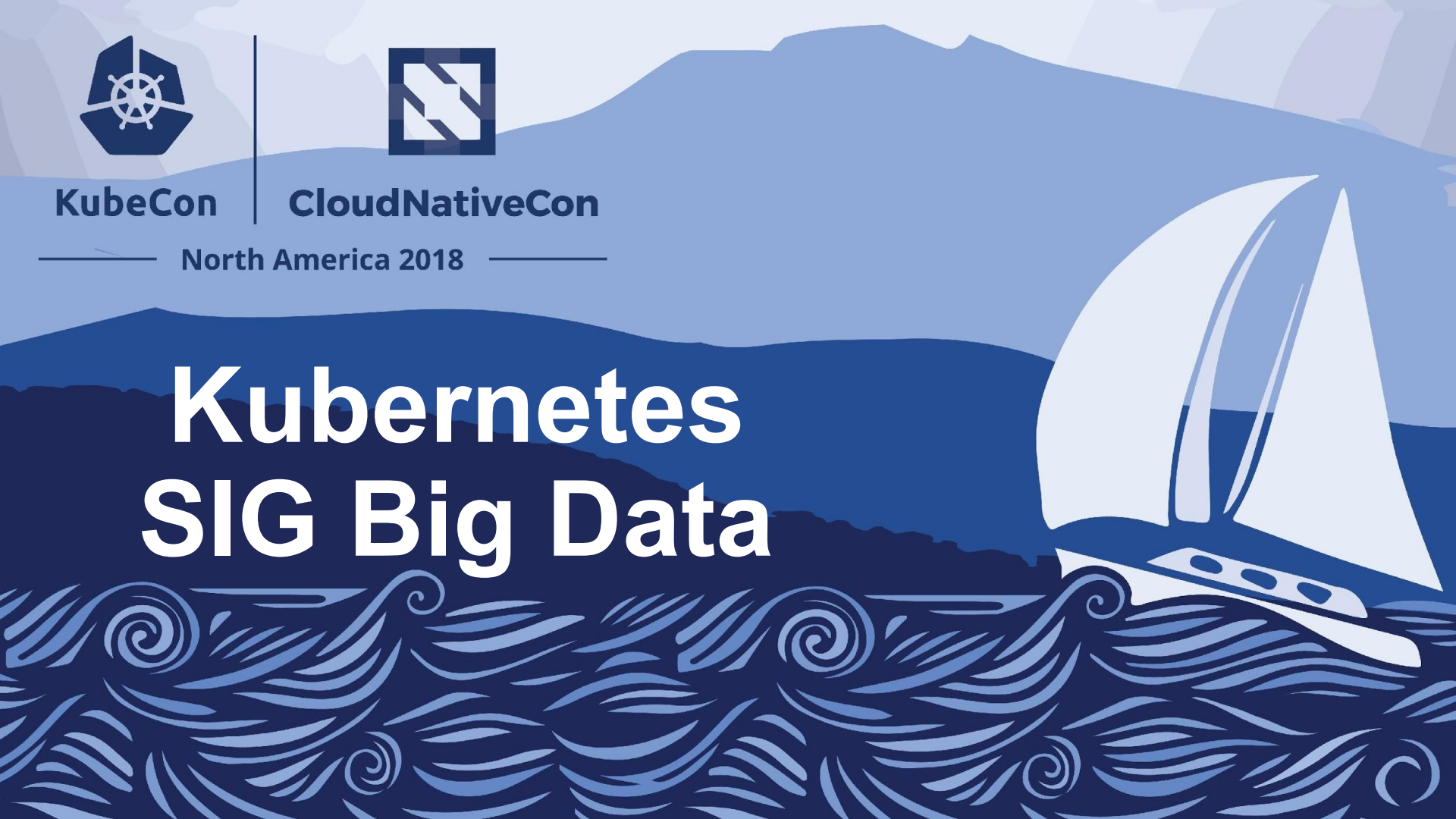

KubeCon
CloudNativeCon
North America 2018
Kubernetes
SIG Big Data

Landscape

Introduce SIG Big Data

Apache Airflow

Apache Spark

Future of this SIG

Audience Dialogue

SIG Mission[1]

Serve as a community resource for advising big data and data science related software projects on techniques and best practices for integrating with Kubernetes.

Represent the concerns of users from big data communities to Kubernetes for the purposes of driving new features and other enhancements, based on big data use cases.

1: https://github.com/kubernetes/community/pull/2988

Chairs

Anirudh Ramanathan (Rockset)

Erik Erlandson (Red Hat)

Yinan Li (Google)

Provenance

Winter 2017: Revived as part of the process of creating a community platform for prototyping development of a Kubernetes scheduler backend for Apache Spark.

Software Projects

Kubernetes scheduler backend for Apache Spark

HDFS deployments for Kubernetes

Apache Airflow operator and executor for Kubernetes

Participating Organizations

Bloomberg

Google

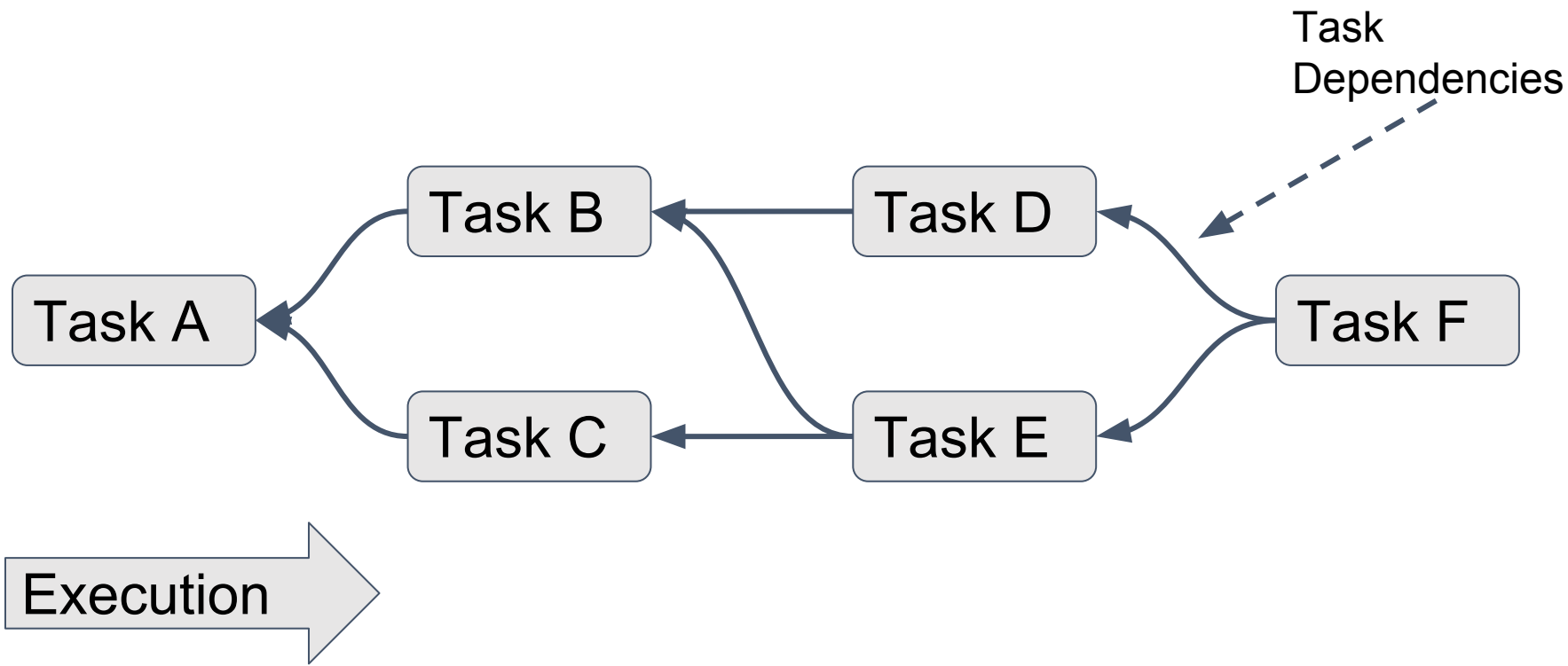
Lightbend

Palantir

Pepperdata

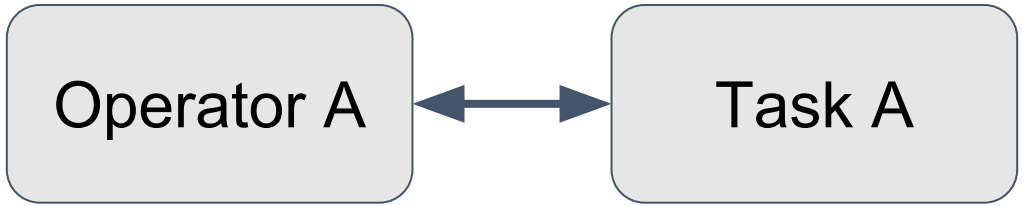
Red Hat

Apache Airflow



Airflow Operators

- Units of work
- Corresponds to a command or functionality
- Associates with a task ID
- Contains parameters and other resources to execute



Operator Flavors

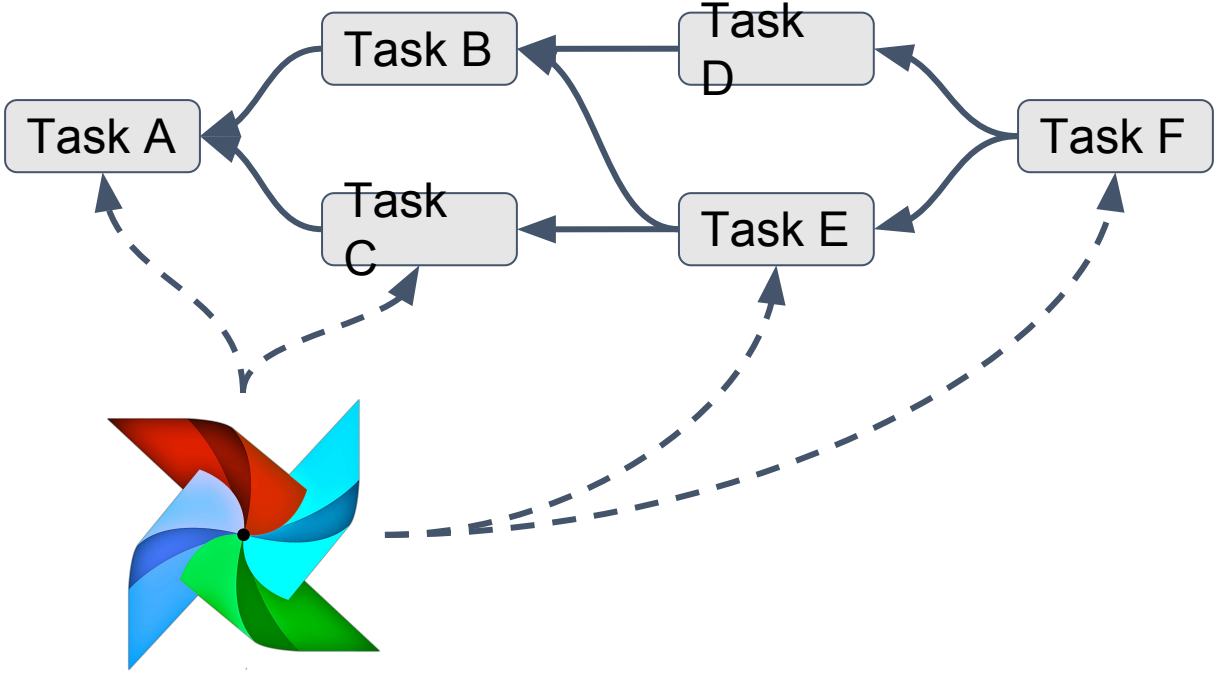
Run a bash command

Invoke a python function

Send an HTTP request

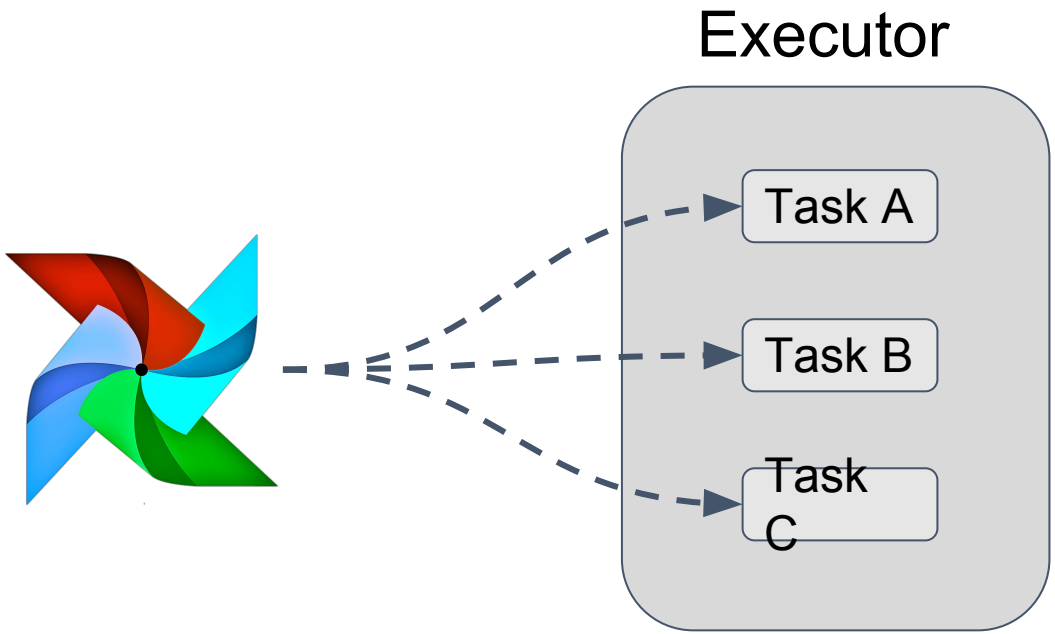
Execute a SQL query

Airflow Scheduler



- Runs tasks in order
- Tracks success and failure

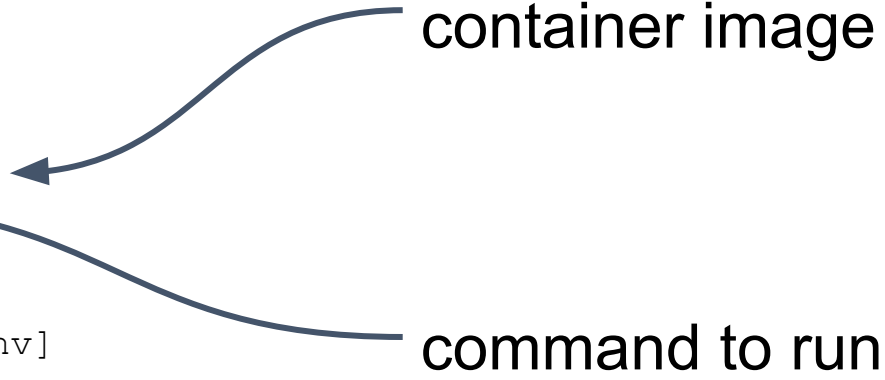
Airflow Executors



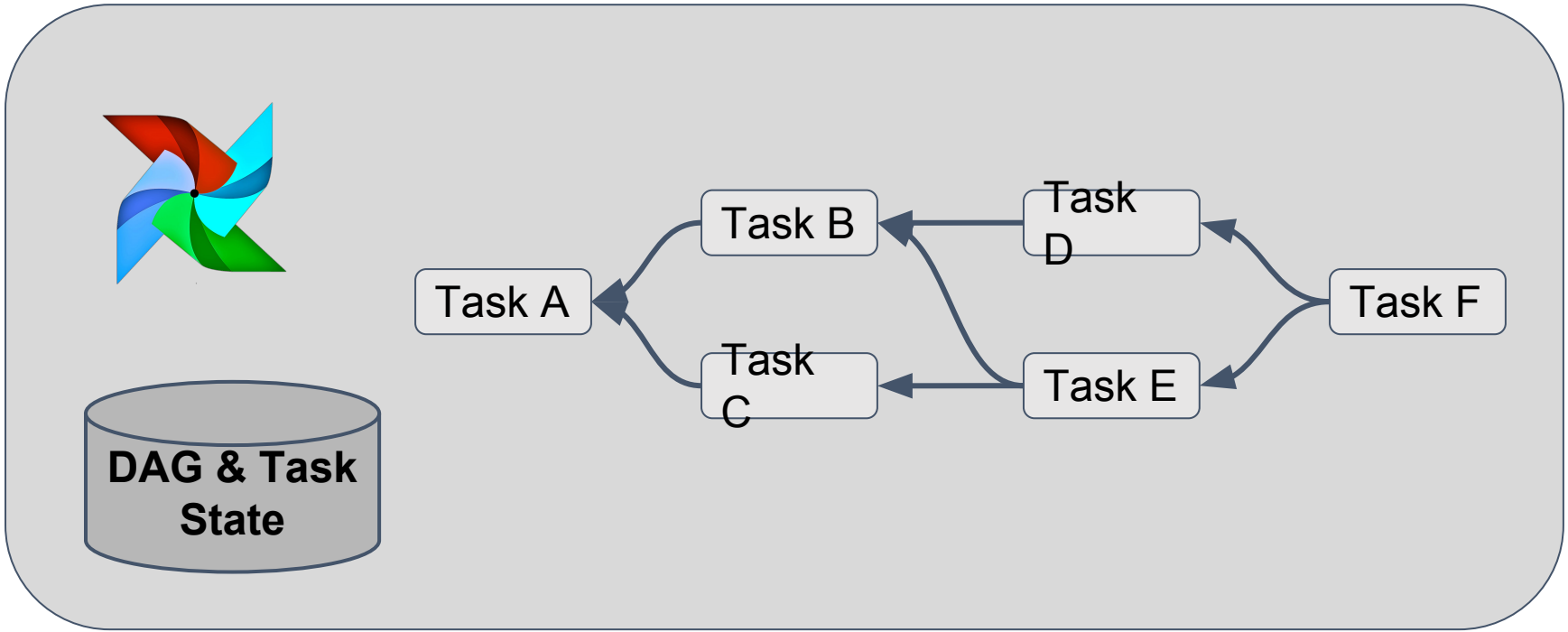
- Local
- Mesos
- Kubernetes

Kubernetes Operator


```
op = KubernetesPodOperator(
    name="example",
    task_id="Task-A",
    namespace='default',
    image=[container_image_name],
    cmds=["bash", "-cx"],
    arguments=["echo", "K8S!"],
    labels={"label": "value"},
    secrets=[secret_file,secret_env]
    volume=[volume],
    volume_mounts=[volume_mount]
    affinity=affinity,
    is_delete_operator_pod=True,
    hostnetwork=False,
    tolerations=tolerations)
```



Airflow On Kubernetes



Airflow DAG Provisioning

GitHub

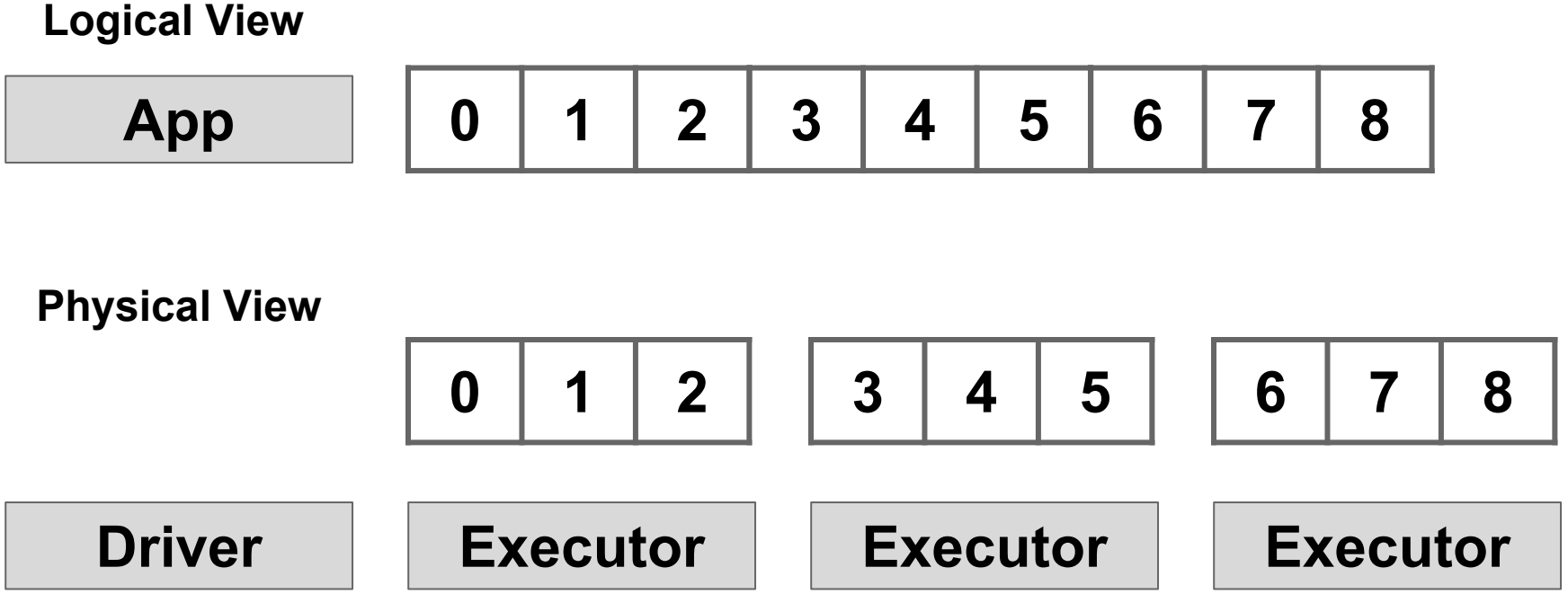
NFS

EFS

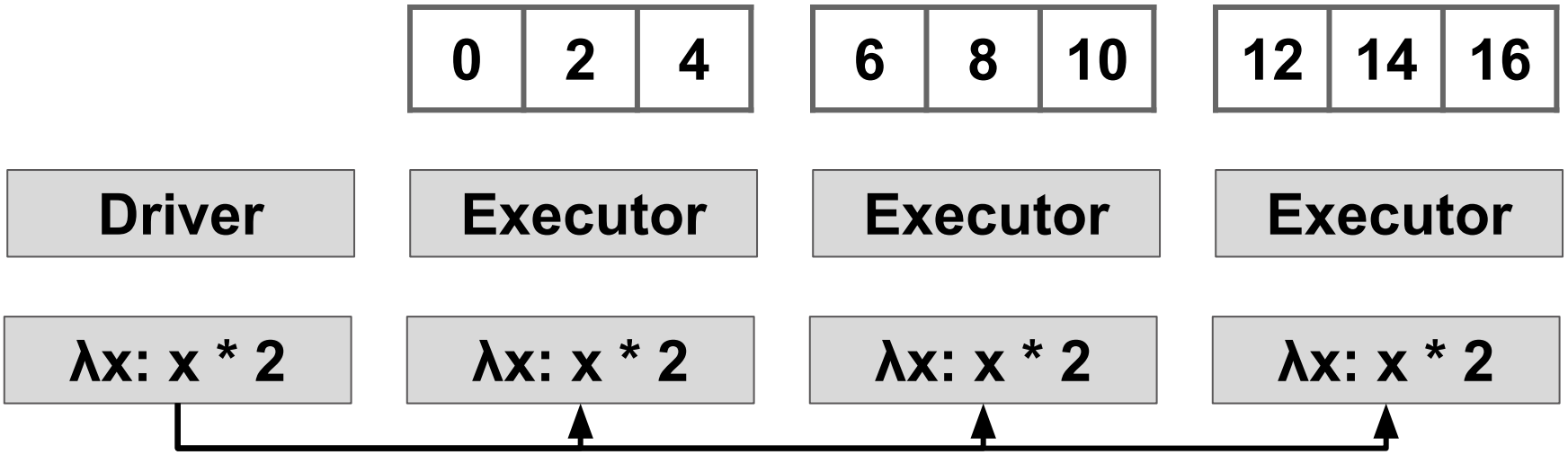
Cinder

[Extend via hooks ...]

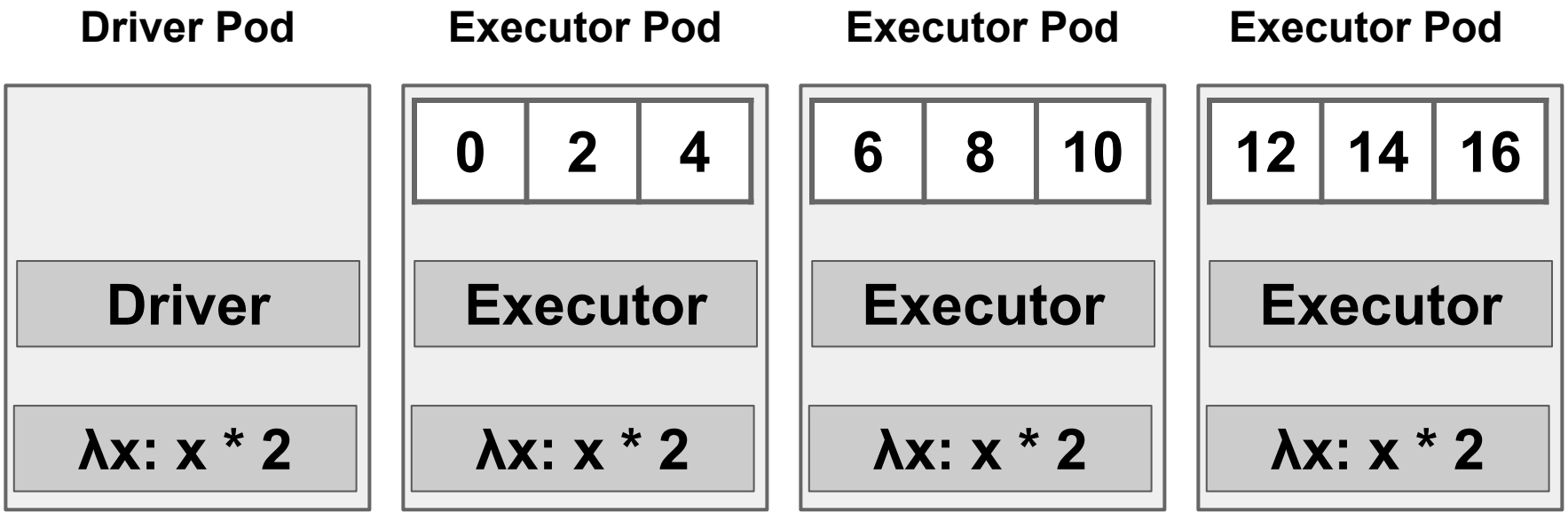
Apache Spark Compute Model



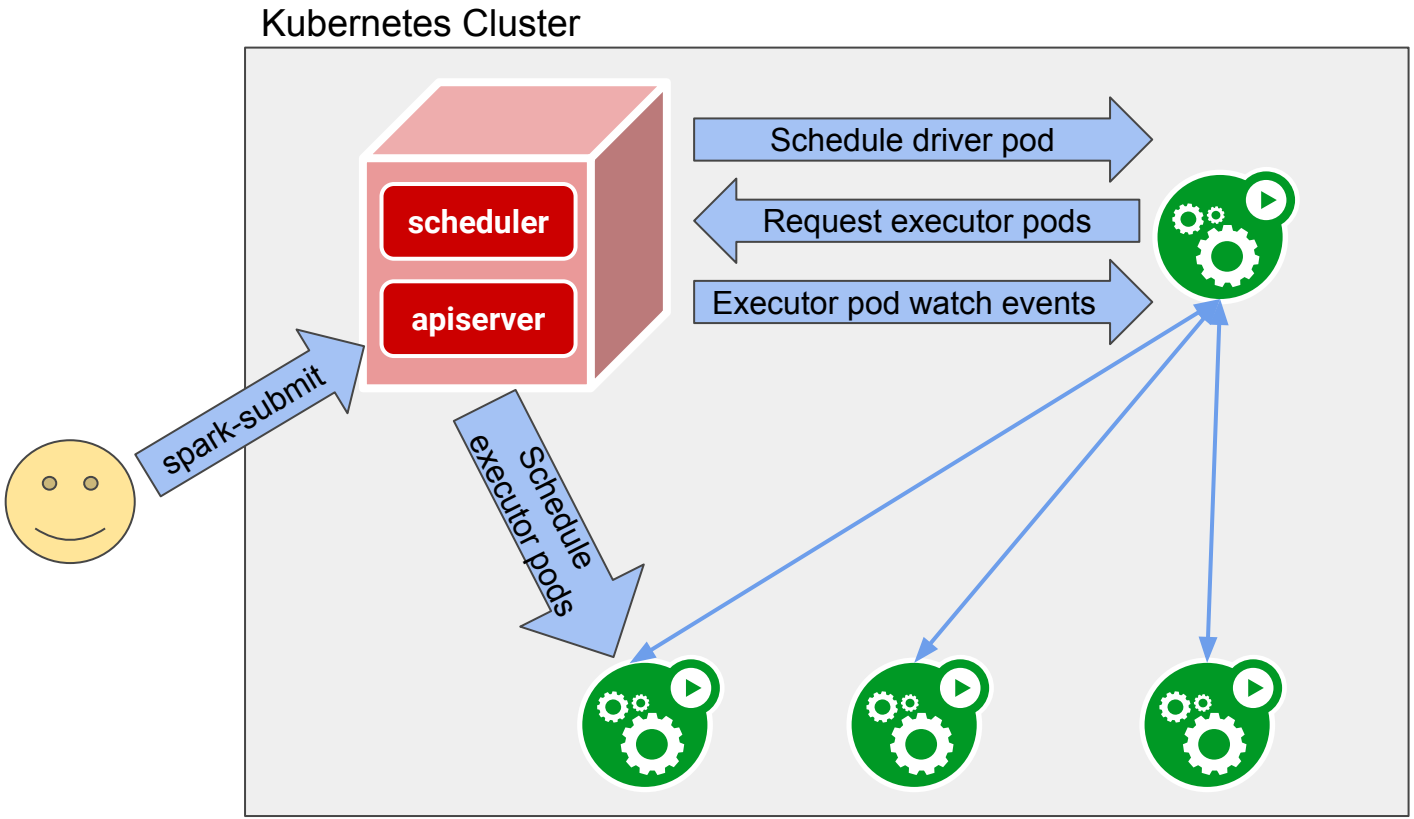
Apache Spark Compute Model



Spark on Kubernetes



Cluster Mode



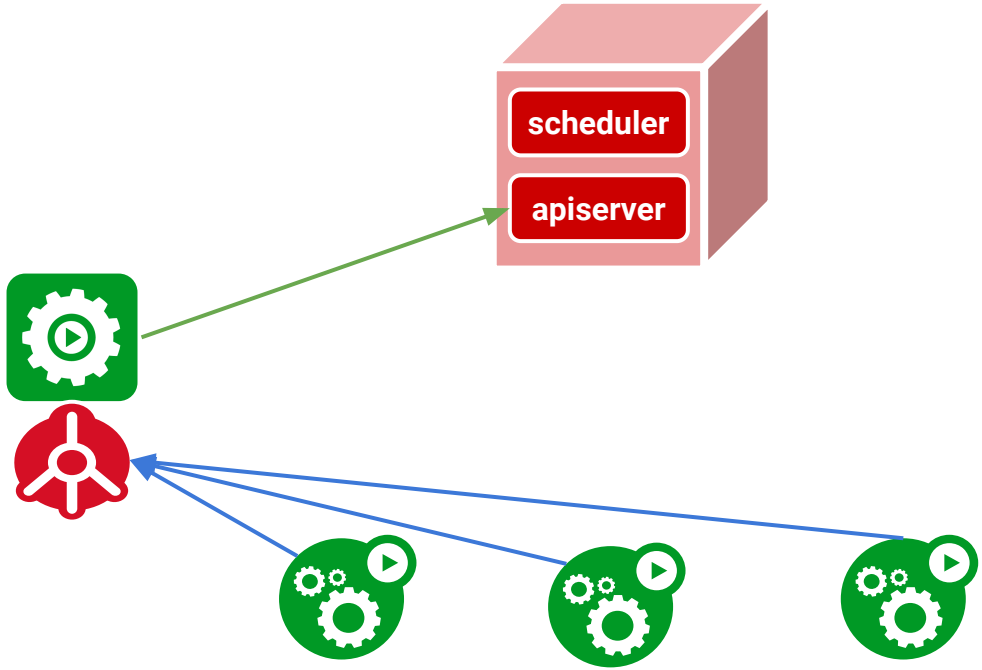
K8S Scheduler Backend for Spark

What we have done so far

- Initial release in Spark 2.3.0 with support for cluster mode, Java/Scala, remote dependencies, and limited pod customization.
- More features in release 2.4.0: Python, R, and limited client mode support.
- New features in upcoming Spark 3.0: Kerberos support and support for pod customization using a pod template.

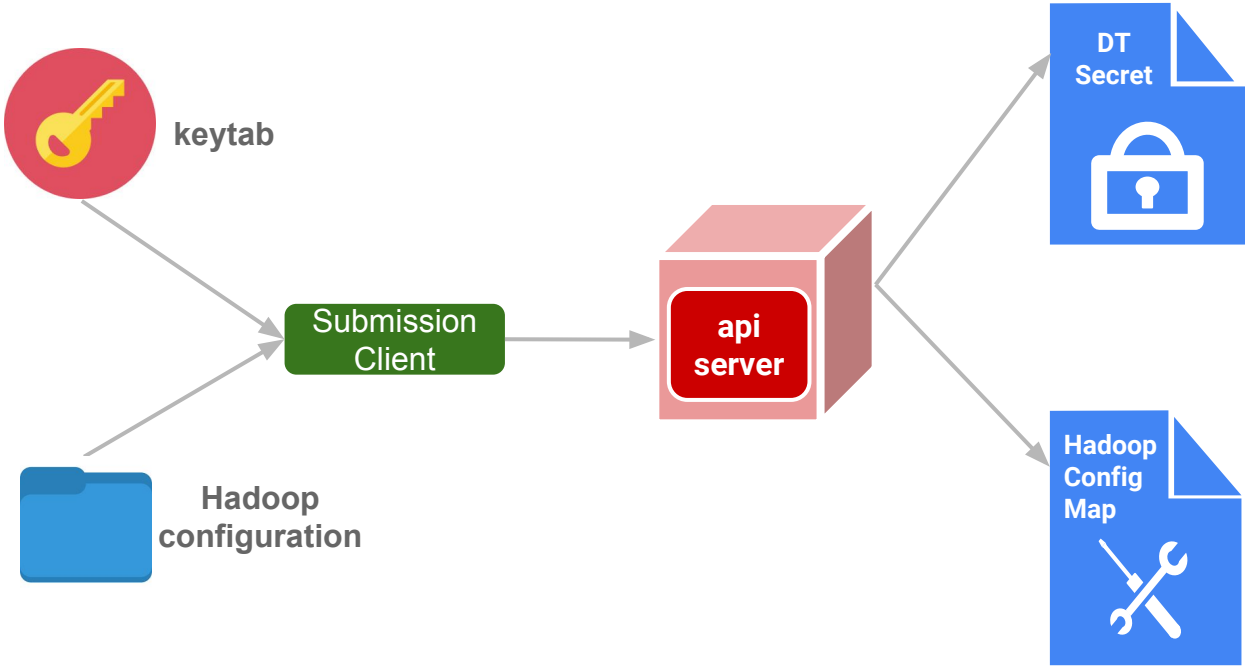
Client Mode (2.4)

- Useful for interactive apps, e.g., notebooks and spark-shell.
- Supports drivers running both inside and outside the cluster
- **Garbage collection** of executor pods supported for in-cluster
- Users are responsible for setting up **network connectivity** from executors to the driver
  - E.g., a headless service for in-cluster

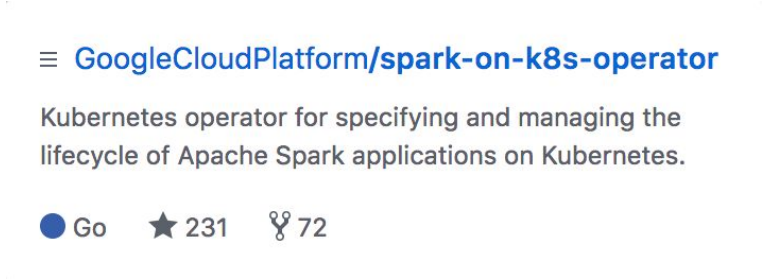


Kerberos Support (3.0)

- Necessary for secure HDFS access.
- Needs both a Delegation Token (DT) and Hadoop configuration
- Does not yet support delegation token renewal.



Kubernetes Operator for Spark



- Kubernetes CRD + custom controller
- Supports extensive pod customization through a mutating admission webhook
- Native Cron support for running scheduled applications
- Automatic application restart with a configurable restart policy
- Supports exporting application-level metrics and driver/executor metrics to Prometheus
- Supports installation with Helm
- Comes with a command-line tool `sparkctl`

```
apiVersion: "sparkoperator.k8s.io/v1alpha1"
kind: SparkApplication
metadata:
  name: spark-pi
  namespace: default
spec:
  type: Scala
  mode: cluster
  image: "gcr.io/spark-operator/spark:v2.4.0"
  mainClass: org.apache.spark.examples.SparkPi
  mainApplicationFile: "..."
  driver:
    memory: "512m"
    serviceAccount: spark
  executor:
    instances: 1
    memory: "512m"
  monitoring:
    exposeDriverMetrics: true
    exposeExecutorMetrics: true
    prometheus:
      port: 8090
  restartPolicy: Never
```

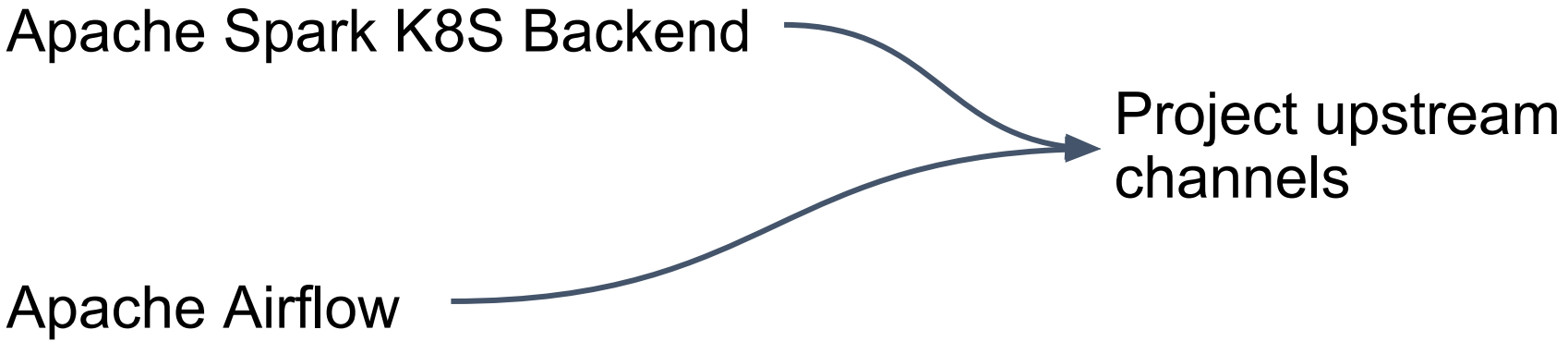
Roadmap (3.0 and Beyond)

- Support for using a **pod template** to customize the driver and executor pods.
  - No more new configuration properties
- Dynamic resource allocation and external shuffle service.
  - New shuffle service work in progress
- Better support for local application dependencies on client machines.
- Driver resilience for Spark Streaming applications.
- Better scheduling support.

Getting Involved

- github.com/apache/spark: code under resource-managers/kubernetes
- Documentation: http://spark.apache.org/docs/latest/running-on-kubernetes.html
- Spark user & dev mailing lists
- Jira (use Kubernetes for Component)
- Slack sig-big-data: https://kubernetes.slack.com/messages/sig-big-data

Trajectory: “Graduating” Projects



Trajectory: Reaching Out To New Communities

Flink Operator Demo

Hazelcast (IMDG & Jet)

SIG Big Data Charter

Charter currently submitted for consideration
https://github.com/kubernetes/community/pull/2988

Kubernetes Definition of a SIG: Owns some component, subsystem or other body of Kubernetes code.

Future Directions

Acquire ownership of Kubernetes code

Working Group

SIG sub-project

User Community

Your Big Data or Data Science Community

Your Community
In This Space

Thank You!

1. New Communities
2. Airflow or Spark
3. Future of SIG Big Data

Erik
eje@redhat.com
@eje on kubernetes.slack.com

Yinan
ynli@google.com
@liyinan926 on kubernetes.slack.com