

Feeding the Kubernetes beast

Bringing data locality back to data workloads

Adit Madan | Lead Engineer | adit@alluxio.com



Agenda

- Alluxio Overview
- Kubernetes Basics
- Alluxio Deployment Options
- Spark + Alluxio on Kubernetes Demo



Alluxio Overview

The Alluxio Story



Originated as Tachyon project, at UC Berkeley AMPLab by then Ph.D. student & now Alluxio CTO, Haoyuan (H.Y.) Li.



Open Source project established & company to commercialize Alluxio founded

ANDREESSEN
HOROWITZ

Goal: **Orchestrate Data at Memory Speed for the Cloud** for data driven apps such as Big Data Analytics, ML and AI.



2018



2019

CRN

2019

Top 10 Big Data

CRN

2019

Top 10 Cloud
Software

Fast-growing Open Source Community



1000+ Contributors



4000+ Github Stars



Apache 2.0 Licensed



Join the community on Slack
alluxio.io/slack



Contribute to source code
github.com/alluxio/alluxio

Data Ecosystem - *Beta*



Data Ecosystem 1.0

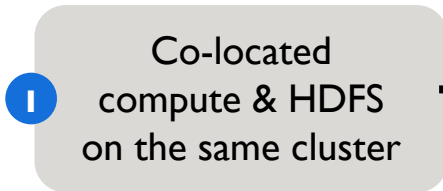


Data stack journey and innovation paths

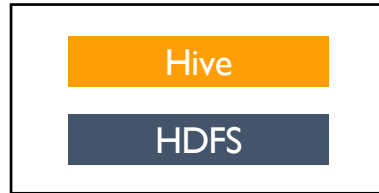
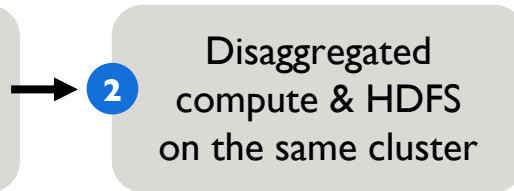
Co-located

Disaggregated

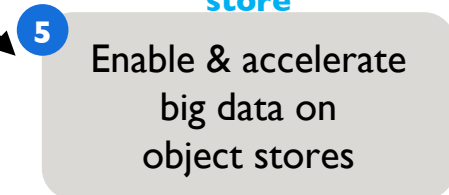
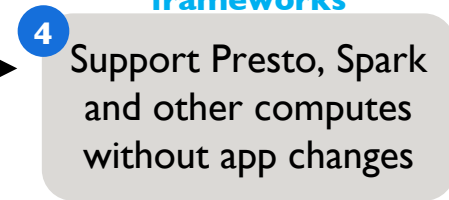
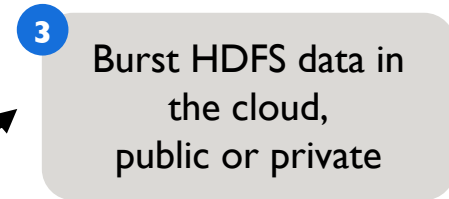
HDFS for Hybrid Cloud



- Typically compute-bound clusters over 100% capacity
- Compute & I/O need to be scaled together even when not needed



- Compute & I/O can be scaled independently but I/O still needed on HDFS which is expensive



Independent scaling of compute & storage



Java File API

HDFS Interface

S3 Interface

POSIX Interface

REST API



Data Orchestration

for the Cloud

HDFS Driver

Swift Driver

S3 Driver

NFS Driver



Alluxio – Key innovations

Data Locality with Intelligent Multi-tiering

Accelerate big data
workloads with transparent
tiered local data

Data Accessibility for popular APIs & API translation

Run Spark, Hive, Presto, ML
workloads on your data
located anywhere

Data Elasticity with a unified namespace

Abstract data silos & storage
systems to independently scale
data on-demand with compute

Flexible APIs to Interact with data in Alluxio

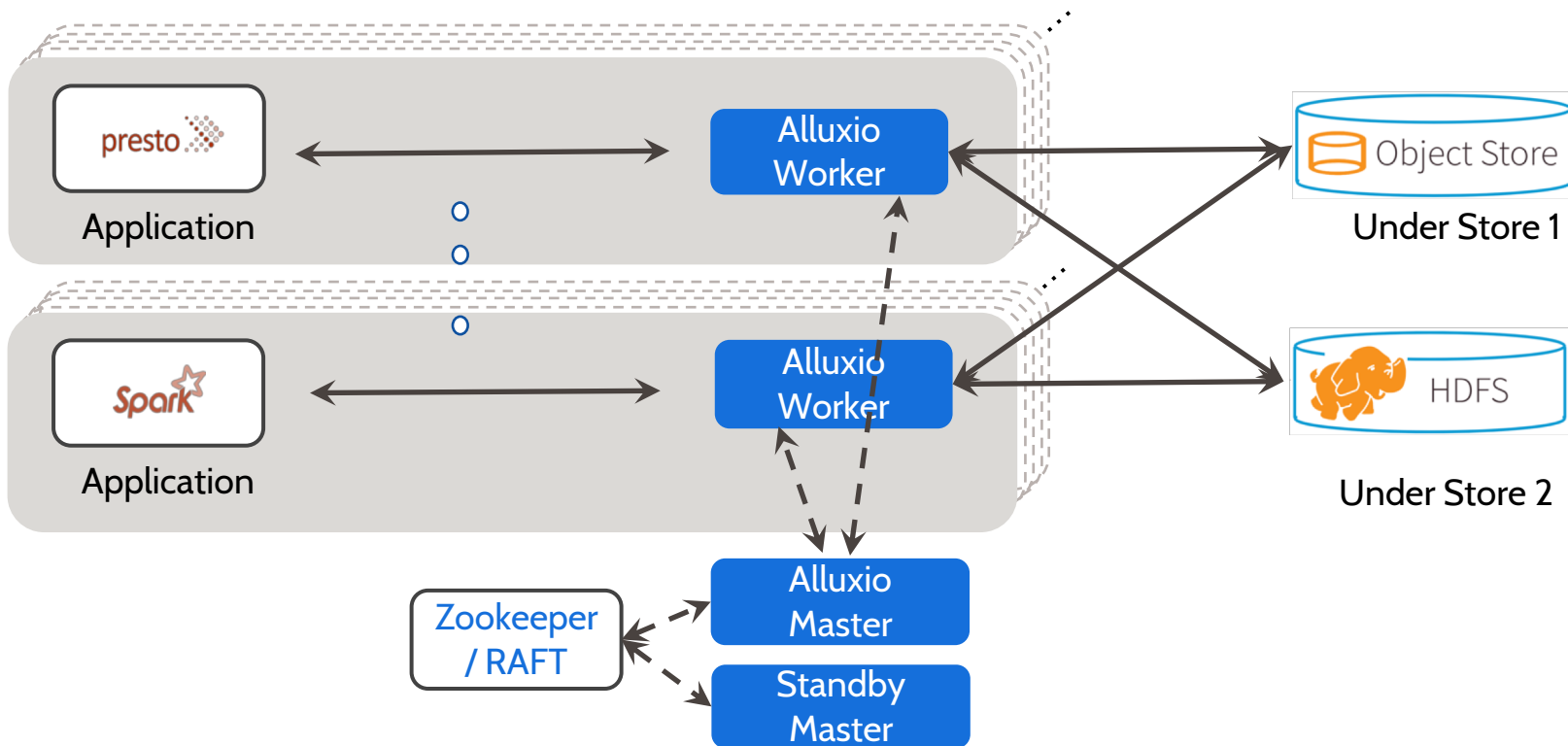
Spark > rdd = sc.textFile("alluxio://localhost:19998/myInput")

Presto CREATE SCHEMA hive.web
WITH (location = 'alluxio://master:port/my-table/')

POSIX \$ cat /mnt/alluxio/myInput

Java FileSystem fs = FileSystem.Factory.get();
FileInputStream in = fs.openFile(new AlluxioURI("/myInput"));

Alluxio Reference Architecture



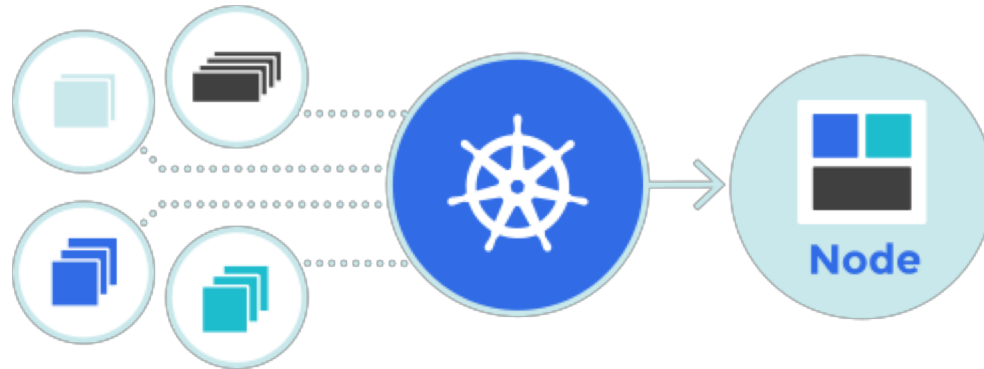


Kubernetes Overview

Kubernetes (k8s) is...



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



What we'll cover

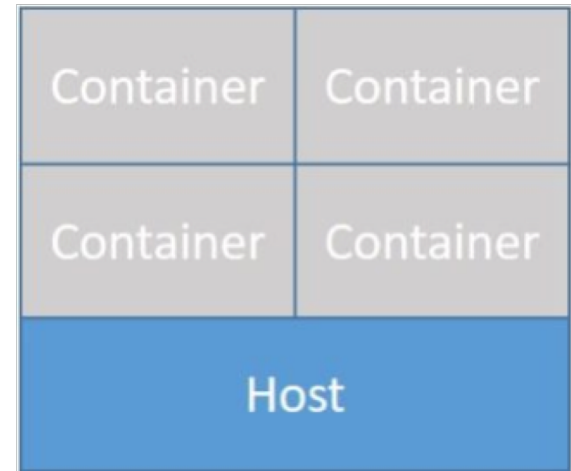


- Kubernetes Basics
- Alluxio Deployment Options
- Spark on Alluxio Demo

Container Orchestration Platform



- **Abstract Physical Infrastructure**
 - Platform Agnostic
 - On-premise, hybrid or in the public cloud
- **Service Discovery**
 - Networking abstraction
- **Self-healing**
 - Resilience to failures
- **Secret Management**
 - Management of sensitive credentials
- **Storage Management**
 - Lifecycle management tied to applications



Key Concepts



- Containers
 - Docker Image = Lightweight OS and application execution environment
 - Container = Image once running on Docker Engine
- Pods
 - Schedulable unit of one or more containers
 - Containers share resources and network
- Controllers
 - Controls the desired state such as copies of a Pod
- Persistent Volumes
 - Storage provisioned by admin with lifecycle independent of a Pod



Some more K8s Basics...

- **Declarative Specs**

- Configure, deploy and manage an application on K8s using a declarative language

- **Helm Charts**

- Thin wrapper over declarative specs
- Reduce complexity using a single configuration file

- **Operators**

- Another abstraction layer over declarative specs to
- Built-in domain knowledge
- Manage upgrades and improve troubleshooting



Alluxio on Kubernetes

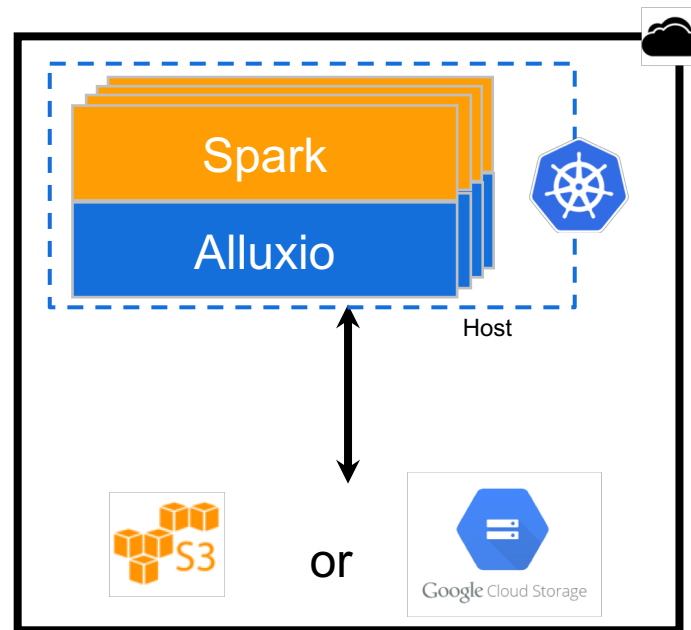
Data challenges

- **Data Copy**
 - Multiple Copies of Data
- **A Stateful Application on K8s can be hard**
 - Data Migration and Rebalancing on elasticity
- **Changing applications for a new storage system can be hard**
 - Lack of familiar API
 - Tuning can be challenging

Why Alluxio in K8s?

Elastic Data for Elastic Compute with Tight Locality

- **Data locality**
 - Big data analytics or ML
 - Cache data close to compute
- **Enable high-speed data sharing across jobs**
 - A closer staging storage layer for compute jobs
- **Unification of persistent storage**
 - Same data abstraction across different storage



Use Case: Zero Copy Bursting

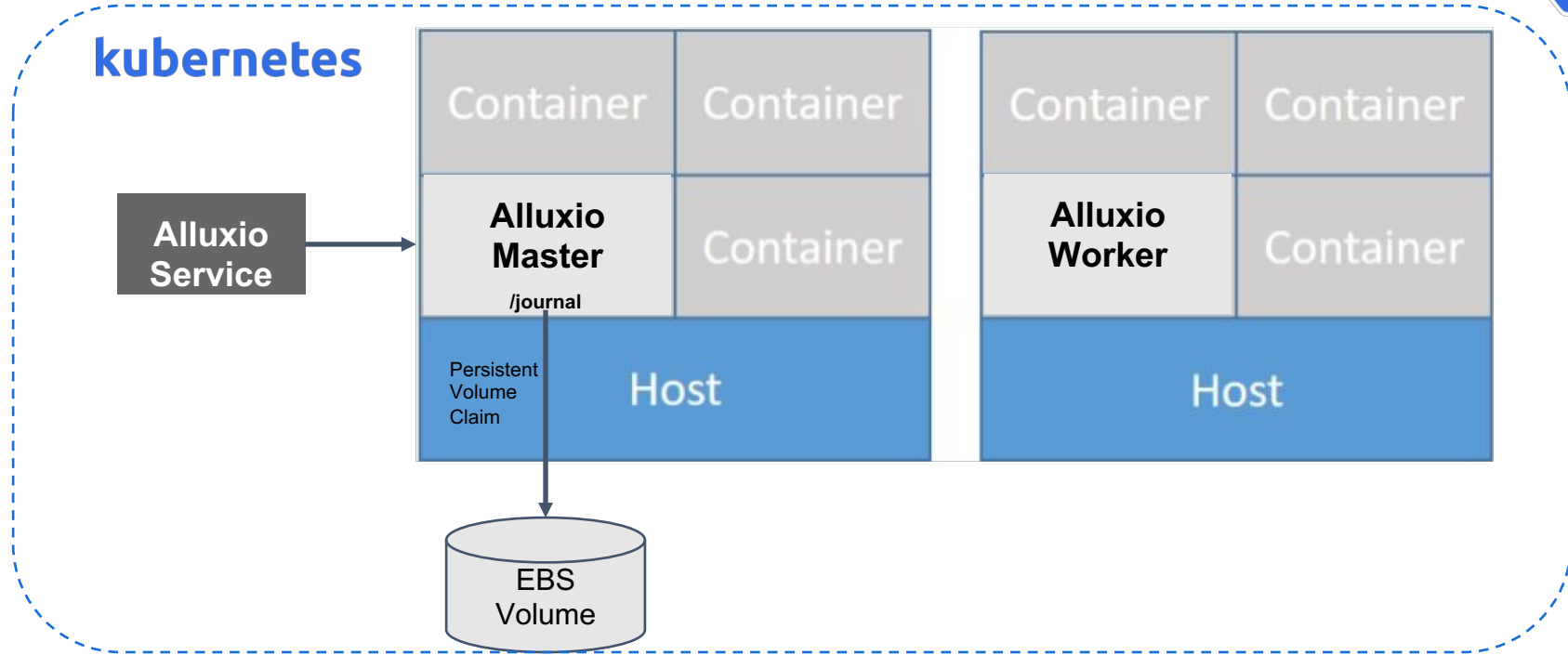
Scenario:

- **Data Resides on-prem**
 - Traditionally HDFS
- **Compute bursts into the Cloud**
 - Resources added on-demand
 - Economical

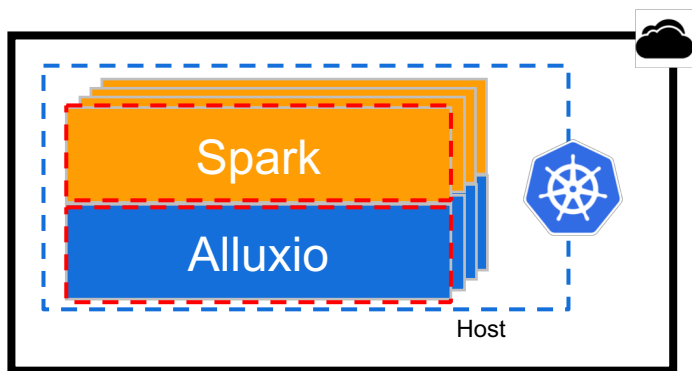
Solution w/ Alluxio:

- **Data Accessible Immediately**
 - No ETL pipeline needs to be setup
- **Data fetched on access**
 - No persistent copies in the Cloud

Alluxio on K8s Architecture



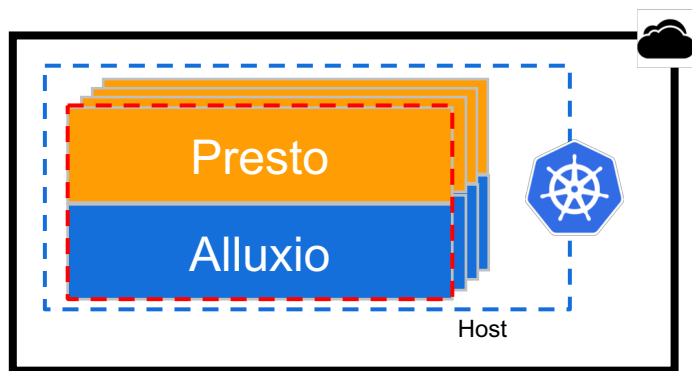
Deploying Alluxio in K8s



Alluxio and Compute framework in different pods on the same host

When do you use this?

- Compute, like Spark, is short running and ephemeral
- Alluxio data orchestration & access layer is long running and used across many jobs



Alluxio and Compute framework in the same pod

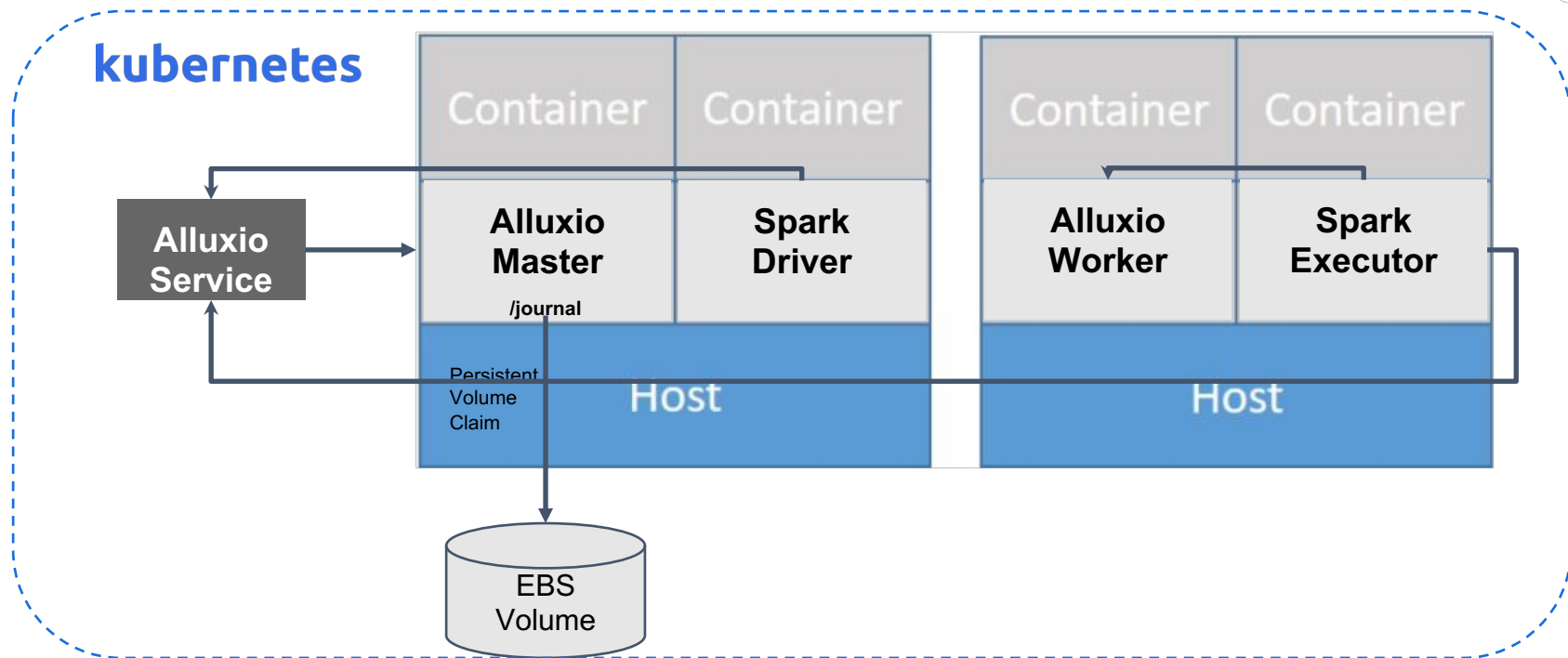
When do you use this?

- Compute, like Presto, is long running
- Data tier with Alluxio needs to be scaled along with compute tier

Legend:



Spark on Alluxio on K8s



Deploying Alluxio

- **Provision a Persistent Volume for Journal**

```
$ kubectl create -f alluxio-journal-volume.yaml
```

- **Specify Configuration**

```
$ kubectl create -f alluxio-configMap.yaml
```

- **Deploy Master**

```
$ kubectl create -f alluxio-master.yaml
```

- **Deploy Workers**

```
$ kubectl create -f alluxio-worker.yaml
```

Alluxio Helm Chart

Now simplified...

```
$ cat << EOF > config.yaml
properties:
  alluxio.mount.table.root.ufs: "<under_storage_address>"
  aws.accessKeyId: "<accessKey>"
  aws.secretKey: "<secretKey>"
EOF
```

```
$ helm install -f config.yaml alluxio-repo/alluxio --version 2.1.0-SNAPSHOT
```

Ongoing Effort

Upcoming Alluxio on K8s Features

- **Scaling Alluxio for Large Production Deployments**
 - High-Availability in the absence of Zookeeper
 - Off-Heap Metadata Layer
- **Helm Chart**
 - Greater Flexibility
 - Parity w/ Alluxio Features in a non-containerized environment
- **CSI**
 - POSIX API Access
 - Ease of Access to Alluxio



Demo: Running Spark & Alluxio on K8s

Summary

- An Overview of Data Orchestration
- Alluxio enables elastic data for elastic compute in Kubernetes
 - Data Locality on Demand, Data Abstraction & Unification, Data Sharing
- A guide to run Alluxio in Kubernetes environment
- A demo of running Spark on Alluxio in Kubernetes

Questions?



Join the community on Slack
alluxio.io/slack

DATA ORCHESTRATION SUMMIT

November 7, 2019 | Computer History Museum | Mountain View, CA

Organized by  ALLUXIO

[Register Here!](#)





Appendix



Kubernetes w/ Alluxio Components

- Containers
 - Master, Job Master, Worker, Job Worker
- Pods
 - Master, Worker
- Service
 - Master
- Daemon Sets
 - Workers
- Persistent Volumes
 - Master Journal, Worker Tiered Storage (optional)
- HostPath Volume
 - Worker Domain Socket Directory