

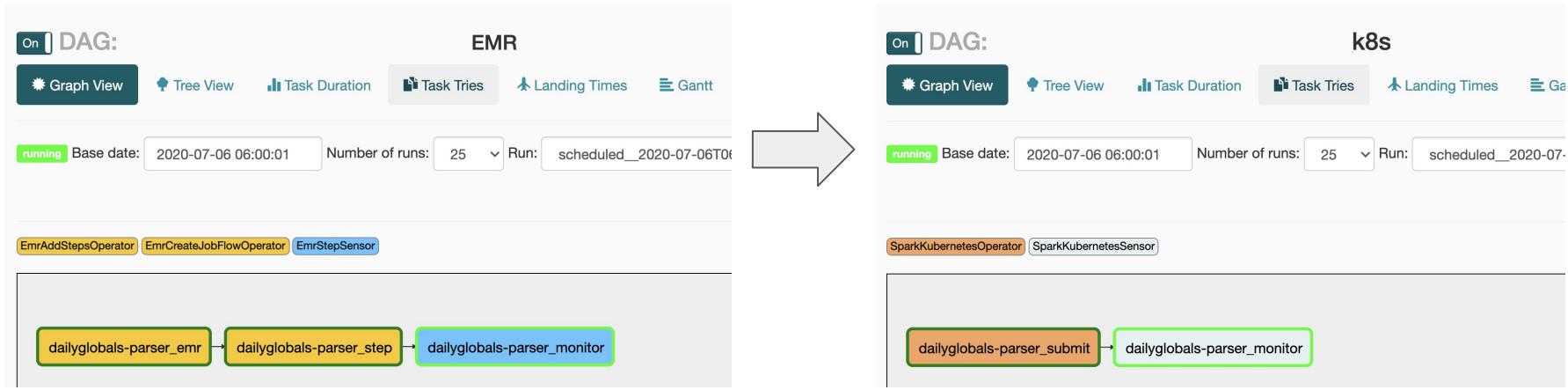
Airflow Summit 2020



July 6-17, 2020. Join from anywhere.

Join us!

Same-same...



... but different!



\$30,000/month Savings



Open your eyes



Robust



10

Migrating Airflow-based Spark jobs to K8s the native way

Roi Teveth & Itai Yaffe
Nielsen

Introduction



Roi Teveth

-  Big Data developer
-  Kubernetes evangelist
-  [Roi Teveth](#)  [@RTeveth](#)

[@ItaiYaffe](#), [@RTeveth](#)



Itai Yaffe

-  Tech Lead, Big Data group
-  Dealing with Big Data challenges since 2012
-  [Itai Yaffe](#)  [@ItaiYaffe](#)

What will you learn?

01.



=



02.

ETLs with
Airflow &



03.



-on-kubernetes



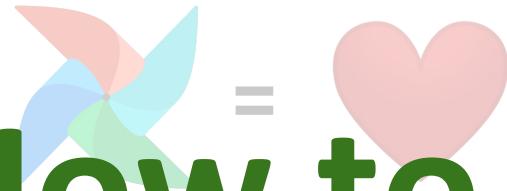
04.

Airflow Integration
& Contribution



What will you learn?

01.



=



02.

ETLs with
Airflow &



How to leverage that

03.

in your organization

-on-

kubernetes

04.

Airflow Integration
& Contribution

Nielsen Identity Engine

- eXelate was acquired by Nielsen on March 2015
- A Data company
- Machine learning models for insights
- Targeting
- Business decisions

Nielsen Identity Engine in numbers



~3000 nodes/day



\$100K's/month



~30TB ingested/day



**10's of TB
ingested/day**

The challenges

Scalability

Fault-tolerance

Cost Efficiency

Why did we need Airflow?

- Dozens of ETL workflows running around the clock
- Originally used AWS Data Pipeline for workflow management
- But we also wanted:
 - Better **visibility** of configuration and workflow
 - Better **monitoring** and **statistics**
 - Share **common configuration/code** between workflows

Why do we ❤️ Airflow?

Met all requirements & more

~2 years in production

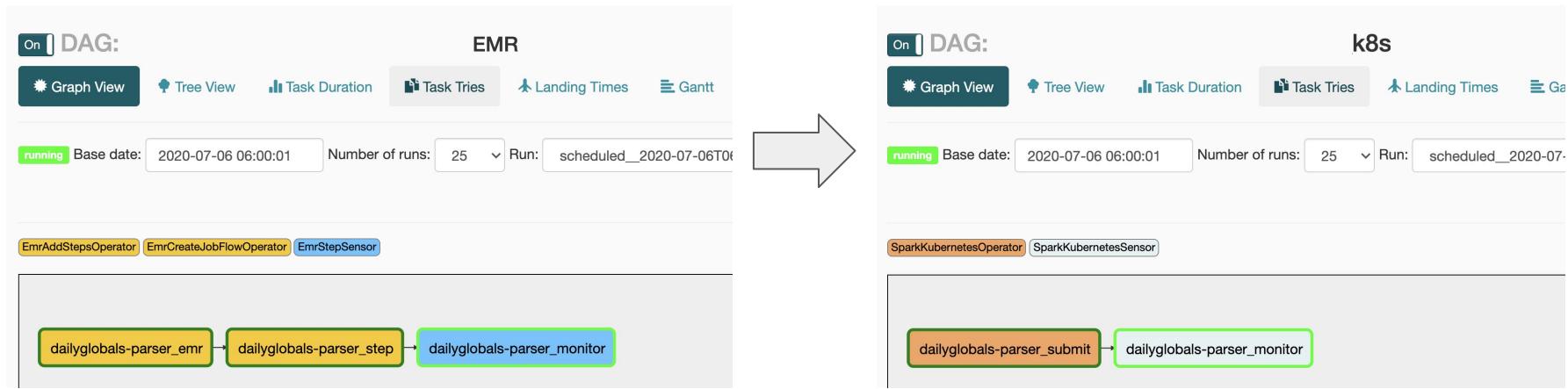
~40 users across 4 groups

~1000 DAG Runs/day

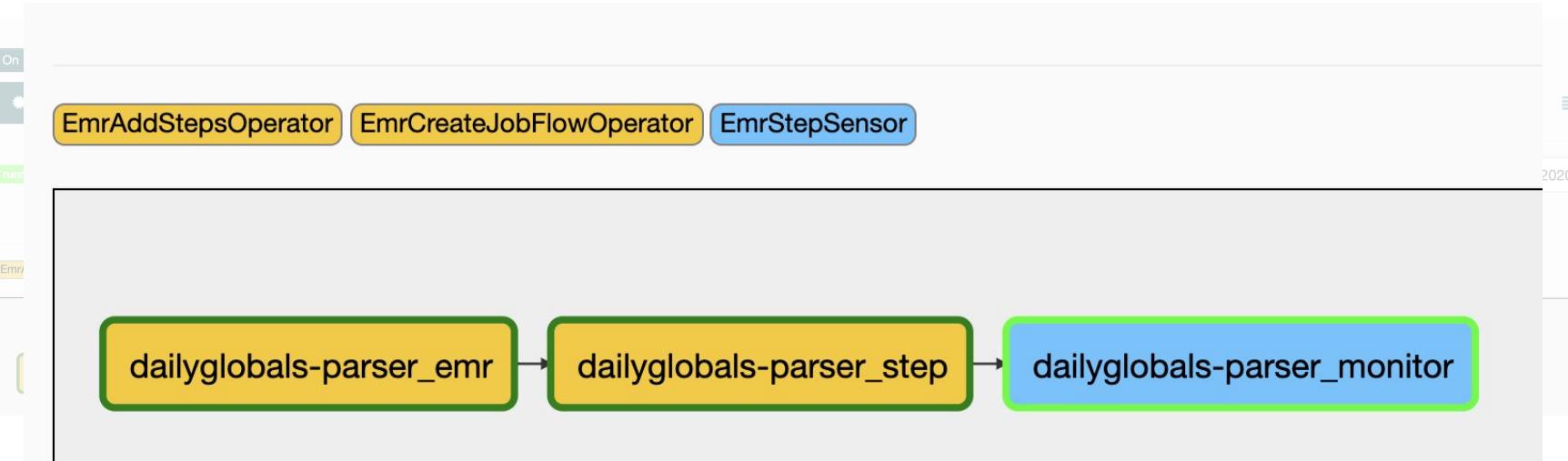
~20 automatic DAG deployments/day

6 contributions to open-source

Remember this slide?

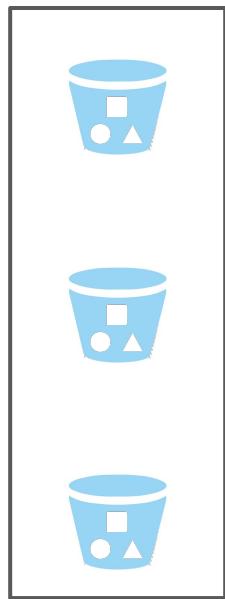


Let's take a closer look...

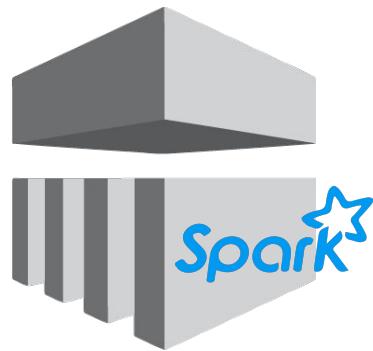


Common data pipeline pattern - high-level architecture

Data Lake

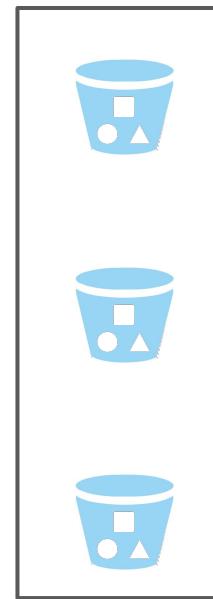


Data Processing



1.
Read input
files
»»

Intermediate Storage



2.
Write output
files
»»

OLAP



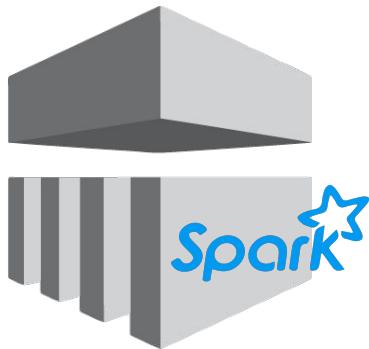
3.
Ingest
to DB
»»

Common data pipeline pattern - high-level architecture

Data Lake



Data Processing



1.
Read input
files
»»

Intermediate Storage



2.
Write output
files
»»

OLAP



3.
Ingest
to DB
»»

What is Spark?

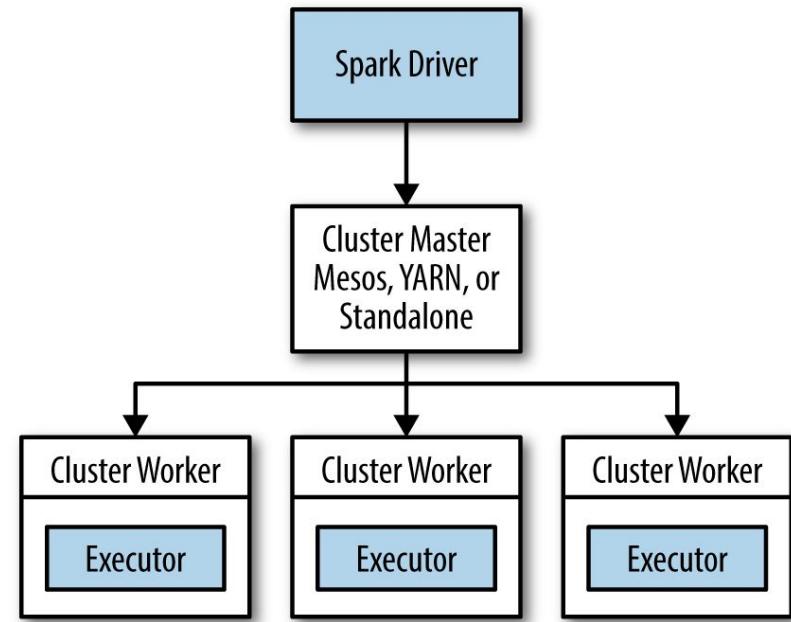


- An analytics engine for large-scale **data processing**
- **Distributed** and highly **scalable**
- A **unified framework** for batch, streaming, machine learning, etc.

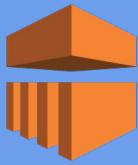
Basic Spark terminology



- Driver
- Executor
- Cluster manager
 - Mesos, YARN or Standalone
- Managed Spark on public clouds
 - AWS EMR, GCP Dataproc, etc.



What is EMR?



EMR is an AWS managed service to run Hadoop & Spark clusters



Allows you to reduce costs by using Spot instances



Charges management cost for each instance in a cluster

Running Airflow-based Spark jobs on EMR

- EMR has official Airflow support

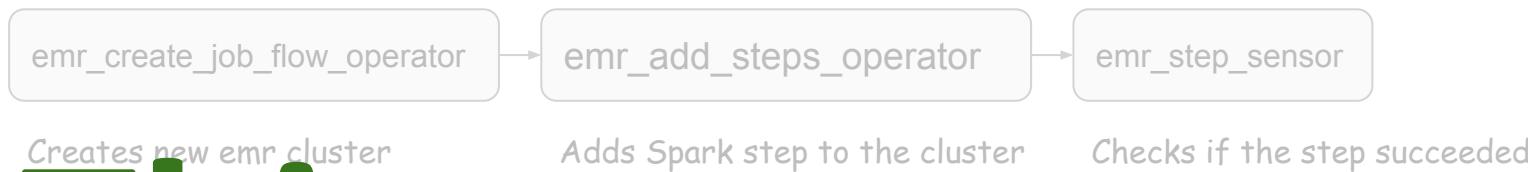


- Open-source, remember?

- Allows us to **fix** existing components
 - EmrStepSensor fixes ([AIRFLOW-3297](#))
- ... As well as add **new** components
 - AWS Athena Sensor ([AIRFLOW-3403](#))
 - OpenFaaS hook ([AIRFLOW-3411](#))

Running Airflow-based Spark jobs on EMR

- EMR has official Airflow support



- Open source, community driven?

- Allows us to **fix** existing components

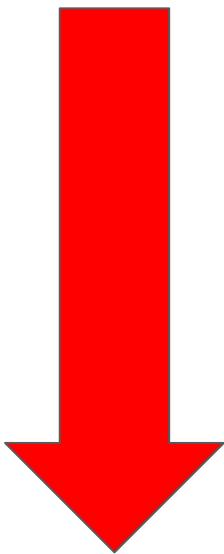
- EmrStepSensor fixes ([AIRFLOW-3297](#))

- ... As well as add **new** components
- AWS Athena Sensor ([AIRFLOW-3403](#))
 - OpenFaaS hook ([AIRFLOW-3411](#))

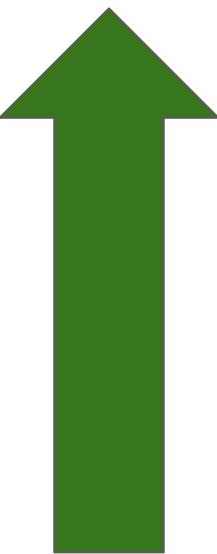
This was great...

But we wanted MORE!

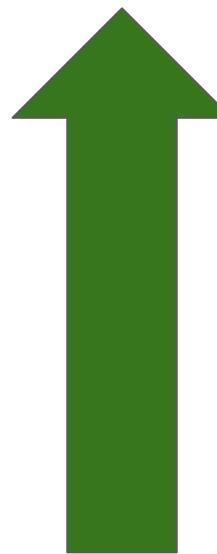
\$\$\$



Visibility



Robustness



Introducing - Spark-on-Kubernetes

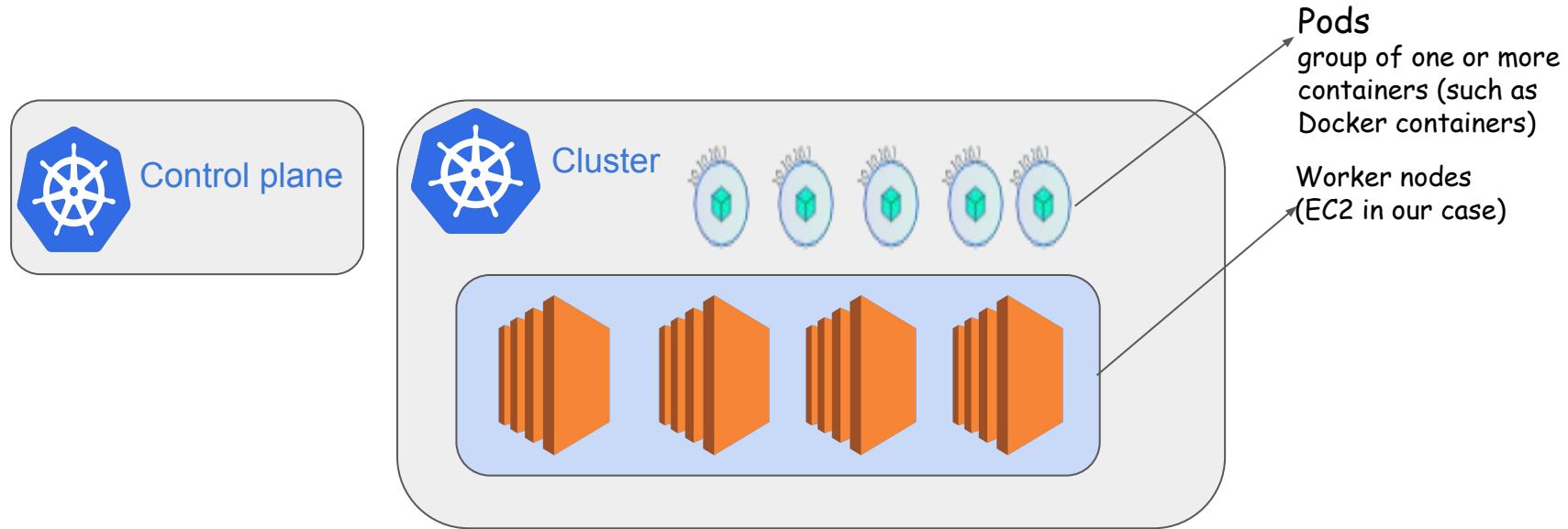


Let's explain what is Kubernetes (a.k.a K8s)

- Open source platform for running and managing **containerized workloads**
- Includes
 - **Built-in controllers** to support various workloads (e.g micro-services)
 - **Additional extensions** (called “**operators**”) to support custom workloads
- Highly scalable



Basic Kubernetes terminology



Basic Kubernetes terminology

- **kubectl** - K8s CLI

```
(base) ➔ ~ kubectl get po  
NAME                 READY   STATUS    RESTARTS   AGE  
app1-67d75d7f9-bj8fh   1/1     Running   0          11s
```

```
(base) ➔ ~ kubectl run ubuntu --image ubuntu:18.06 --generator=run-pod/v1  
pod/ubuntu created
```

Basic Kubernetes terminology

- The term “operator” exists both in Airflow and in Kubernetes

Basic Kubernetes terminology

- The term “operator” exists both in Airflow and in Kubernetes
-  operator
 - Represents a **single task**
 - Operators determine what is actually executed when your DAG runs
 - Example:
 - **bash-operator** - executes a bash command

Basic Kubernetes terminology

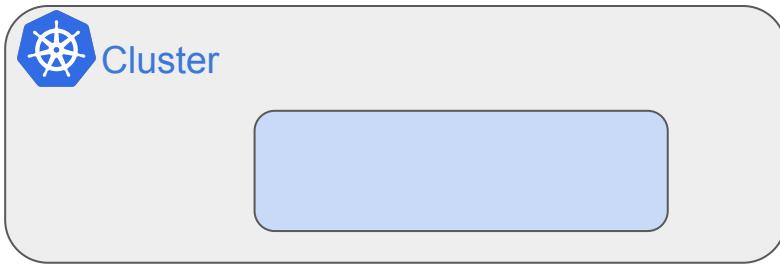
- The term “operator” exists both in Airflow and in Kubernetes
-  operator
 - Additional extensions to Kubernetes
 - Holds the knowledge of how to manage a **specific application**
 - Example:
 - **postgres-operator** - defines and manages a PostgreSQL cluster

Basic Kubernetes terminology

- The term “operator” exists both in Airflow and in Kubernetes
-  operator
 - A more Kubernetes controller
 - Holds the knowledge of how to manage a **specific application**
 - Example:
 - **postgres-operator** - defines and manages a PostgreSQL cluster

Kubernetes auto-scale

Phase 1: no applications are running on the cluster

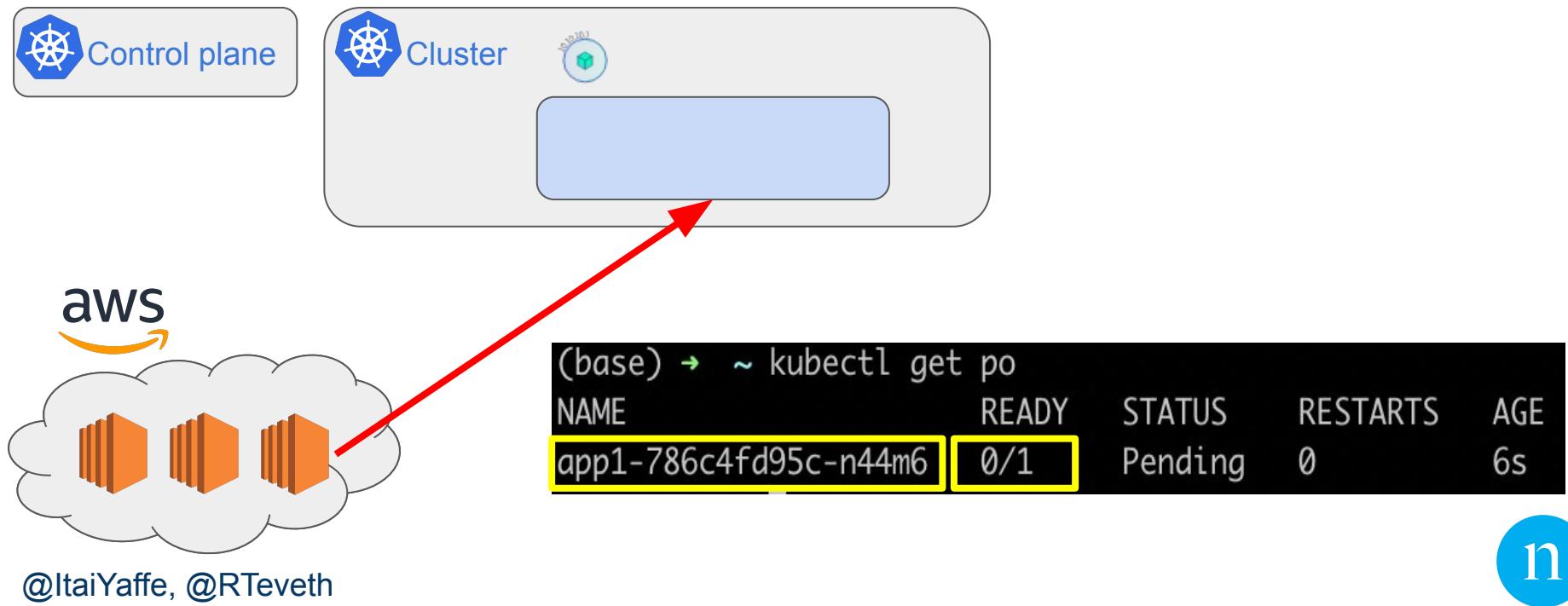


@ItaiYaffe, @RTeveth

```
(base) ➔ ~ kubectl get pods  
No resources found in default namespace.
```

Kubernetes auto-scale

Phase 2: application #1 starts running



Kubernetes auto-scale

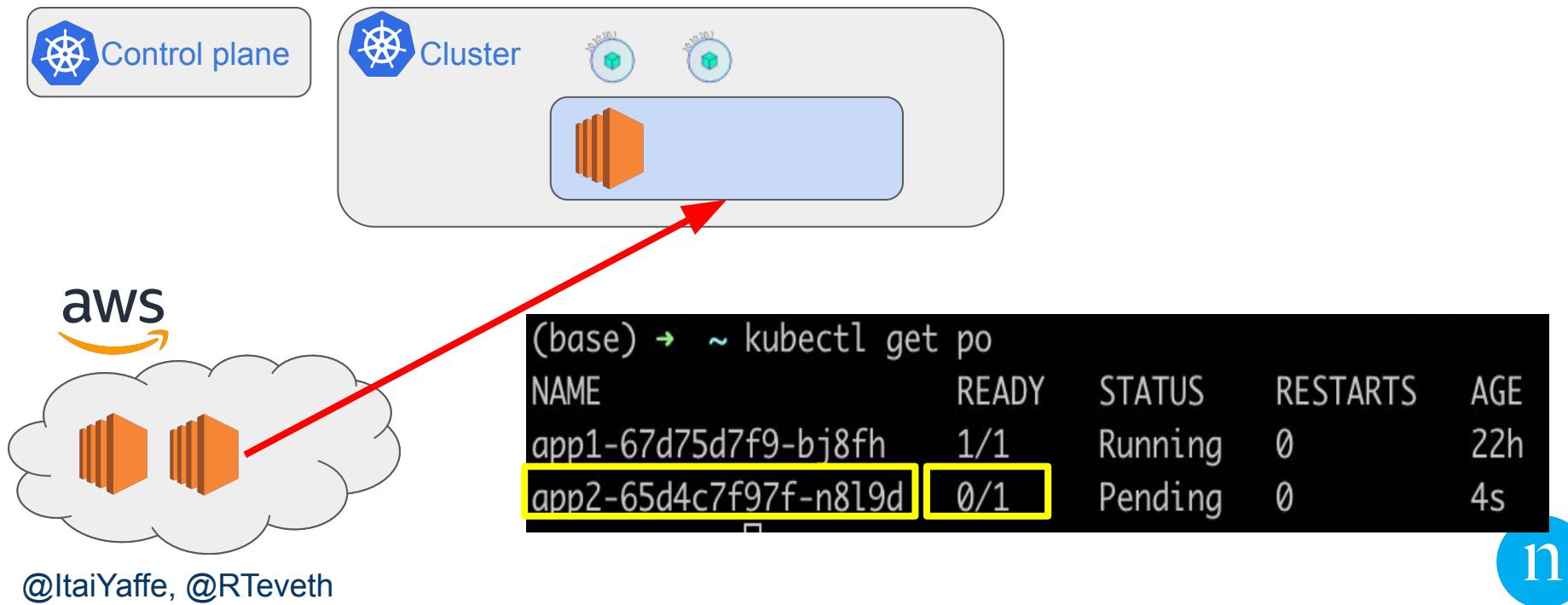
Phase 3: the cluster scales-up as needed



```
(base) → ~ kubectl get po
NAME                  READY   STATUS    RESTARTS   AGE
app1-67d75d7f9-bj8fh  1/1     Running   0          11s
```

Kubernetes auto-scale

Phase 4: application #2 starts running



Kubernetes auto-scale

Phase 5: the cluster scales-up as needed

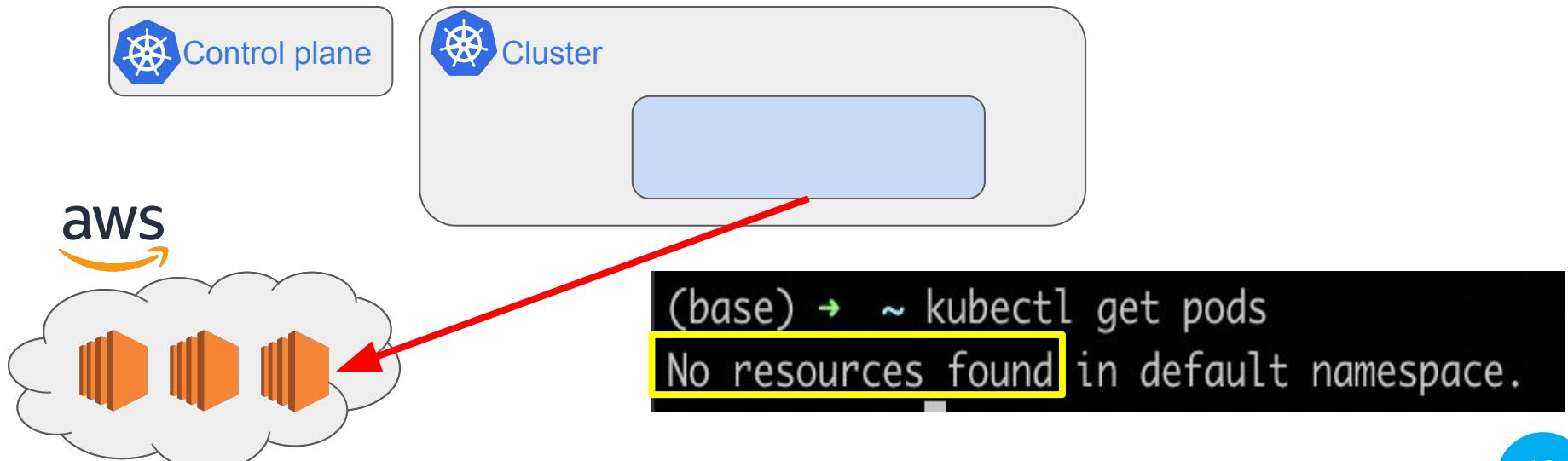


```
(base) ➔ ~ kubectl get po
```

NAME	READY	STATUS	RESTARTS	AGE
app1-6589f85558-w8pdm	1/1	Running	0	12s
app2-5499955bb4-pplgl	1/1	Running	0	7s

Kubernetes auto-scale

Phase 6: applications finished running, cluster scales down



Kubernetes in a nutshell

- A platform for running and managing **containerized workloads**
- Each cluster has
 - 1 control plane
 - 0..X worker nodes
 - 0..Y pods
 - 0..Z applications running concurrently
- Kubernetes operator != Airflow operator
- Automatically scales out and in

Cool, so... Back to Spark-on-Kubernetes?

Spark-On-Kubernetes overview

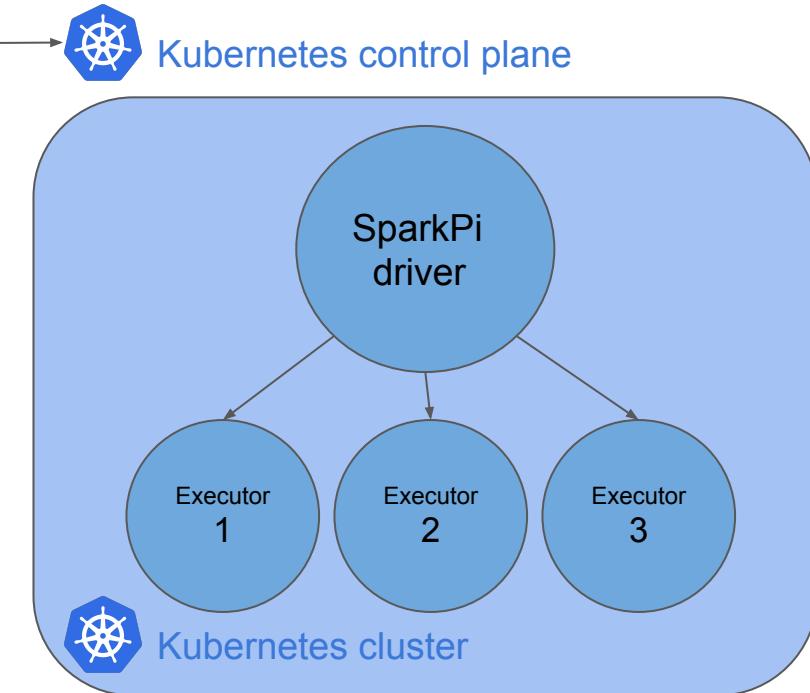
- From Spark 2.3.0, K8s is supported as a **cluster manager**
- This is still **experimental**, and some features are missing
 - Better auto-scaling is expected in future Spark releases (e.g through dynamic allocation)

Submitting a Spark application to Kubernetes - alternatives

1. Using `spark-submit` script
2. Using Spark-On-Kubernetes Operator

Spark-submit example - SparkPi

```
./bin/spark-submit \  
  --master  
  k8s://https://<k8s-apiserver-host>:<k8s-apiserver-port> \  
  --deploy-mode cluster \  
  --name spark-pi \  
  --class org.apache.spark.examples.SparkPi \  
  --conf spark.executor.instances=3 \  
  --conf spark.kubernetes.container.image=<spark-image>  
  
  local:///path/to/examples.jar
```

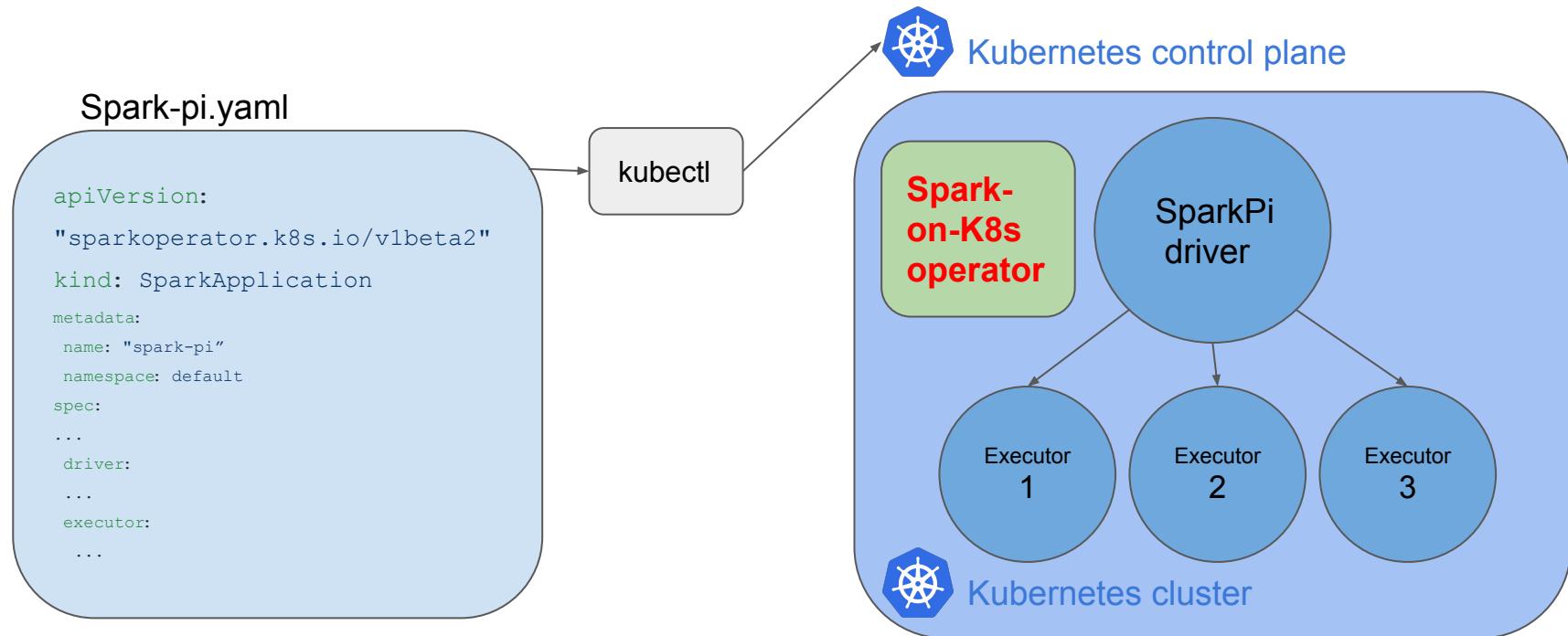


Spark-On-Kubernetes operator

- A Kubernetes operator
- Extends Kubernetes API to support Spark applications
- Built by GCP as an open-source project

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

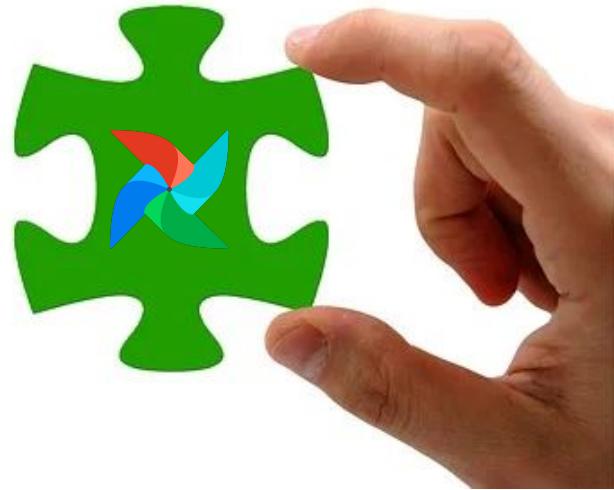
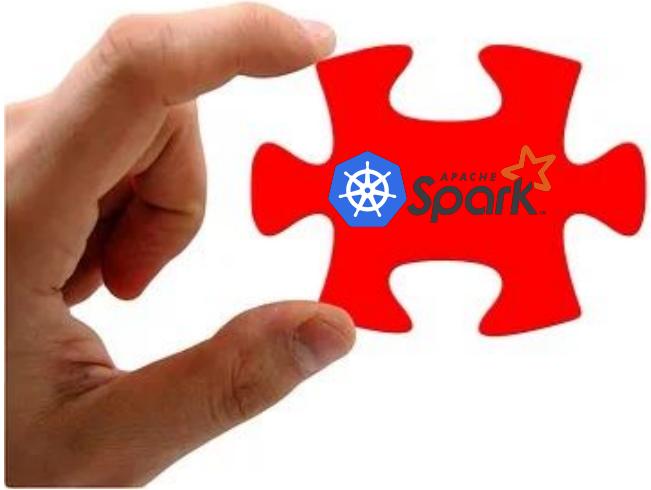
Spark-On-Kubernetes operator example - SparkPi



Submitting a Spark application to K8s

Topic	Spark-submit	Spark-On-K8s operator
Airflow built-in integration	V	X
Customize Spark-pods	X	V
Easy access to Spark UI	X	V
Submit and view application from kubectl	X	V

Integrate it with Airflow



So... we decided to take the road less traveled



pddepot.com
Public Domain Depot

@ItaiYaffe, @RTeveth

n

So... we decided to take the road less traveled

[AIRFLOW-6542] add spark-on-k8s operator/hook/sensor #7163

Merged kaxil merged 1 commit into apache:master from roitvt:master on Mar 10

Conversation 125 Commits 1 Checks 2 Files changed 24 +1,505 -5

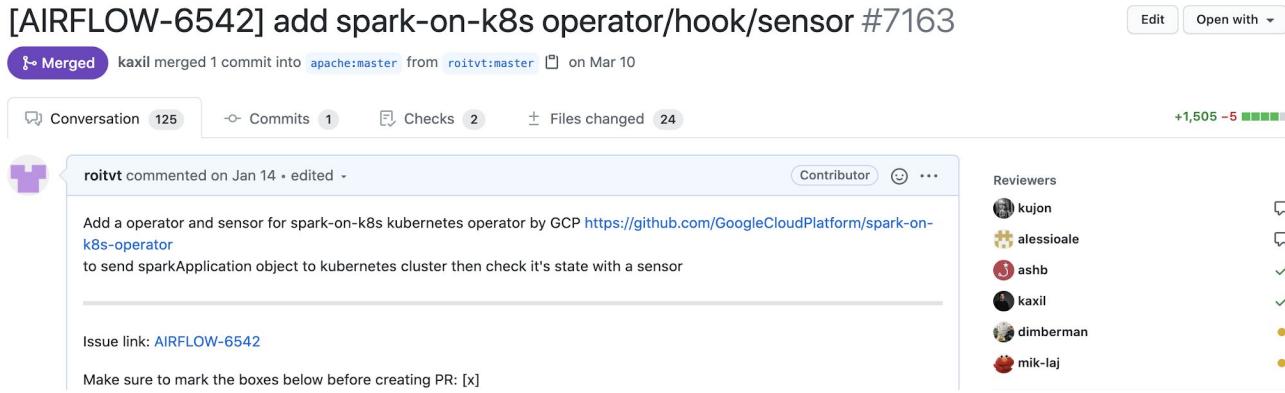
roitvt commented on Jan 14 • edited Add a operator and sensor for spark-on-k8s kubernetes operator by GCP <https://github.com/GoogleCloudPlatform/spark-on-k8s-operator> to send sparkApplication object to kubernetes cluster then check it's state with a sensor

Issue link: [AIRFLOW-6542](#)

Make sure to mark the boxes below before creating PR: [x]

Reviewers

- kujon
- alessioale
- ashb
- kaxil
- dimberman
- mik-laj



<https://github.com/apache/airflow/pull/7163>

A special thanks to Airflow committers

[AIRFLOW-6542] add spark-on-k8s operator/hook/sensor #7163

Merged kaxil merged 1 commit into apache:master from roitvt:master on Mar 10

The screenshot shows a GitHub pull request page for issue [AIRFLOW-6542]. The title is "add spark-on-k8s operator/hook/sensor". The pull request has been merged by kaxil into the apache:master branch from roitvt:master on March 10. The code review section highlights contributions from @CzerwonyElmo (Kamil Breguła) and @kaxil (Kaxil Naik). The pull request includes a detailed description of the changes, a checklist of best practices, and notes about AIP, 3rd party license compliance, and backwards compatibility. It also lists reviewers like kujon, alessioale, ashb, and others, and assignees like mik-laj. Labels include area:Webserver, area:dev, and area:docs. The pull request has received 5 thumbs up and 1 star. A note at the bottom states that merging this pull request may close linked issues.

@CzerwonyElmo (Kamil Breguła)

@kaxil (Kaxil Naik)

@AshBerlin (Ash Berlin-Taylor)

@higrys (Jarek Potiuk)

* For document-only changes commit message can start with [AIRFLOW-XXXX].

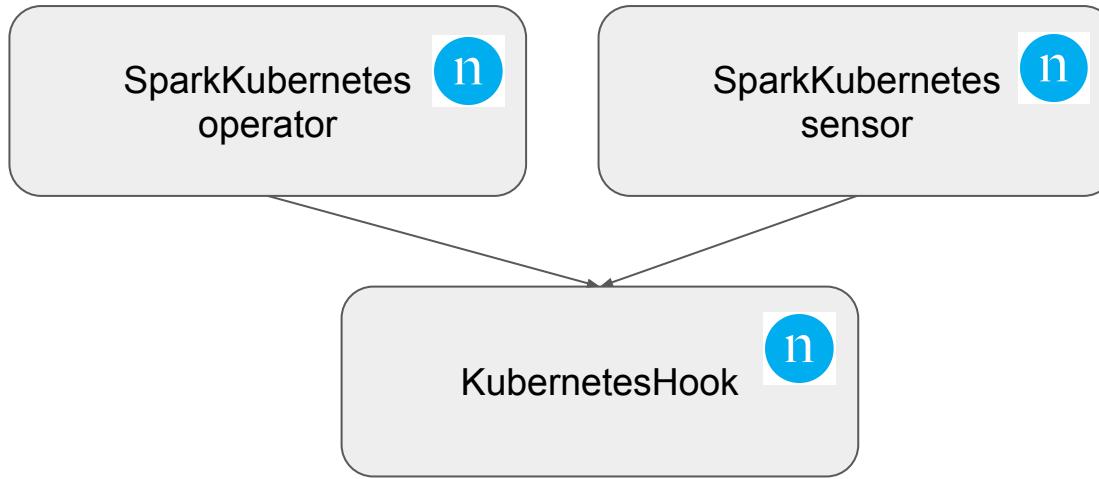
In case of fundamental code change, Airflow Improvement Proposal (AIP) is needed.
In case of a new dependency, check compliance with the ASF 3rd Party License Policy.
In case of backwards incompatible changes please leave a note in UPDATING.md.
Read the Pull Request Guidelines for more information.

5 1

@ItaiYaffe, @RTeveth

n

Airflow SparkKubernetes integration

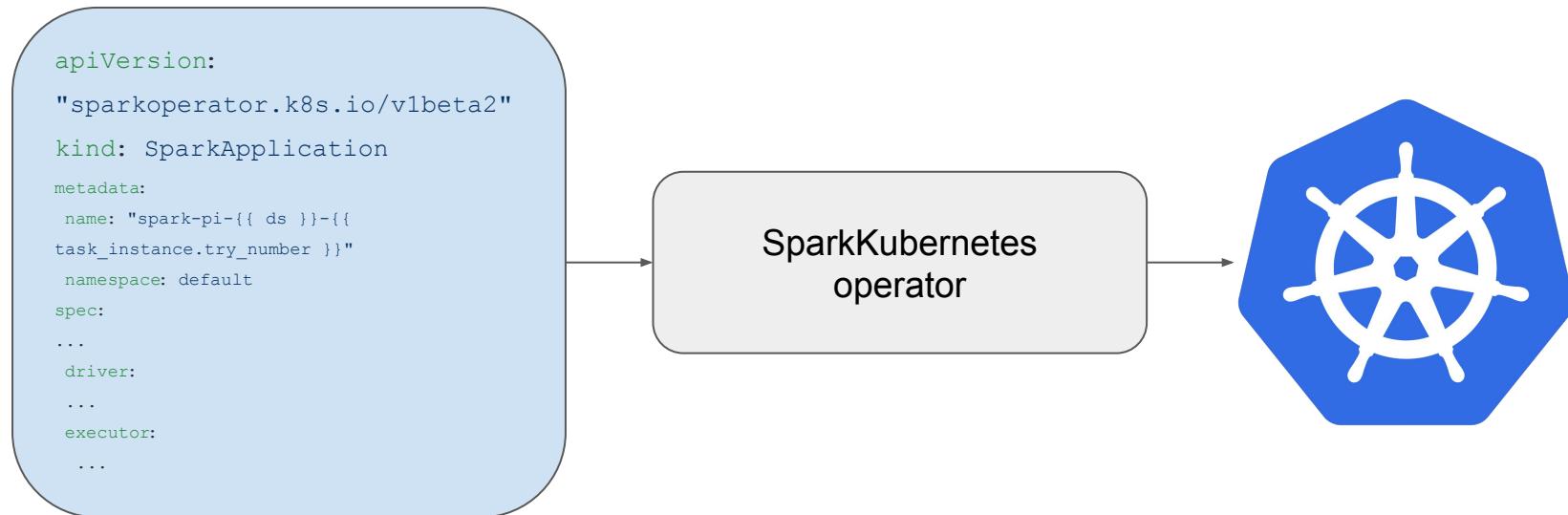


KubernetesHook

- Gets the **connection parameters** from Airflow's Kubernetes connection
 - E.g kubeconfig, namespace or in-cluster configuration
- Creates an **API client** and allows to create and get a Kubernetes `custom_resource_object`
 - *SparkApplication* is of type *customResource*

SparkKubernetes operator

- Gets the *SparkApplication* resource definition
- Templates it using Jinja
- Sends it to the Kubernetes cluster



SparkKubernetes sensor

- Pokes the *SparkApplication* as long as the *applicationState* is SUBMITTED or RUNNING
- Completes successfully if the *applicationState* is SUCCESS, or raises an exception if the *applicationState* is FAILED

What have we gained by building this integration?

- Official built-in Airflow support



- Security

- Save Kubernetes credentials inside Airflow connection mechanism



- Portability

- Use templated Kubernetes object so the same app can be migrated easily to Airflow and also be run manually

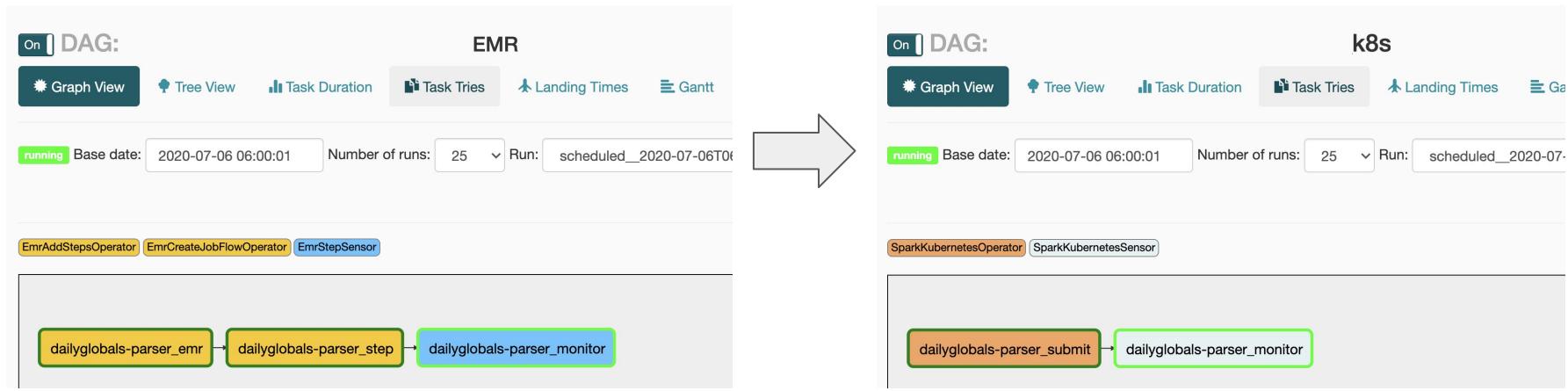


- Kubernetes native

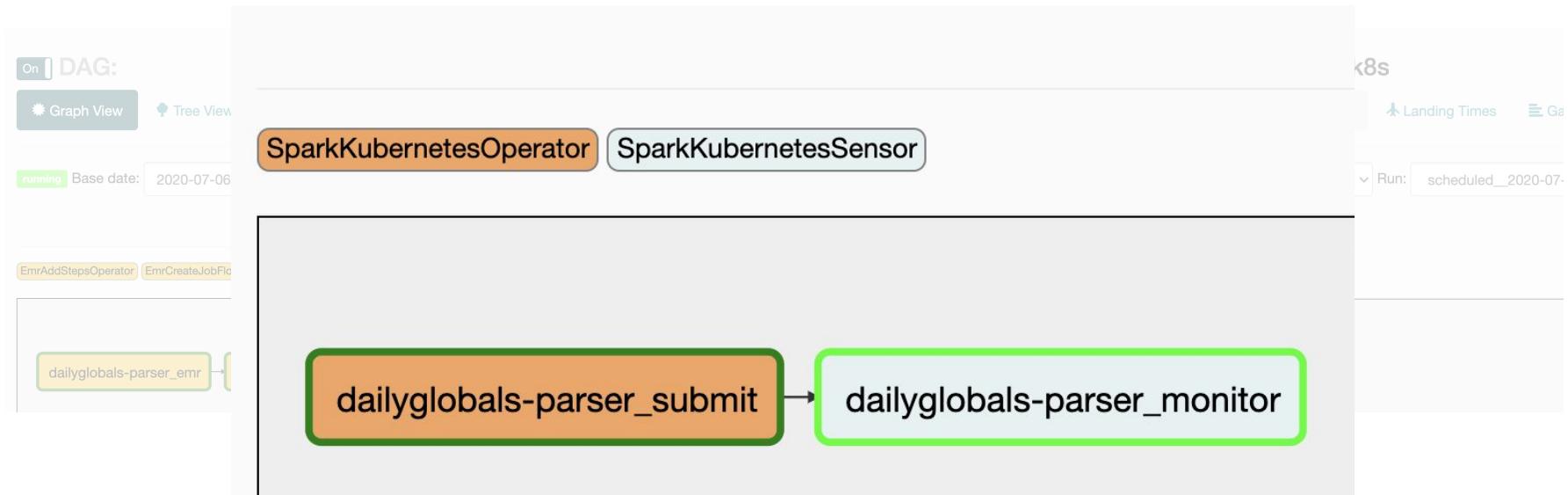
- Communicate directly with the Kubernetes API



Remember this slide?

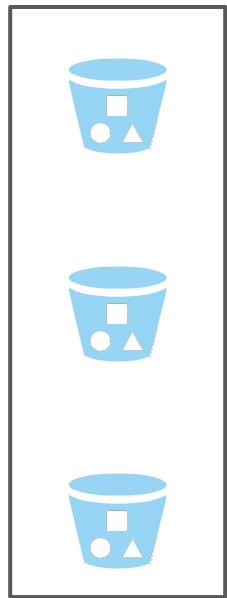


Let's take a closer look again...



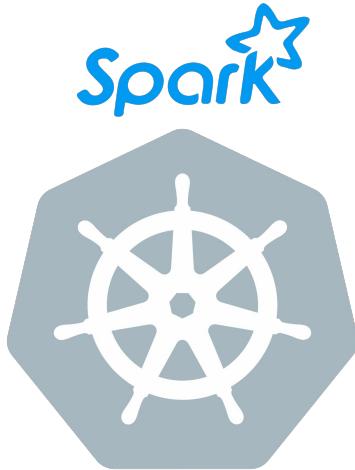
Common data pipeline pattern - revised

Data Lake



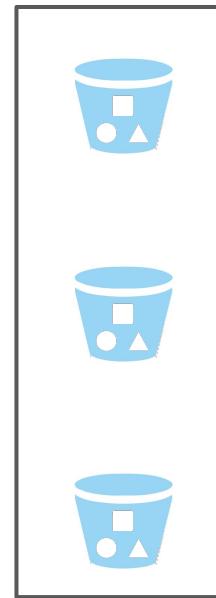
1.
Read input
files
»»

Data Processing



2.
Write output
files
»»

Intermediate Storage



3.
Ingest
to DB
»»

OLAP

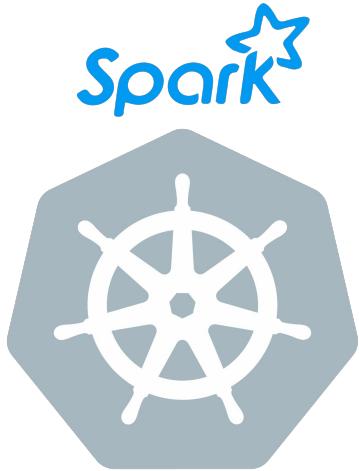


Common data pipeline pattern - revised

Data Lake



Data Processing



1.
Read input
files
»»

Intermediate Storage



2.
Write output
files
»»

OLAP



3.
Ingest
to DB
»»

Benefits from migrating to Kubernetes

- **~30% cost reduction**
 - No additional cost per instance
- Better visibility
 - E.g automatic Prometheus configuration using Spark-on-K8s operator
- Robustness
 - E.g running a Spark job on multiple AZs
(can be beneficial when using Spot instances)



Current status and future plans

- Airflow integration merged to master
 - Will be available on Airflow 2.0
 - Planned to be included as a backport package on Airflow 1.10.11
- Extend the new KubernetesHook and possibly migrate kubernetes_pod_operator to work with it
- We're gradually migrating our DAGs

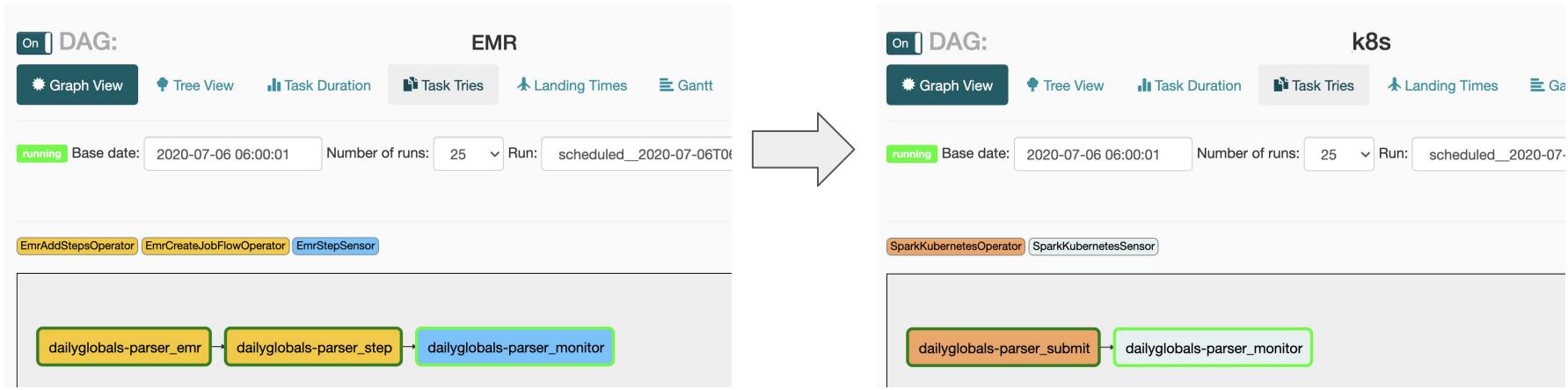
Airflow “pro” tips

- Use **Jinja templating**
 - Provides the DAG developer with a set of built-in parameters and macros
- Make Airflow more **secured** using K8s' role-based access control (RBAC)

Airflow “pro” tips

- **Monitoring and alerting**
 - Use the DAG's `on_success_callback` and `on_failure_callback` (in `default_args`) to **emit metrics** to your monitoring system and **trigger alerts** based on those metrics
- Leverage Airflow's **built-in components**
(Operator/Hook/Sensor)
 - Couldn't find one? Build & **contribute** one 😊

Same-same...



... but different - for you!



Reduce costs



Open your eyes



Robust

- Migrate **your** Spark jobs to Kubernetes
- Leverage the easy-to-use **native Airflow integration** we contributed

Want to know more?



- Women in Big Data
 - A world-wide program that aims :
 - To inspire, connect, grow, and champion success of women in the Big Data & analytics field
 - 30 chapters and 17,000+ members world-wide
 - Everyone can join (regardless of gender), so find a chapter near you -
<https://www.womeninbigdata.org/wibd-structure/>
- Our Tech Blog - medium.com/nmc-techblog
 - Spark Dynamic Partition Inserts part 1 - <https://tinyurl.com/yd94ztz5>
 - Spark Dynamic Partition Inserts Part 2 - <https://tinyurl.com/y8uembml>

QUESTIONS



THANK YOU



[Roi Teveth](#)



[Roi Teveth](#)



[Itai Yaffe](#)



[Itai Yaffe](#)