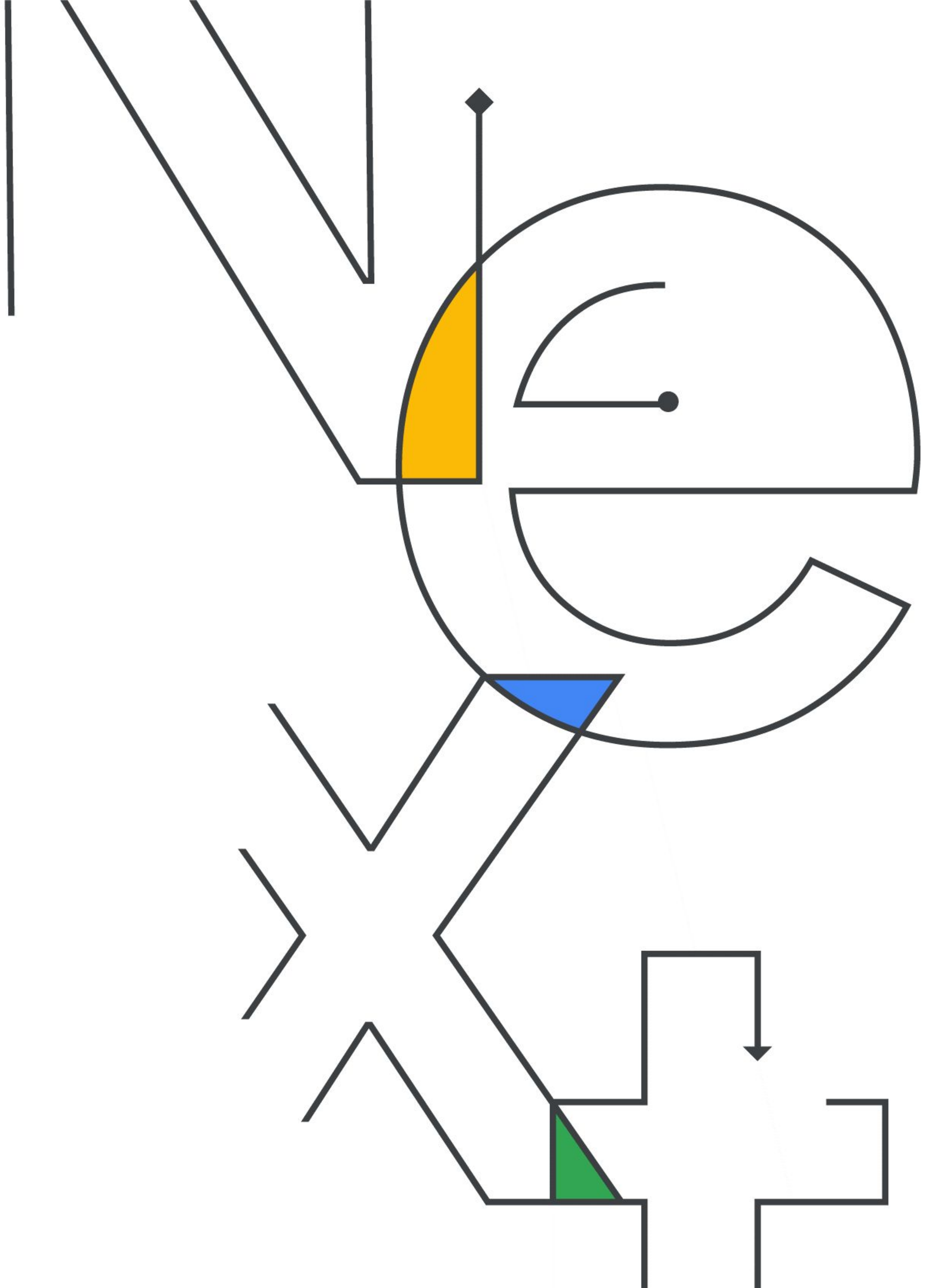
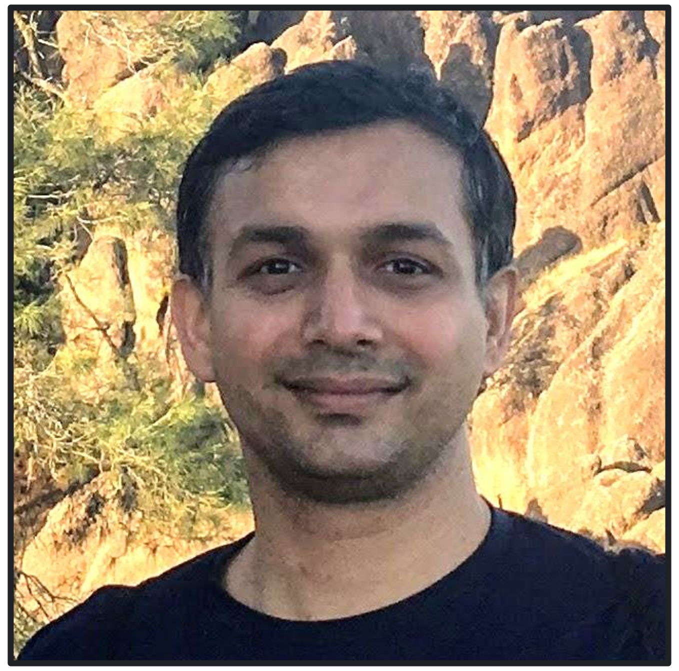
Google Cloud

Next '22

How to scale data analytics securely with Spark on Google Cloud





Abhishek Kashyap

Group Product Manager
Google Cloud

Mithun Bondugula

Sr Engineering Manager
LiveRamp

Agenda

01 Open source data analytics on Google Cloud

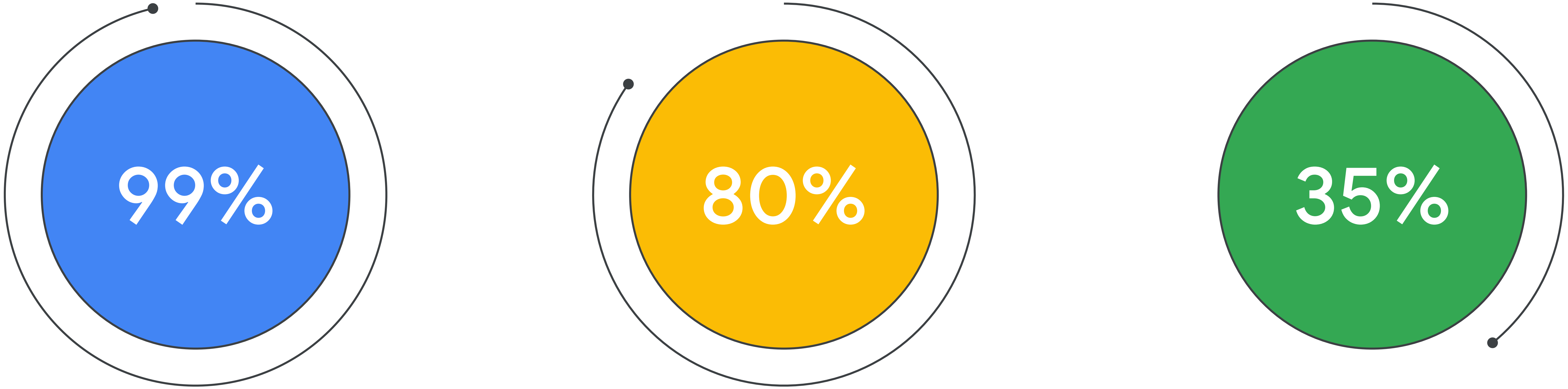
02 Serverless Spark

03 Security & Governance

04 LiveRamp

Organizations are doubling down on their open source investments as part of the overall data architecture.

The rise of open source



99% of Fortune 500 companies currently use open source software

80% of IT departments will increase use of open source in 2021

35% of all enterprise software is based on open source code

Source: BCG, April 2021

At Google, we're committed to helping customers create an open and integrated data platform

METRO

80%

Increase in ecommerce platform stability

30-50%

Reduction in infrastructure cost

Learn more

vodafone

17PB

Of data migrated over to Google

600

Servers of Hadoop eliminated

Learn More

/LiveRamp

30PB

Of storage in Hadoop deployments migrated over to Google

100,000+ YARN applications

To deliver billions in records per day

Learn More

Rapidly develop and run Spark at scale across all use cases

Spark on GCP

Scale without managing infrastructure

- Auto-scale, without manual infrastructure provisioning or tuning
- Comes with latest OSS frameworks

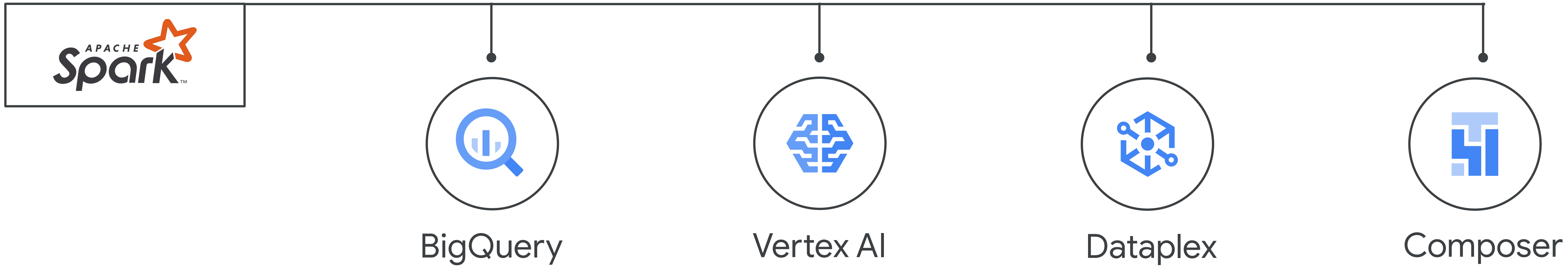
Work with tools you already know

- Connect, analyze, and execute Spark jobs from BigQuery, Vertex AI or Dataplex in 2 clicks
- No custom integrations, using Google-native and Open Source tools

Choose the right deployment model

Choose between:

- Serverless
- Google Kubernetes Engine (GKE)
- Compute clusters for your Spark applications



Serverless Spark for ETL

“

OpenX used serverless Spark to **abstract** away all the **cluster resources** and just focus on the job itself. This significantly helped to **boost** the team’s productivity, **while reducing infrastructure costs.**

Marek Wolczanski, Data Platform Engineer, OpenX

Customers

- PayPal
- ASML
- IAC
- LEVI STRAUSS & CO.
- GENERALI
- orange
- HKT

Partner Ecosystem

- accenture
- Cognizant
- HCL
- TEKsystems Own change

Quote source: Simplify data processing and data science jobs with Serverless Spark, now available on Google Cloud

What’s new

Serverless Spark

- Native Spark support in BigQuery
- Custom executor shapes (CPU:RAM)
- Customizable autoscaling speed
- Docker container streaming

Security & Governance

- Fine grained governance through Big Lake
- Automated Dataproc policy management
- Dataproc Metastore Hive and BigQuery federation

Announcing

Preview now available

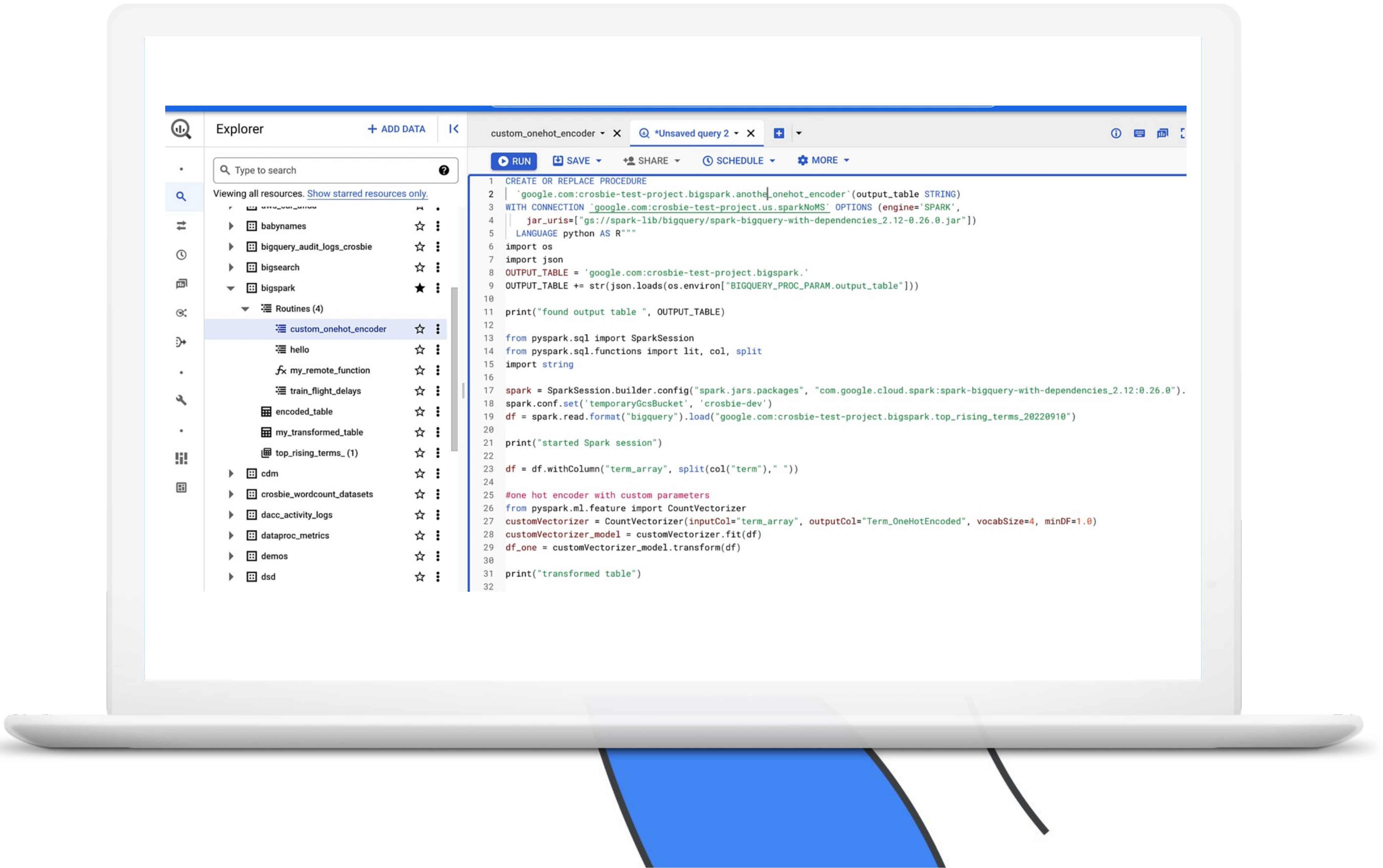
BigQuery stored procedures for Apache Spark

BigQuery moves beyond SQL with new developer extensions

Spark as a first class citizen in BigQuery
Execute Spark optimally with BigQuery SQL, as a stored procedure

Integrated BigQuery billing
Use BigQuery reservations to execute Spark

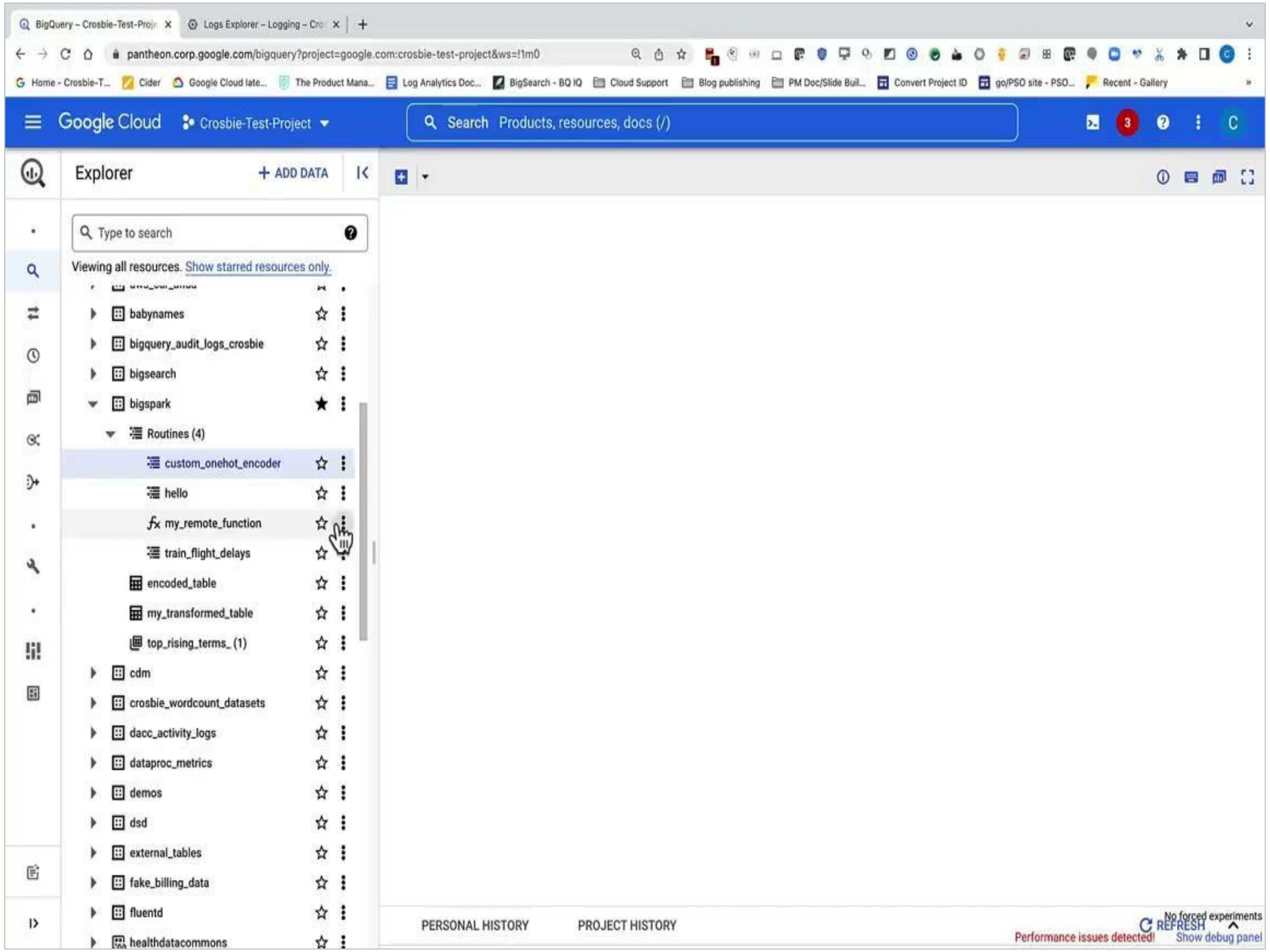
Integrated security and governance
Manage access through BigQuery, no data analyst access to underlying Spark infrastructure



```
CREATE OR REPLACE PROCEDURE
  `google.com:crosbie-test-project.bigspark.anothe_onehot_encoder`(output_table STRING)
WITH CONNECTION `google.com:crosbie-test-project.us.sparkNoMS` OPTIONS (engine='SPARK',
    jar_uris=["gs://spark-lib/bigquery/spark-bigquery-with-dependencies_2.12-0.26.0.jar"])
  LANGUAGE python AS R"""
import os
import json
OUTPUT_TABLE = 'google.com:crosbie-test-project.bigspark.'
OUTPUT_TABLE += str(json.loads(os.environ["BIGQUERY_PROC_PARAM.output_table"]))

print("found output table ", OUTPUT_TABLE)

from pyspark.sql import SparkSession
from pyspark.sql.functions import lit, col, split
import string

spark = SparkSession.builder.config("spark.jars.packages", "com.google.cloud.spark:spark-bigquery-with-dependencies_2.12:0.26.0").
spark.conf.set('temporaryGcsBucket', 'crosbie-dev')
df = spark.read.format("bigquery").load("google.com:crosbie-test-project.bigspark.top_rising_terms_20220910")

print("started Spark session")

df = df.withColumn("term_array", split(col("term")," "))

#one hot encoder with custom parameters
from pyspark.ml.feature import CountVectorizer
customVectorizer = CountVectorizer(inputCol="term_array", outputCol="Term_OneHotEncoded", vocabSize=4, minDF=1.0)
customVectorizer_model = customVectorizer.fit(df)
df_one = customVectorizer_model.transform(df)

print("transformed table")

```

Google Cloud

DSN201_Demo1_AbhishekKashyap



Google Cloud

Serverless Spark
Interactive + Vertex AI for Data Science

Preview now available

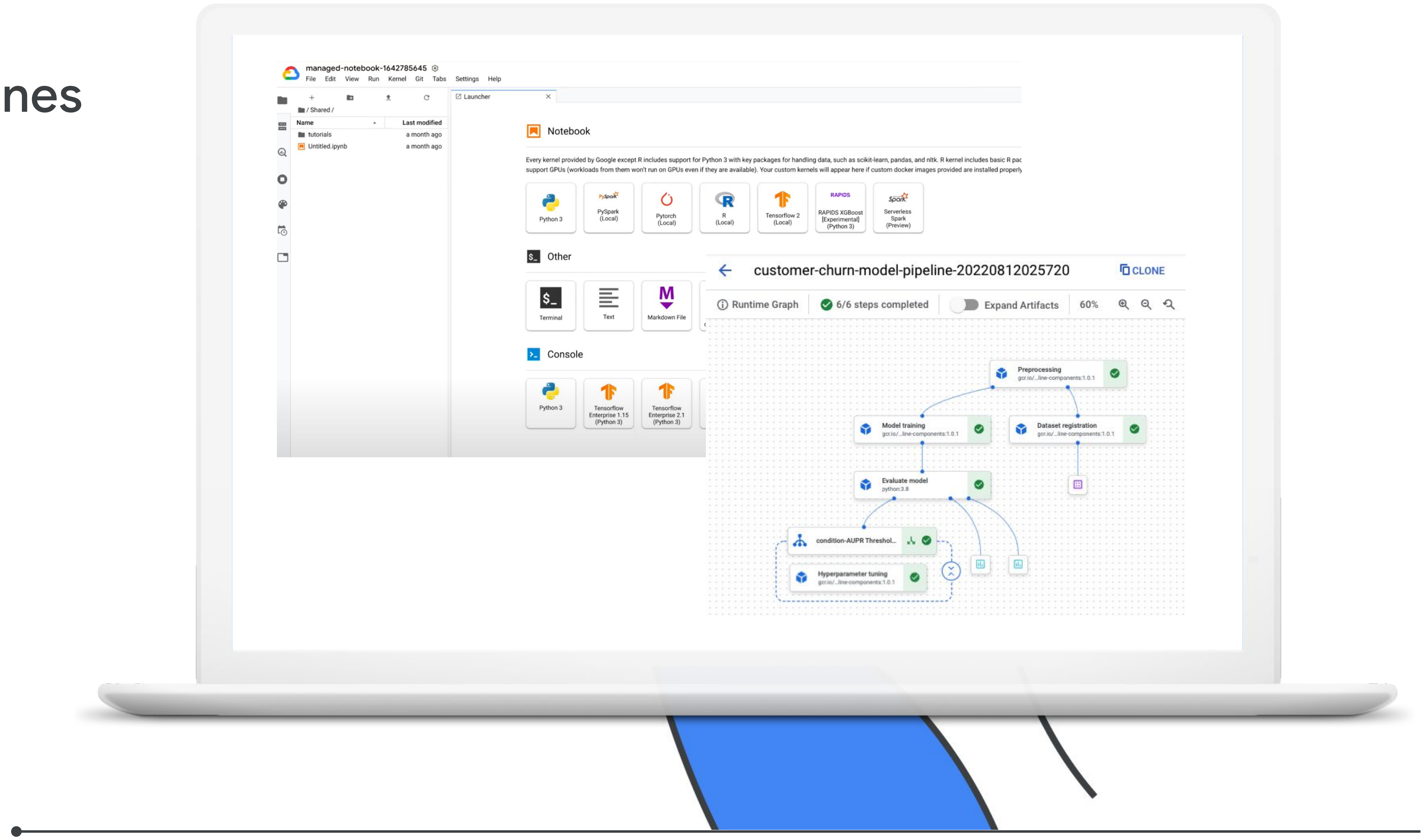
Accelerate data science development and MLOps pipelines

Spark for Data Science in 1 click
Data scientists can use Spark for development from notebooks and Vertex AI workbench seamlessly

No cluster creation needed

Built-in security and authentication
GCP security and user access are automatically applied from Vertex AI to Spark

Integrate Spark with MLOps
Execute Spark code through Kubeflow pipelines



managed-notebook-1642785645

/ Shared /

Name	Last modified
tutorials	a month ago
Untitled.ipynb	a month ago

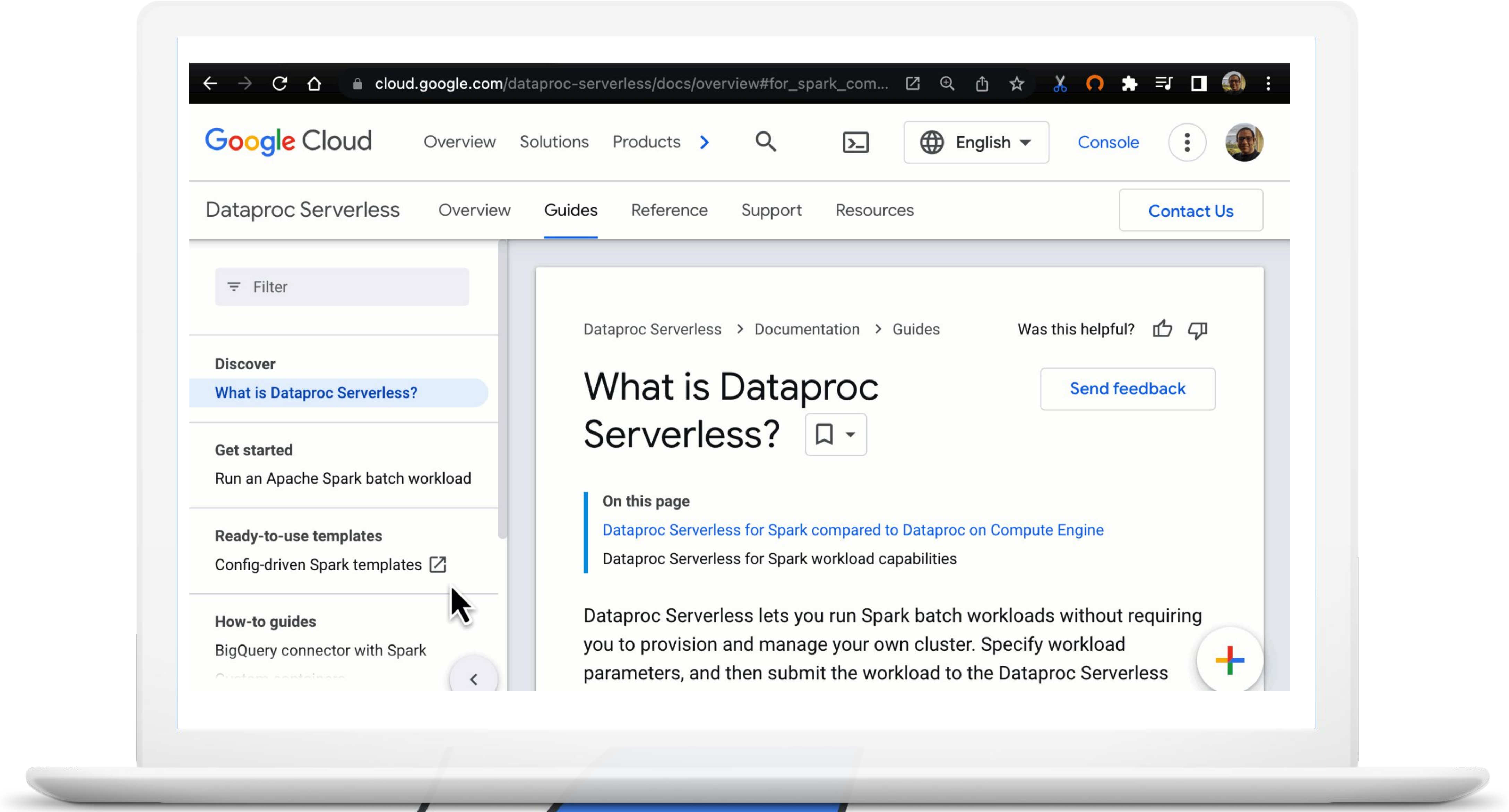
Every kernel provided by Google except R includes support for Python 3 with key packages for handling data, such as scikit-learn, pandas, and nltk. R kernel includes basic R pac
support GPUs (workloads from them won't run on GPUs even if they are available). Your custom kernels will appear here if custom docker images provided are installed properly

customer-churn-model-pipeline-20220812025720

6/6 steps completed

Open source templates

tinyurl.com/dtproc-templates


cloud.google.com/dataproc-serverless/docs/overview#for_spark_com...
Google Cloud
Overview
Solutions
Products
English
Console
Dataproc Serverless
Overview
Guides
Reference
Support
Resources
Contact Us
Filter
Discover
What is Dataproc Serverless?
Get started
Run an Apache Spark batch workload
Ready-to-use templates
Config-driven Spark templates
How-to guides
BigQuery connector with Spark
Dataproc Serverless > Documentation > Guides
Was this helpful?
Send feedback
On this page
Dataproc Serverless for Spark compared to Dataproc on Compute Engine
Dataproc Serverless for Spark workload capabilities

What is Dataproc Serverless?

Dataproc Serverless lets you run Spark batch workloads without requiring you to provision and manage your own cluster. Specify workload parameters, and then submit the workload to the Dataproc Serverless

Easily get started with serverless Spark for your use cases

Templates

- 16+ Java templates
- 16+ Pyspark templates
- Notebooks

Easy to use

- Open source
- Launch from cloud shell using inbuilt scripts

Row and column ACLs for GCS files now possible

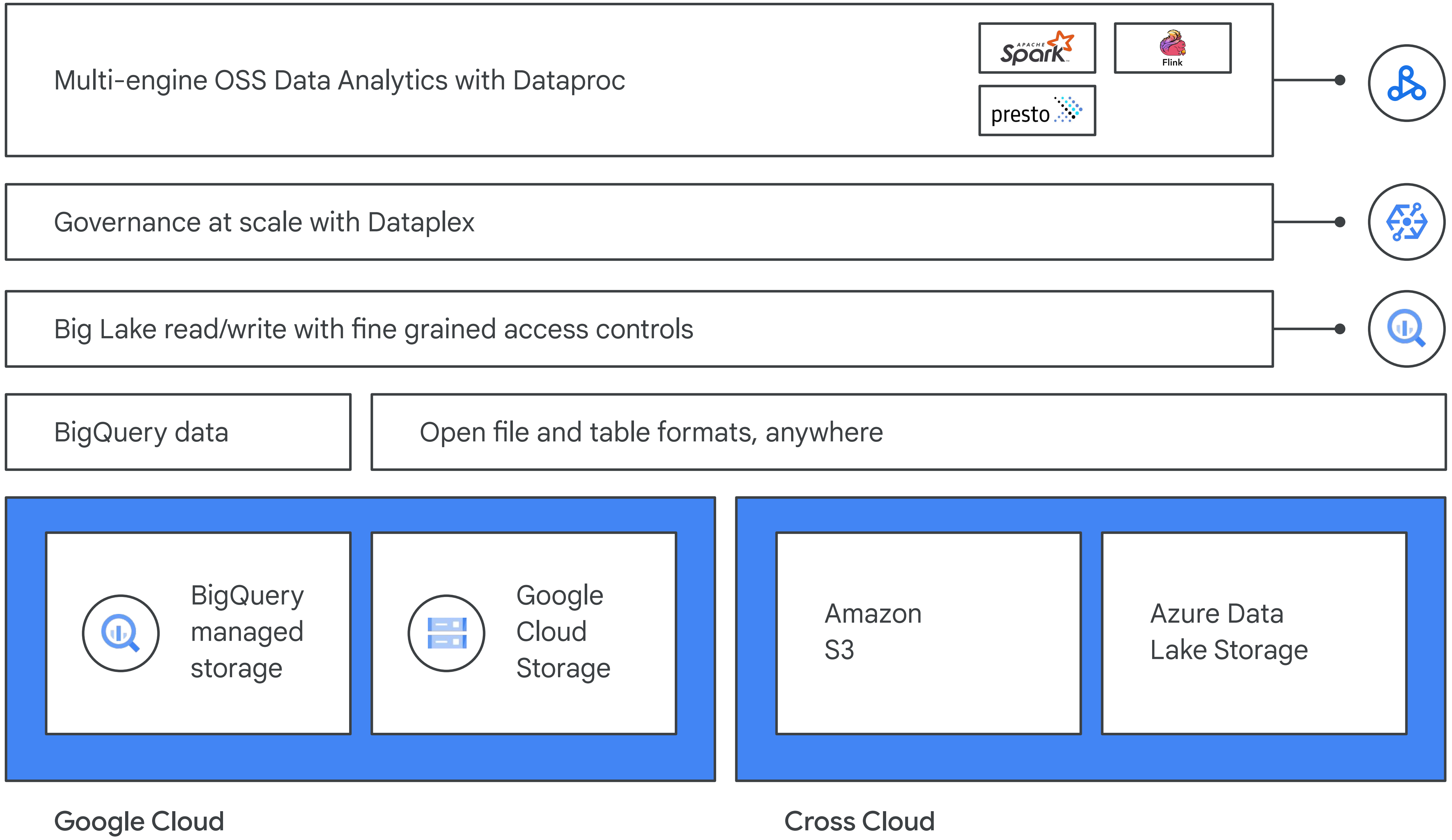
Now available

Fine grained governance for Spark through BigLake

OSS Analytics on all of your data

Any OSS engine on any data, anywhere, with a unified governance and access layer

- Dataproc runs your OSS workloads
- Dataplex scales your data governance
- BigLake provides a standardized access layer with fine grained access control to any data



Multi-engine OSS Data Analytics with Dataproc

Governance at scale with Dataplex

Big Lake read/write with fine grained access controls

BigQuery data

Open file and table formats, anywhere

BigQuery managed storage

Google Cloud Storage

Amazon S3

Azure Data Lake Storage

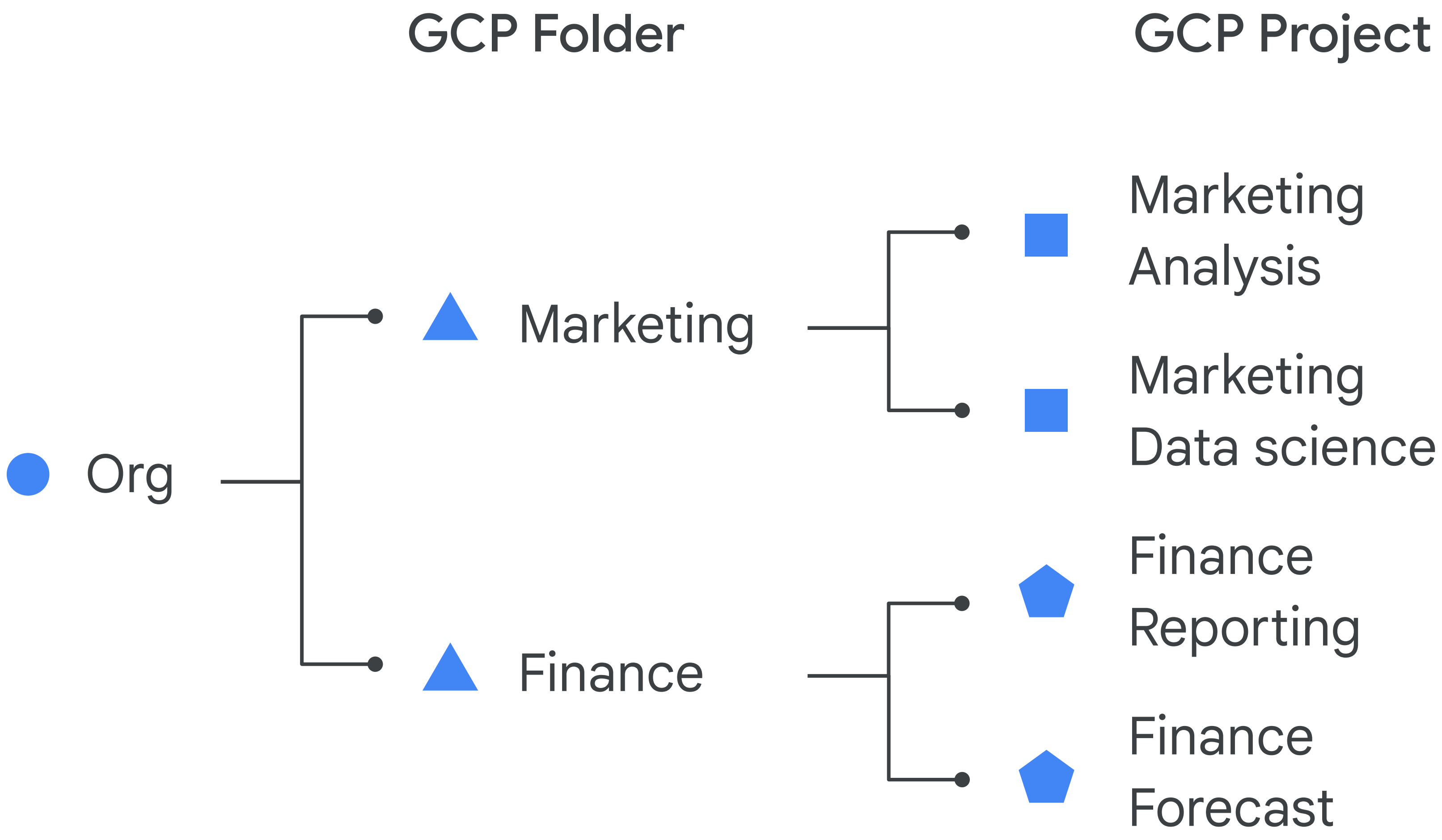
Google Cloud

Cross Cloud

Announcing

Preview now available

Automated Dataproc policy management



GCP Folder

GCP Project

Org

Marketing

Finance

Marketing Analysis

Marketing Data science

Finance Reporting

Finance Forecast

Standardize config per Org, Folder, or Project

- Resource policies for cost management, e.g., GPUs restricted to data science, VM configs
- Security policies, e.g., more stringent for projects dealing with PII data
- Network policies, e.g., internal IP only
- Images and components
- Metastore configuration

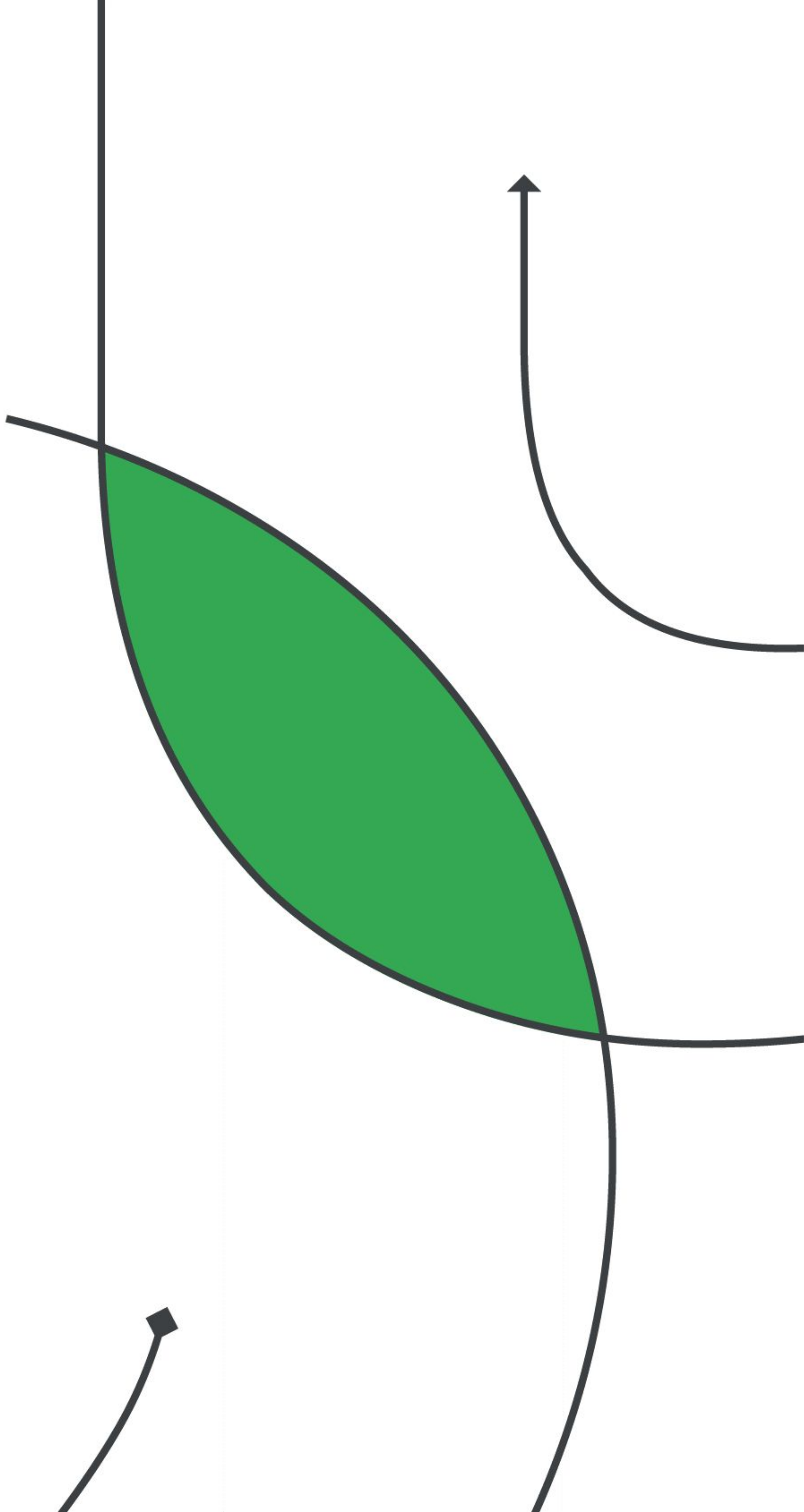
Real-time access to BigQuery metadata from Spark jobs

Dataproc Metastore BigQuery Federation

- Read and write access to BQ tables from Hive metastore
- Fully integrated with BQ permissions
- Support for both DDL and DML statements from Spark

```
scala > spark.sql("create database bqdataset");
scala > spark.sql("show databases").show();
+------------------+
|         namespace|
+------------------+
|         default  |
|         bqdataset|
+------------------+
scala > spark.sql("create table bqtable(id int, name string);
scala > spark.sql("desc bqtable").show();
+--------+---------+-------+
|col_name|data_type|comment|
+--------+---------+-------+
|      id|   bigint|   null|
|    name|   string|   null|
+--------+---------+-------+
```

/LiveRamp



We connect consumer data with durable privacy-conscious post-cookie identifiers for more accurate customers views, improved measurement, and secure data collaborations.

At LiveRamp, we make it safe and easy for companies to use data effectively.

Identity resolution

Data enrichment and audience activation

Data collaboration and media networks

LiveRamp identity platform with embedded cloud services

TOYOTA	NORDSTROM	SAMSUNG
TD Ameritrade	★macy's	IHG
Boston Scientific	UNITED	20
WARNER BROS. DISCOVERY	CONDÉ NAST	SEPHORA
MARRIOTT	Uber	LANDS'END
Office DEPOT	TIME	hp
Constellation Brands		

LiveRamp engineering

Team

Spans 11 countries with a total of > 450 engineers. LiveRamp is proud to be named in Fortune’s Best Workplaces in Technology™ 2021.

Data processing

Not map-reduce but map-join

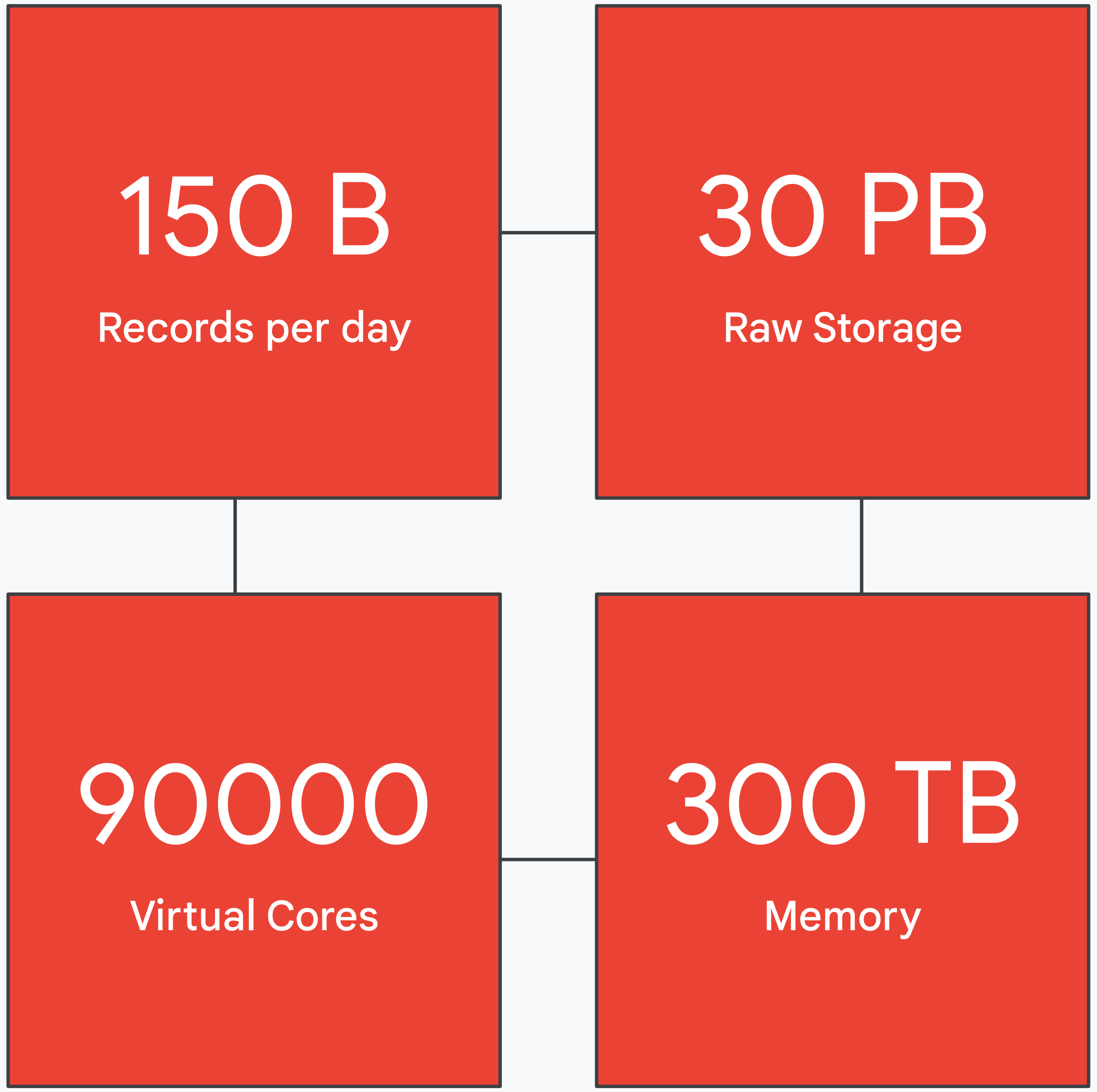
Biggest Identity graph

Technology stack

Open source technologies

Multi-cloud support (GCP, AWS, Snowflake)

Problem and infrastructure scale



150 B

Records per day

30 PB

Raw Storage

90000

Virtual Cores

300 TB

Memory

Migrating to GCP

Architectural Decisions

Decentralized team ownership

HDFS -> GCS

Autoscaling clusters

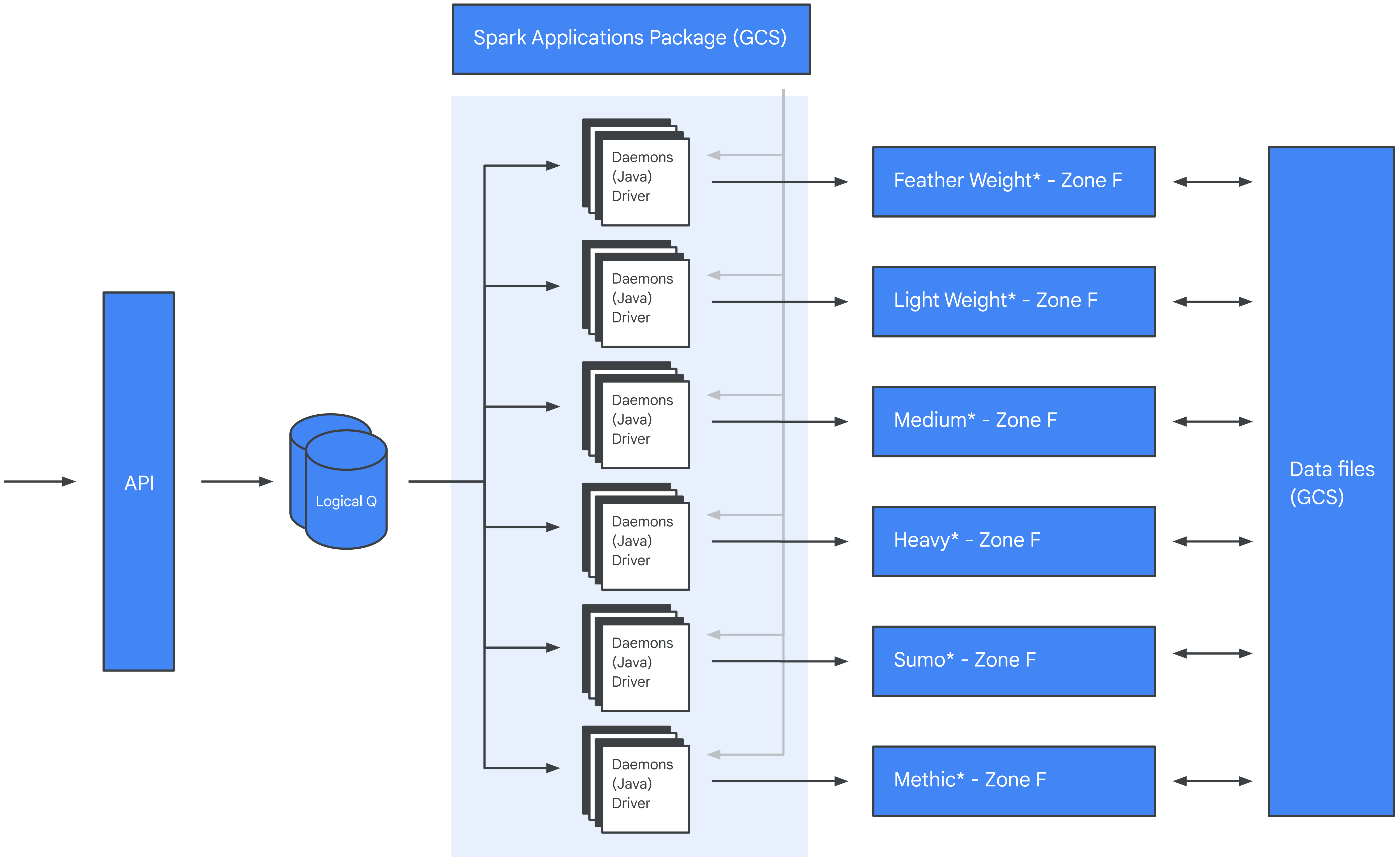
Infrastructure as Code

Self-service tooling for data engineers to deploy infrastructure as easily and safely as possible

Map Side Join (MSJ) Library

Schema that defines virtual partitions, a strategy for assigning records to those partitions, and a library that handles writing/reading that data in a distributed fashion.

LiveRamp's largest workload Infrastructure



Key benefits on GCP

Support

- Weekly sync-ups
- Roadmap collaboration
- Clear escalation path

Cost attribution

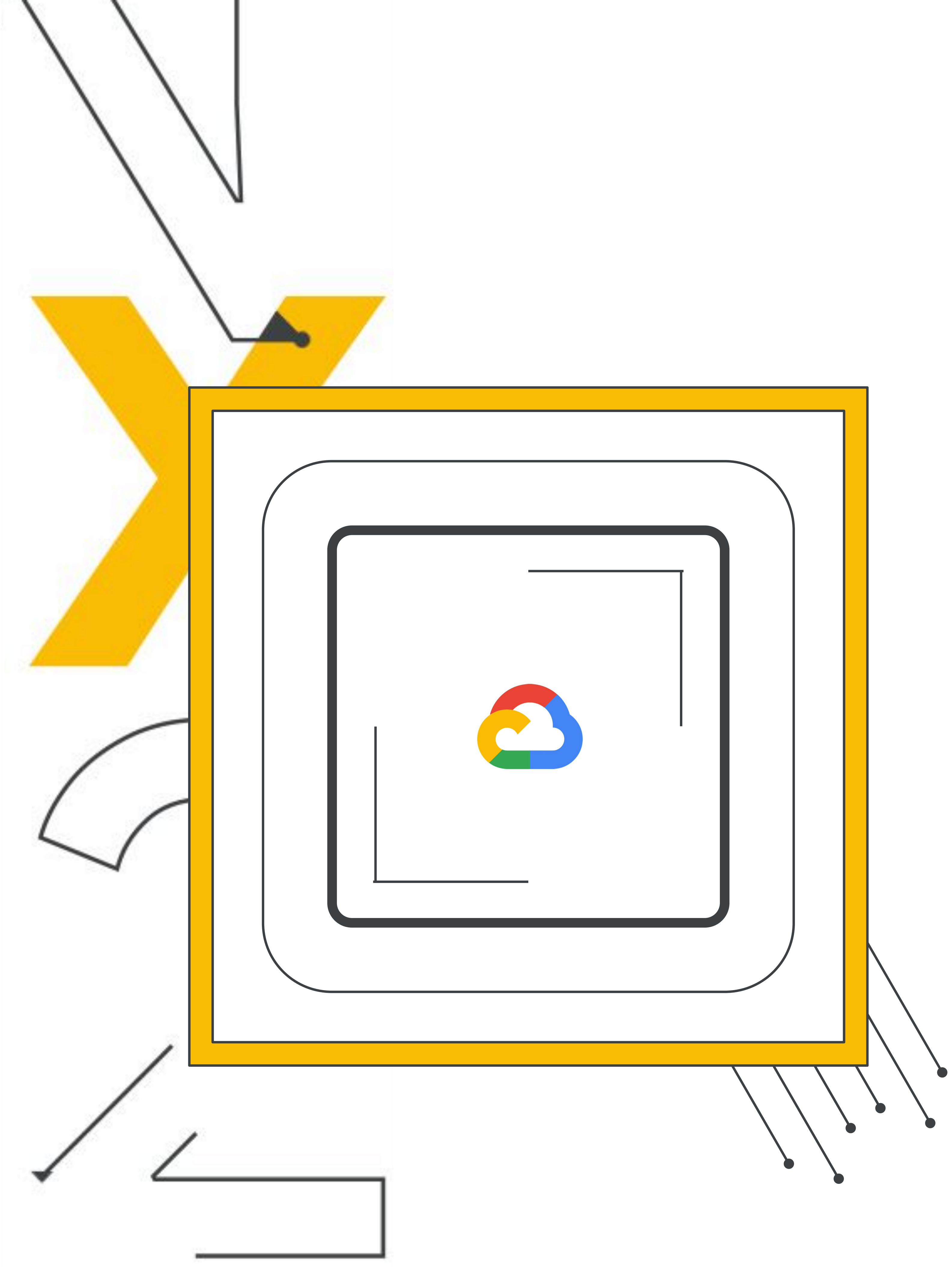
- Cost attributed at Project/Cluster/Asset (VM etc.) using tagging
- Monitoring and alerting to protect against the cost overruns

Flexibility

- Self-service Terraform module
- Agility in cluster management - create, delete etc.
- Environments for A/B testing

Cost savings

- ~ 30% cost savings in some clusters
- Some applications is now 10x faster



Where we’re headed

Performance

GPUs and VM shape

Scale

Driver Pools

Management

Enhanced Monitoring

Alerting

Agility

VertexAI

Serverless DataProc

Lessons learned on migrating to the Cloud

Benchmark

Ensure current behavior of key workloads is clearly understood before migration

Focus

For us, reliability first then cost optimization post-migration

Preview < > GA

Understand feature stability to judge risk and time for adoption

Quotas

We had to increase IP space and change quotas as we tried out new VM and disk settings

Discounts

Understand impact on Committed Use Discounts (CUDs) when you start to migrate to Spot VMs

Google Cloud

Next '22

How to scale data analytics securely with Spark on Google Cloud