CLOUDERA

KubeCon | CloudNativeCon

North America 2023

Migrating Hybrid Big Data Platforms at Scale to Kubernetes

Sunil Govindan

8th Nov 2023

INTRODUCTION

Sunil Govindan

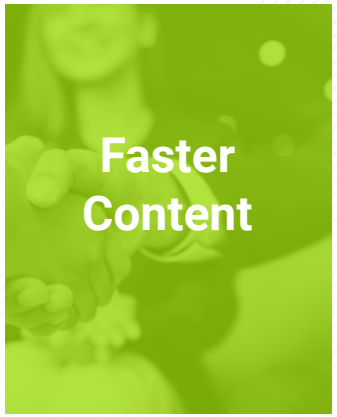
Senior Engineering Manager, Cloudera

- Apache YuniKorn (Batch Scheduler for K8s) PMC & Committer
- A decade with Big Data
- Deep commitment to Open Source

- @sunilgovind
- @sunilgovindan

AN “ENTERPRISE” COMPANY’S JOURNEY TO ADOPT AI

Get value out of below categories within their organization



R&D

SALES & MARKETING

CUSTOMER

LEGAL

HR

YOUR “**DATA**” DIFFERENTIATES YOUR AI

DATA - WHY IS IT RELEVANT?

1

Data Explosion

Most (75%) Data is semi-structured and unstructured

Cloudera is managing 25 EB of data

200 Trillion Objects in S3 as of 2023*

2

Access for All

10 - 50X increase in data consumers

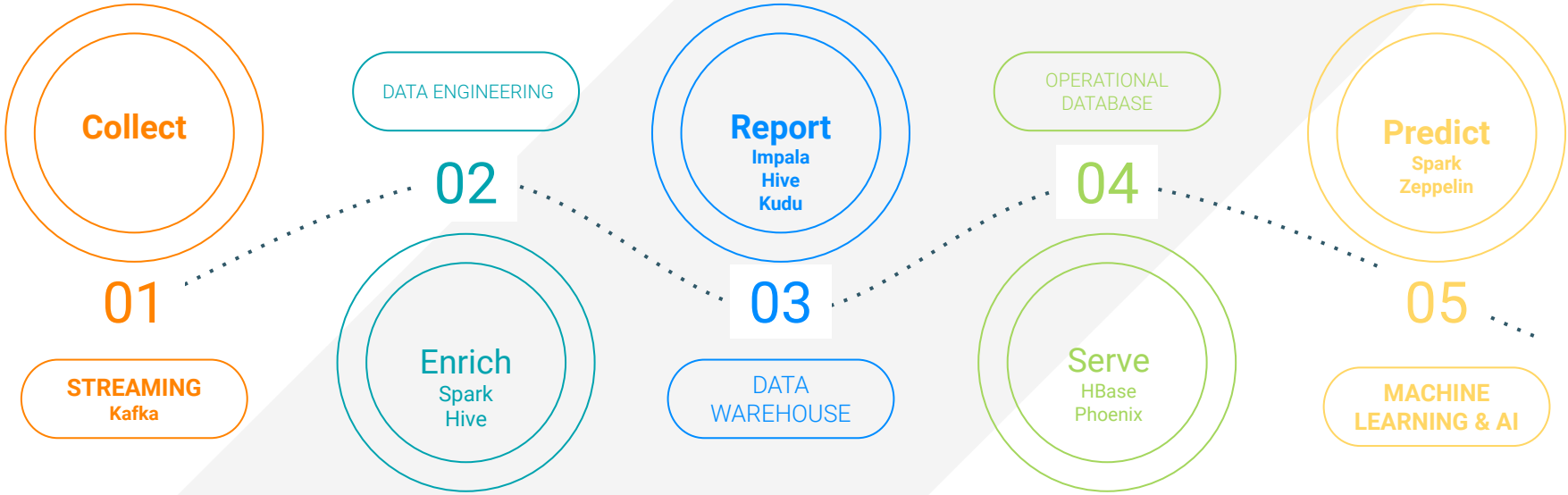
Every employee and every application (AI or non-AI) needs access to data

3

Faster Value prop

Storing and managing data is not enough for getting the value of data, we need powerful engines to process the data

DATA LIFE CYCLE



CLOUDERA SDX

SECURITY | GOVERNANCE | LINEAGE | MANAGEMENT | AUTOMATION

DATA USE CASE

Data Engineer & Data Scientist

Clean and curate data pipelines sourced from operational systems & feed to ML models

DATA FROM MULTIPLE SOURCES	APPLY UPDATES AFTER DATA PROCESSING	TAKE BACKUPS - AUDITING	RUN HISTORIC QUERIES	PREDICTIVE SERVICES FROM ML ENGINE
Petabyte scale & partition evolution	ACID transactions, updates, deletes & compaction	Snapshots & Expiration	Time-travel	Powerful Models & Data Context

DATA USE CASE

Data Engineer & Data Scientist

Clean and curate data pipelines sourced from operational systems & feed to ML models

DATA FROM MULTIPLE SOURCES → **APPLY UPDATES AFTER DATA PROCESSING** → **TAKE BACKUPS - AUDITING** → **RUN HISTORIC QUERIES** → **PREDICTIVE SERVICES FROM ML ENGINE**

30+ OPEN SOURCE PROJECTS

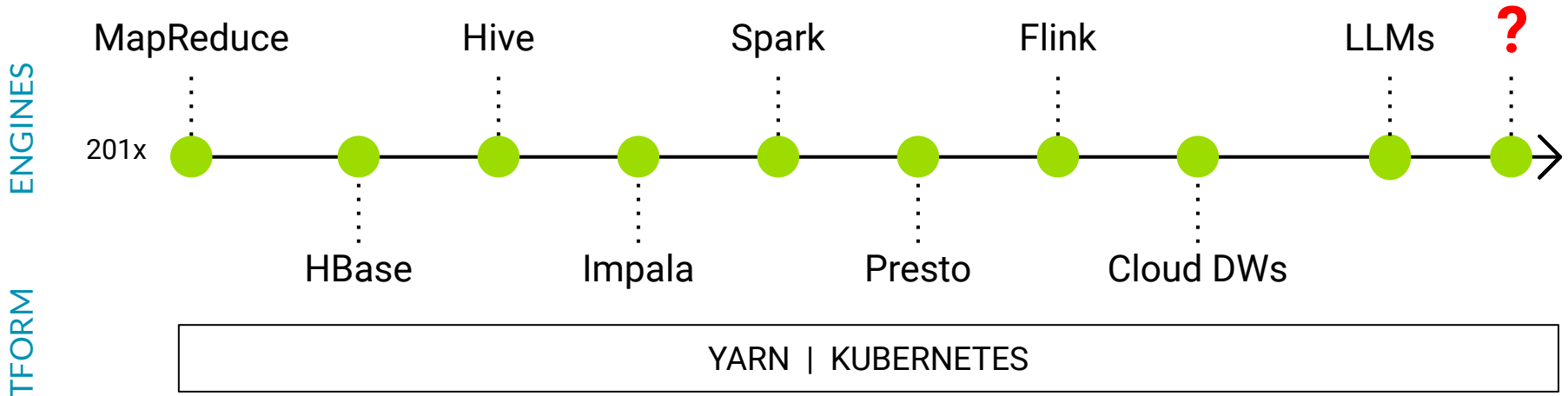
A Story from the last decade

Compute in Big Data world is complex

- Long running & stateful services - HBase, Hive-LLAP, Impala, Flink, Kafka, Solr ..
- Batch workloads - MR, Spark, Tensorflow …

And solves a ton of use cases that are relevant in the genAI revelation era

30+ OPEN SOURCE PROJECTS - JOURNEY



KEY TAKEAWAYS

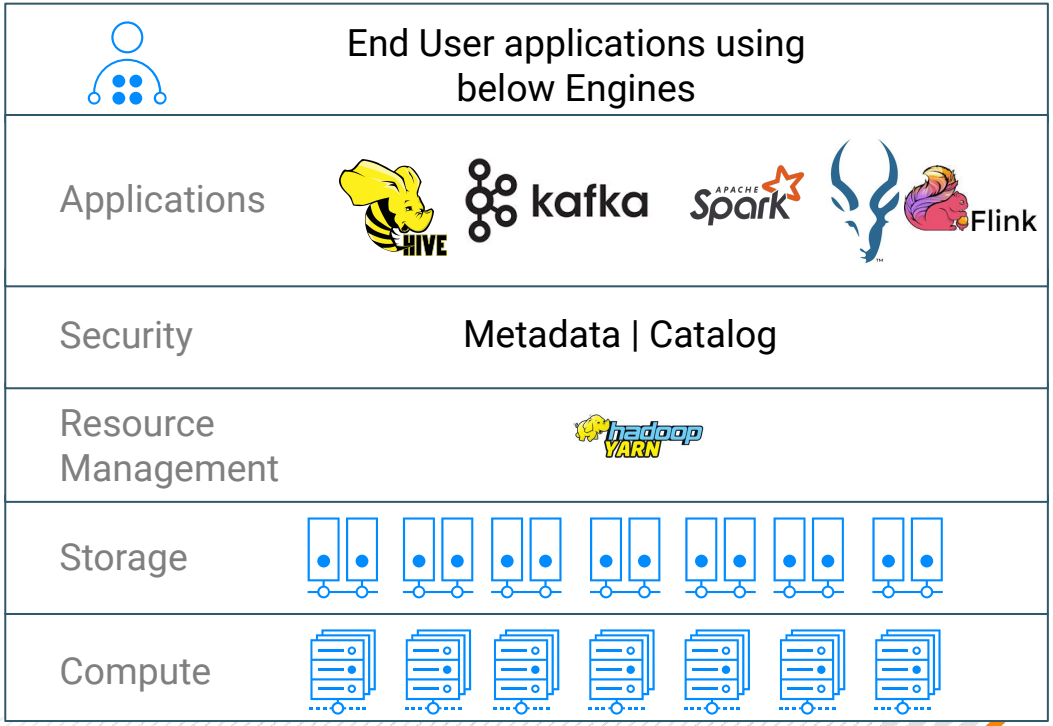
- Data is constant
- Emerging new Engines
- Flexible Platform

ACTIONS

- Embrace the change (Engine & Platform)
- Keep the data & engine w/o vendor lock in – Open Source

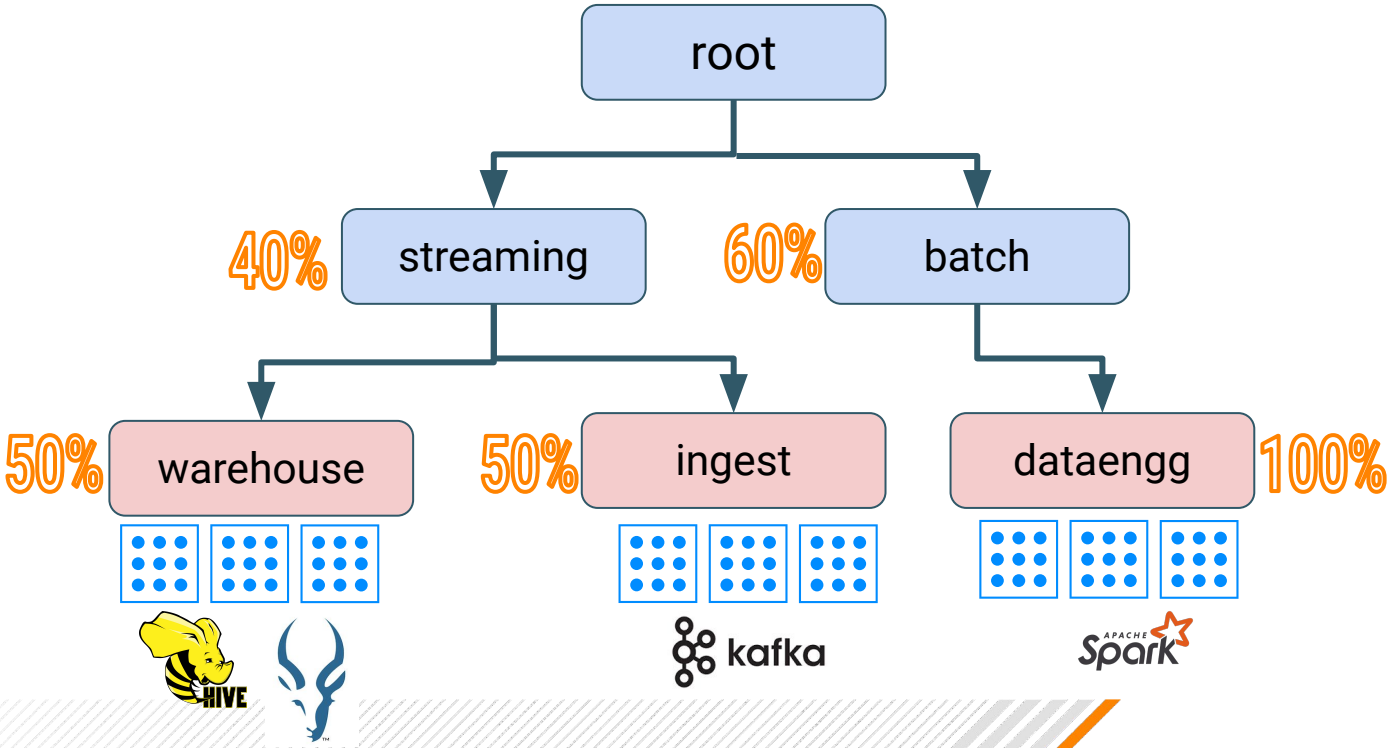
DATA USE CASE - STACK OVERVIEW

Platform Administrator View



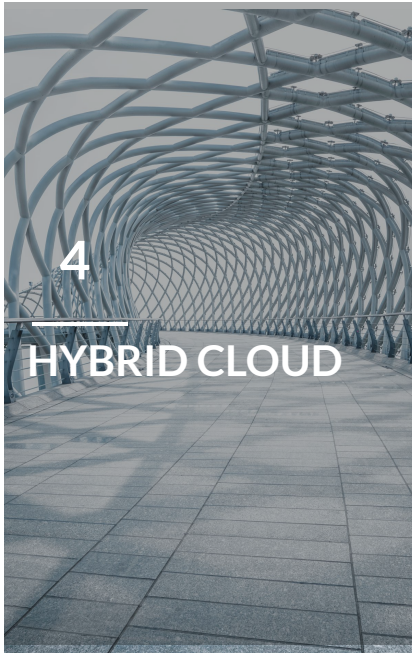
DATA USE CASE - RESOURCE OVERVIEW

Platform Administrator View



CHALLENGES FOR MANAGED DATA PLATFORM

Journey to Kubernetes & beyond



DATA SERVICES ON **K8s** IS the FIRST STEP

Compute-layer innovation in Data Services

DATA CLUSTERS

SPARK, HIVE, IMPALA, CDSW

- Servers
- Monolithic
- Co-Located Storage & Compute
- Optimized for Existing Applications
- Forklift Upgrades

DATA SERVICES

DATA ENGINEERING

DATA WAREHOUSE

MACHINE LEARNING

- Services
- Modular
- Cloud Arch - Separated Storage & Compute
- Optimized for New Applications
- Independent Upgrades

ARE WE MISSING SOMETHING WITH “DATA SERVICES”?

Did we add any new challenges?

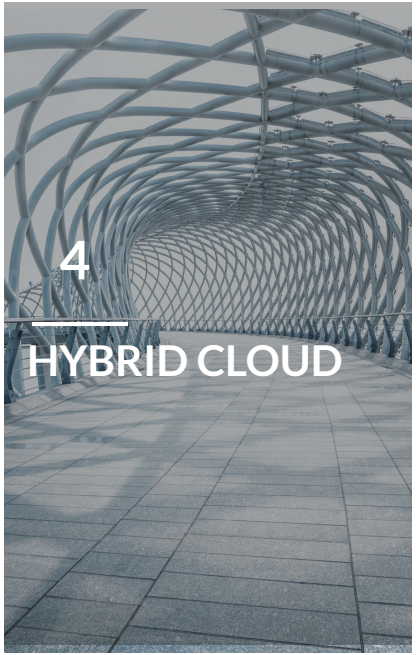
- Do we have the same flexibility of managing resources with **hierarchical queues** across different organizations for different use cases?
- Are these new services purely **multi-tenant**?
- Can I achieve the same or **better SLA** for my applications?
- Can this new service **scale** for batch type of jobs (Spark) where I need 10k+ containers for huge data processing?
- Is this **cost effective** compared to previous cluster form factor?

Are the Data engines ready to adapt Kubernetes & Cloud Native architecture?

How did Cloudera Solve Problems on Cloud Native Stack?

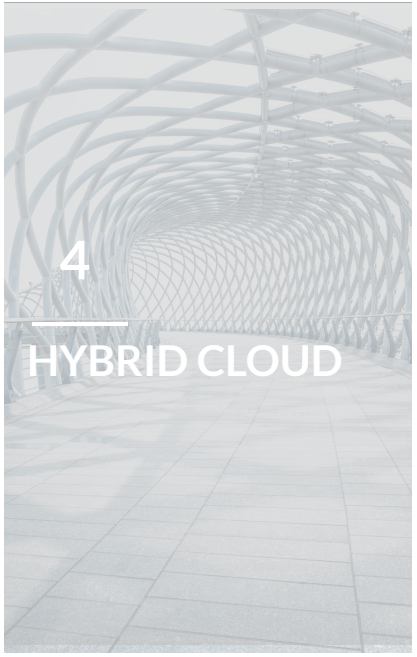
MIGRATING DATA PLATFORMS TO THIS NEW ERA

At every stage, there are challenges, and they can be categorized under



MIGRATING DATA PLATFORMS TO THIS NEW ERA

At every stage, there are challenges, and they can be categorized under



WHAT DID WE GET FROM KUBERNETES?

- Support faster **Autoscaling** (cost effective)
- Running **Containerized** workloads
- Mix different **application versions** with **dependency management**
- Solve **resource isolation**

CHALLENGES WITH K8s

SCHEDULING BATCH WORKLOADS

HIERARCHICAL QUEUES
How to assign quotas for teams and users based on use case?

FIRST CLASS APPLICATION
How to submit Spark job with driver and executors as a single unit?

APP AWARE PREEMPTION
How do I maximize resource utilization in fixed size clusters?

LIMITATIONS WITH K8s SCHEDULER

Quota

- Kubernetes Quotas are an add-on, not part of the scheduler
- Enforcement as part of resource creation
- Quotas & Limits are Namespace based
- Lack of Application aware preemption

CLOUDERA SOLUTIONS

SCHEDULING BATCH WORKLOADS

Schedules any Batch or Service workload

- K8s (job, daemon set, deployment etc)
- Apache Spark, Hive, Impala, Tensor Flow, Ray etc

Simple integration:

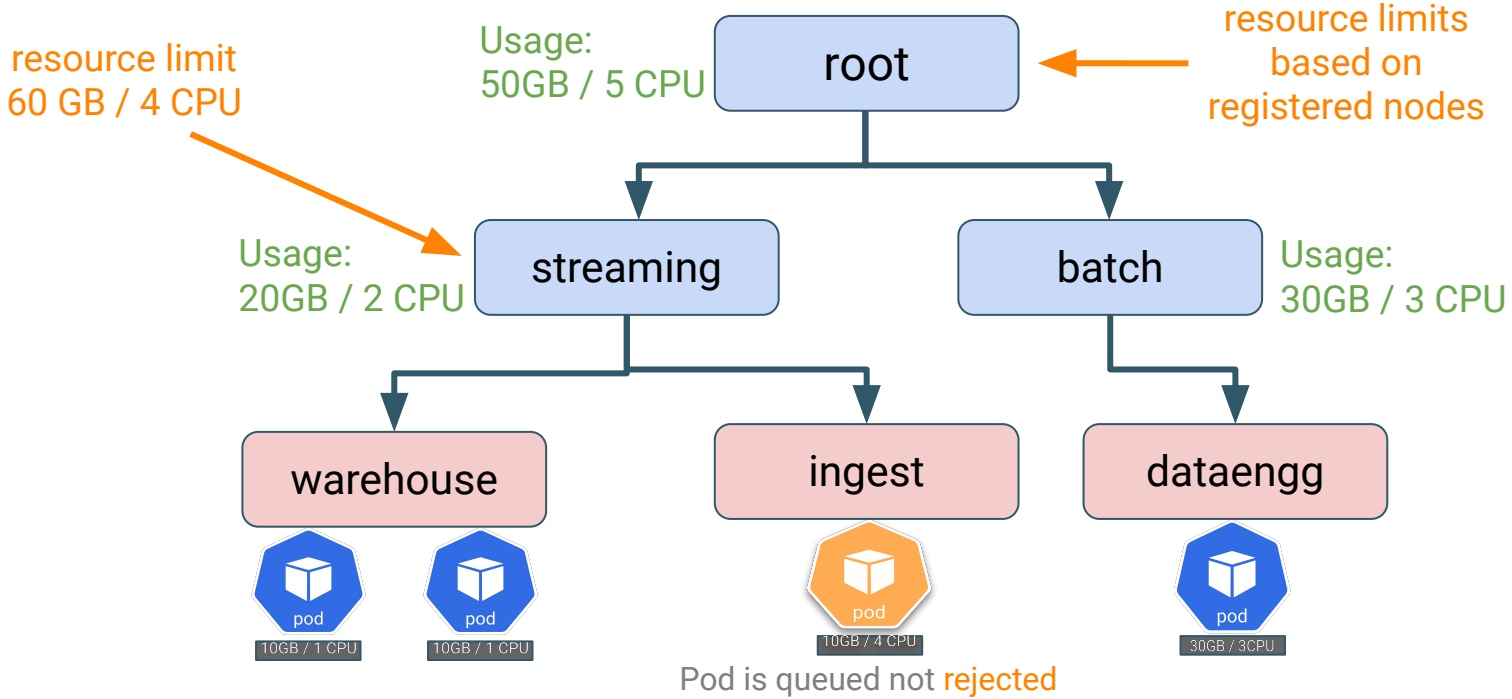
- No code changes needed from the Engines
- Annotations and labels only

Supports Hierarchical queues with

- Guaranteed resources & Quotas
- Various Scheduling policies (FIFO, Fair, Gang Scheduling etc)

CLOUDERA SOLUTIONS

SCHEDULING BATCH WORKLOADS WITH **YUNIKORN**



CHALLENGES WITH K8s

PREEMPTION

K8s and preemption

- In K8s, all pods for the entire cluster are sorted based on priority for scheduling
- Pods considered for **eviction** are ranked on priority
- **Opt-out** capability is NOT possible, hence possibility of killing *originator (long running Spark driver, etc.)* pods are high, resulting in large costs
- PriorityClass (defines the priority in a K8s cluster)
  - PriorityClass objects are cluster-wide objects
  - There are no limitations on the priority that can be set on a pod
  - Any rogue user can create a pod with the highest defined priority

CLOUDERA SOLUTIONS

PREEMPTION WITH **YUNIKORN**

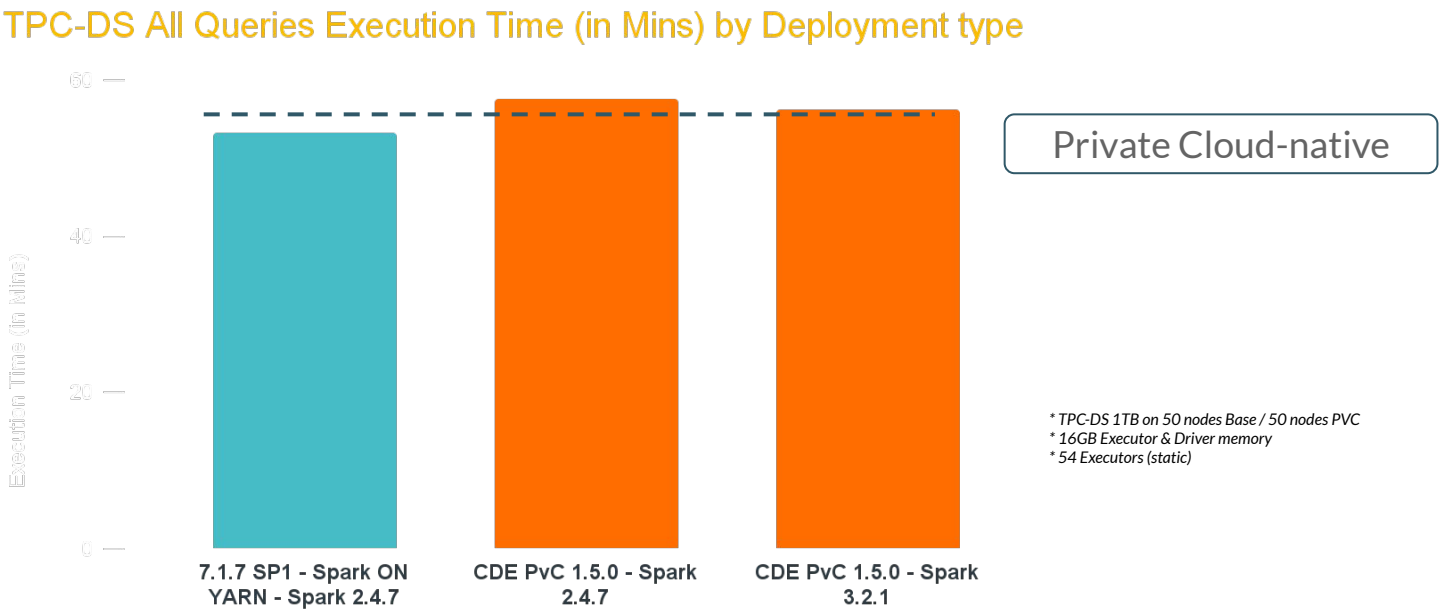
- YuniKorn uses a hierarchical queue model
  - Multi-level resource limits (both *guaranteed* and *max*)
- Preemption adjusts queue usage towards ***guaranteed*** resource capacity
  - Preempt tasks in queues **above** guaranteed limits in favor of tasks in queues **below** guaranteed limits
- Application-aware
  - Avoid preempting *originating* pod (i.e. Spark driver)
    Avoid pods which *opt out* of preemption

DEMO

Elastic Queue Capacity

PERFORMANCE

Comparison between Spark on YARN vs Spark in K8s on a limited data set



SUMMARY

Kubernetes + YuniKorn modernized our Data Platform

MIGRATING DATA PLATFORMS TO THIS NEW ERA

At every stage, there are challenges, and they can be categorized under

1

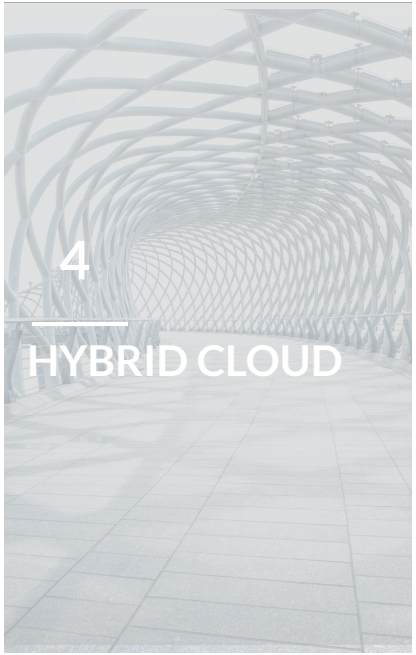
RESOURCE MANAGEMENT

2

SCALABLE CLUSTERS

3

OPERATIONAL EFFICIENCY



4

HYBRID CLOUD

CHALLENGES WITH THE BIG DATA PLATFORM

COMPUTE & STORAGE CLUSTERS

CO LOCATED

How to independently scale compute without storage?

Data Locality — For performance, data co-located with compute was helpful. Does it matter, given better n/w is possible with today's h/w spec?

Cost — In order to scale compute alone, forced to buy storage, and increased cost to overall cluster.

SPECIALITY HARDWARE

How to cost effectively handle GPU resources?

Scarce resources — Heavy cluster maintenance cost to place them along with storage nodes & isolate them for mission critical use cases

Operation cost — Developers need to involve Platform IT team whenever they need special h/w.

AUTO SCALING

How do I scale my cluster based on compute demand?

Cost Effective — Based on the exact compute demand, how to get the desirable compute under budget?

WHAT DID WE GET FROM KUBERNETES?

CLUSTER MANAGEMENT

- **Container Storage Interface** (CSI) helped to manage different storages through single abstraction
- K8s helped to extract the best of **speciality hardware** (GPU) for dedicated use cases
- Less administrative overhead for spinning up or shutting down K8s cluster
  - Also, Containerized deployments help a lot in **reducing spin up time** significantly with K8S
- Determining the right cluster size can be challenging for workloads, hence an **autoscaling** cluster based on workload demand will be effective.

HOW DID CLOUDERA ADDRESS THE REMAINING ISSUES?

DISAGGREGATED COMPUTE & STORAGE CLUSTERS

ON PREM

Disaggregate storage cluster with **Apache Ozone**

- Distributed Key Value Object Store with native S3 and FS interfaces.
- Ozone can scale up to billions of objects (30x of HDFS)
- Reduce cost per TB using commodity hardware

CLOUD

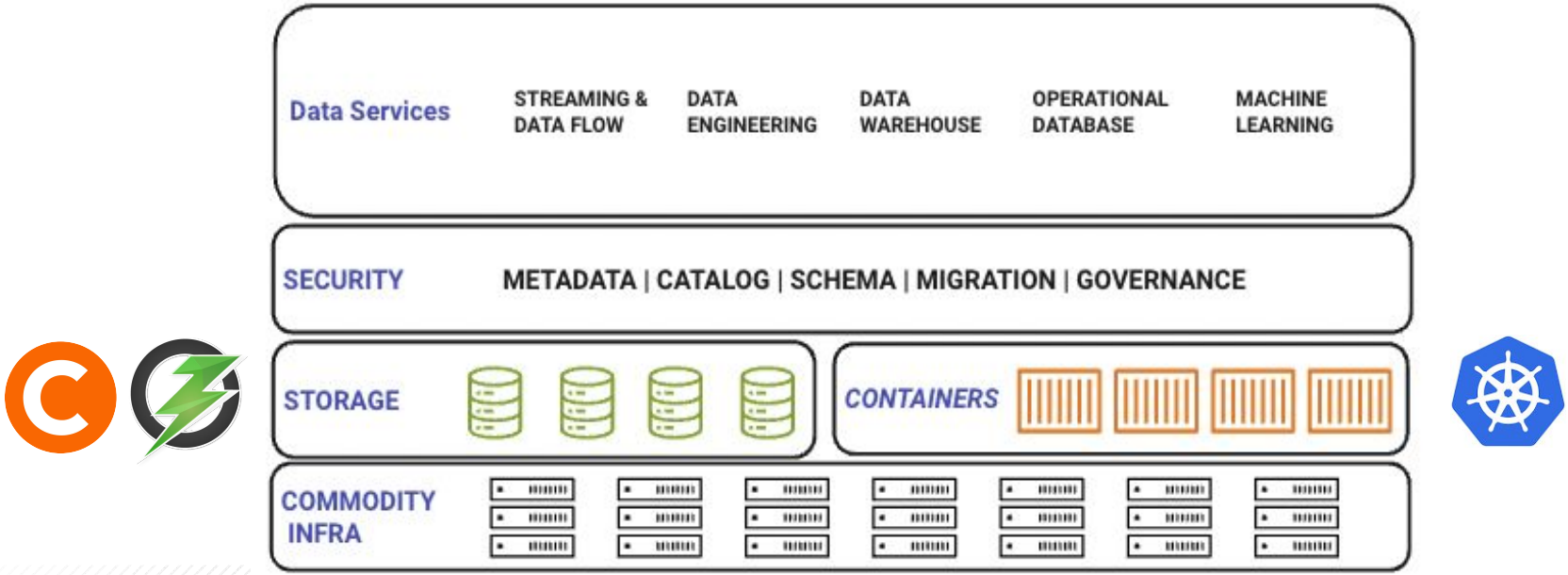
Use **Cloud Connectors** with Cloud Storages

- S3A connector for enabling parallel and asynchronous reading of different data blocks

SUMMARY

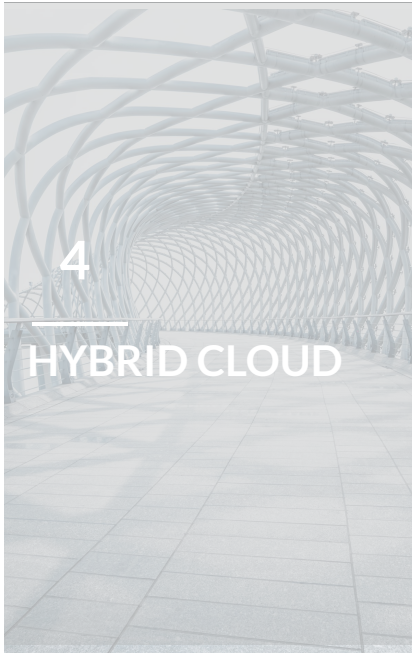
POWERED BY OZONE & CLOUD CONNECTORS WITH K8s

Upgraded Multi-Public/Private/Hybrid Cloud Arch



MIGRATING DATA PLATFORMS TO THIS NEW ERA

At every stage, there are challenges, and they can be categorized under



CHALLENGES WITH BIG DATA PLATFORM

COST OF MAINTENANCE

HIGH UPGRADE COST

How to upgrade services & engines without down time?

ALWAYS ON "SERVICES"

How to keep the services high available?

COST OF ADDING NEW APPS

Is it easy to add a new app to this ecosystem?

CHALLENGES WITH K8s

- Faster K8s Version releases
  - Upgrades are not cheaper, because workloads are running always
- Evolution of APIs
  - Breaking API changes impact our upgrade cycles

CLOUDERA SOLUTIONS

COST OF MAINTENANCE

- Clean separation of K8s API usages from internal services & apps
  - Avoid mixing APIs, Client binaries, Helm from multiple parts of you code
  - Always keep clean API abstractions for platform access

- Use CSI storage as ephemeral / scratch space wherever possible.
  - For persistent storage, use clean data abstractions

- Monolith vs Microservices
  - Dependency between services are critical, hence design that carefully
  - Design a quiesce mode option for your service to enter into silo mode

LOT MORE WORK NEEDS TO BE DONE

MIGRATING DATA PLATFORMS TO THIS NEW ERA

At every stage, there are challenges, and they can be categorized under

1

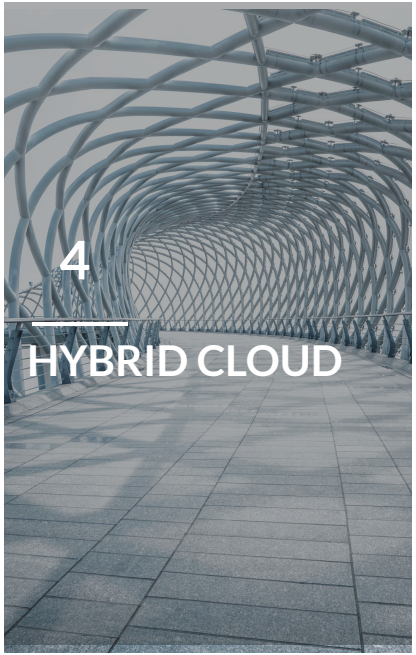
RESOURCE MANAGEMENT

2

SCALABLE CLUSTERS

3

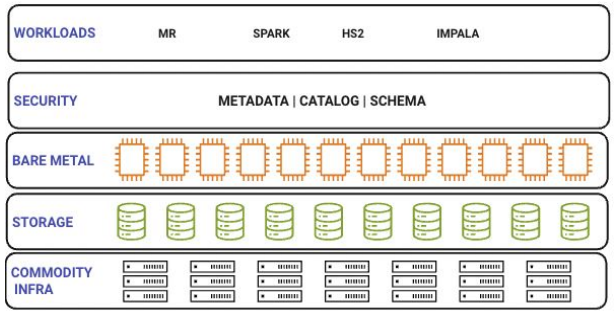
OPERATIONAL EFFICIENCY



EVOLUTION OF THE ARCHITECTURE

Journey

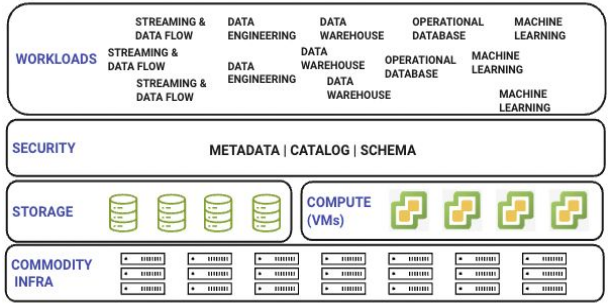
On Premise



Co-located Storage/Compute
Large, Shared Clusters
(CDH / HDP)

Generation 1

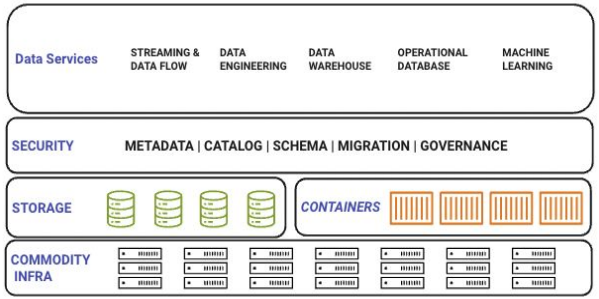
Public Cloud



Disaggregated Storage/Compute
Multiple Clusters

Generation 2

Multi-Public/Private/Hybrid Cloud



Disaggregated Storage/Compute
Multi-Tenant, Containerized Data Services

Generation 3

Evolution

SUMMARY

K8s & Iceberg helped us to achieve

Cloud-Native

To get Cloud agility and flexibility on premises

Hybrid Portability

To run your workloads on the right infrastructure, write once run anywhere

Private Lakehouse

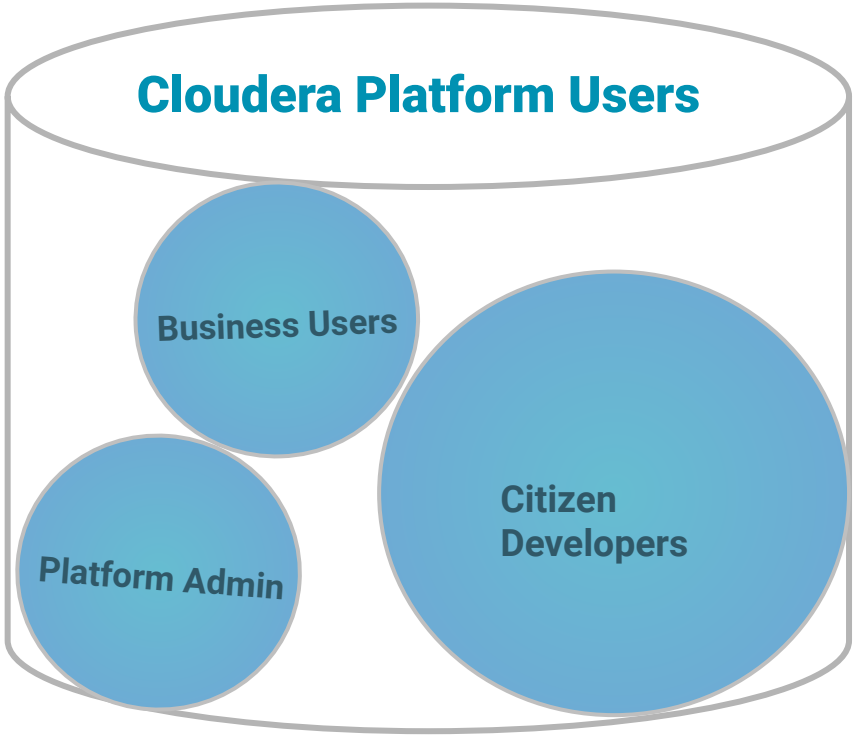
Lakehouse oriented full data lifecycle, powered by Iceberg

Optimized Processing

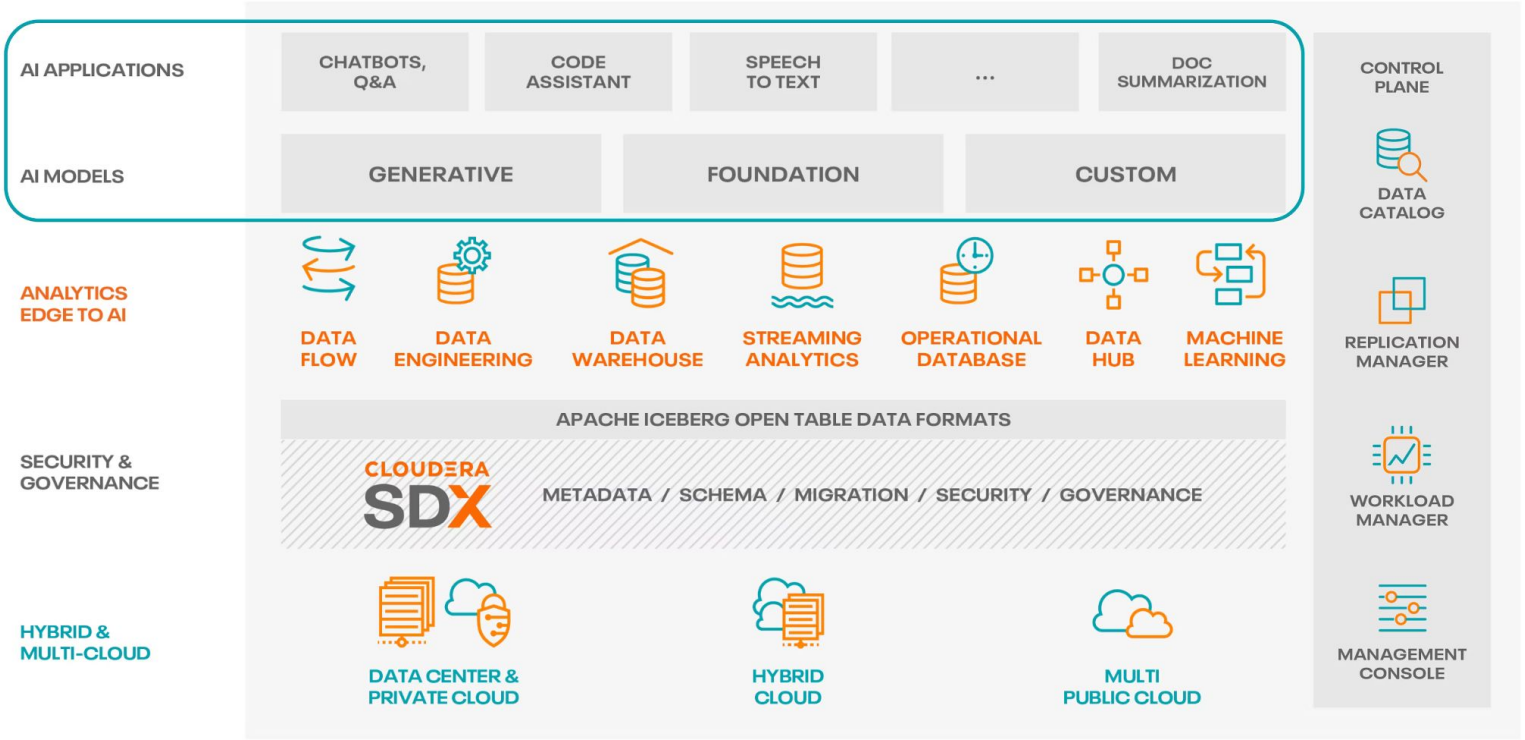
Self-service workload provisioning accelerates onboarding of new teams and applications within hours to enable LoBs

How do we see the new architecture now?

CLOUDERA PLATFORM SERVES “EVERYONE”



CLOUDERA DATA PLATFORM



THANK YOU

CLOUDERA

KubeCon CloudNativeCon North America 2023

Please scan the QR Code above to leave feedback on this session