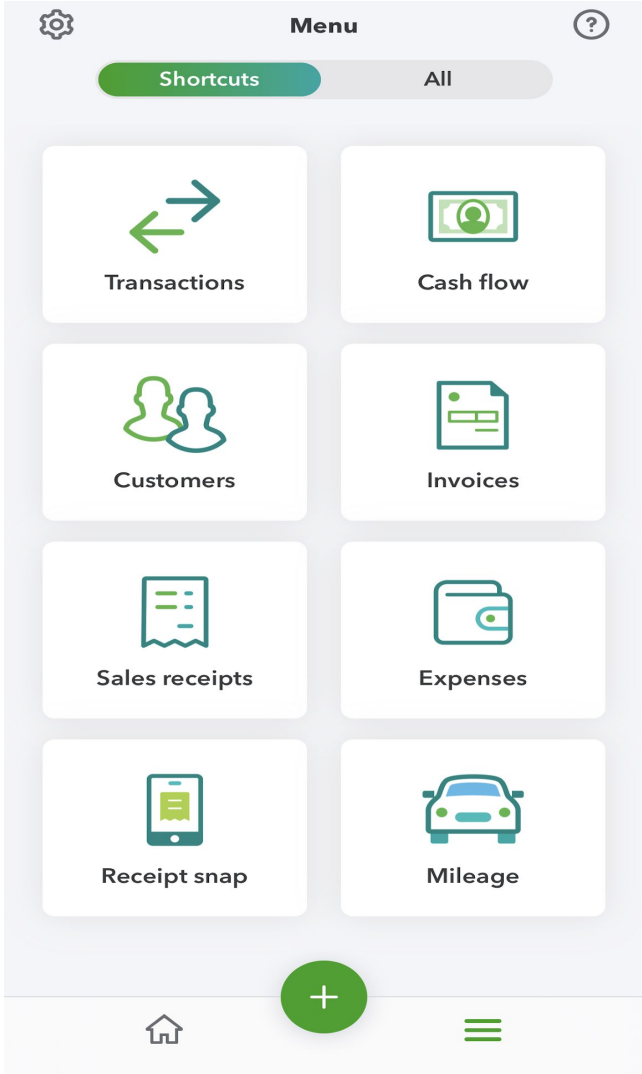


Use case: Model training and Batch inference

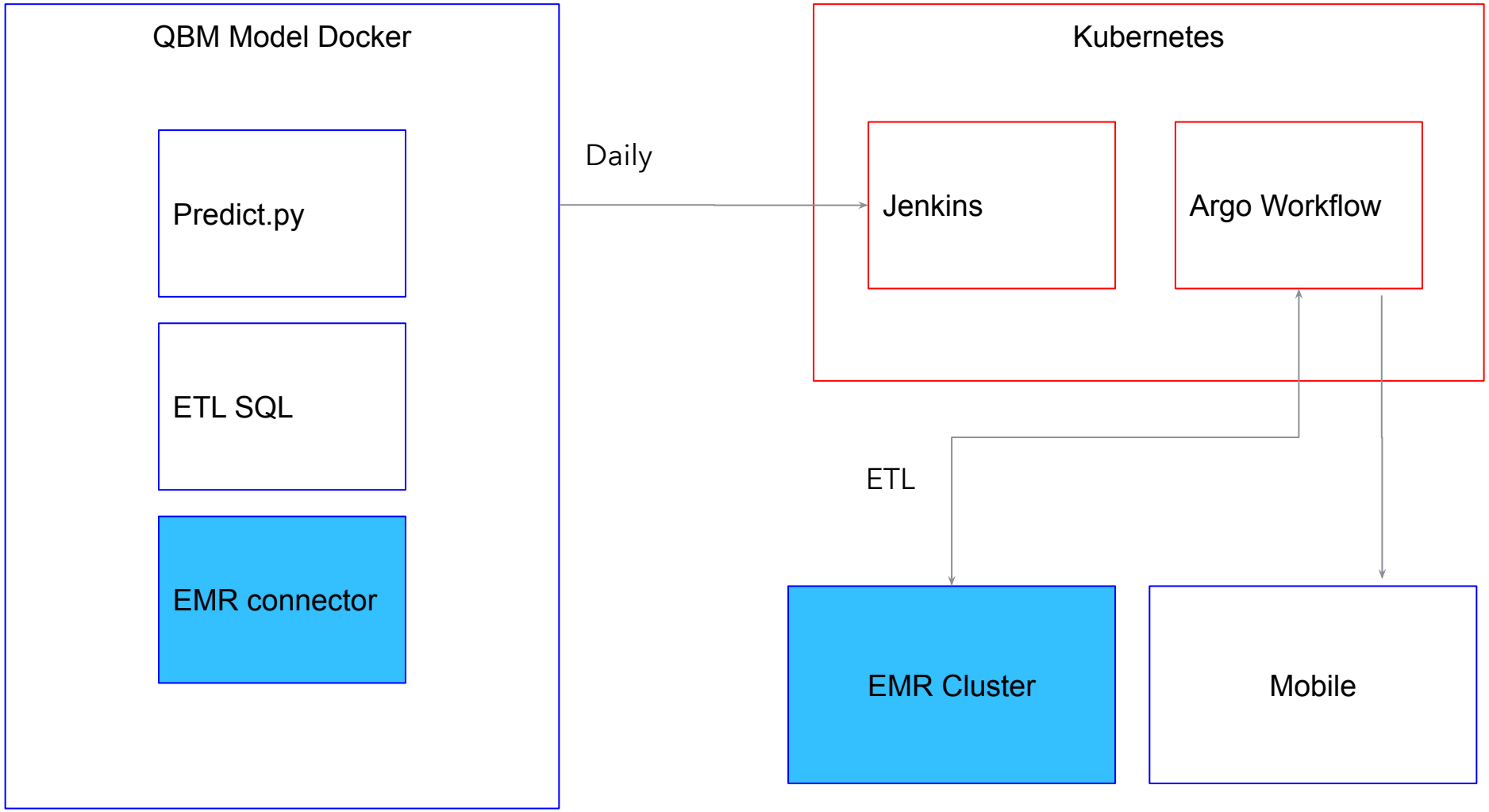


Example: QuickBooks Mobile Push notification model

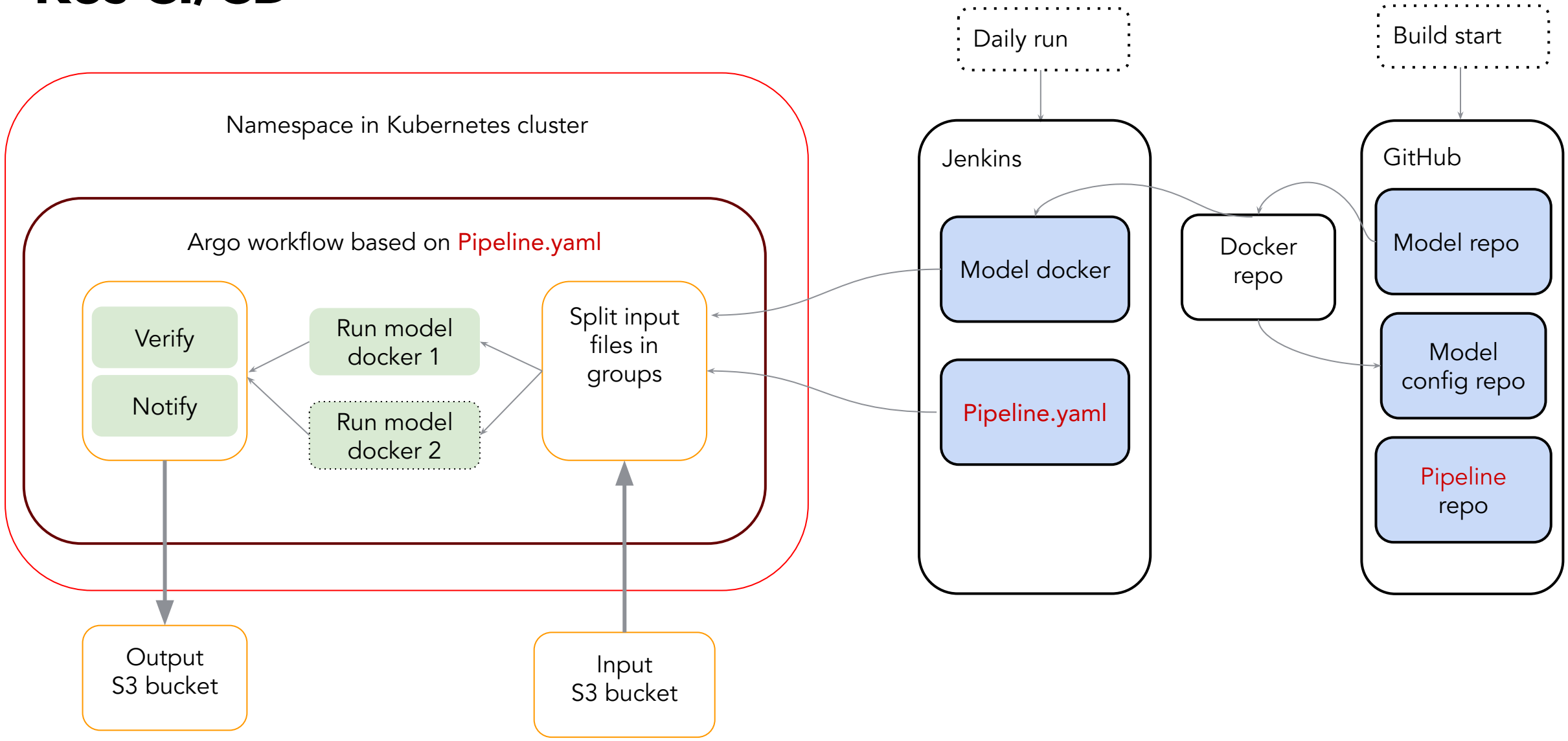
- **An XGBoost** model that predicts
  - Conversion to paid after the 45 days trial
  - Personalized top 5 actions that could drive conversion (i.e. “Add bank” or “Track mileage”)
- **Deploy** the model to personalize messaging to help Free Trialers get to know the product and convert to Premium



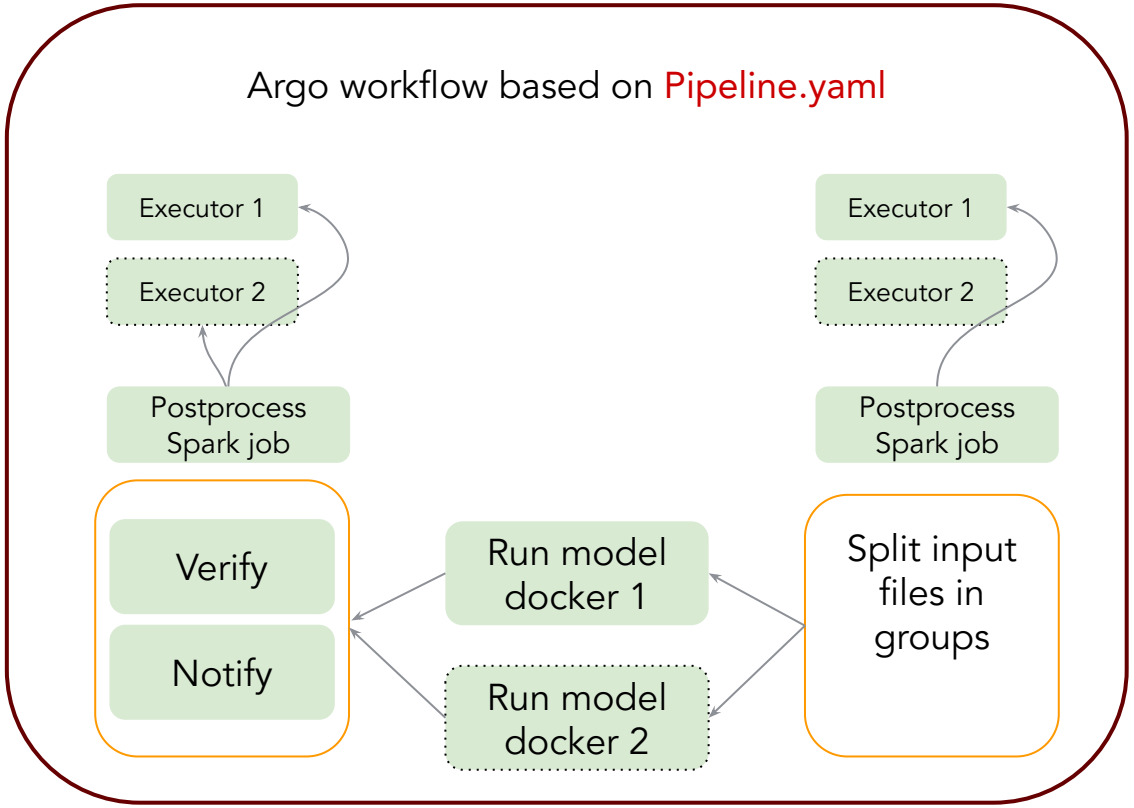
Daily batch inference using EMR

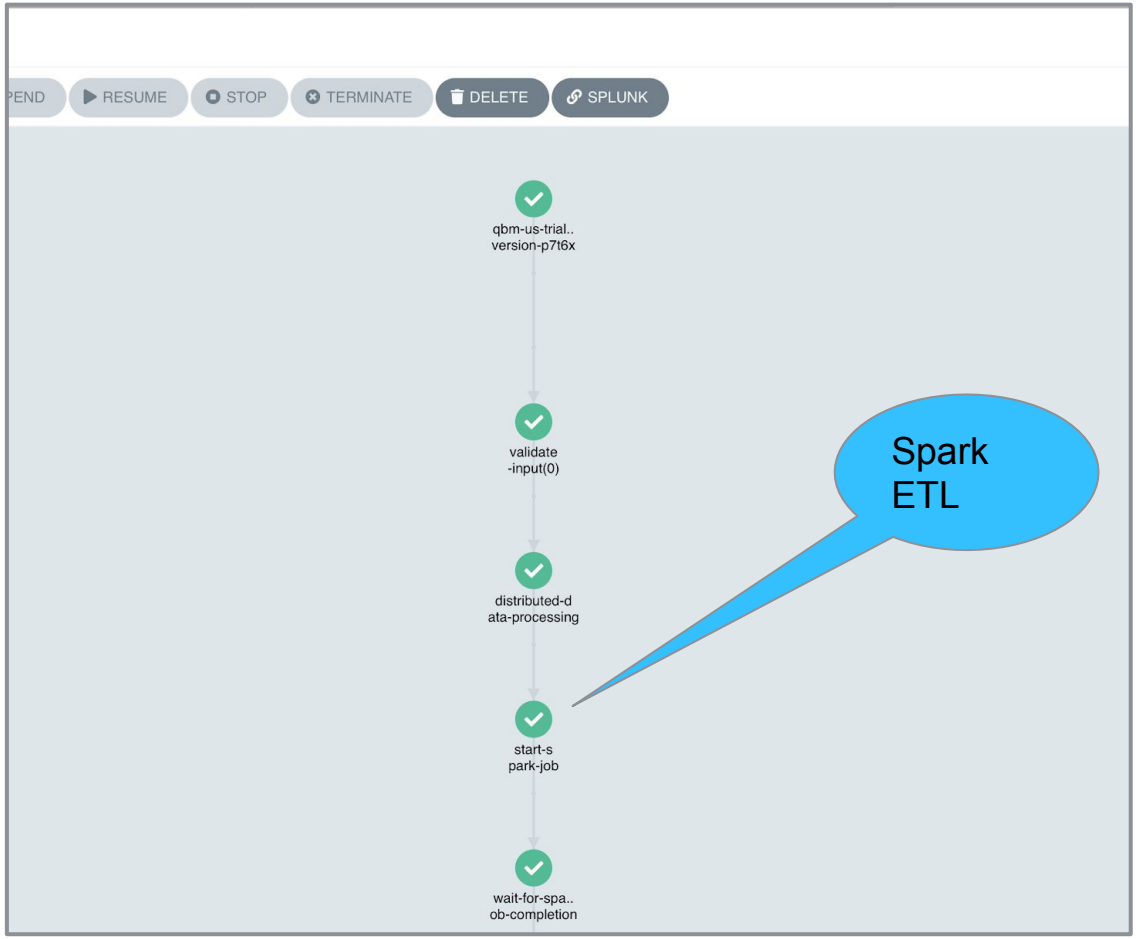


K8s CI/CD


Daily run
Build start
Namespace in Kubernetes cluster
Argo workflow based on Pipeline.yaml
Verify
Notify
Run model docker 1
Run model docker 2
Split input files in groups
Output S3 bucket
Input S3 bucket
Jenkins
Model docker
Pipeline.yaml
Docker repo
GitHub
Model repo
Model config repo
Pipeline repo

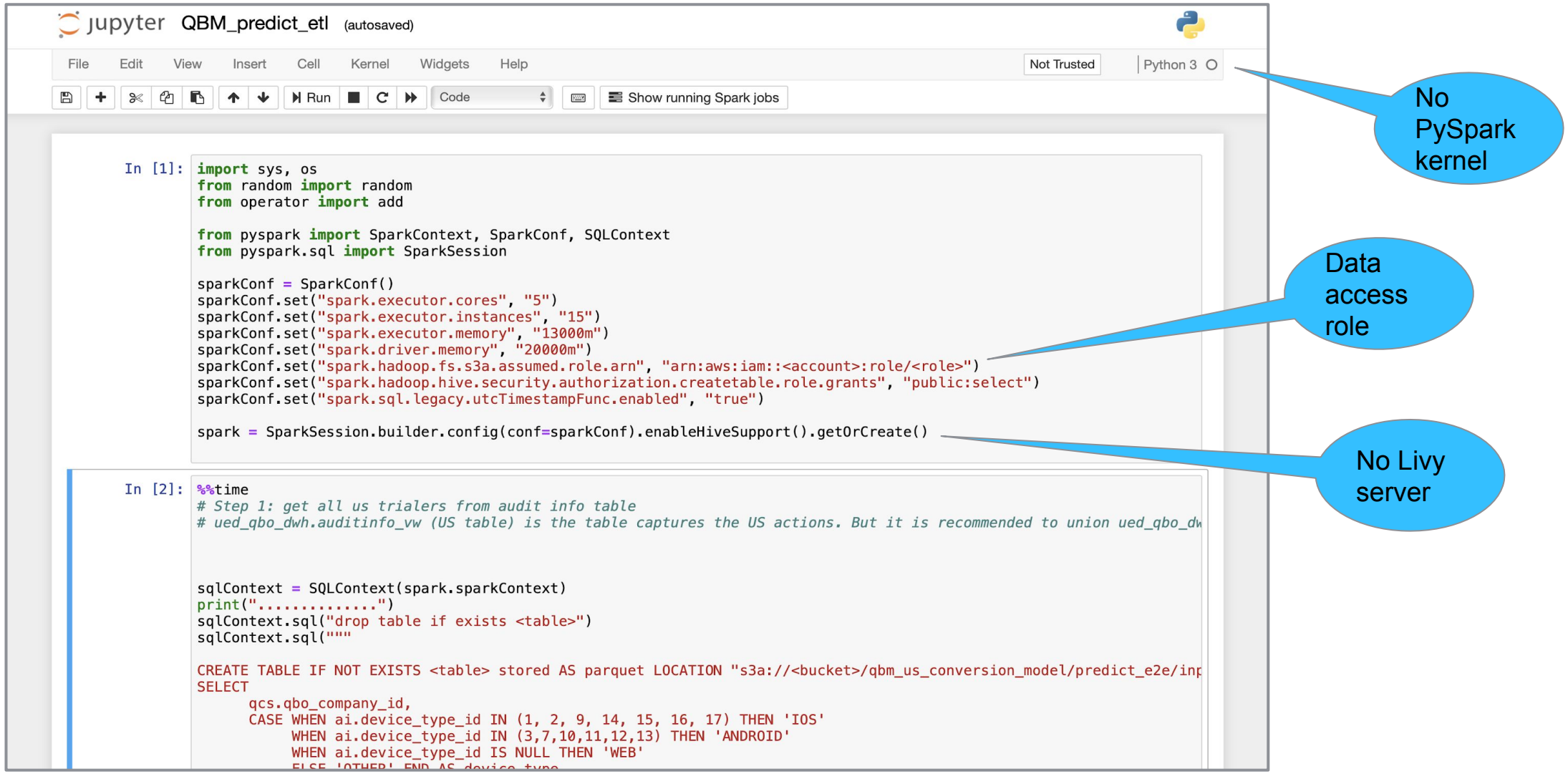
Daily batch inference using SpoK



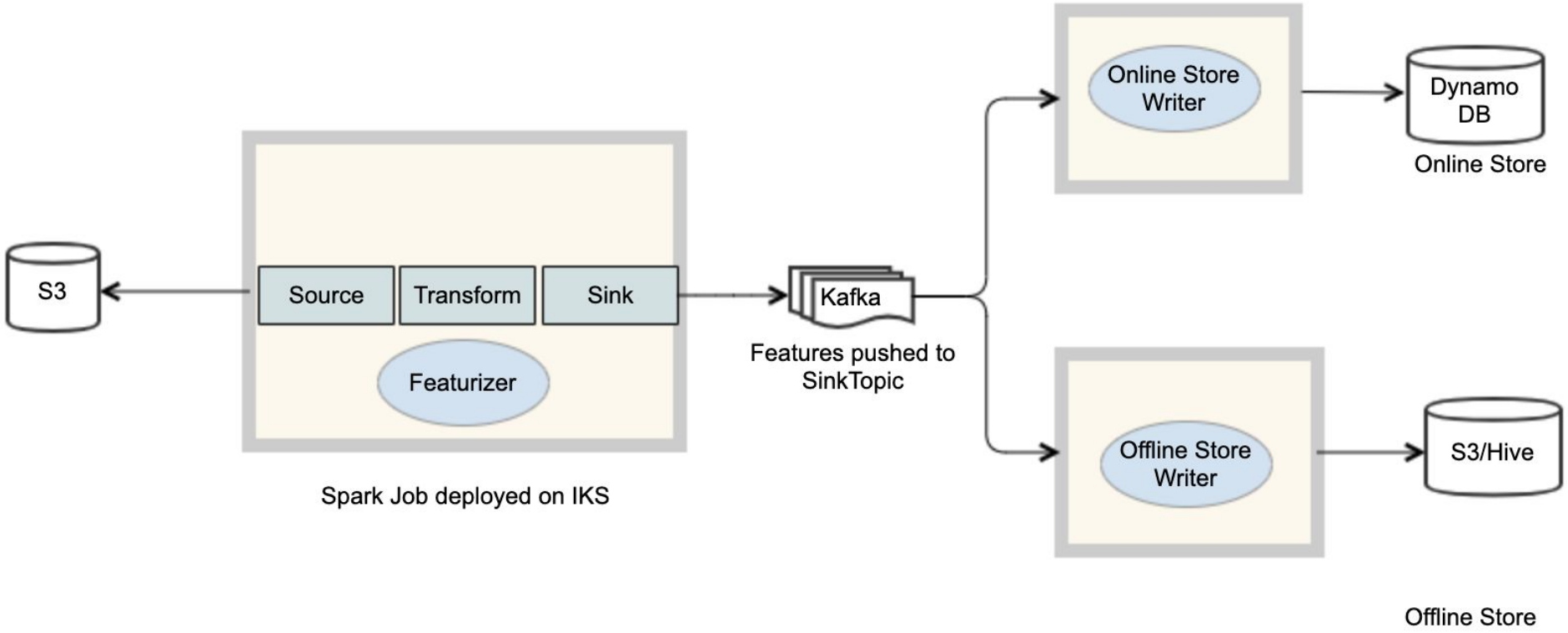


Performance ~ comparable to EMR. Takes 15-20 min

SpoK can be invoked from notebooks

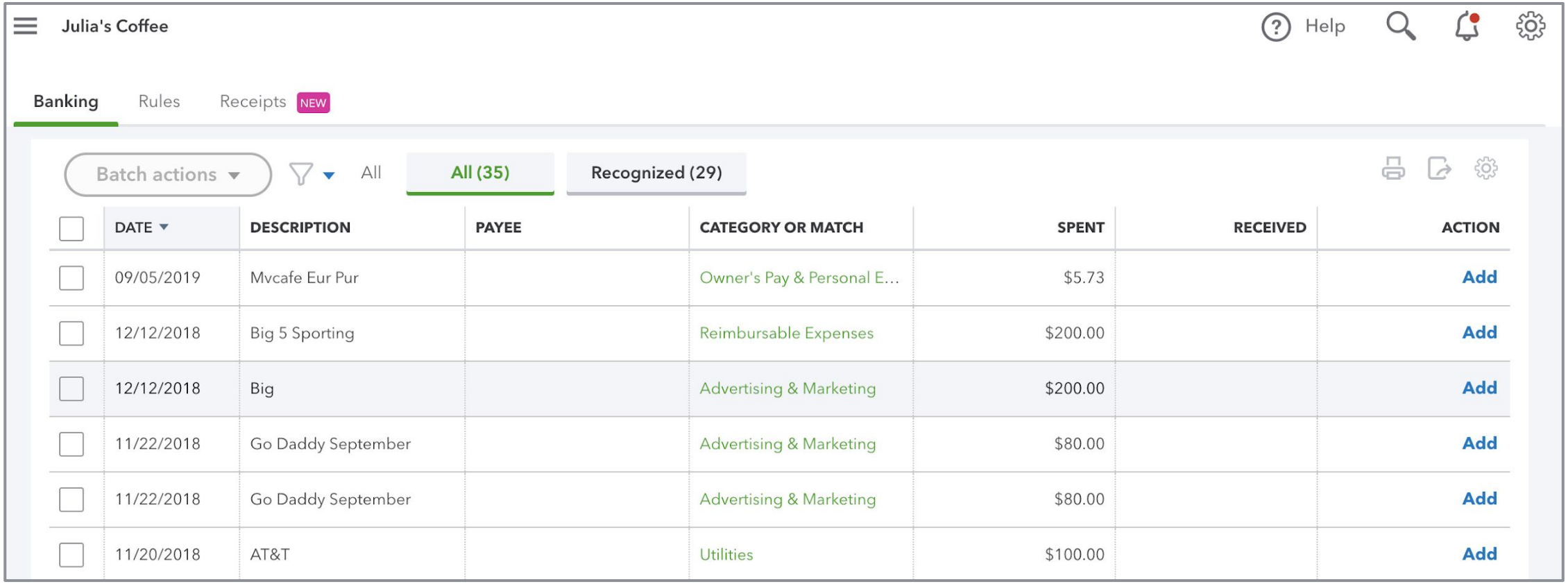


Use case: Feature Management Platform (FMP)



Example: QuickBooks Online transaction categorization

Categorization of QBO banking transactions



DATE	DESCRIPTION	PAYEE	CATEGORY OR MATCH	SPENT	RECEIVED	ACTION
09/05/2019	Mvcafe Eur Pur		Owner's Pay & Personal E...	$5.73		Add
12/12/2018	Big 5 Sporting		Reimbursable Expenses	$200.00		Add
12/12/2018	Big		Advertising & Marketing	$200.00		Add
11/22/2018	Go Daddy September		Advertising & Marketing	$80.00		Add
11/22/2018	Go Daddy September		Advertising & Marketing	$80.00		Add
11/20/2018	AT&T		Utilities	$100.00		Add

Challenge: Different Expense Categories for each user

User A

Sales	157
Auto:Fuel	132
Foxbusiness	112
West Coast Business Forms	108
Partners LLC	105
LOAN	69
Other General and Admin Expenses:Owners Draw	56
Office Supplies	30
Computer & Internet Fees	21
PAYROLL EXPENSE:Payroll Taxes	20
Meals and Entertainment	19
Office Expenses	17
Travel	16
Insurance -Workers Comp	15

User B

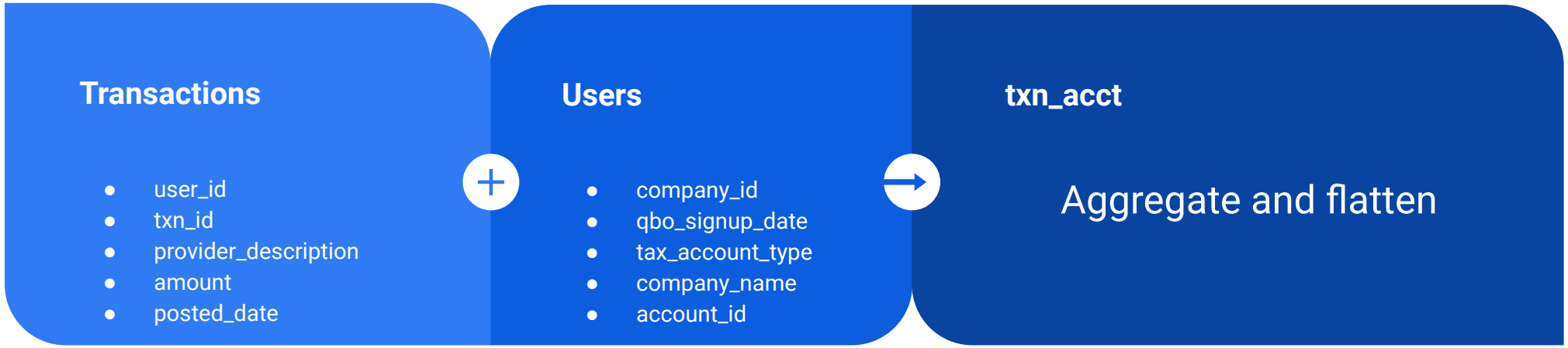
Supplies:Food	1278
Supplies:Curriculum	414
Supplies:Extra-Curricular	268
Automobile Expense:Bus Fuel	258
Field Trips	187
Supplies:Computer	145
Advertising	137
Travel & Ent Meals	126
Payroll Expense Wages	109
Employee Benefits:Gift Cards	59
Bus Tolls	54
Repairs & Maintenance:Building Repairs	51
Repairs & Maintenance:Pest Control	51
Bus maintenance	47
Cleaning	47

THE **HOME DEPOT** 6948 PHOENIX AZ325677 01/08 $15.90

For a **contractor** ⇒ Materials and Supplies

For a **retailer** ⇒ Repairs and Maintenance

Feature ETL logic is a simple JOIN

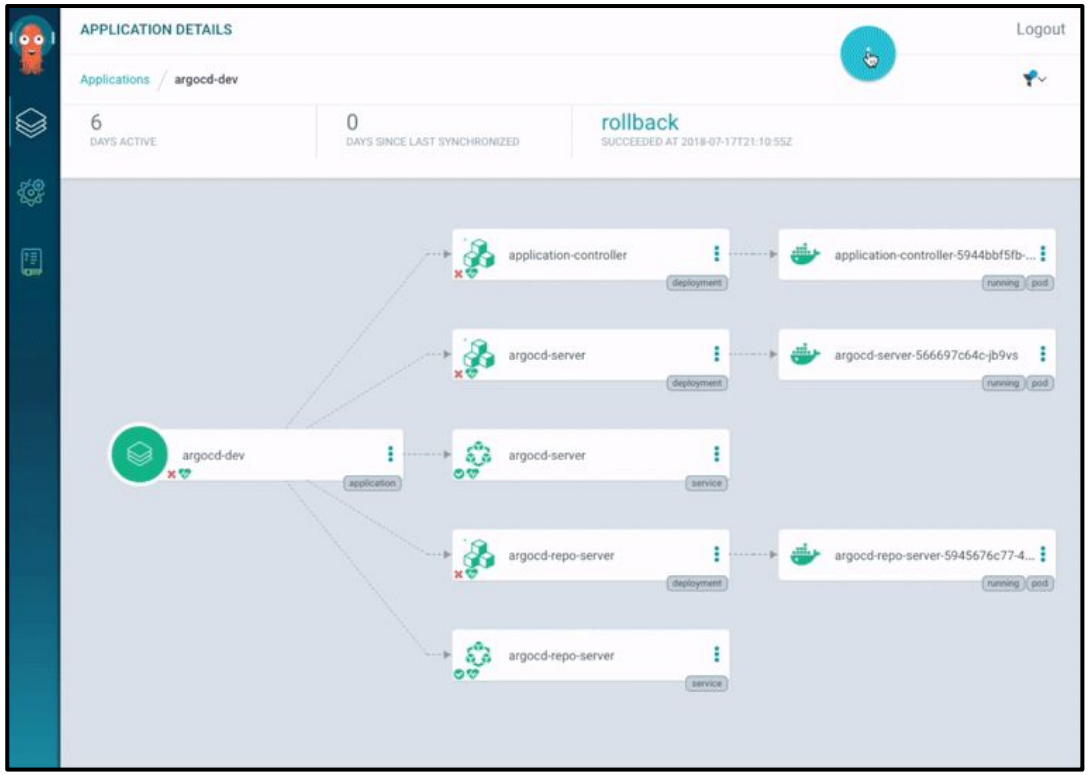


example output:

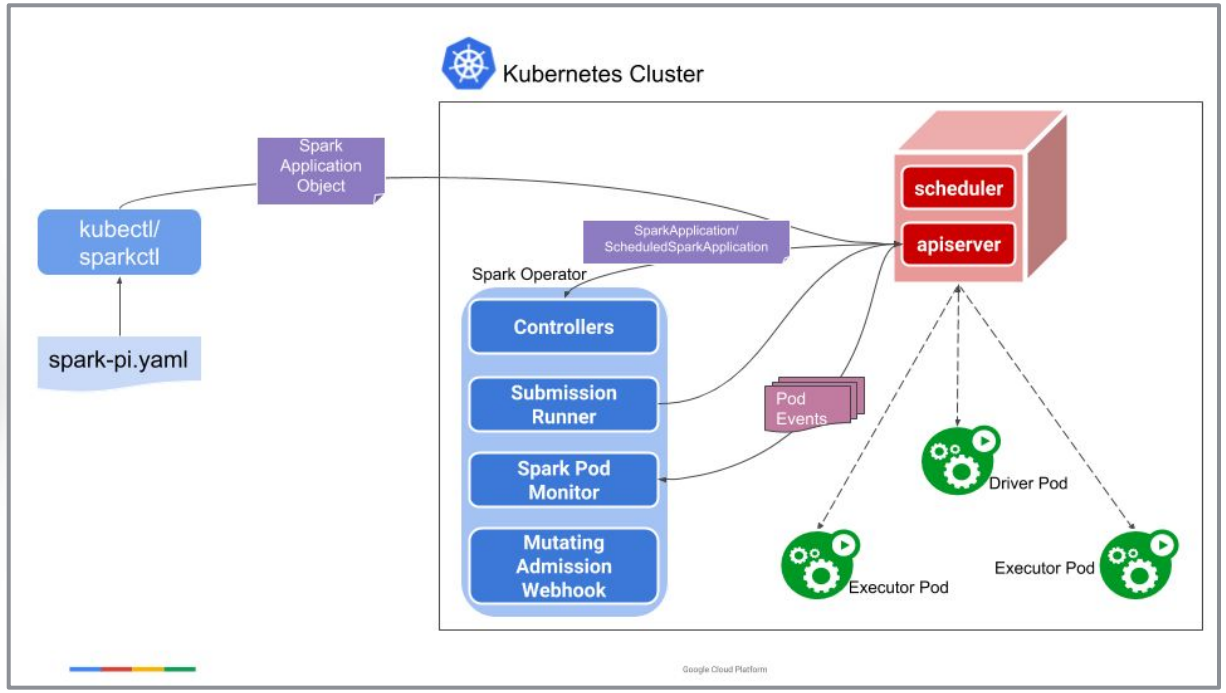
1000360580 ["2019-12-16 08:00:00,65,11901,128,131,-1234.3,colonial life ins","2019-12-24 08:00:00,54,10020,130,131,...", ...]

ArgoCD + SparkOperator

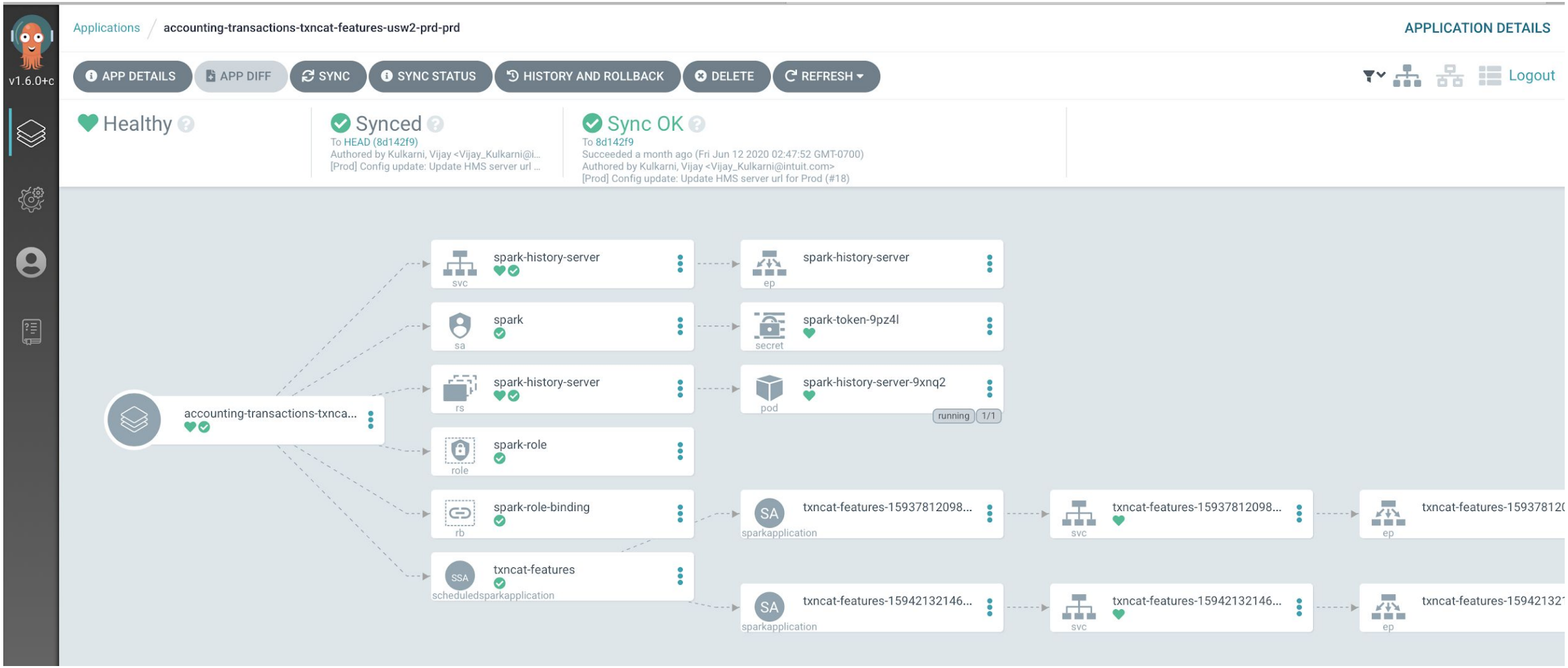
https://argoproj.github.io/argo-cd/



https://github.com/GoogleCloudPlatform/spark-on-k8s-operator



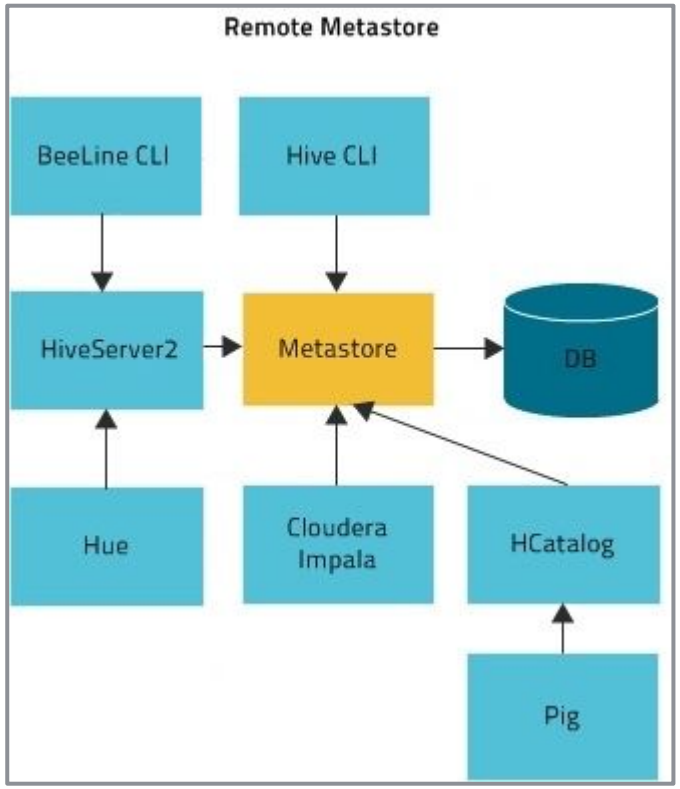
Transaction Categorization SparkApplication in ArgoCD



Learnings from Spark On Kubernetes

- External Hive Metastore
- S3A is required, in place of EMRFS
- Hadoop-AWS Jars are missing from the Spark docker
- Spark 2.4.5 required patching. Recommend 3.0.0
- Spark History Server as K8s Deployment
- Spark Operator mounts */opt/spark/conf/spark.properties*

Hive Metastore on Kubernetes



Name
hive-metastore-service2
Namespace
spark-iks-ged-usw2-ppd-prd
Created
Jun 16, 2020
Age
22 days ago
UID
48bd14ab-ee46-4bc7-b7ba-2adbb085c745
Labels
app: hive-metastore2

Resource information

Type
ClusterIP
Cluster IP
100.69.160.211
Session Affinity
None
Selector
app: hive-metastore2

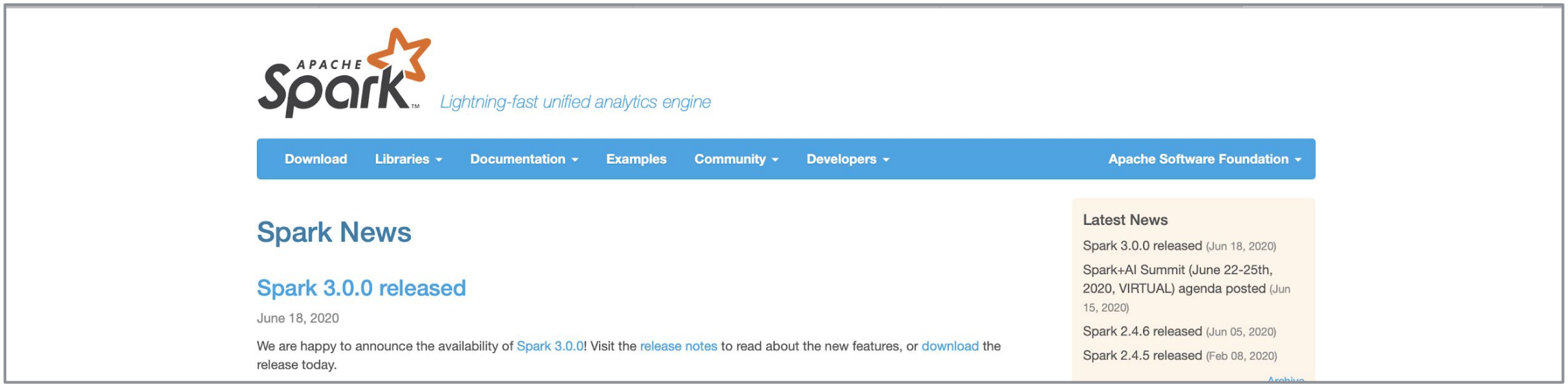
Endpoints

Host	Ports (Name, Port, Protocol)	Node	Ready
100.114.119.138	<unset>,9083,TCP	ip-10-68-99-191.us-west-2.compute.internal	true

Pods

Name	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
hive-metastore-deployment2-55b9db6bd-km8dk	app: hive-metastore2 pod-template-hash: 55b9db6bd Show all	ip-10-68-99-191.us-west-2.compute.internal	Running	0	1.00m	241.66Mi	8 days ago

Spark 3 is best for Kubernetes



Spark 2.4.5 can be configured to run on Kubernetes, but requires custom patching:

- Dynamic Allocation only added in Spark 3.0 https://issues.apache.org/jira/browse/SPARK-27963
- A critical Parquet bug https://issues.apache.org/jira/browse/HADOOP-16109

Spark Operator mounts over */opt/spark/conf/spark.properties*

Spark Operator tracks IPs and other dynamic properties of driver and executors

Name	Namespace	Created	Age
pyspark-select-1594329722894-driver-conf-map	spark-iks-ged-usw2-ppd-prd	Jul 9, 2020	40 seconds ago

UID
e6c9d1a2-14ed-4359-a63b-5985522f6872

Data

```
{
  "spark.properties": "#Java properties built from Kubernetes config map with name: pyspark-select-1594329722894-driver-conf-map
    #Thu Jul 09 21:22:03 GMT 2020
    spark.kubernetes.submission.waitAppCompletion=false
    spark.hadoop.fs.s3a.aws.credentials.provider=org.apache.hadoop.fs.s3a.auth.AssumedRoleCredentialProvider
    spark.driver.port=7078
    spark.kubernetes.driver.label.sparkoperator.k8s.io/submission-id=3bbb52e3-a5d9-4a74-a50d-f5220101b059
```

We need to mount additional ConfigMap and merge it in the Docker

Summary: Advantages of SpoK over EMR/YARN

- **Fits** into the larger K8s architecture
- **Simpler**, more **homogeneous**. Everything is a container.
- **Cost effective**. due to shared resources and Opex.
- **Declarative Infrastructure**, CICD
- **ML** is neatly *containerized* and separated from the *platform* (K8s) and the *compute* (Spark)

Future

- **Custom ML Operator** to orchestrate and manage ML Resources. (Spark, SageMaker Training/Deploy, Feature Store)
- **Declarative Management** of MDLC such as re-training of models, monitoring of features etc.
- **Managed Notebook Service**