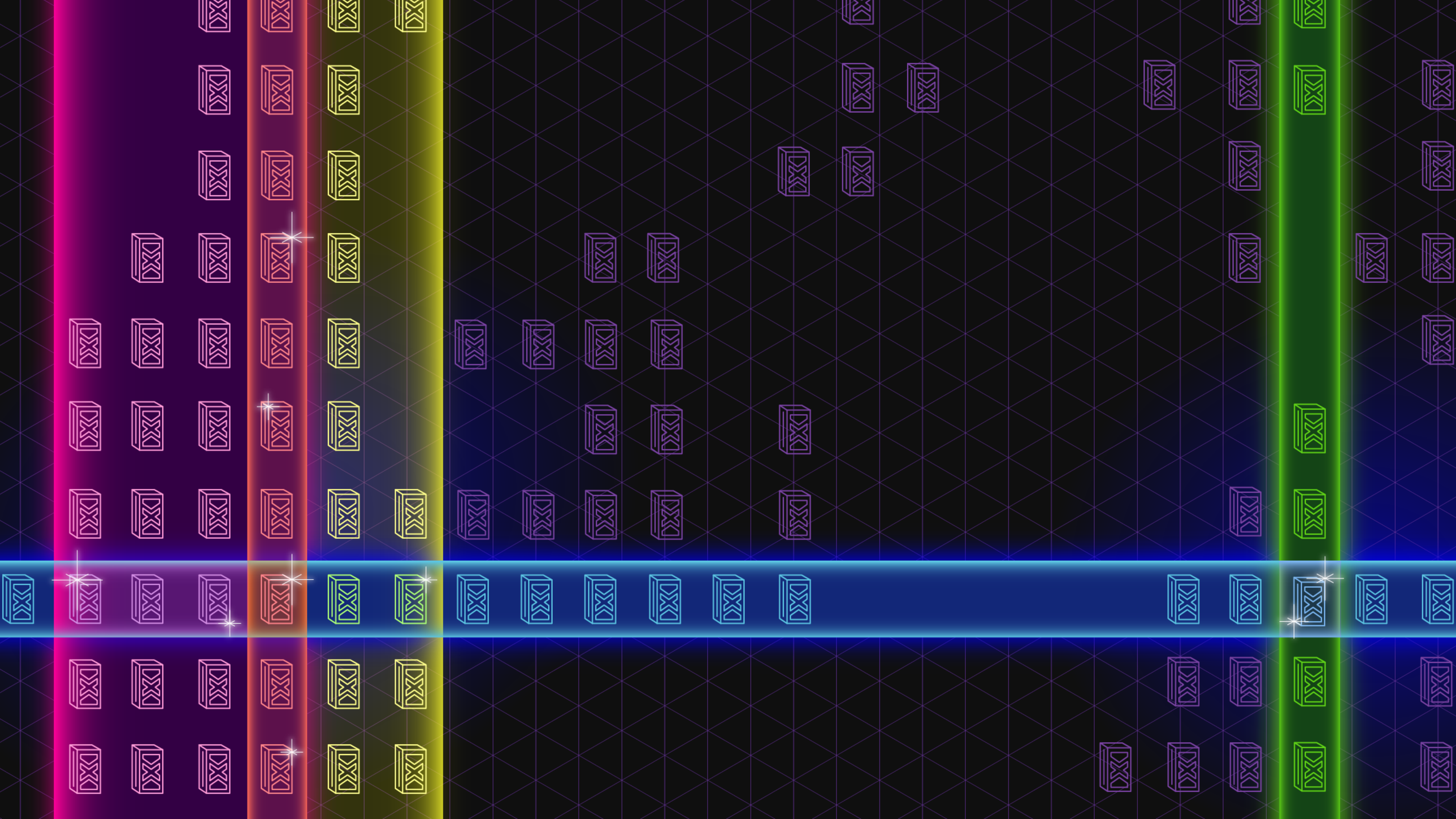
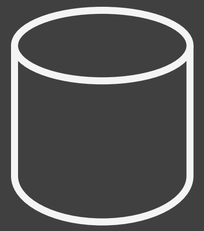
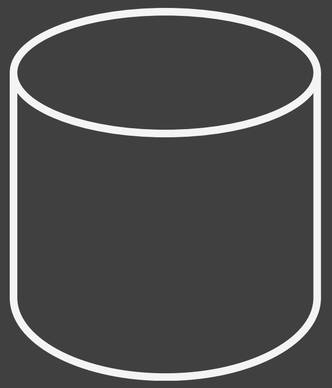


The Evolution of a Data Project

Python script

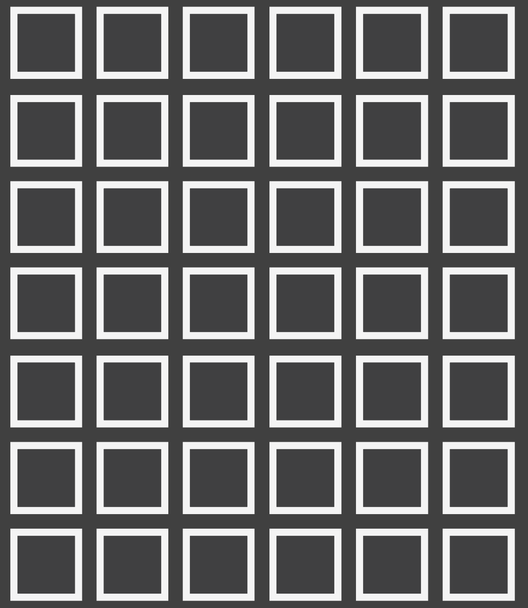


SQL on live DB



SQL on reporting DB

Terrible confusion



Hadoop / Spark cluster

What needs fixing

- Data inaccessible outside single cluster
- Want cluster time? You have to wait.
- Clusters are underutilized and EXPENSIVE

image: Pexels

Elastic Big Data Platform @ Datadog

Doug Daniels
Director, Engineering

DATADOG

What’s our big data platform do?

WHOM

Data Engineers
Data Scientists

do

WHAT

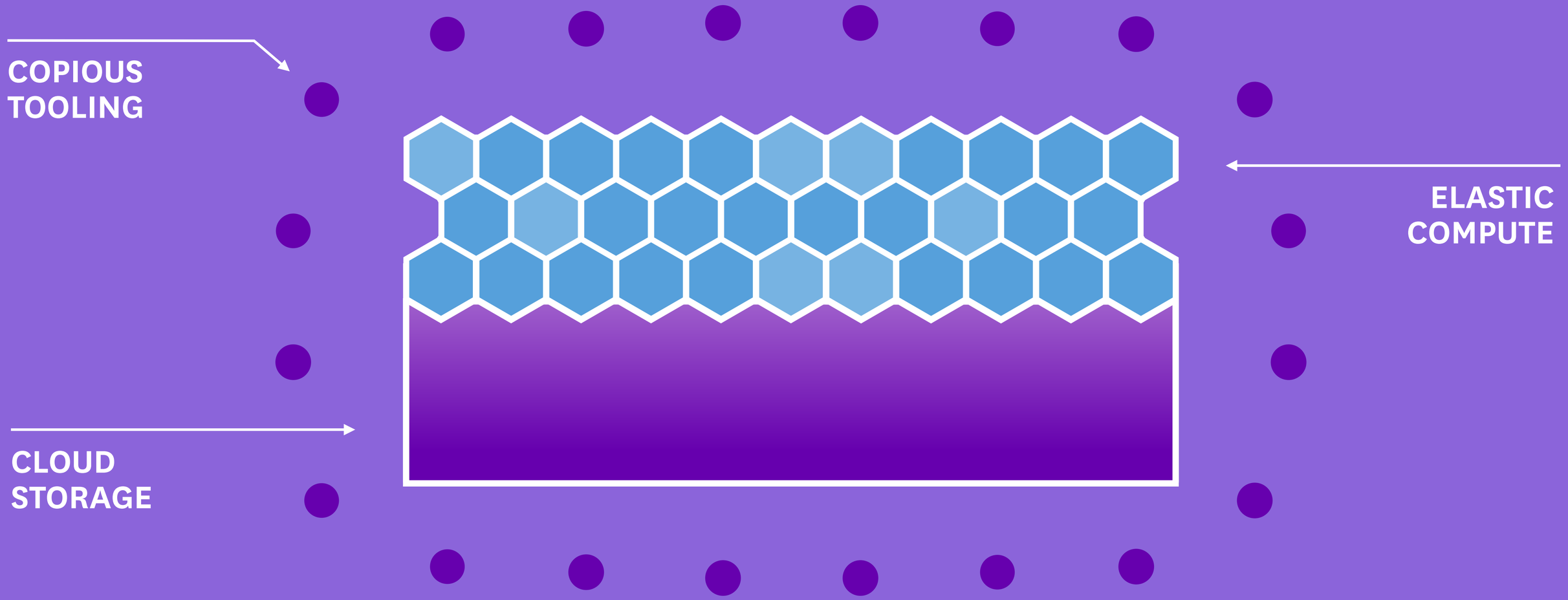
App features
Statistical Analysis/ML
Ad-hoc investigation

with

WITH

Spark
Hadoop (Pig)
Python (Luigi)

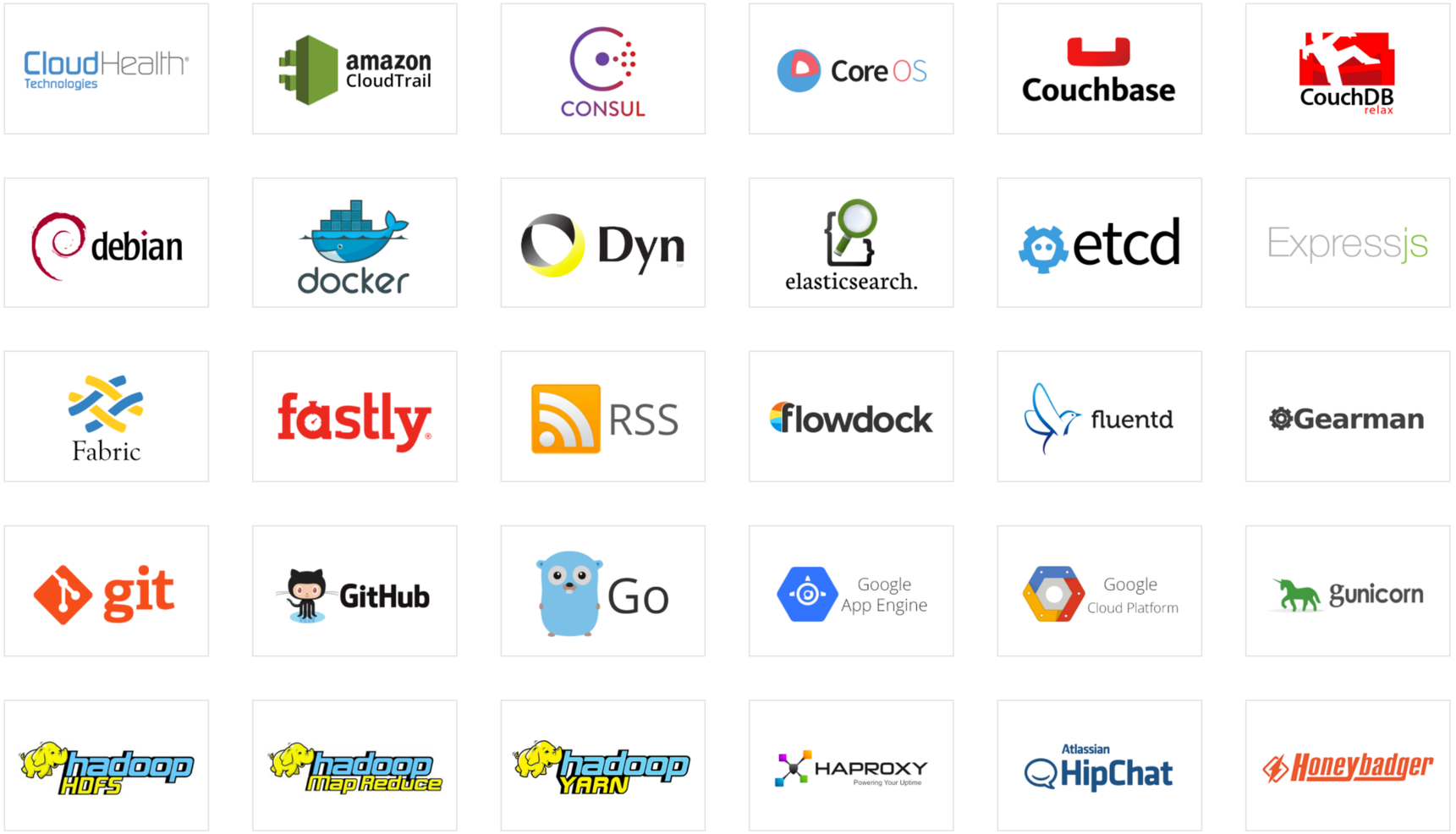
Exploring the platform



CLOUD STORAGE

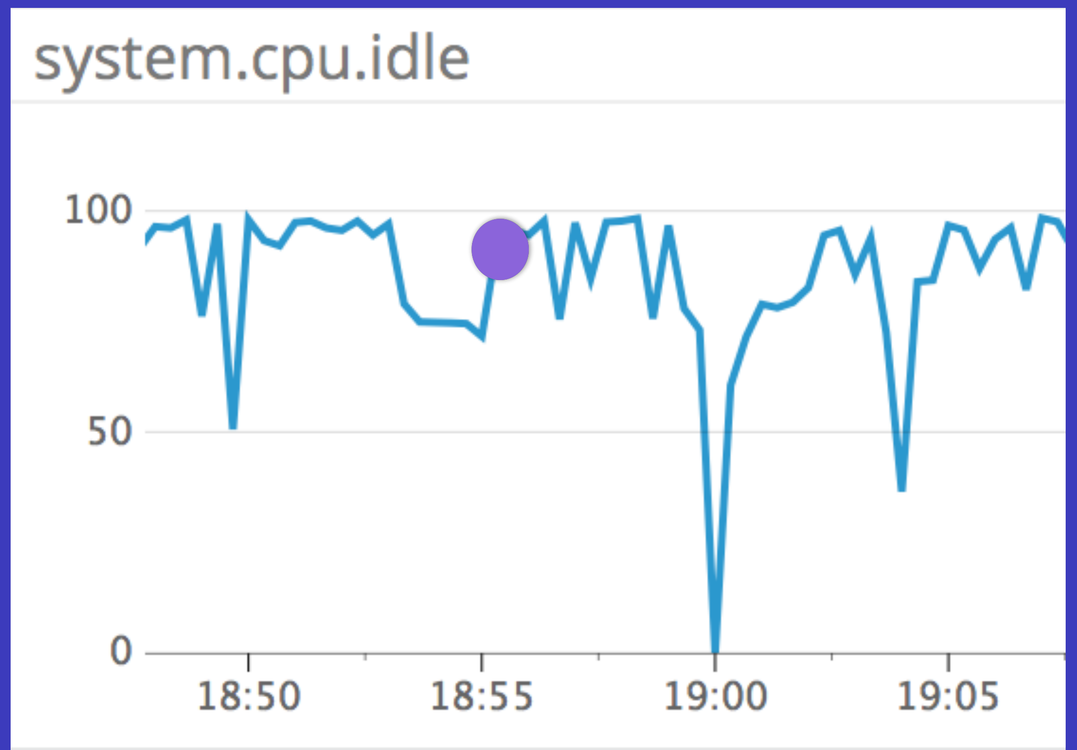
What do we store?

150 Integrations



...and more

What’s time series data?

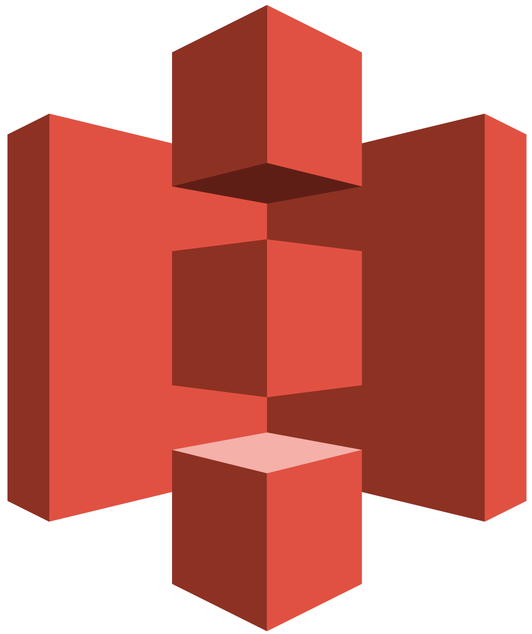


timestamp	1447020511
metric	system.cpu.idle
value	98.16687
tags	host:i-xyz, role:cassandra, …

We collect over a trillion of these per day

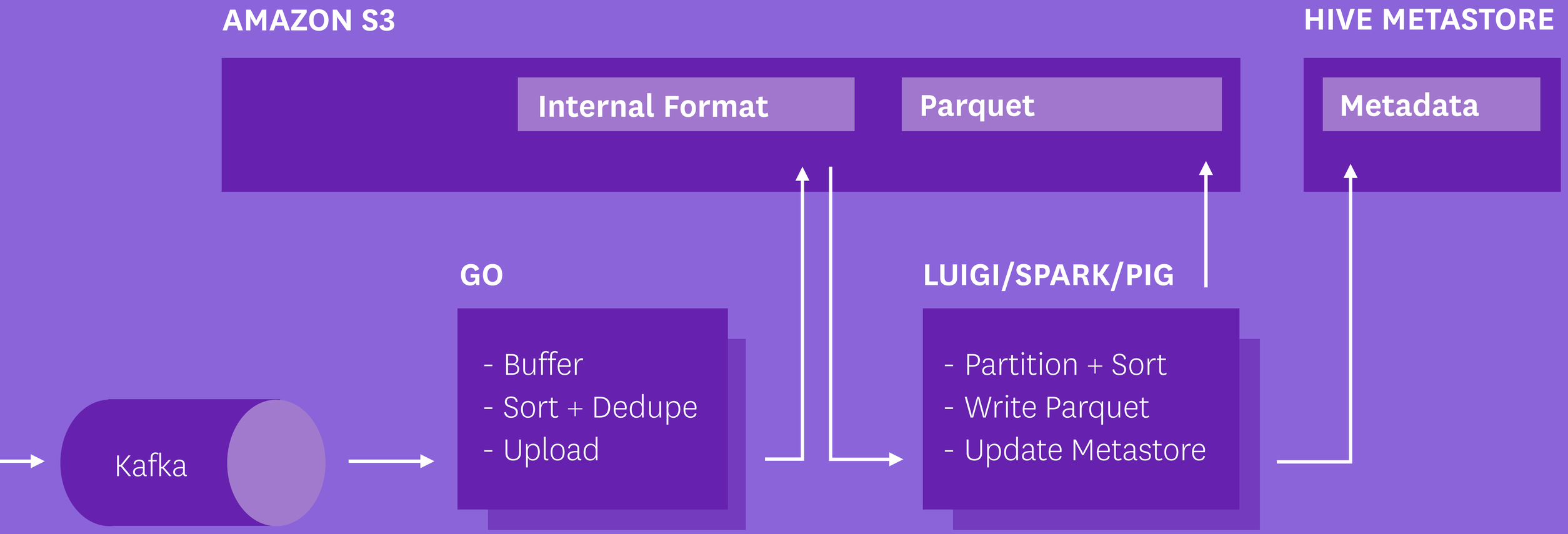
...and growing!

Where to put the petabytes?



Amazon S3

How data gets to S3



Isn’t this a job for HDFS?

What we don’t love about HDFS

- The “one cluster” problem
- Come for the storage, get stuck with the servers
- No Java? No data!

S3 is flexible

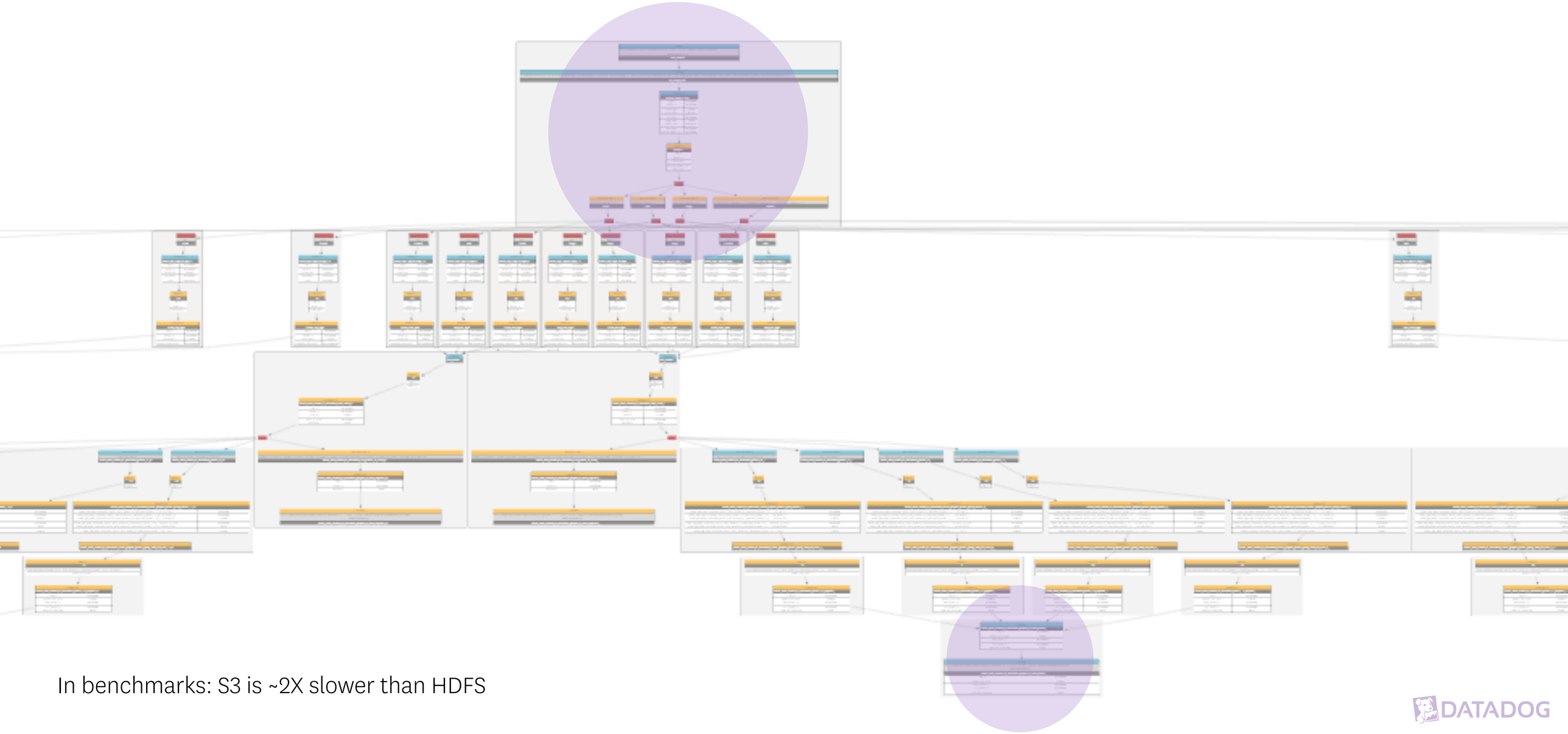
- Read data from as many clusters as you want
- Store unlimited stuff(*) with no management
- Rock solid: durability (99.999999999), availability (99.99)
- Access from any programming language

* Accepting laws of physics and your credit card limit

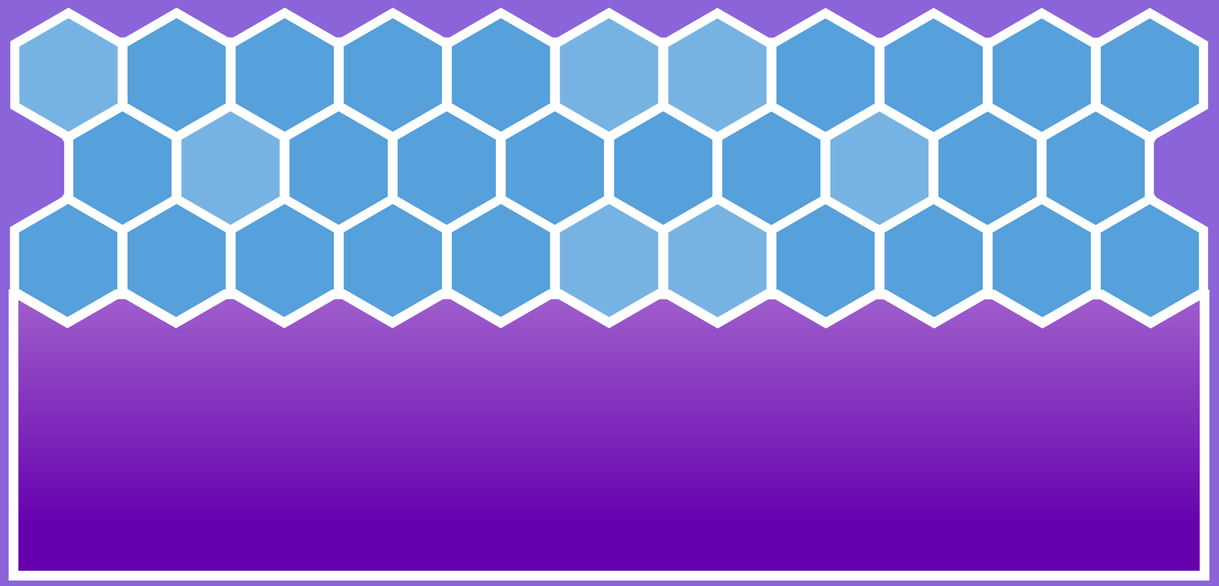
Decouple data and compute

(BREAK THE RULES!)

Breaking the rules is fine.



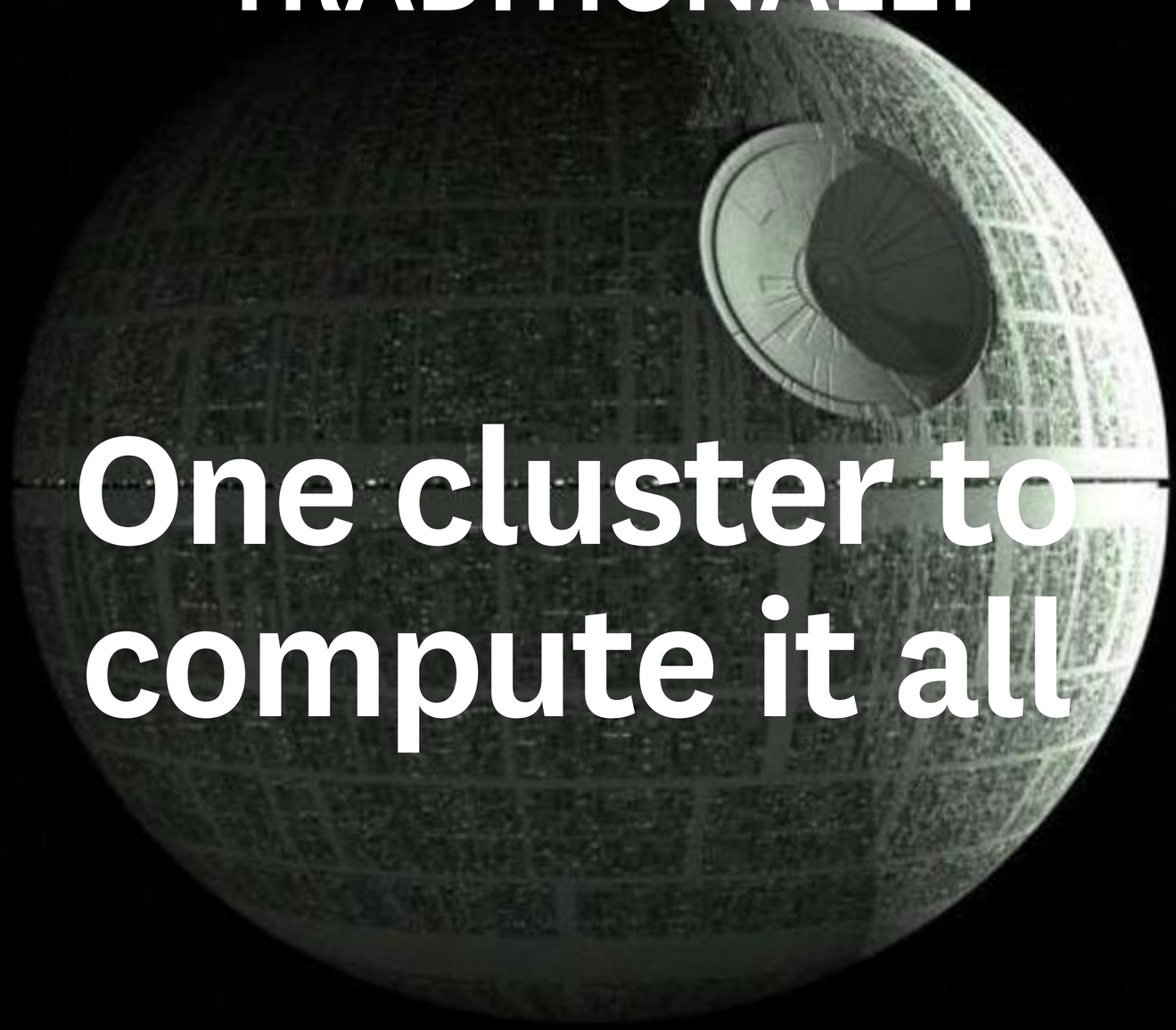
In benchmarks: S3 is ~2X slower than HDFS



ELASTIC
COMPUTE

CLOUD
STORAGE

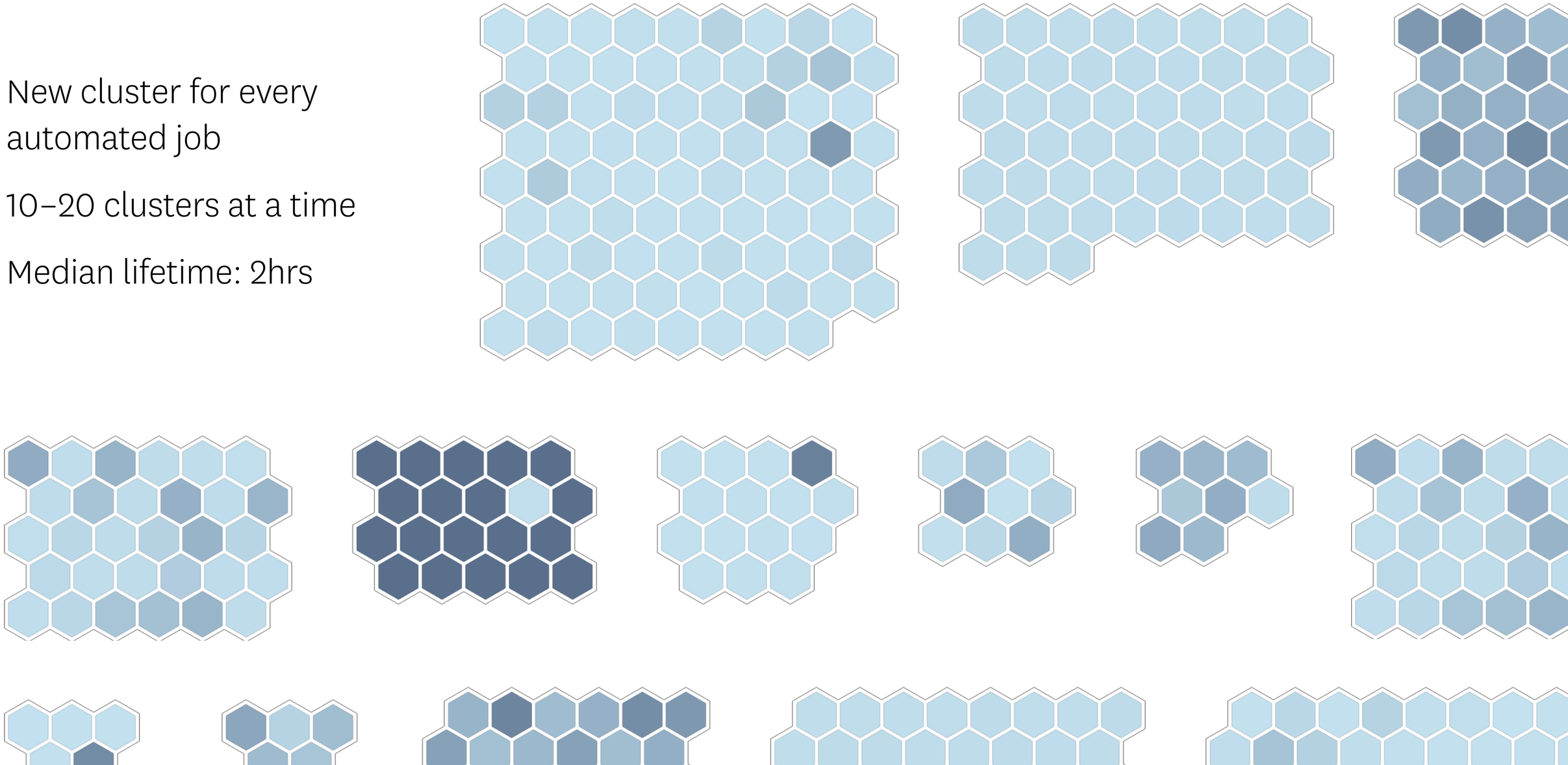
TRADITIONALLY



One cluster to compute it all

Instead, we run many, many clusters

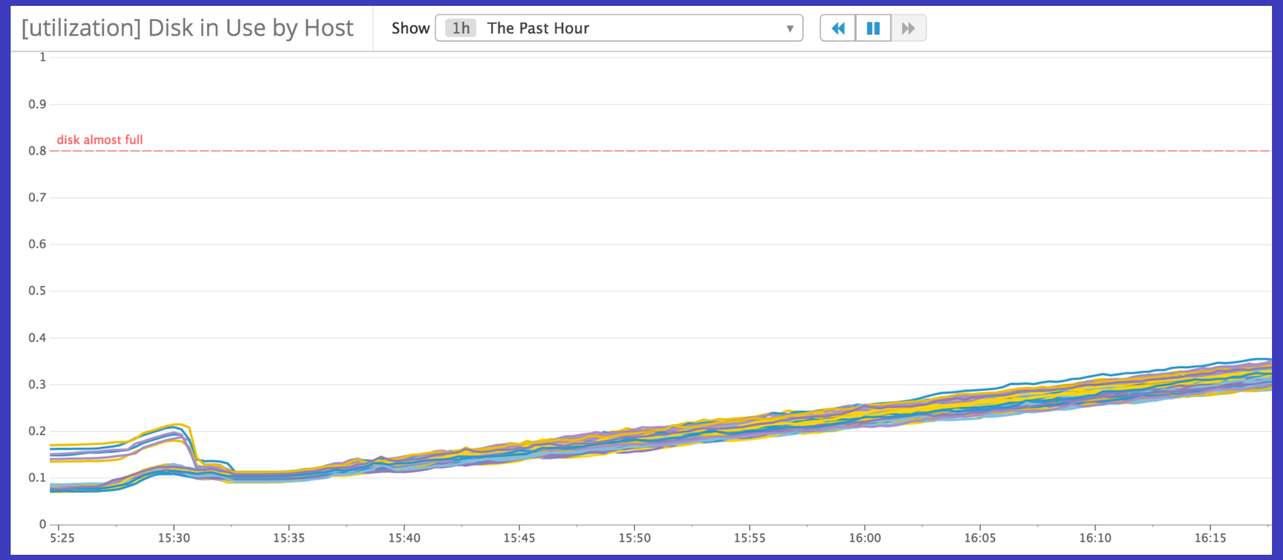
- New cluster for every automated job
- 10–20 clusters at a time
- Median lifetime: 2hrs

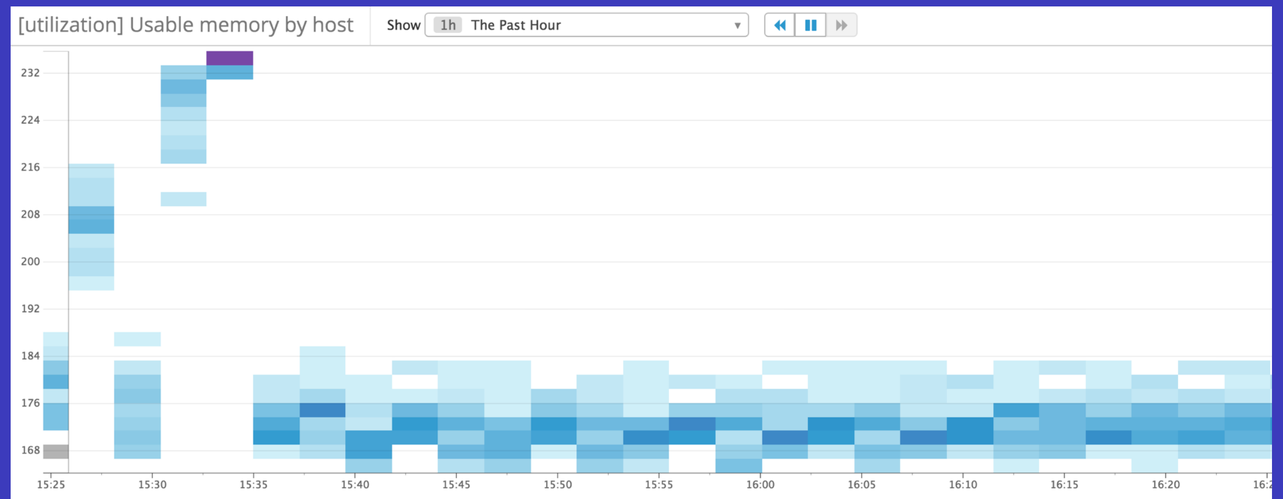


Why so many clusters?

Total isolation

We know what’s happening and why



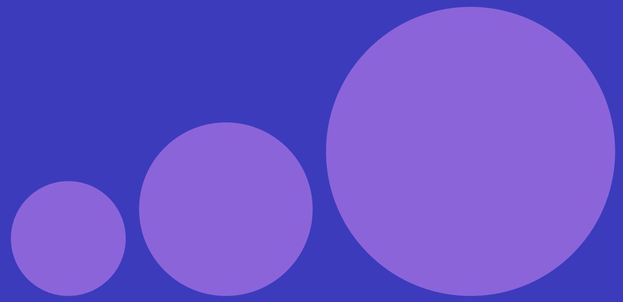


No more waiting on loaded clusters

- Tailor each cluster to the work you want to do
- Scale up when you need results faster
- Data scientists and data engineers don’t have to wait

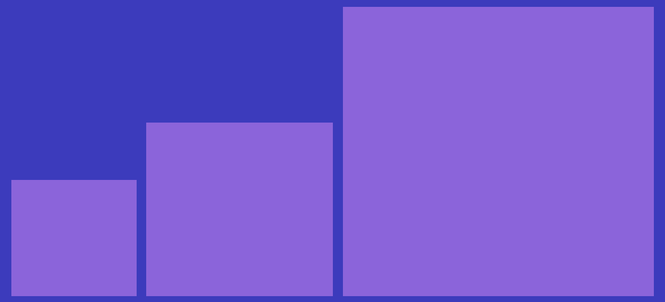
Pick the best hardware for each job

== ~30% savings over general purpose hardware



c3
for CPU-bound jobs

r3
for memory-bound jobs



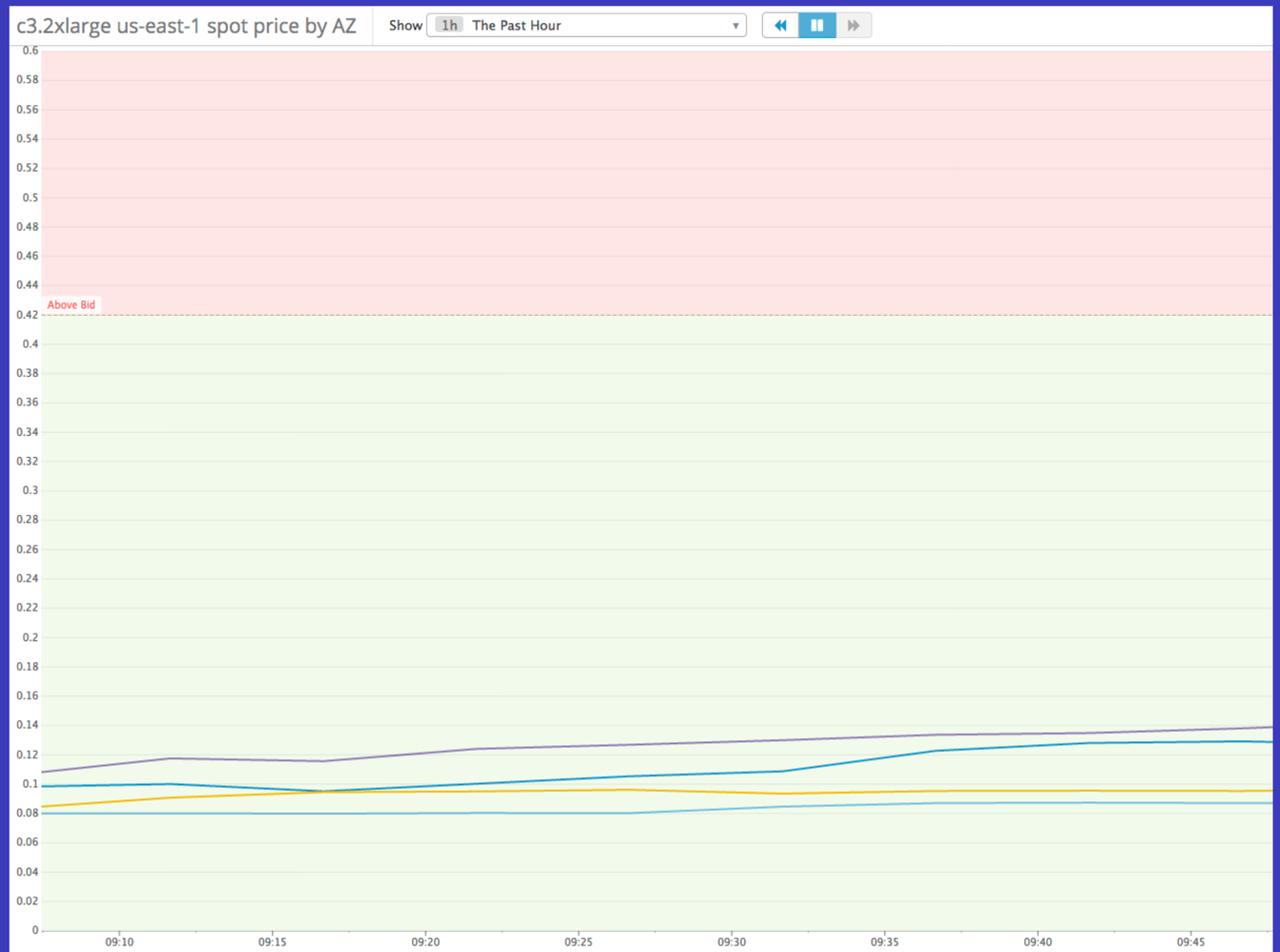
m1.xlarge
if you don’t care (cheap!)

100% spot-instance clusters, all the time.*

* (ok, most of the time)

How we do spot clusters

- Bid the on-demand price, pay the spot price
- Fallback to on-demand instances if you can’t get spot
- Monitor everything: jobs, clusters, spot market
- ☞ Save up to 80% off the on-demand price



Monitor the spot price

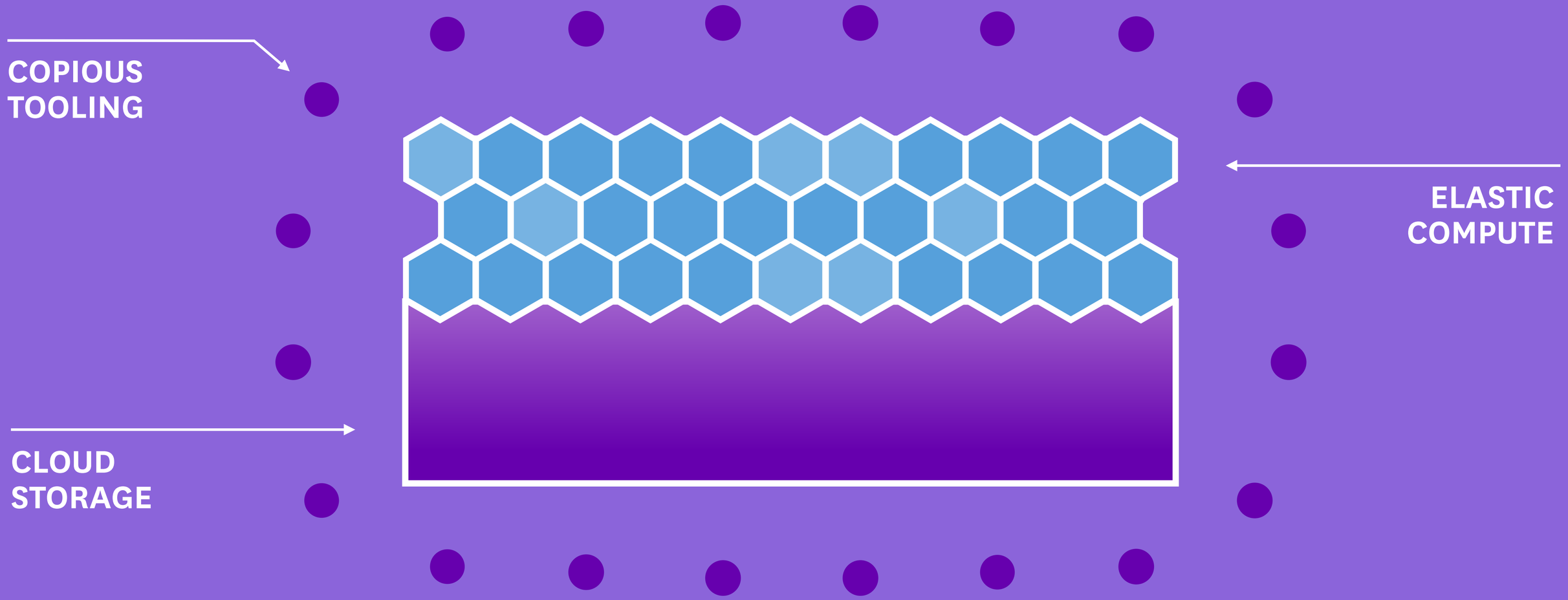
Switch hardware when the market gets volatile

We like this strategy a lot!

- ✓ No waiting for the cluster you need
- ✓ No waste from hardware sitting idle
- ✓ Spot clusters are affordable enough to use everywhere

What’s challenging, though?

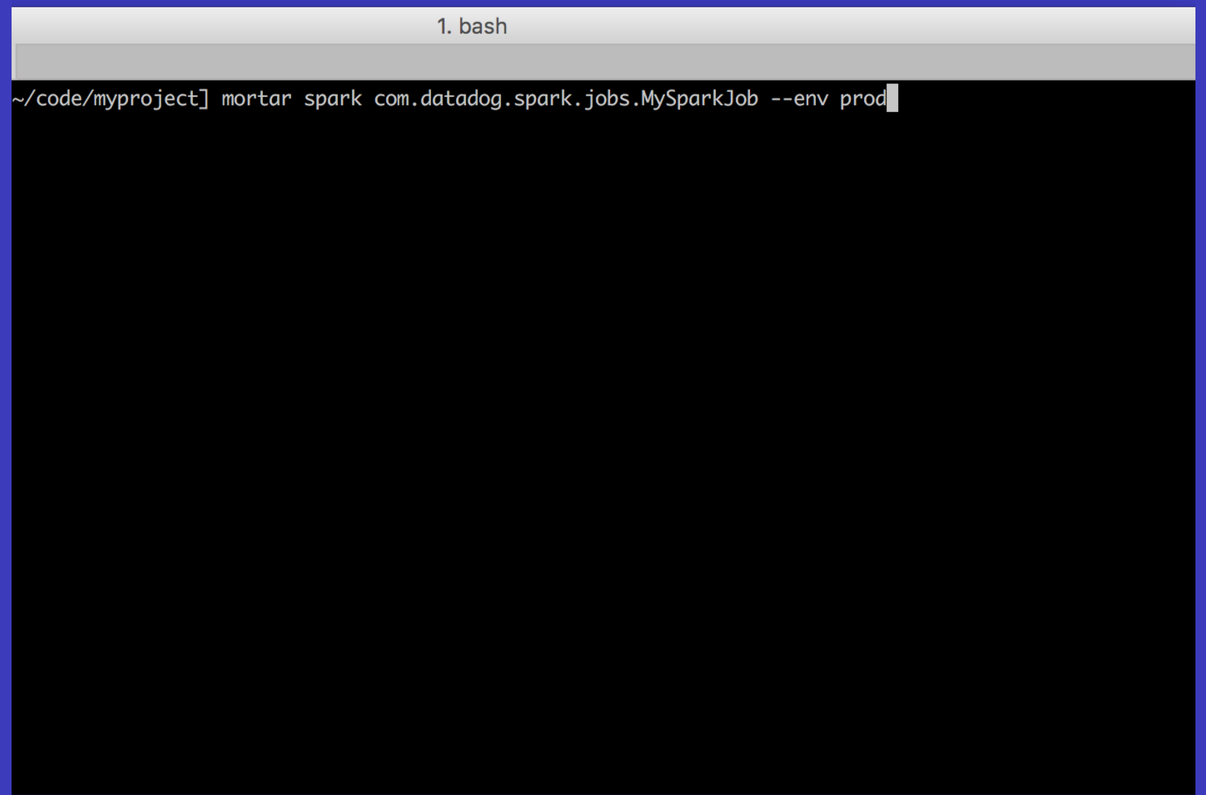
Many things that disappear.


COPIOUS
TOOLING
CLOUD
STORAGE
ELASTIC
COMPUTE

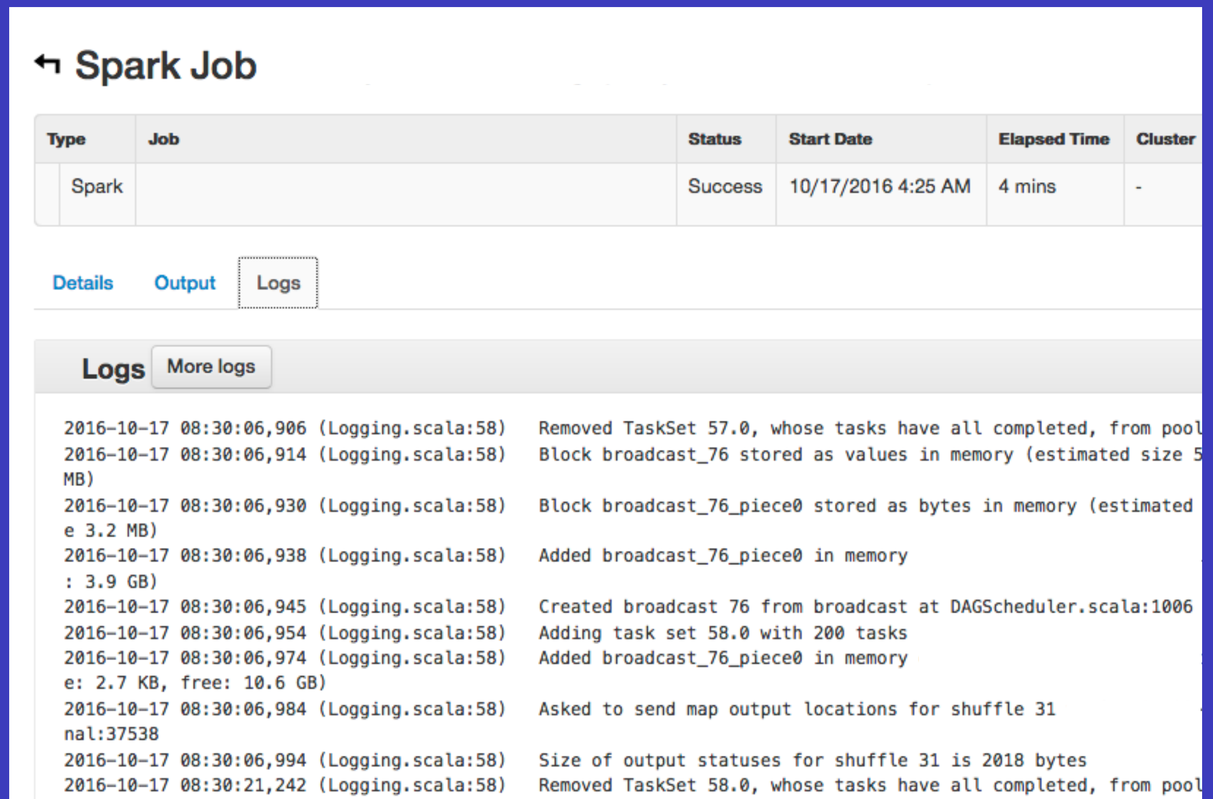
Platform as a service

Jobs, Clusters, Schedules, Users, Code, Monitoring, Logs, and more

CLI



Web and APIs

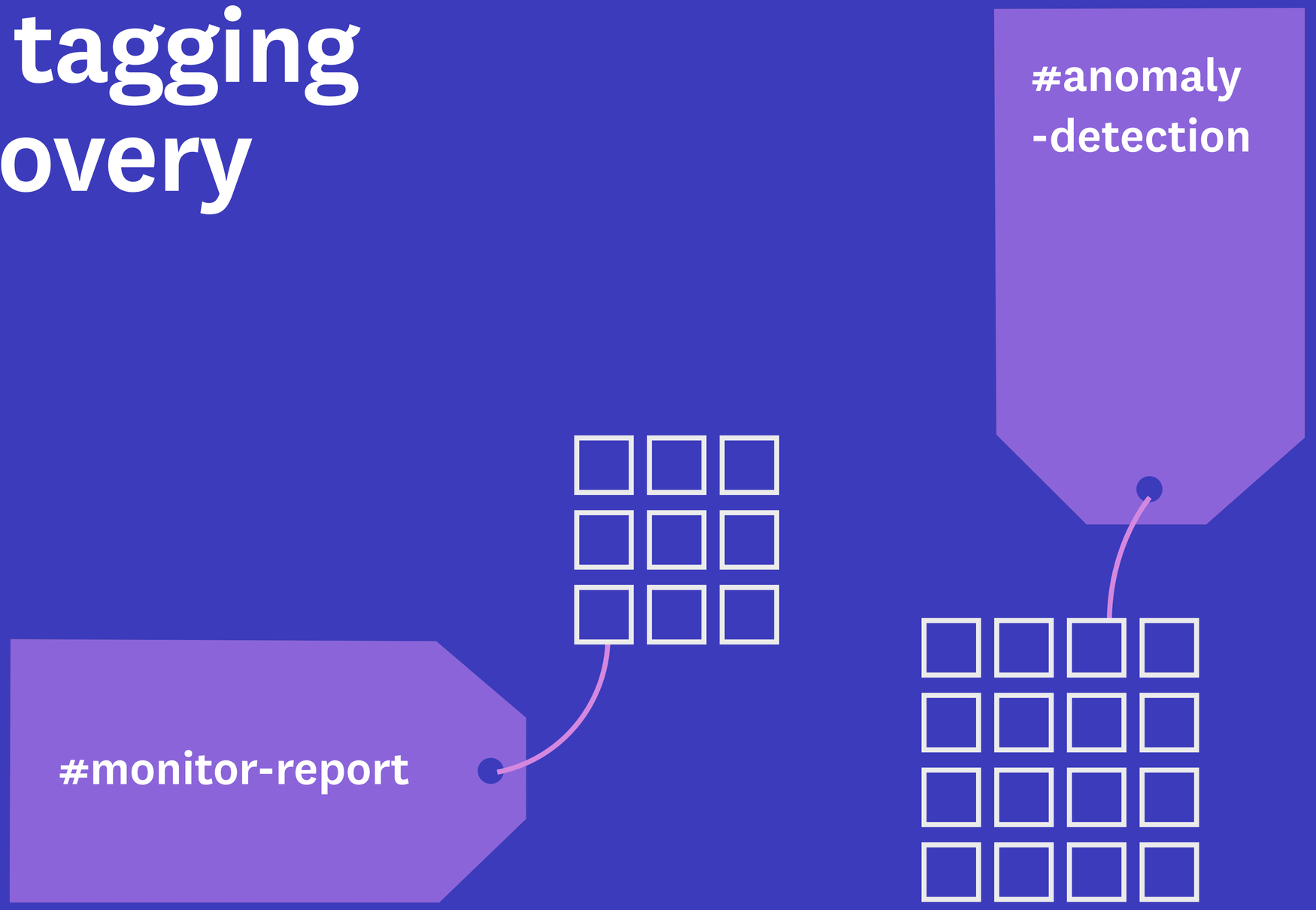


Big Data Platform Architecture

USER	CLI		API Clients		Job Scheduler			Datadog Monitoring 
WEB	Web			API				
STORAGE	Metadata DB		Queueing		Logs			
WORKER	Pig Workers		Spark Workers		Luigi Workers			
CLUSTER	EMR							
DATA	Amazon S3							

How to find the right cluster when they disappear?

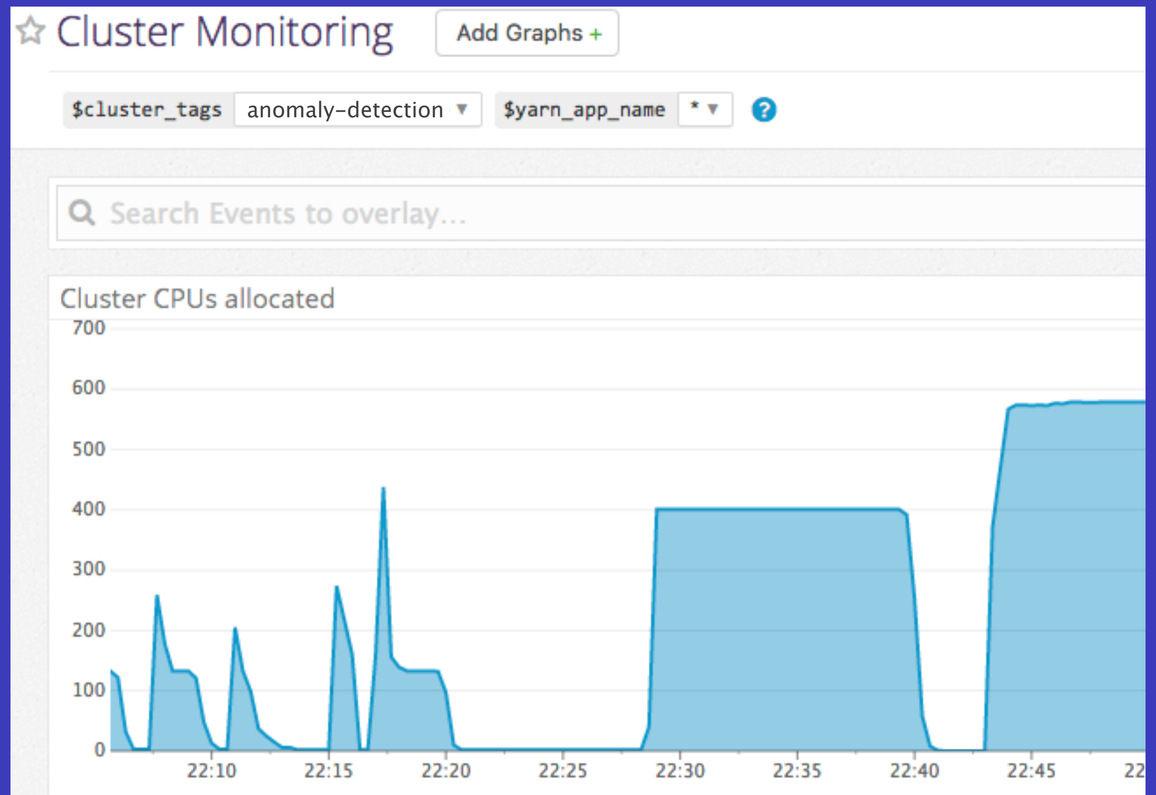
Cluster tagging for discovery



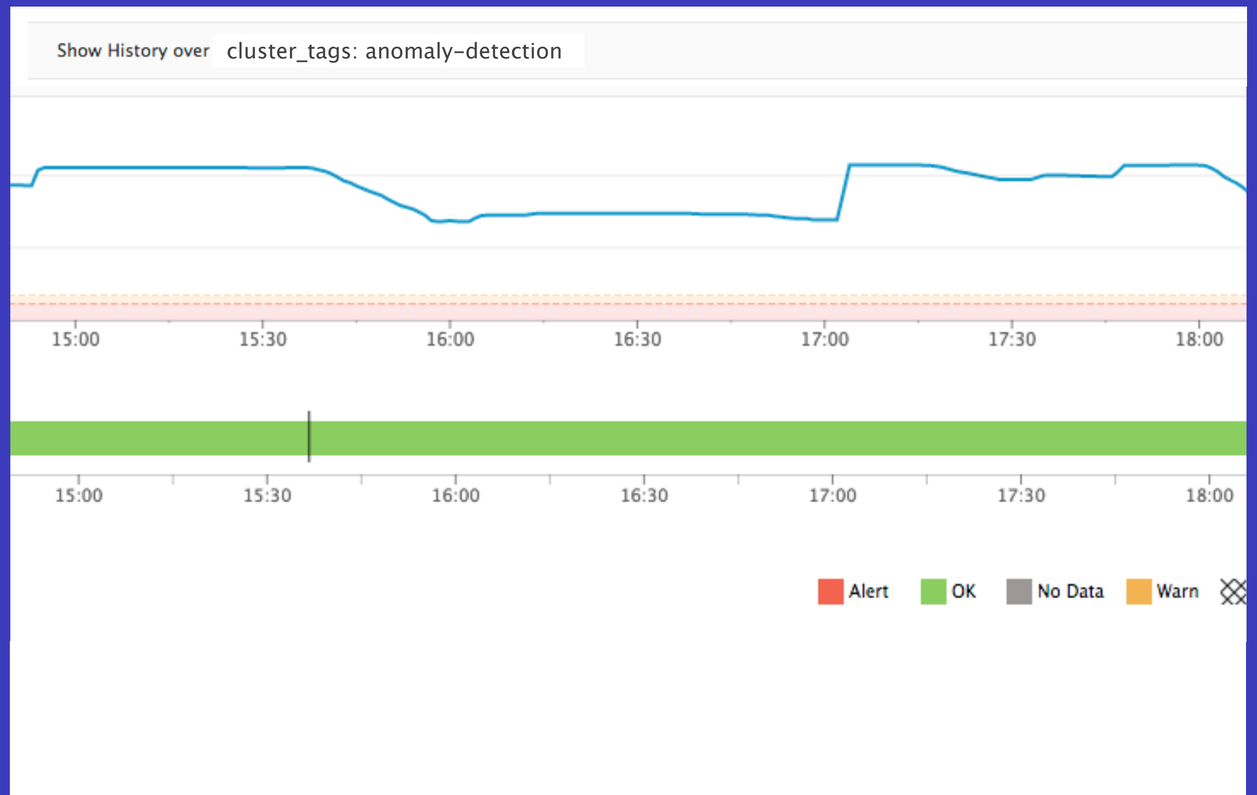
How to monitor many disappearing clusters?

Dynamic Monitoring on Tags

Dashboards



Monitors



How to avoid an ever-growing army of clusters?

Shut off the lights when you’re done

Single-job

1 job and done

Persistent cluster

Shut off after idle
for N minutes

Permanent cluster

Shut off yourself
(alerting on > 24 hours)

How to debug problems when the cluster’s gone?

Debugging In a Post-Cluster World

Send all logs to S3

- HDFS
- YARN
- Pig
- Spark

Visualize the pipeline

- Lipstick for Pig
- Spark History Server
- Luigi task flow

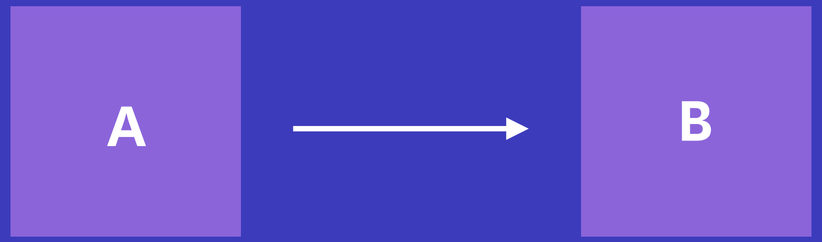
Preserve historical monitoring data

Keep history, by tag, after the cluster disappears

How to handle certain cluster failure in your jobs?

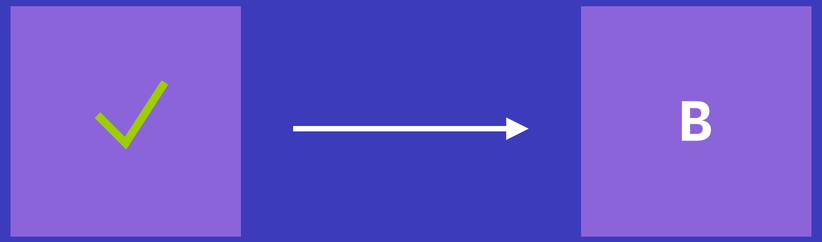
Luigi: design for failure.

Automatic cleanup and restart



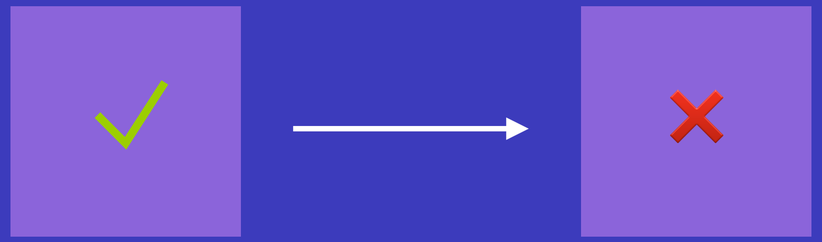
Luigi: design for failure.

Automatic cleanup and restart



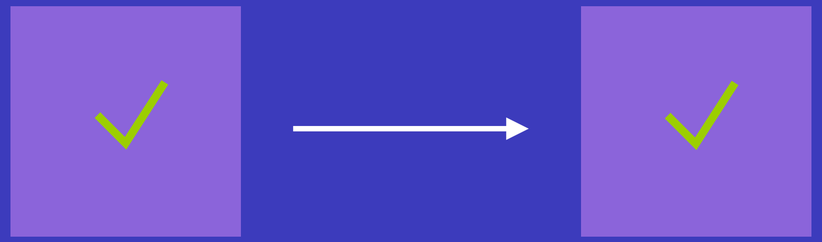
Luigi: design for failure.

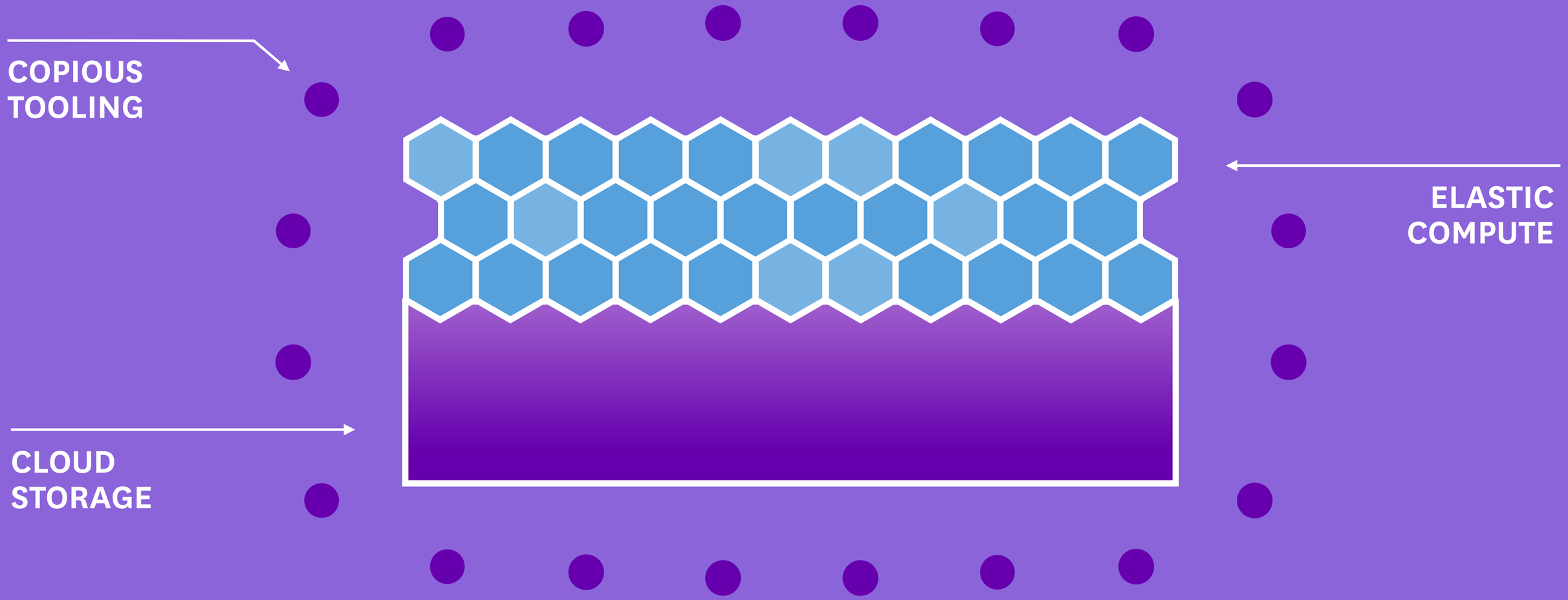
Automatic cleanup and restart



Luigi: design for failure.

Automatic cleanup and restart




COPIOUS
TOOLING
CLOUD
STORAGE
ELASTIC
COMPUTE

Recommendations for Cloud Big Data

- Use S3 for permanent data, not HDFS
- Start from EMR if building yourself
- Look into a PaaS: Netflix Genie, Qubole, Databricks
- Tag your clusters for dynamic monitoring
- Design for failure with a workflow tool (Luigi, Airflow)

Thanks!

Want to work with us on Spark, Hadoop, Kafka, Parquet, and more?
jobs.datadoghq.com

DM me @ddaniels888 or doug@datadoghq.com