Apache Spark on Kubernetes

Anirudh Ramananathan (foxish@google.com)
Software Engineer (Kubernetes)

Timothy Chen (tim@hyperpilot.io)
Co-founder & CTO (HyperPilot)

Agenda

- Kubernetes & Containers
- Motivation
- Design
- Demo
- Deep Dive
- Roadmap

What is Kubernetes?

Kubernetes

Kubernetes is an open-source system

Kubernetes

Kubernetes is an open-source system for automating deployment, scaling, and management

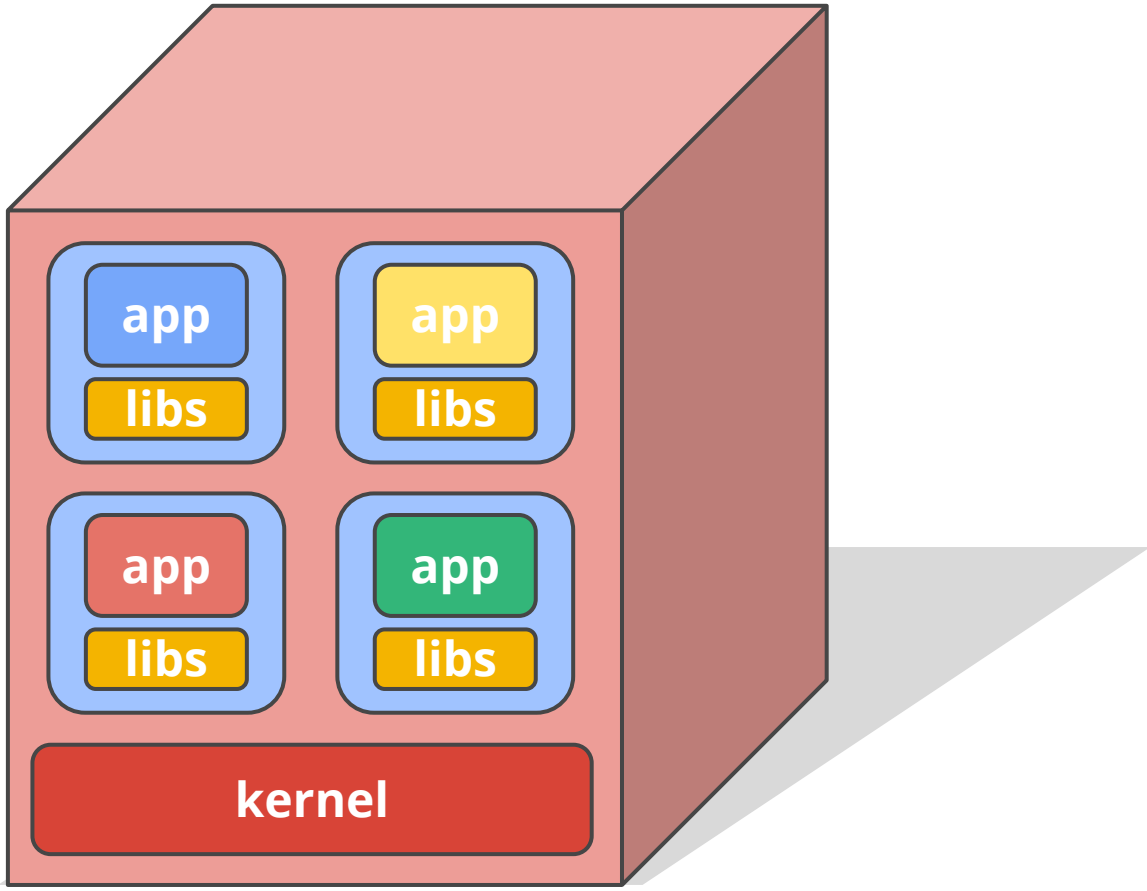
Kubernetes

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

‘Containerized’

Containers

- Repeatable Builds and Workflows
- Application Portability
- High Degree of Control over Software
- Faster Development Cycle
- Reduced dev-ops load
- Improved Infrastructure Utilization

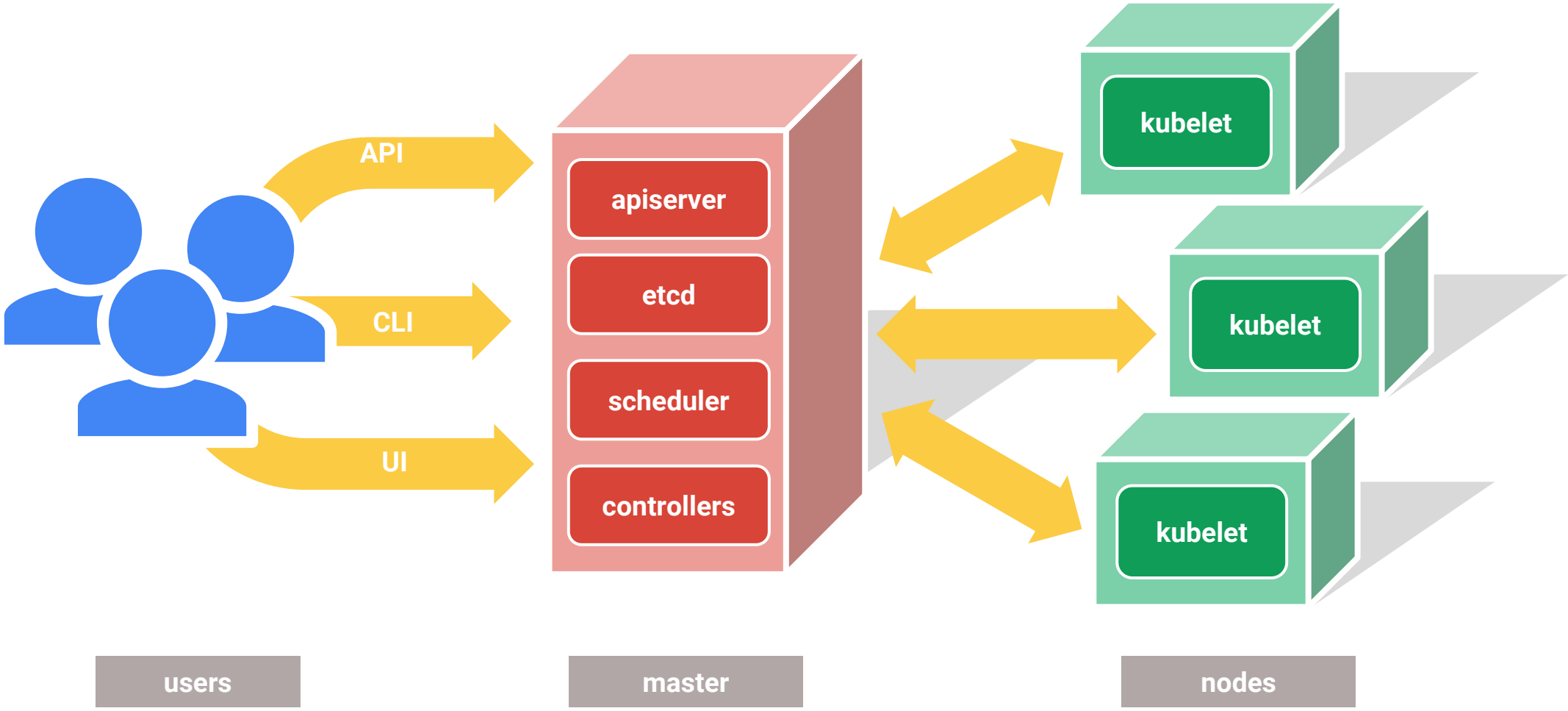


Kubernetes

- Large OSS Community - **1200+ contributors and 45k+ commits**
- Ecosystem and Partners - **100+ organizations involved**
- One of the top 100 projects **overall** on GitHub - **23k+ stars**
- Large production deployments on-prem and on various cloud providers
- Built with multi-tenant and multi-cloud deployments in mind

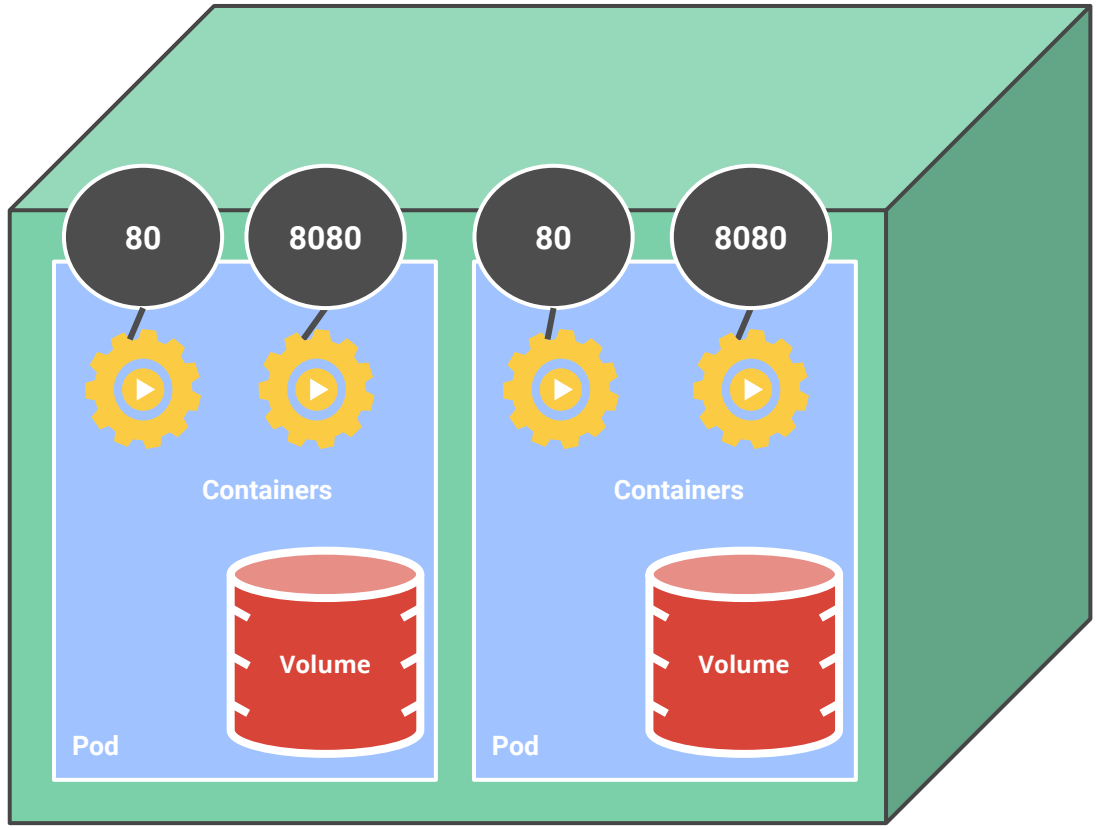
Overview

At a Glance


API
CLI
UI
apiserver
etcd
scheduler
controllers
kubelet
kubelet
kubelet
users
master
nodes

Nodes and Pods

- A **pod** is a set of co-located containers
- Created by a declarative specification supplied to the master
- Each pod has its own IP address
- Volumes can be local or network-attached



Node

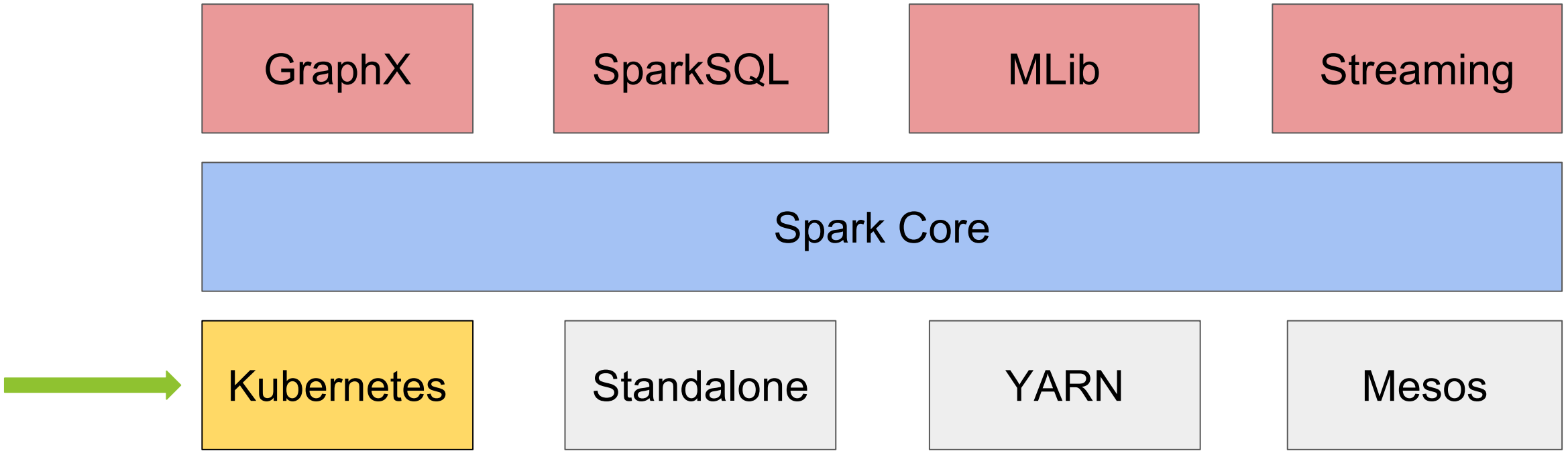
Motivation

Why Spark on Kubernetes?

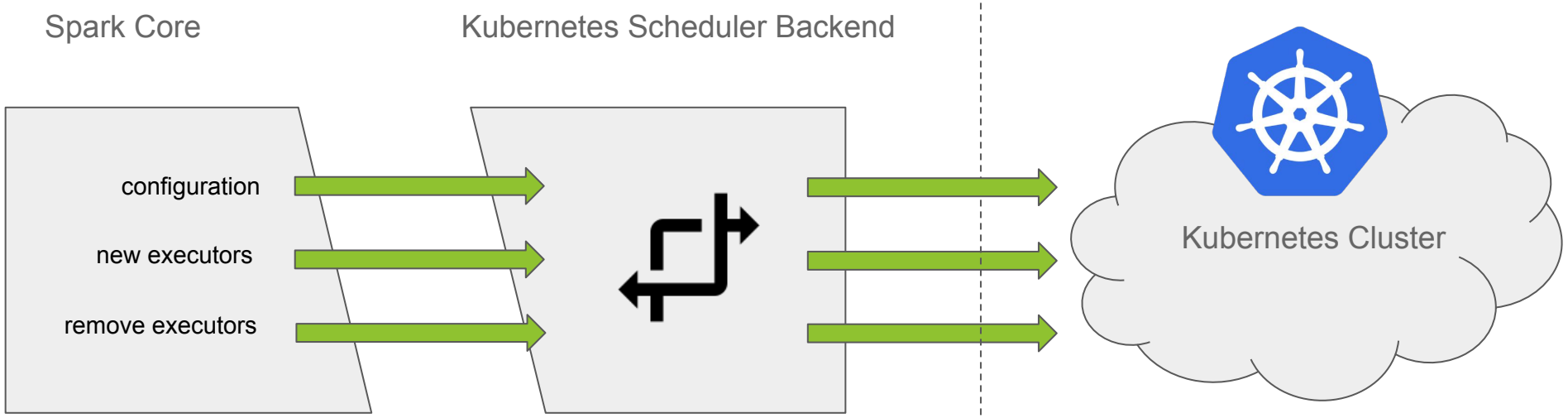
- Docker and the Container Ecosystem
- Kubernetes
  - Lots of addon services: third-party logging, monitoring, and security tools
  - For example, the Istio project, announced May 24, by IBM, Google and Lyft, provides a service mesh for authenticating, authorizing, tracing, and timing, and rate-limiting container-to-container communication, and more.
- Resource sharing between batch, serving and stateful workloads
  - Streamlined developer experience
  - Reduced operational costs
  - Improved infrastructure utilization

Design

Spark, meet Kubernetes!



Spark, meet Kubernetes!

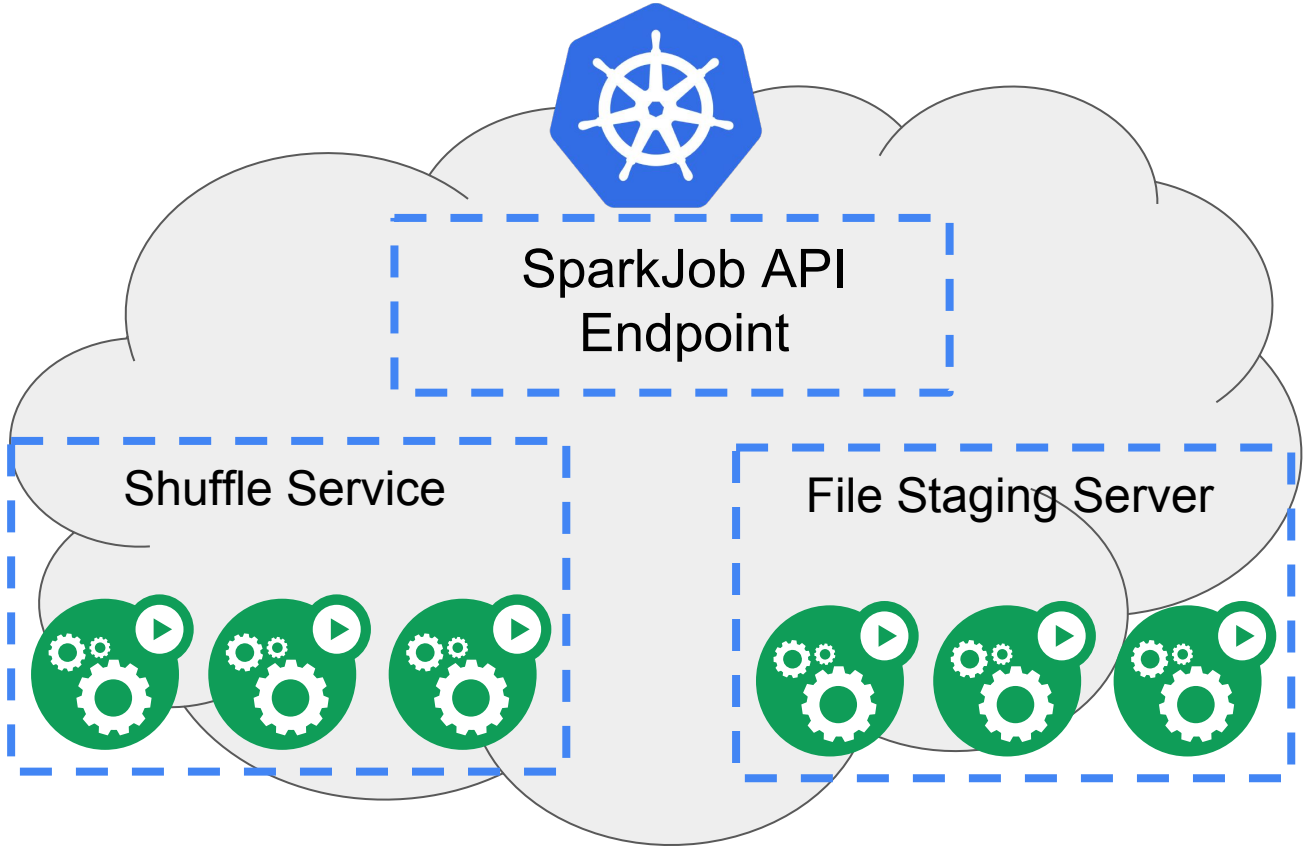


- Resource Requests
- Authnz
- Communication with K8s

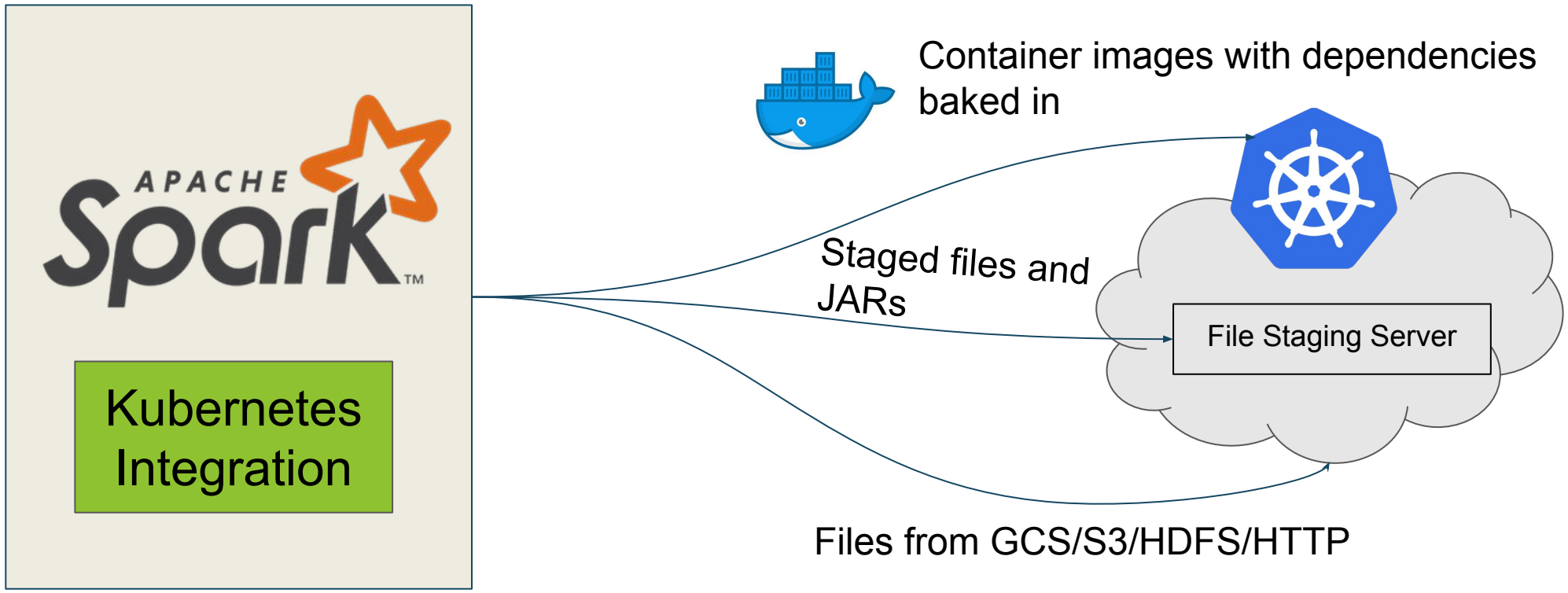
- Runs Spark Drivers/Executors
- Runs Shuffle Service
- Runs Additional Components for Spark jobs

Kubernetes, meet Spark!

- **Staging server**: component to stage local files
- **Spark Shuffle service**: component to store shuffle data for dynamic allocation
- **ThirdParty/CustomResources:** extend Kubernetes API with Spark Knowledge



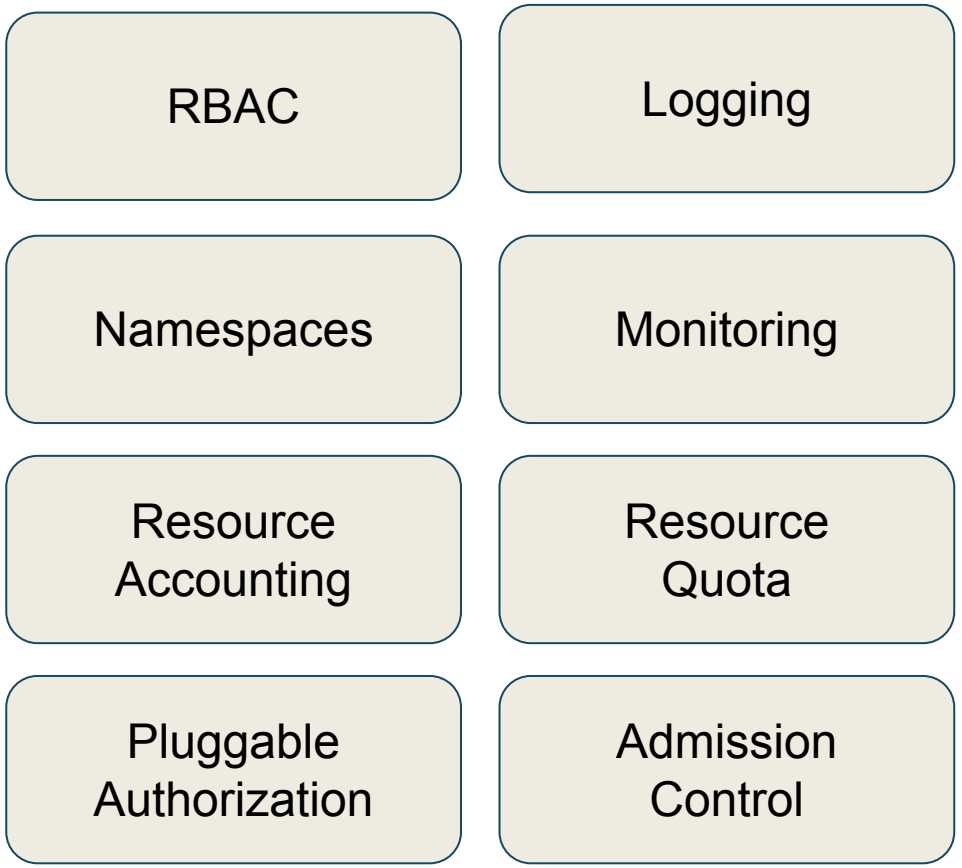
Dependencies



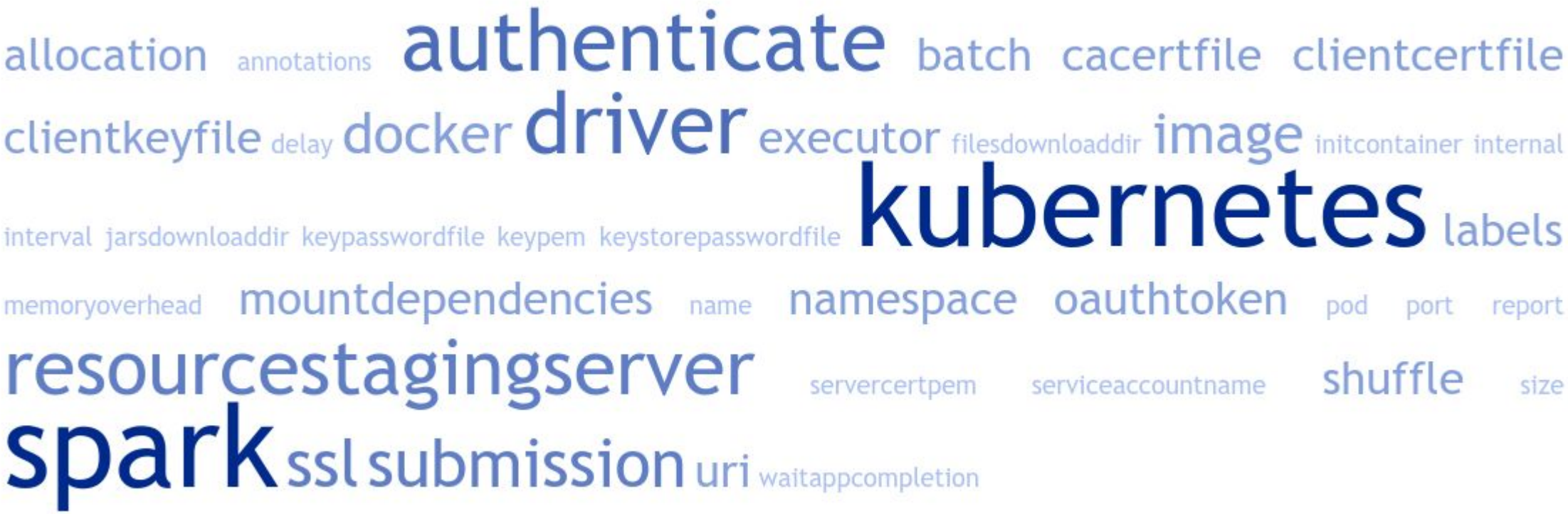
Several ways of running Spark Jobs along with their dependencies on Kubernetes

Administration

- Launch Spark Jobs as a particular user into a specific namespace
- RBAC and Namespace-level resource quotas
- Audit logging for clusters
- Several monitoring solutions to see node, cluster and pod-level statistics



Focus Areas



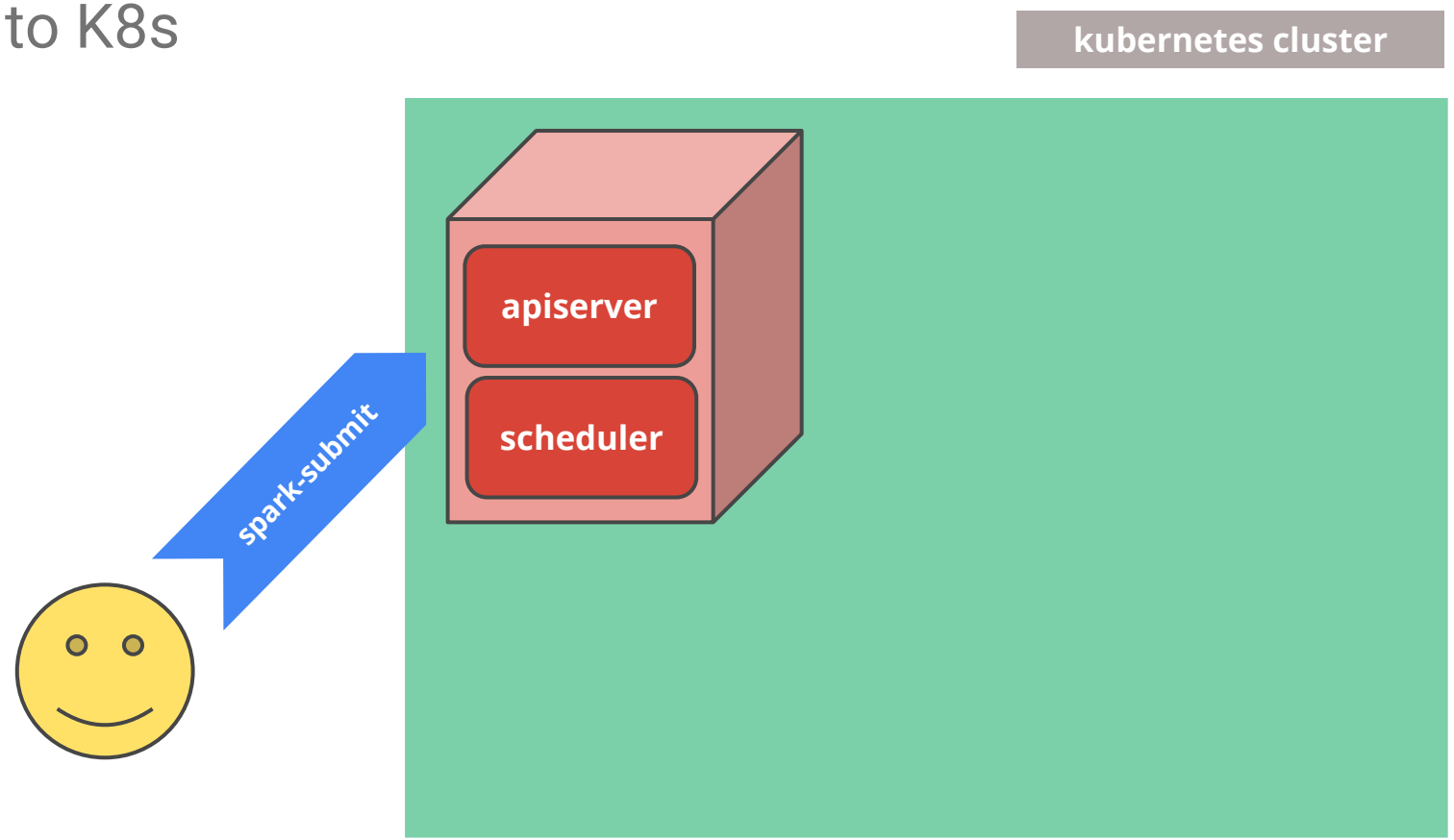
Wordcloud of the command-line options we added to spark-submit on Kubernetes

Demo

Deep Dive

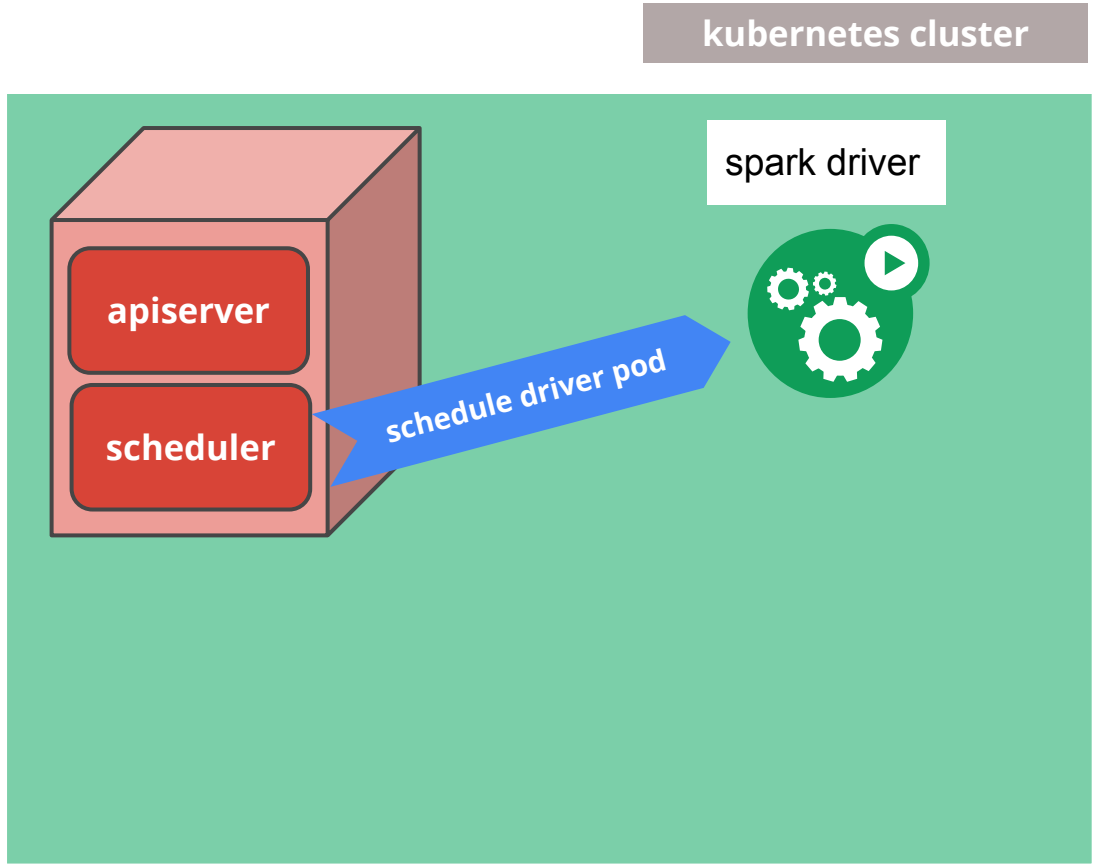
Deep Dive

- Spark Submit submits job to K8s



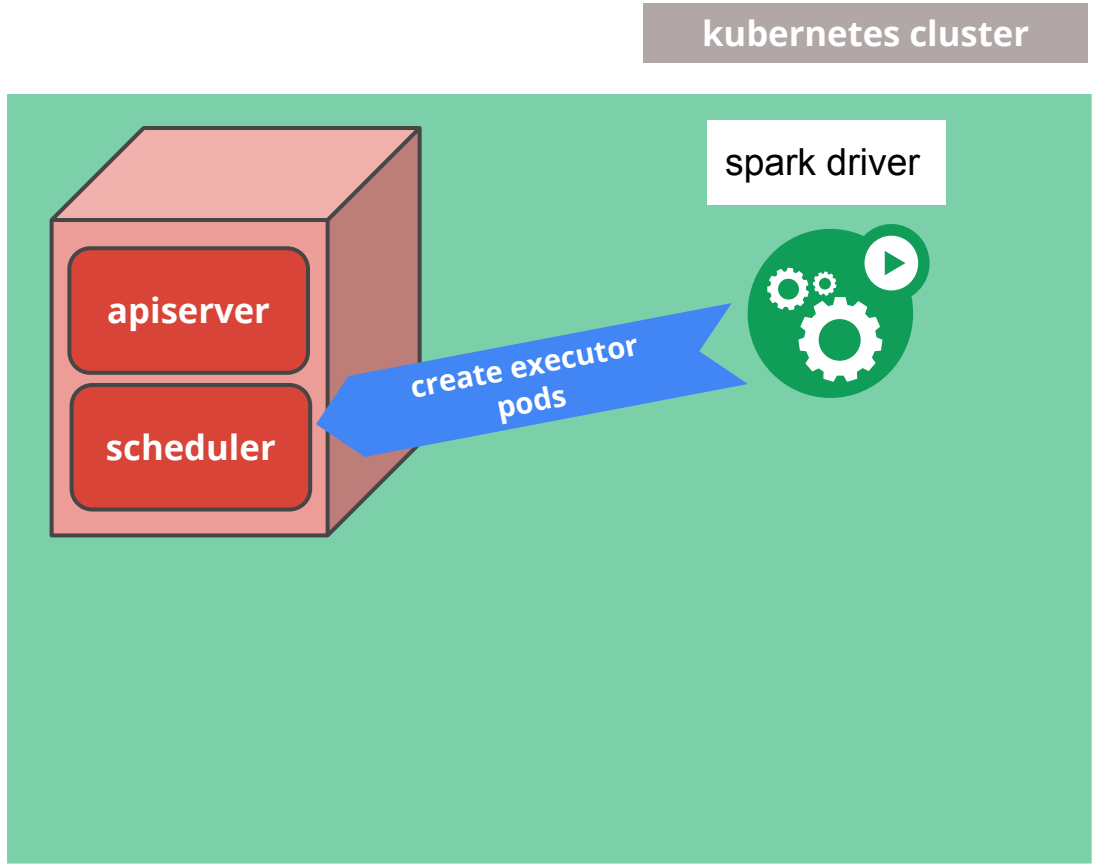
Deep Dive

- Spark Submit submits job to K8s
- K8s schedules the driver for job



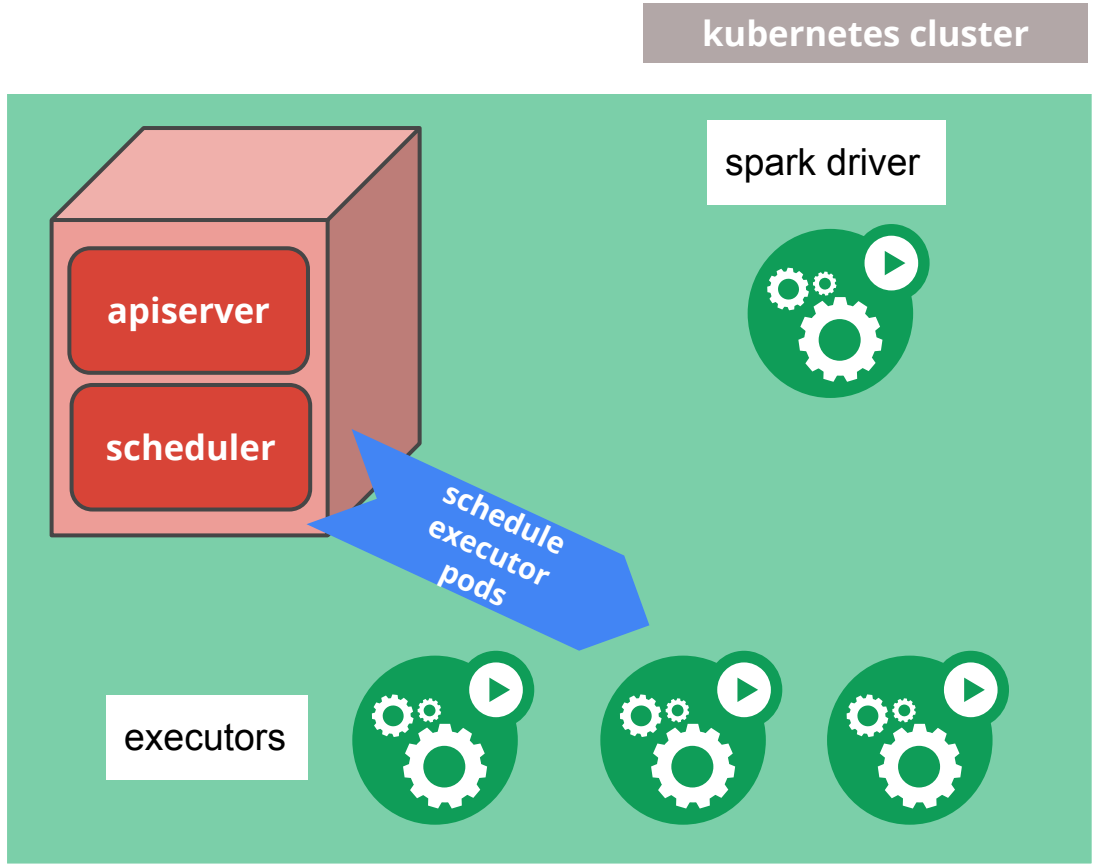
Deep Dive

- Spark Submit submits job to K8s
- K8s schedules the driver for job
- Driver requests executors as needed



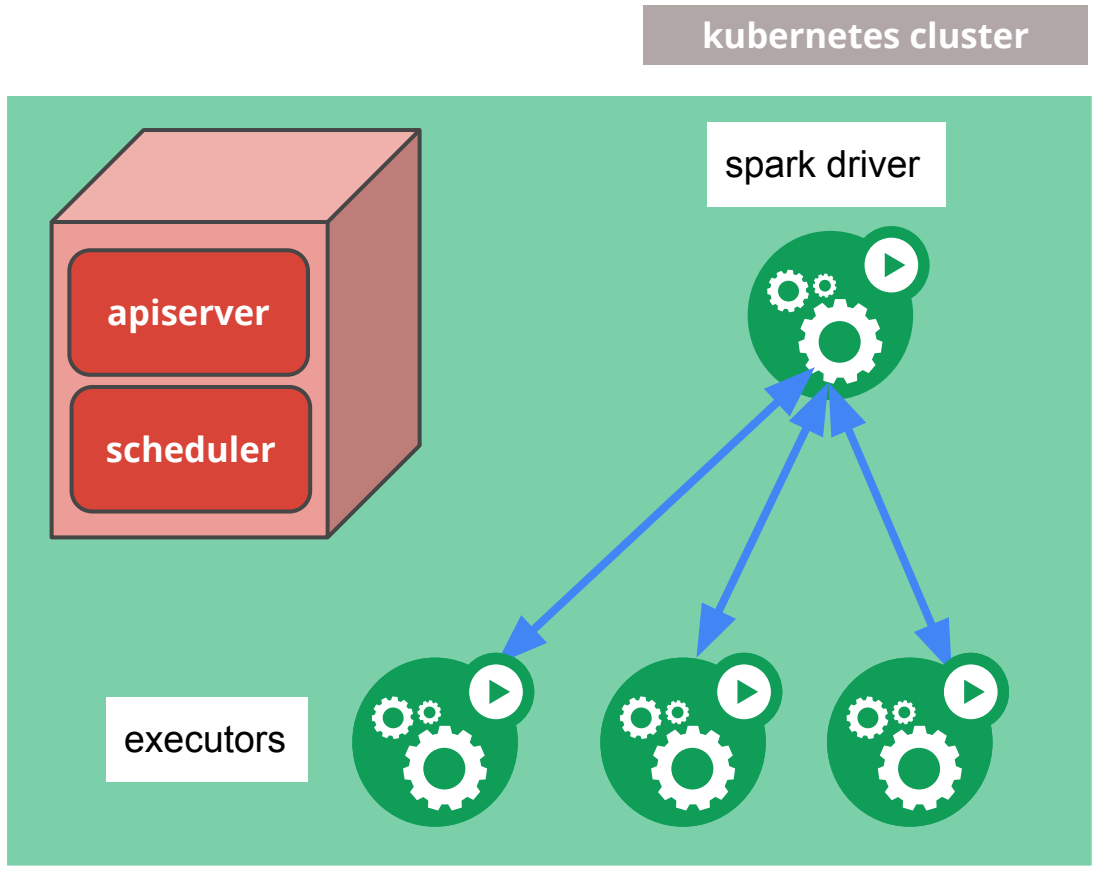
Deep Dive

- Spark Submit submits job to K8s
- K8s schedules the driver for job
- Driver requests executors as needed
- Executors scheduled and created



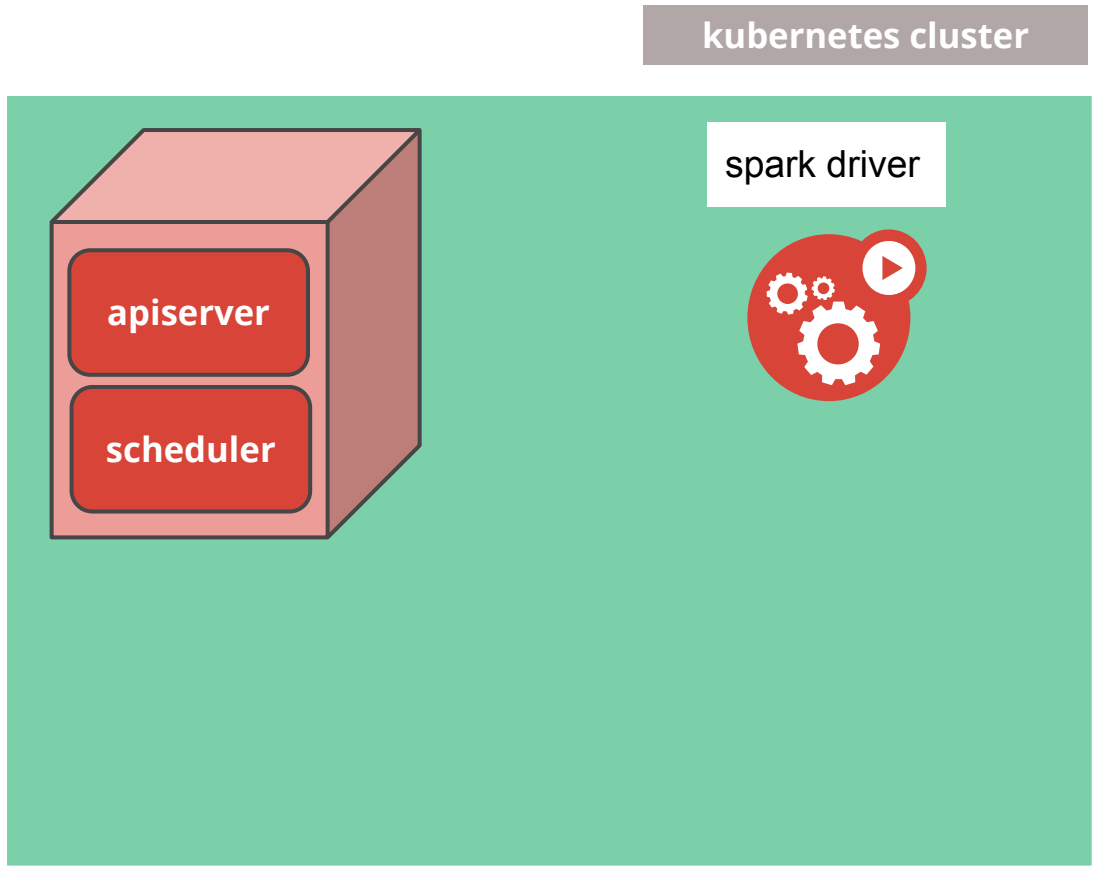
Deep Dive

- Spark Submit submits job to K8s
- K8s schedules the driver for job
- Driver requests executors as needed
- Executors scheduled and created
- Executors run tasks



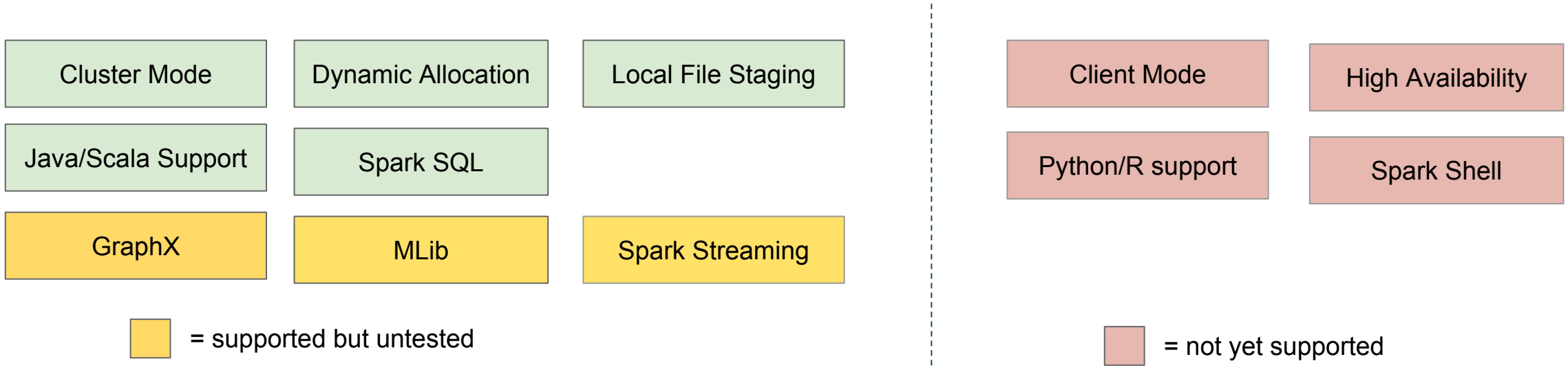
Deep Dive

- Spark Submit submits job to K8s
- K8s schedules the driver for job
- Driver requests executors as needed
- Executors scheduled and created
- Executors run tasks
- Driver “completes” job and persists logs



Roadmap

Spark Roadmap



We’re just getting started...

- Kubernetes CustomResources
- Priorities and Preemption for Pods
- Batch Scheduling and Resource Sharing
- Cluster Federation and Multi-cloud deployments
- Ecosystem: Kafka, Cassandra, HDFS, etc

Contributors

Organizations Alphabetically:

- Google
- Haiwen
- Hyperpilot
- Intel
- Palantir
- Pepperdata
- Red Hat

Links:

- Spark 2.2.0 Documentation
- https://issues.apache.org/jira/browse/SPARK-18278
- https://github.com/apache-spark-on-k8s/spark
- https://github.com/kubernetes/kubernetes/issues/34377

Thank You.

HDFS on Kubernetes - Lessons Learned
June 7 at 11:00 AM in Room 2003

Join us Wednesdays at 10am PT at the SIG BigData meeting
https://github.com/kubernetes/community/