The Evolution of a Data Project

The Evolution of a Data Project

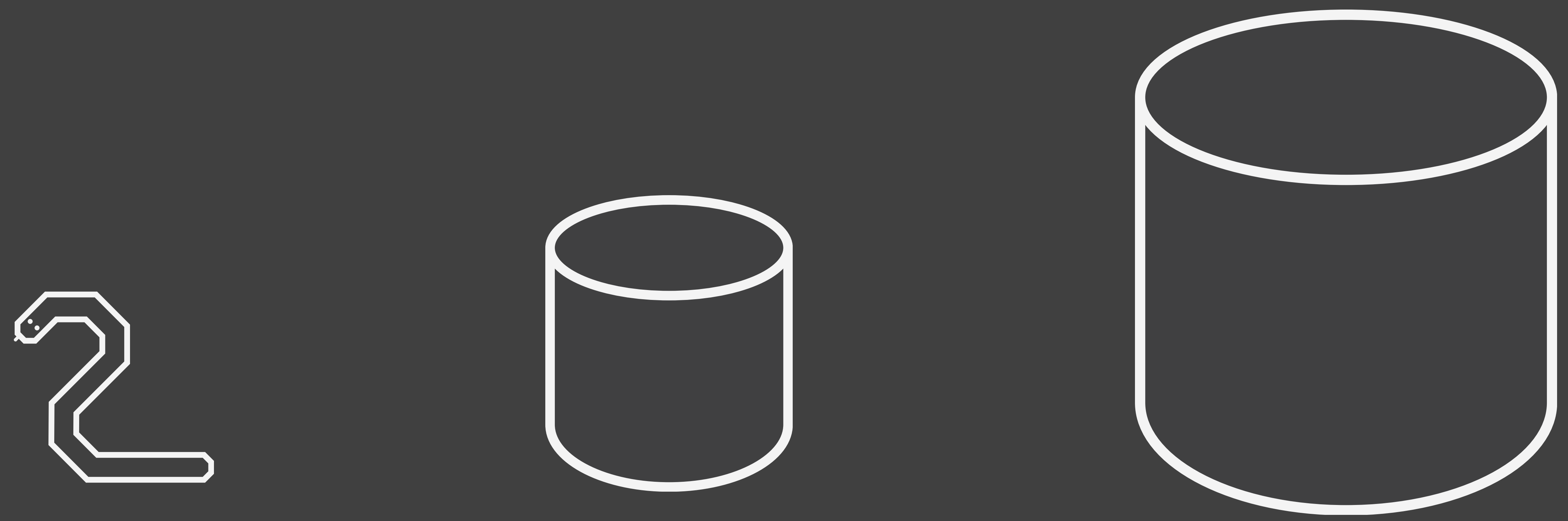
Python script

The Evolution of a Data Project

Python
script

SQL on
live DB

The Evolution of a Data Project



Python script

SQL on live DB

SQL on reporting DB

The Evolution of a Data Project

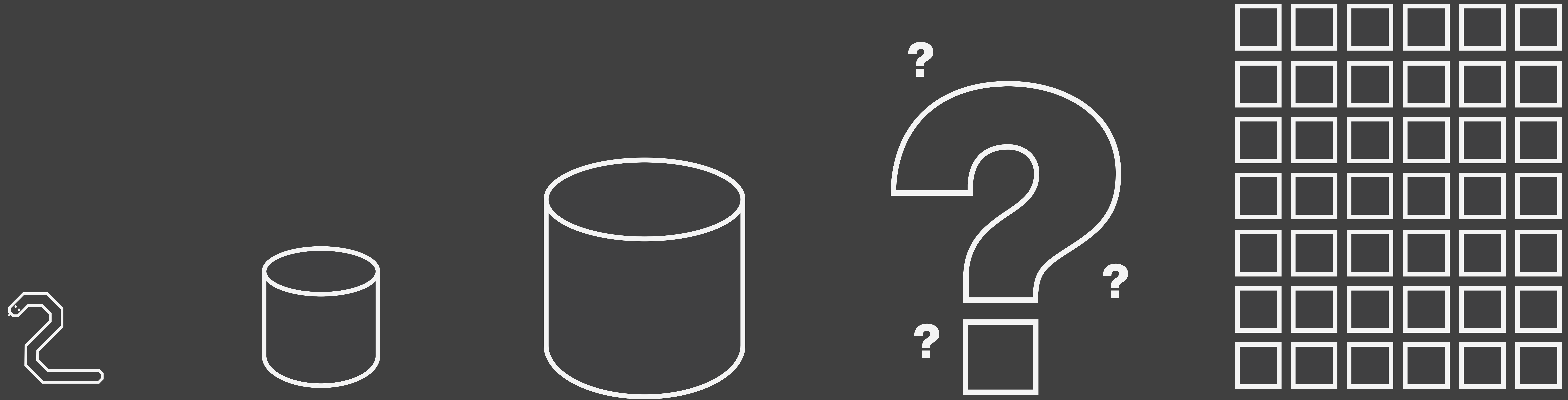
Python script

SQL on live DB

SQL on reporting DB

Terrible confusion

The Evolution of a Data Project



Python script

SQL on live DB

SQL on reporting DB

Terrible confusion

Hadoop / Spark cluster

What needs fixing

image: Pexels

What needs fixing

- One cluster: data lock-in.

image: Pexels

What needs fixing

- One cluster: data lock-in.
- Want cluster time? You have to wait.

image: Pexels

What needs fixing

- One cluster: data lock-in.
- Want cluster time? You have to wait.
- Clusters are underutilized and EXPENSIVE

image: Pexels

Elastic Big Data Platform @ Datadog

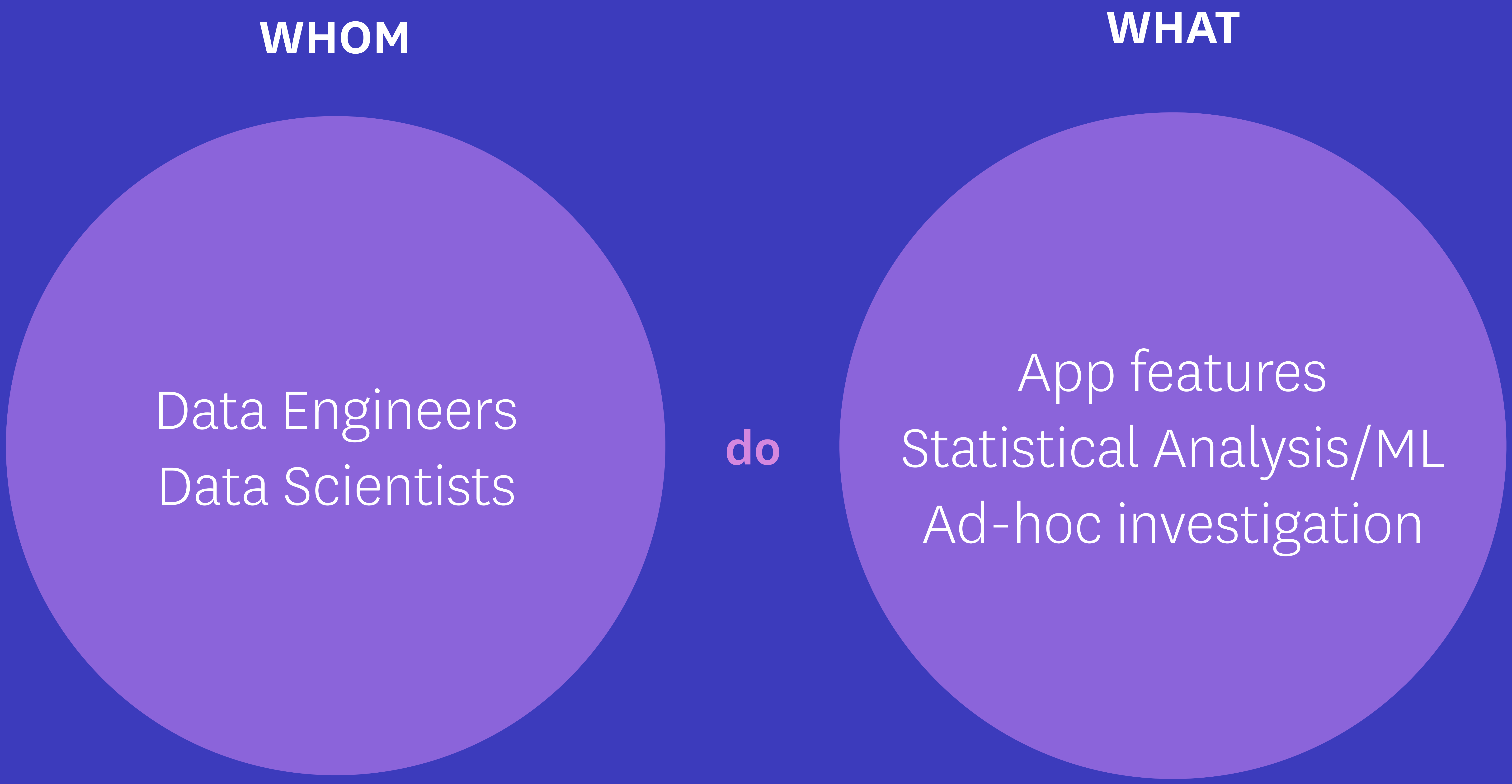
Doug Daniels
Director, Engineering

What’s our big data platform do?

WHOM

Data Engineers
Data Scientists

What’s our big data platform do?



WHOM

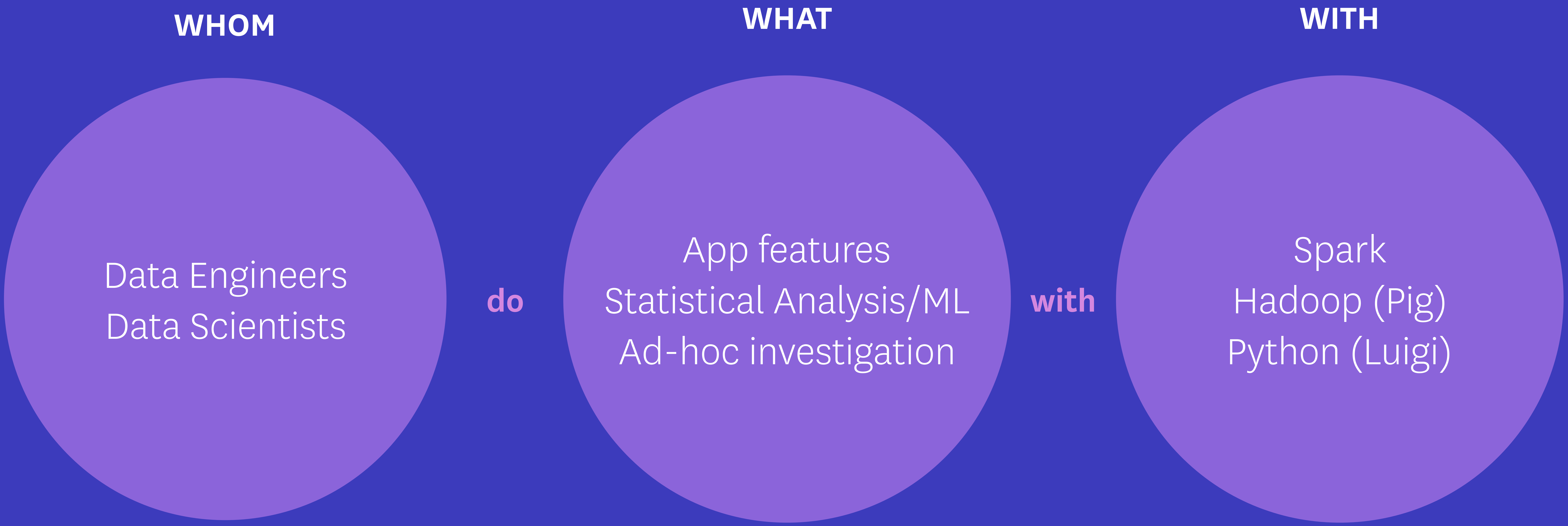
Data Engineers
Data Scientists

do

WHAT

App features
Statistical Analysis/ML
Ad-hoc investigation

What’s our big data platform do?



WHOM

Data Engineers
Data Scientists

do

WHAT

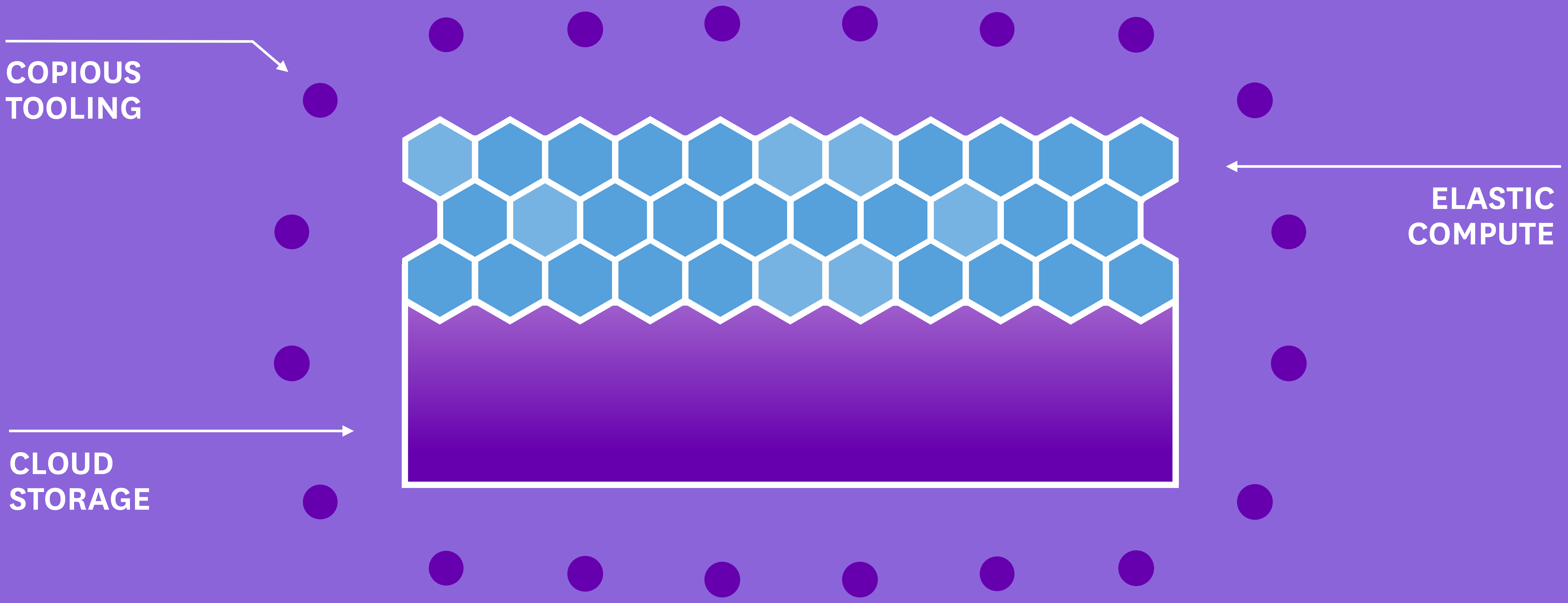
App features
Statistical Analysis/ML
Ad-hoc investigation

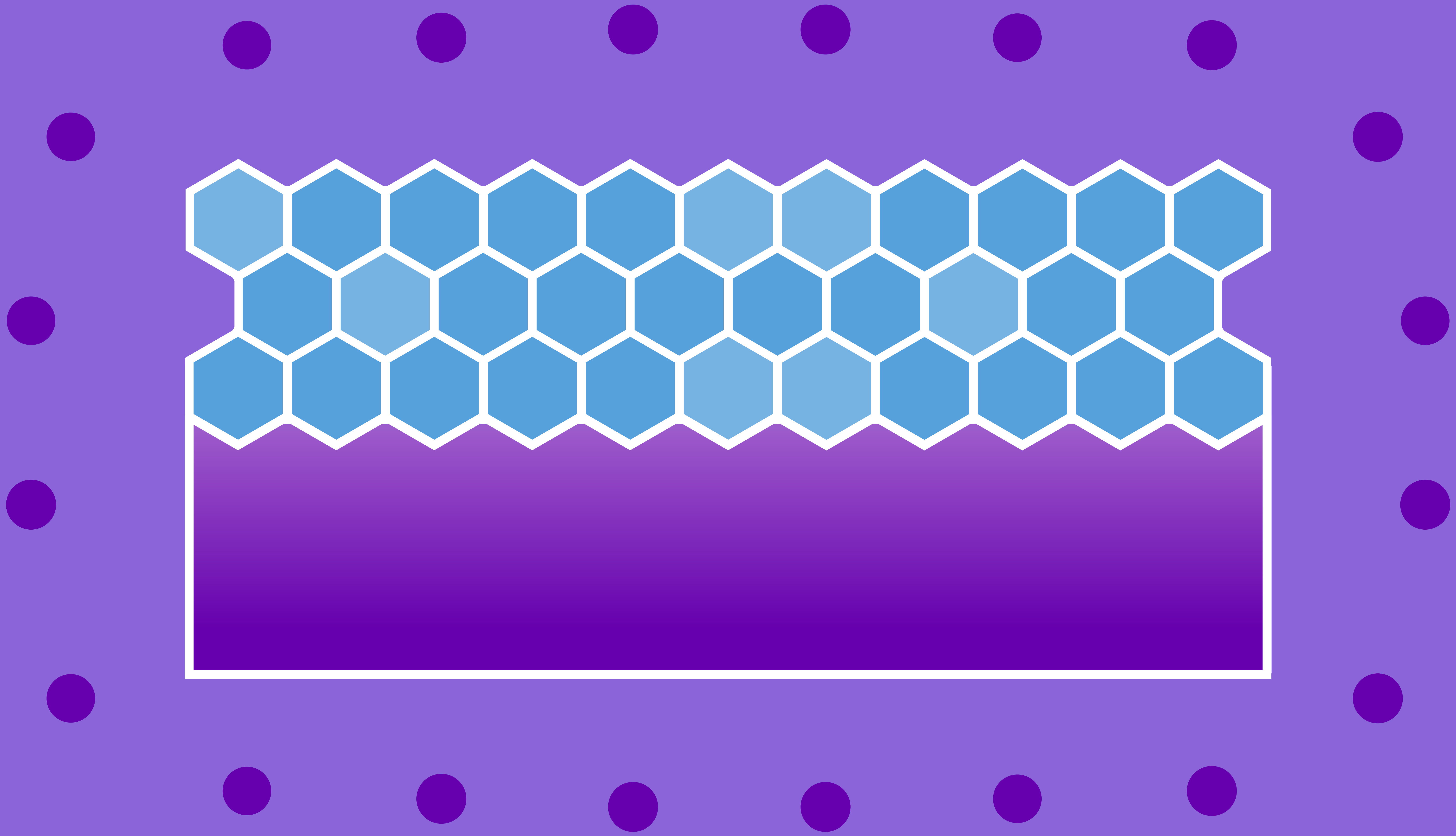
with

WITH

Spark
Hadoop (Pig)
Python (Luigi)

Exploring the platform

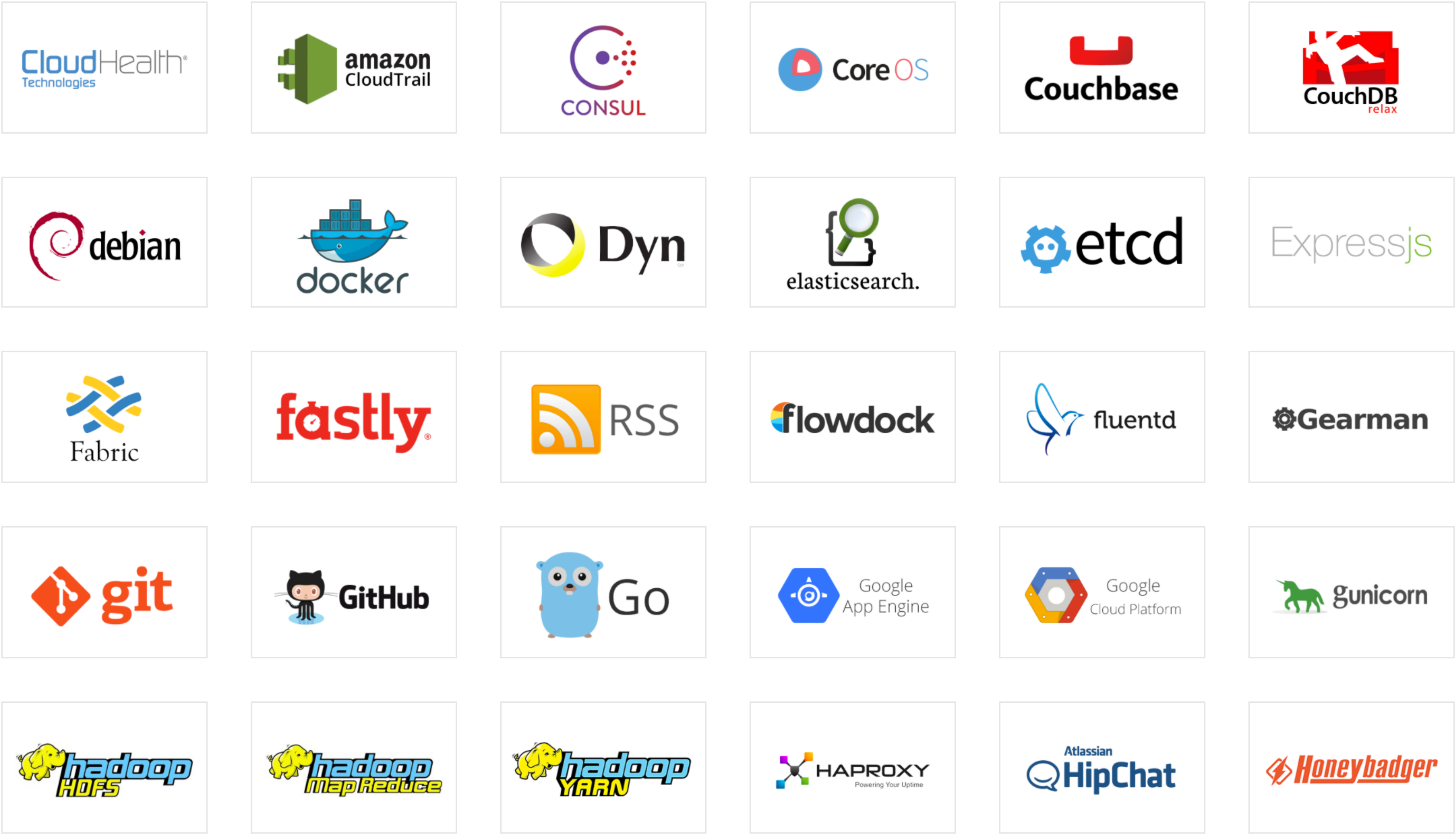




CLOUD STORAGE

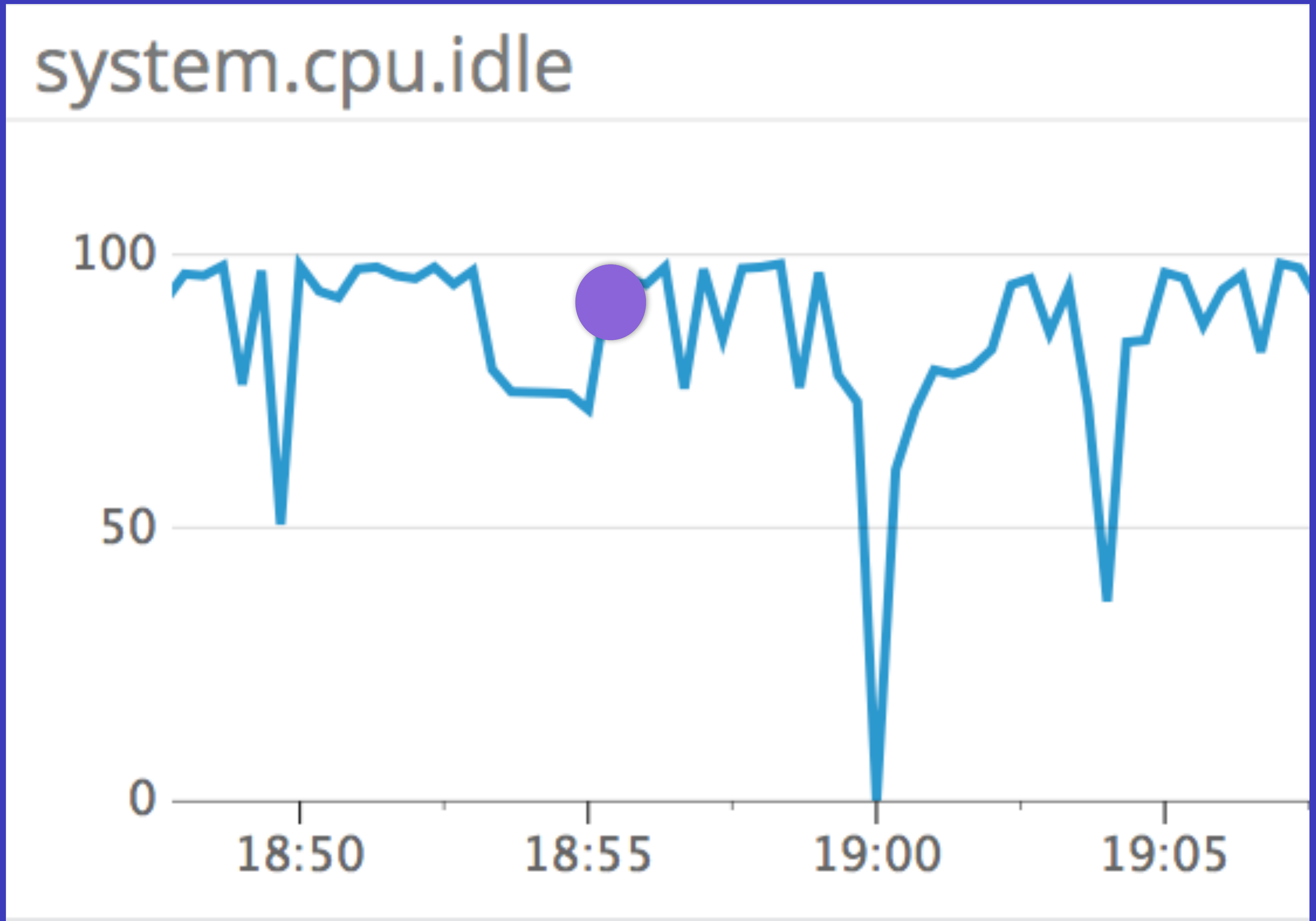
What do we store?

150 Integrations



...and more

What’s time series data?

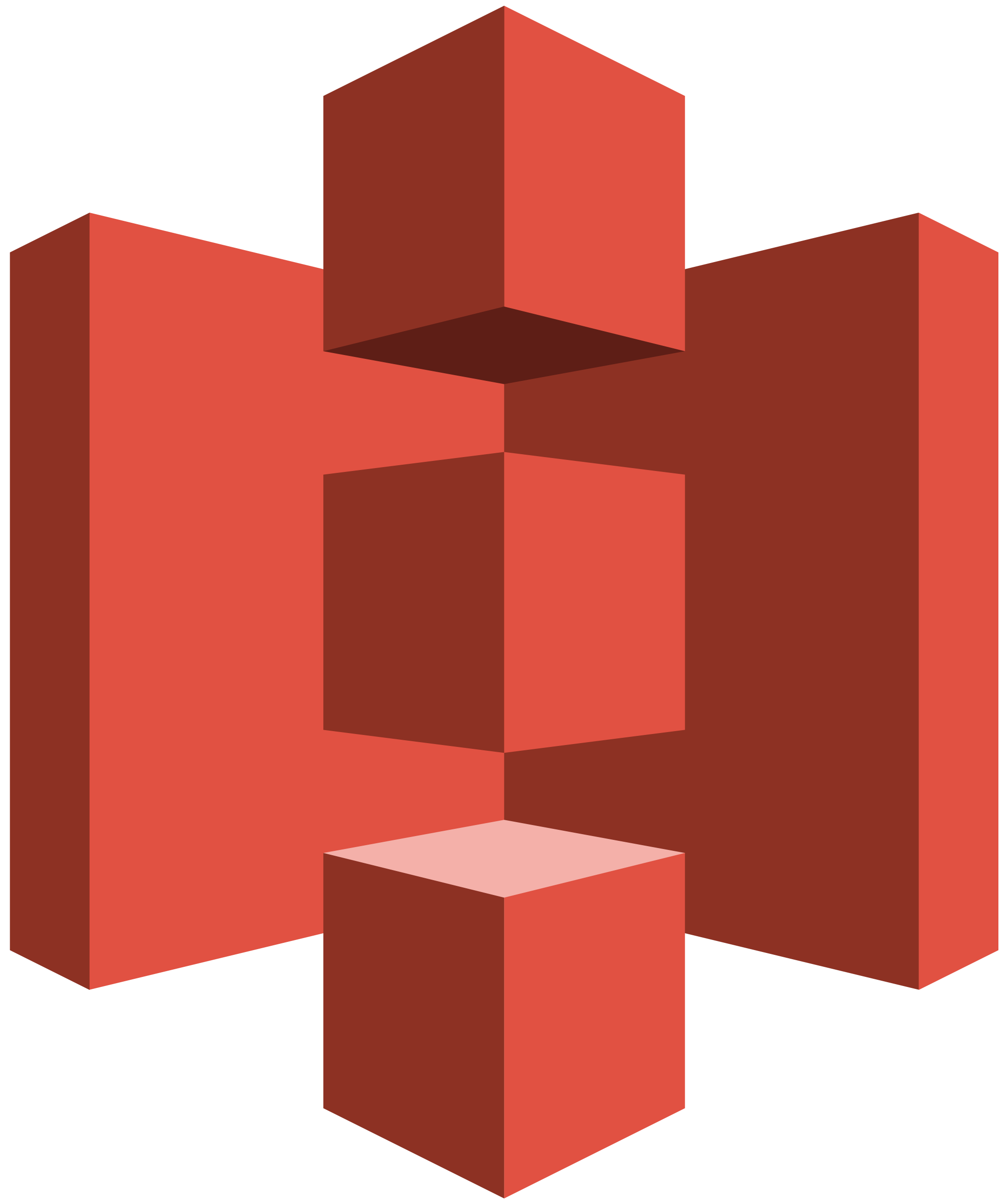


timestamp	1447020511
metric	system.cpu.idle
value	98.16687
tags	host:i-xyz, role:cassandra, …

We collect over a trillion of these per day

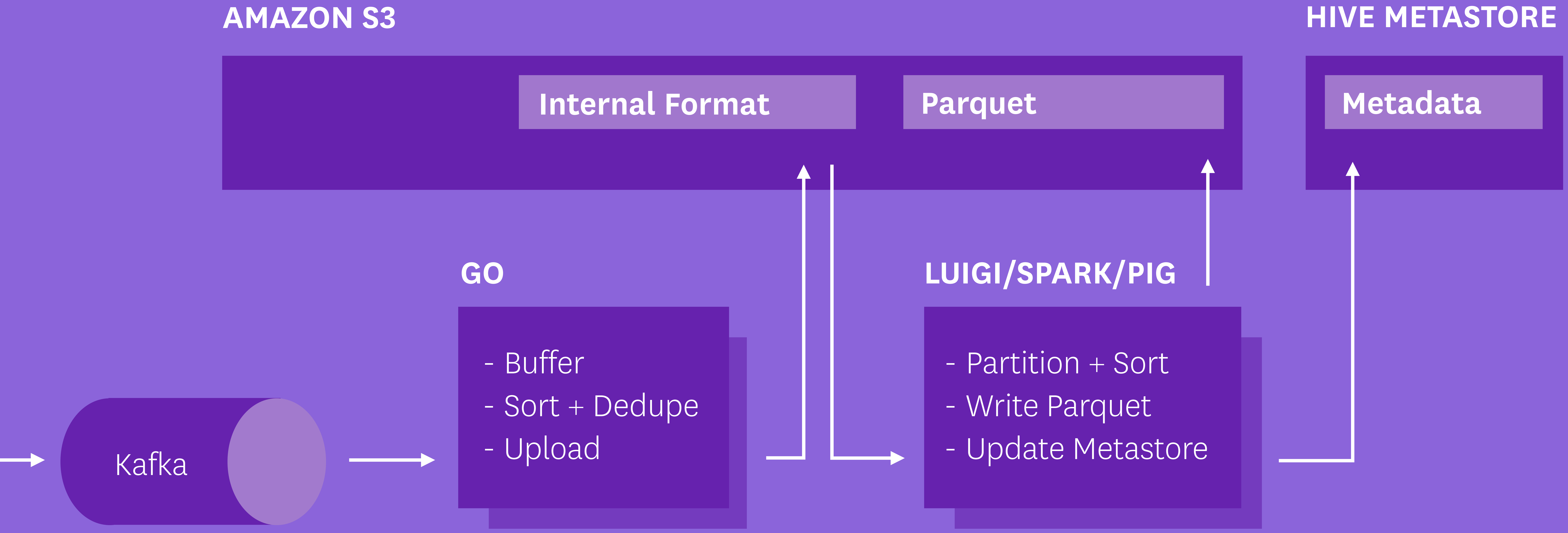
…and growing!

Where to put the petabytes?



Amazon S3

How data gets to S3



AMAZON S3

Internal Format

Parquet

HIVE METASTORE

Metadata

GO

- Buffer
- Sort + Dedupe
- Upload

LUIGI/SPARK/PIG

- Partition + Sort
- Write Parquet
- Update Metastore

Isn’t this a job for HDFS?

What we don’t love about HDFS

What we don’t love about HDFS

- Causes the “one cluster” problem

What we don’t love about HDFS

- Causes the “one cluster” problem
- Come for the storage, get stuck with the servers

What we don’t love about HDFS

- Causes 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

S3 is flexible!

- Read data from as many clusters as you want
- Store unlimited stuff(*) with no management

* Accepting laws of physics and your credit card limit

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)

* Accepting laws of physics and your credit card limit

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

Breaking the rules is fine.

In benchmarks: S3 is ~2X slower than HDFS

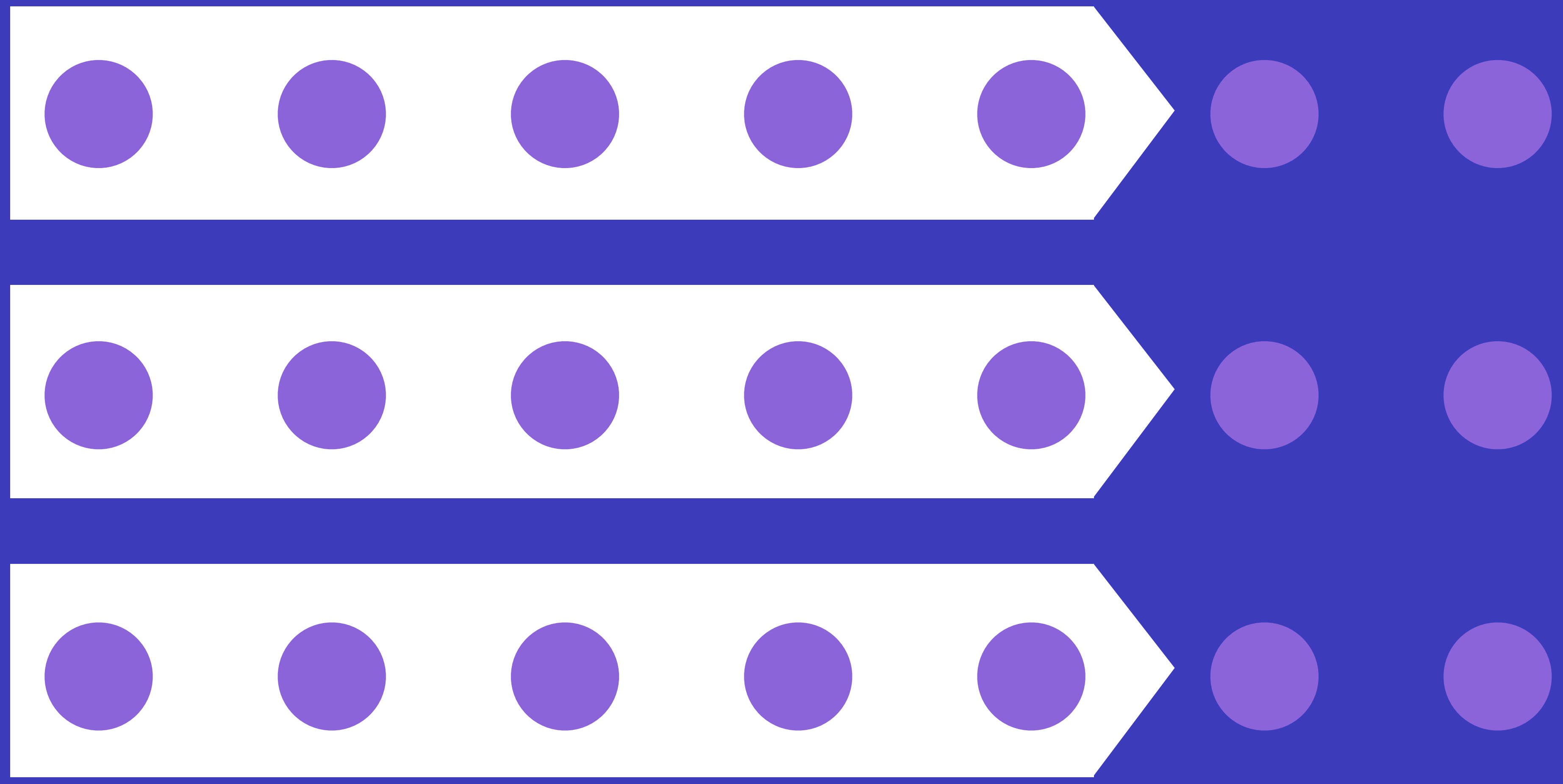
It’s not *all* roses

Listing is slooooow

(A CAUTIONARY TALE)

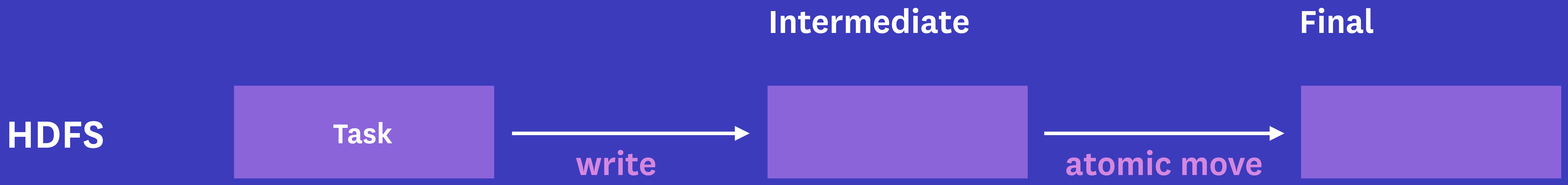
How to fix slow listing

Parallelize it



Bigger files

No way to quickly move data



HDFS

Task

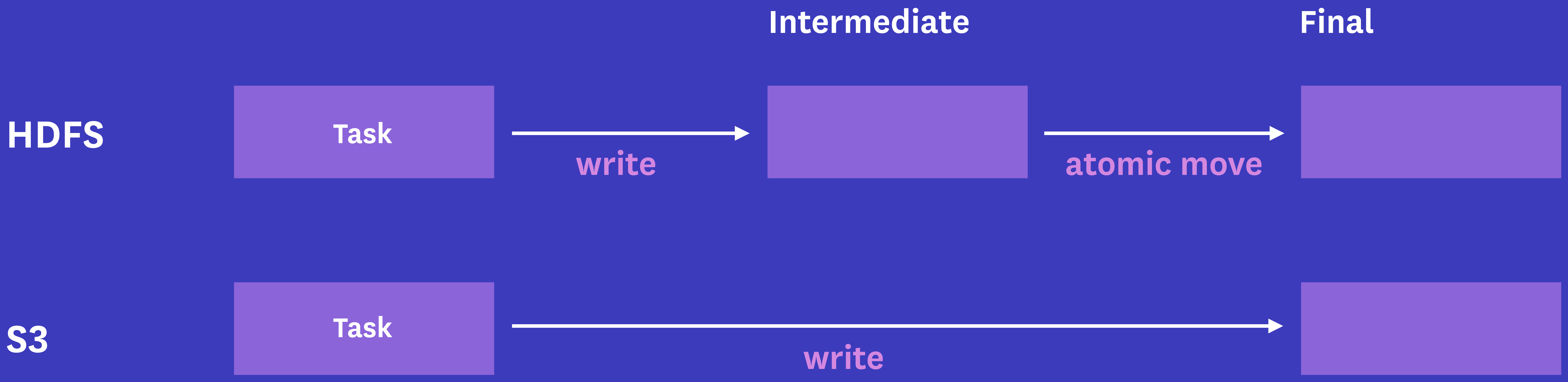
write

Intermediate

atomic move

Final

No way to quickly move data



Intermediate

Final

HDFS

Task

write

atomic move

S3

Task

write

No way to quickly move data

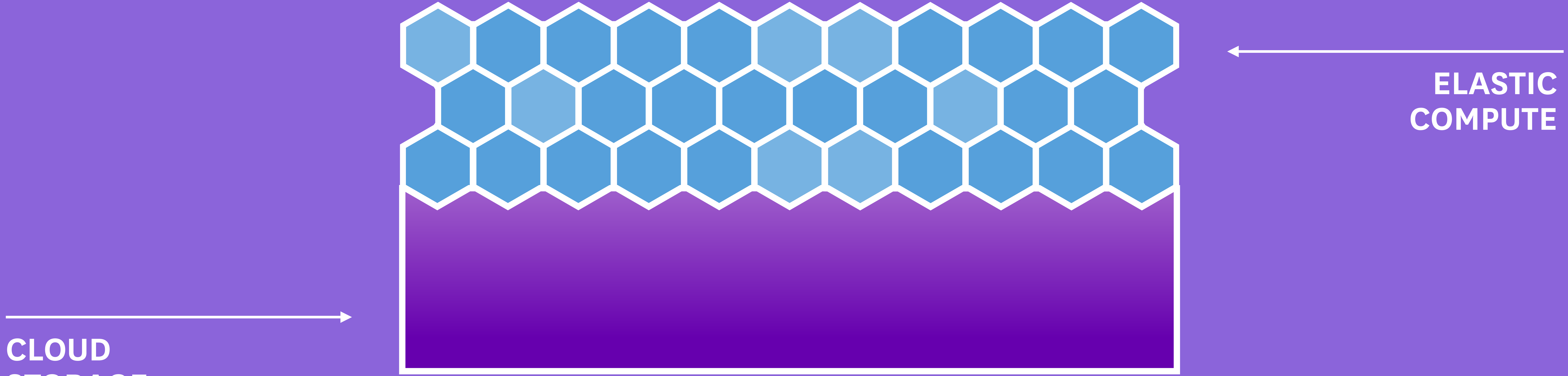
- Say goodbye to speculative execution

No way to quickly move data

- Say goodbye to speculative execution
- Say hello to better task timeouts

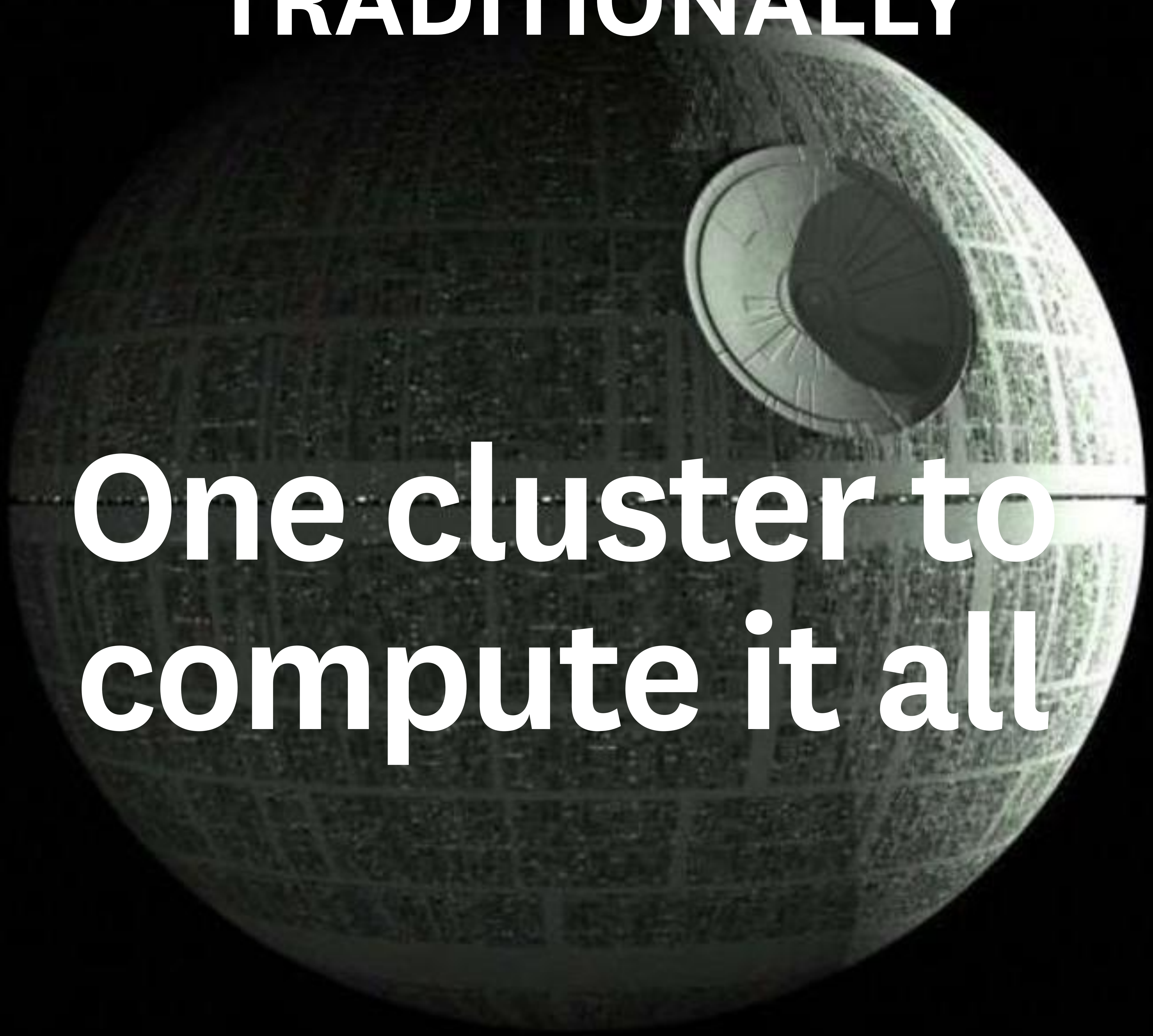
But really: We💜S3
This is a great system.

- ✓ Data accessible from many clusters
- ✓ Storage is easy to manage
- ✓ It’s a multi-language paradise up in here



ELASTIC
COMPUTE

CLOUD
STORAGE

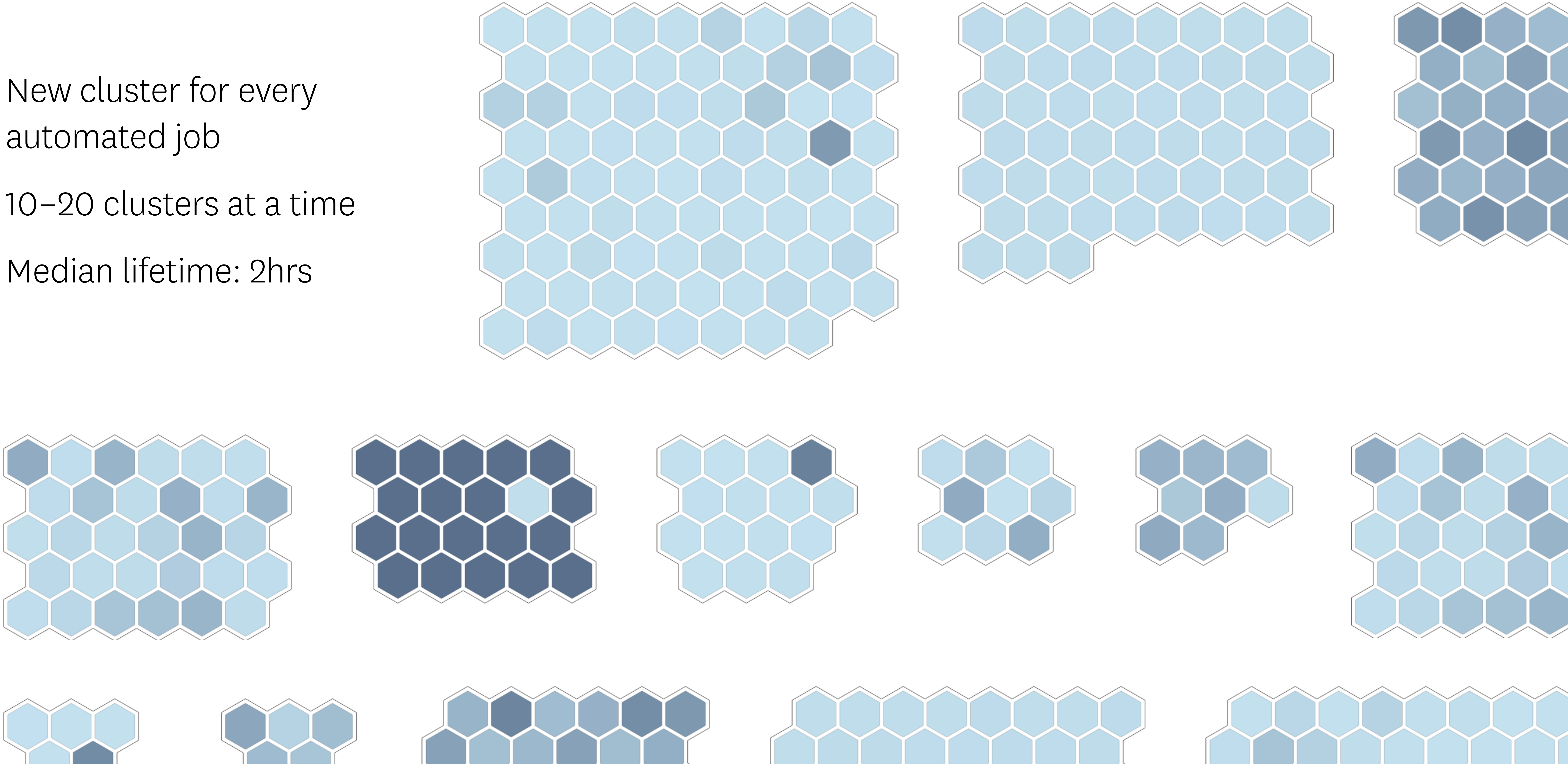


TRADITIONALLY

One cluster to compute it all

Instead, we run many, many clusters

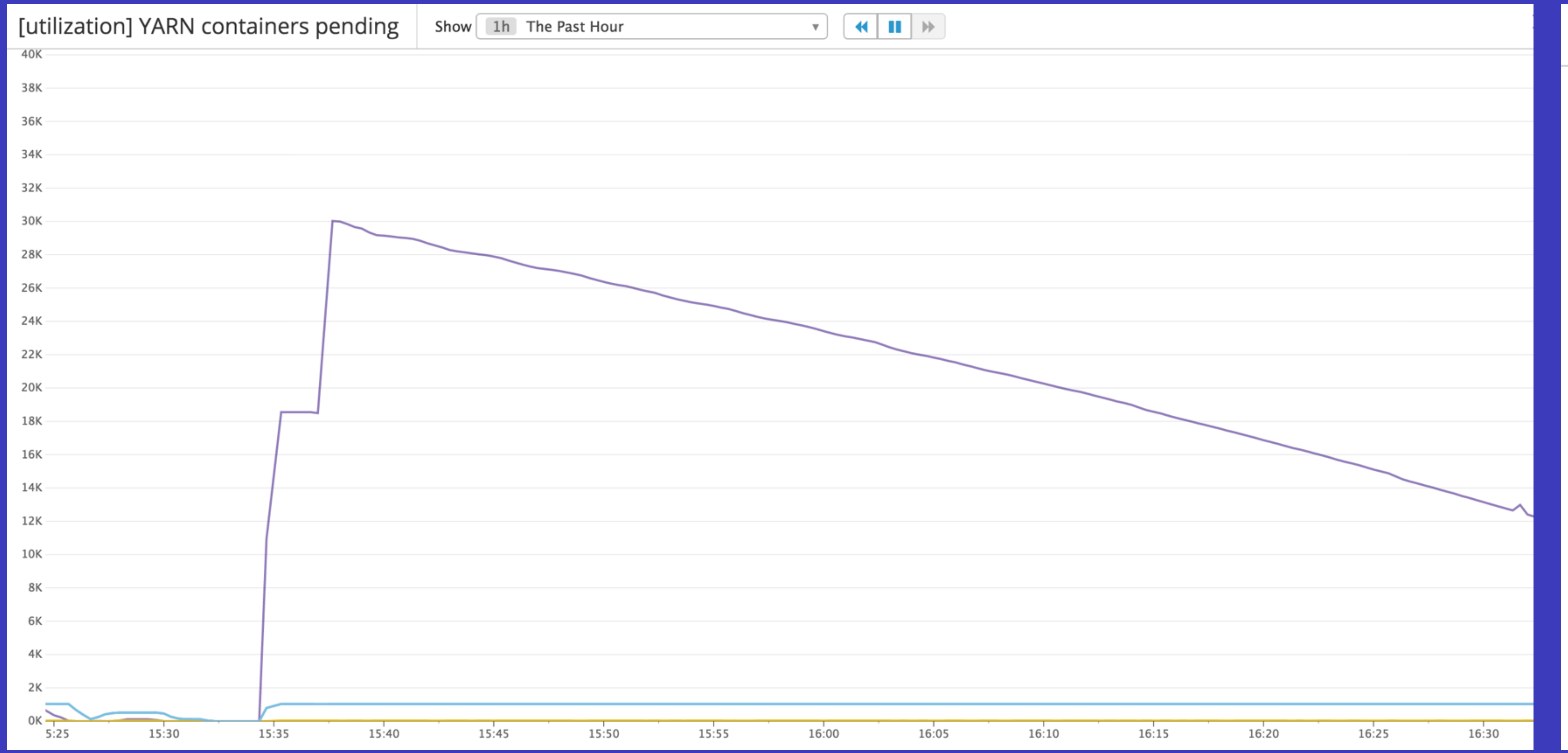
- New cluster for every automated job
- 10–20 clusters at a time
- Median lifetime: 2hrs

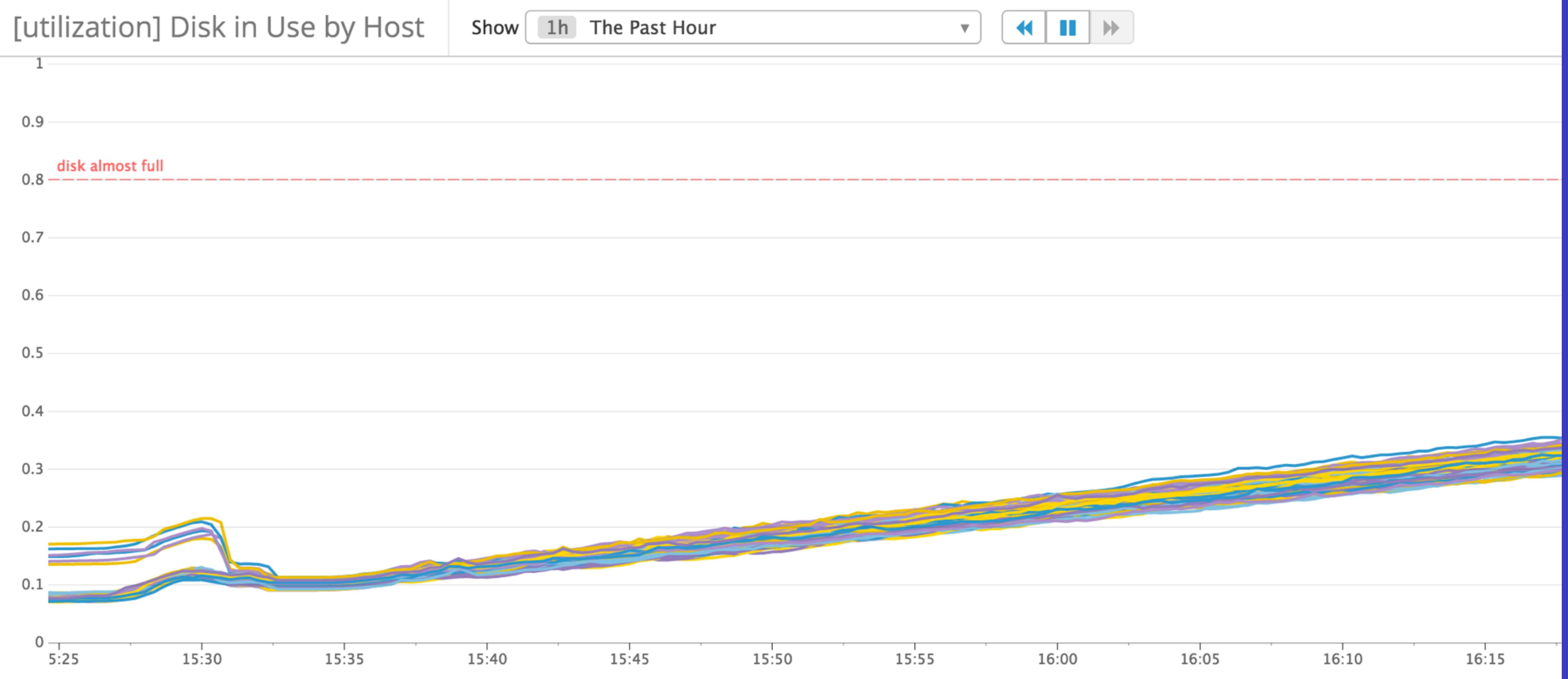


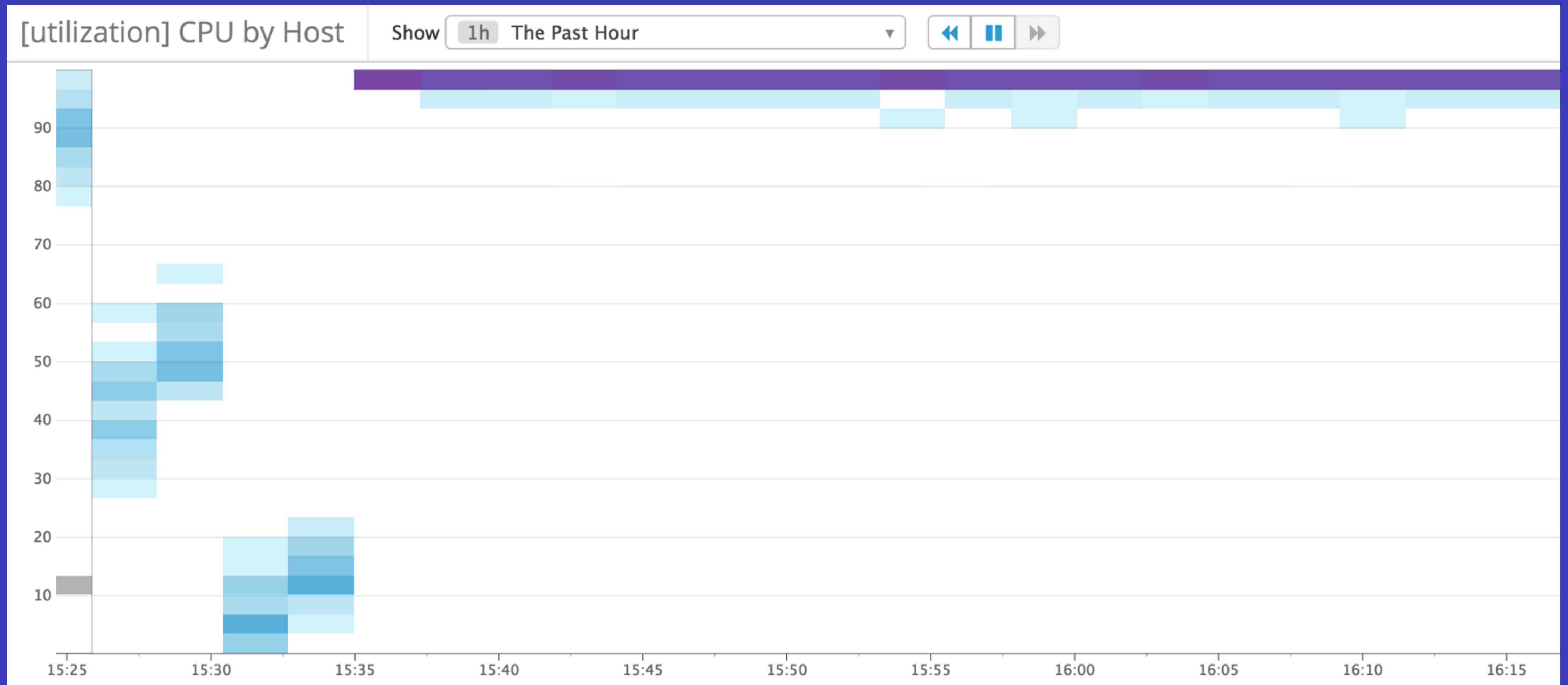
Why so many clusters?

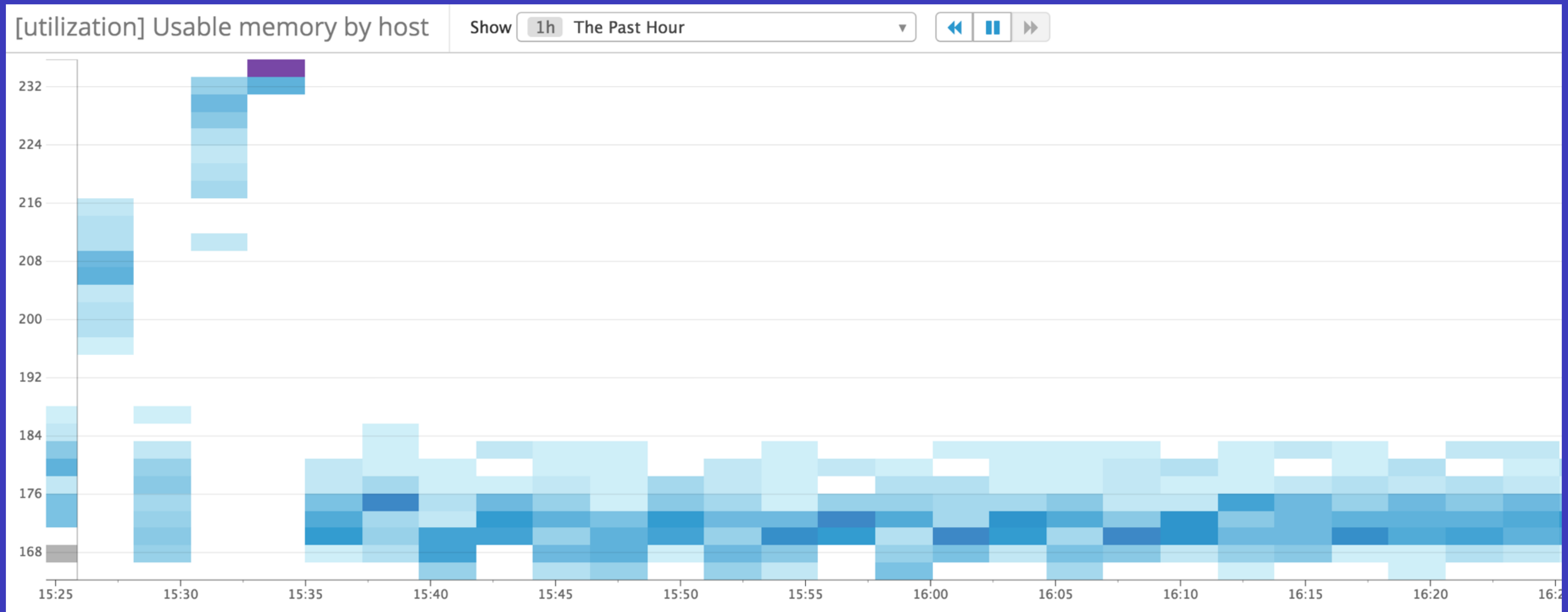
Total isolation

We know what’s happening and why


[utilization] YARN containers pending


[utilization] Disk in Use by Host


[utilization] CPU by Host

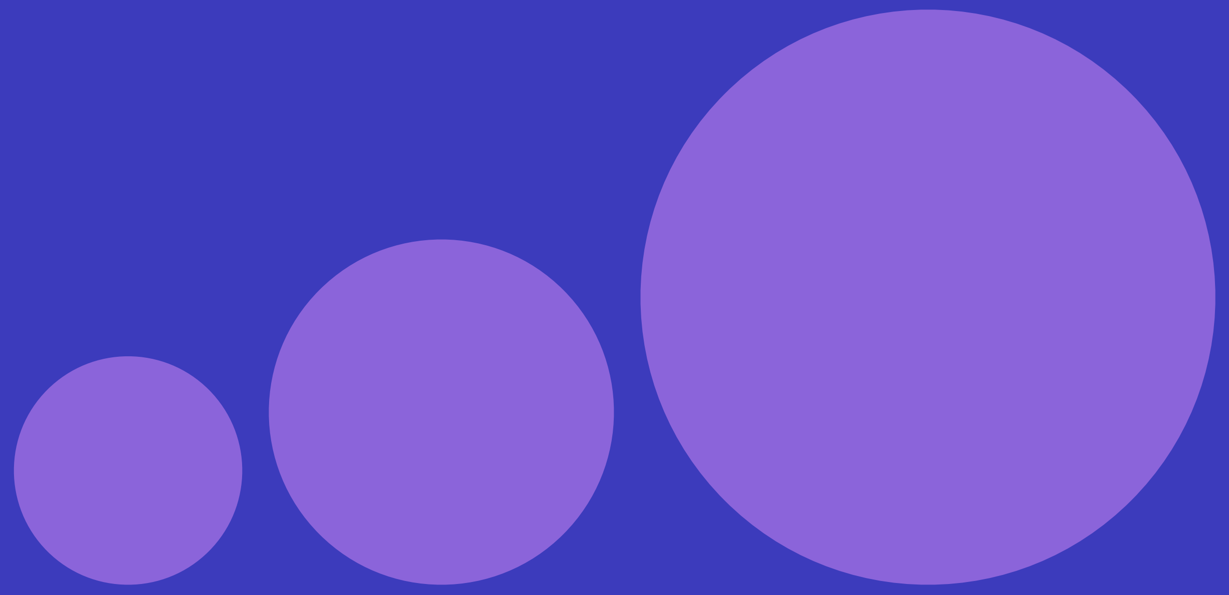

[utilization] Usable memory by host

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

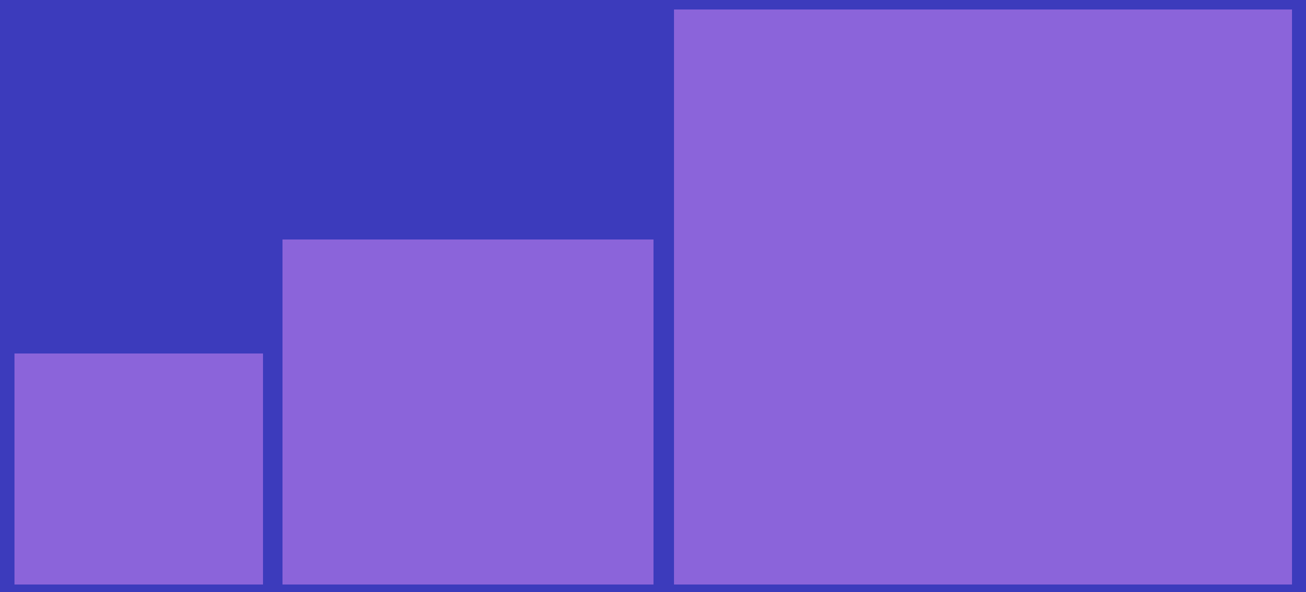
Pick the best hardware for each job

== ~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)

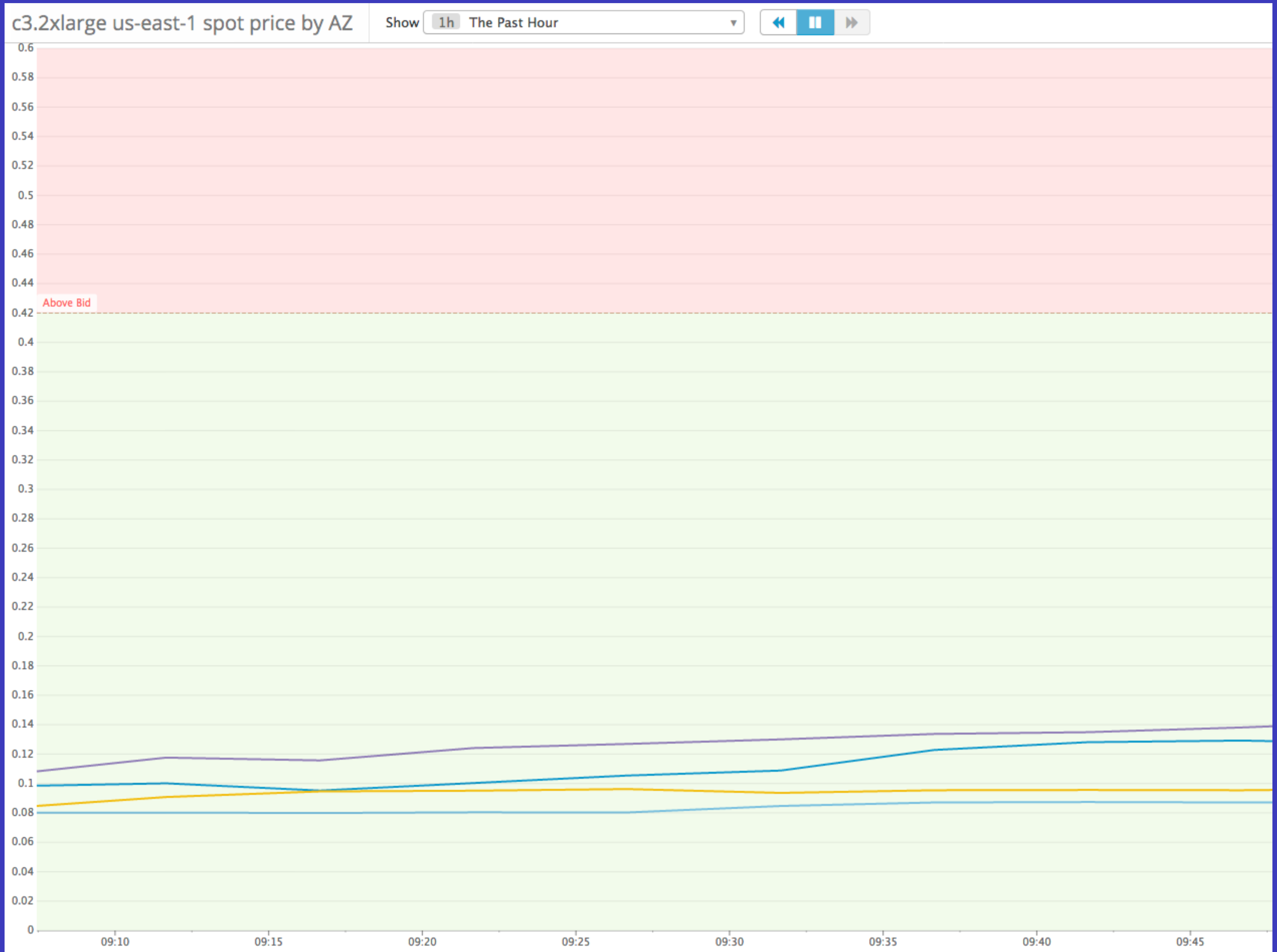
100% spot-instance clusters, all the time.*

* (ok, most of the time)

How we do spot clusters

In the big data platform

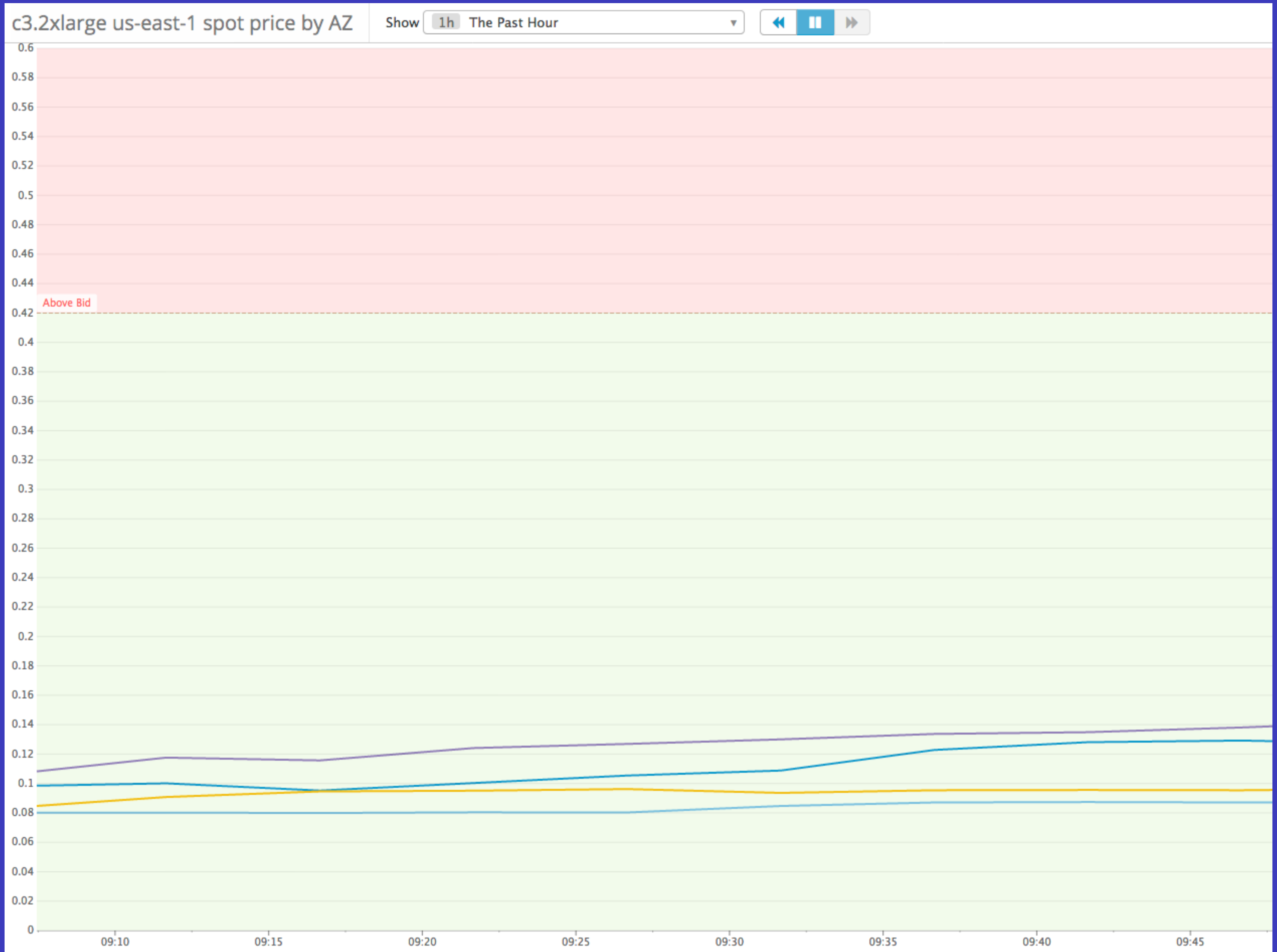
- Bid the on-demand price, pay the spot price



How we do spot clusters

In the big data platform

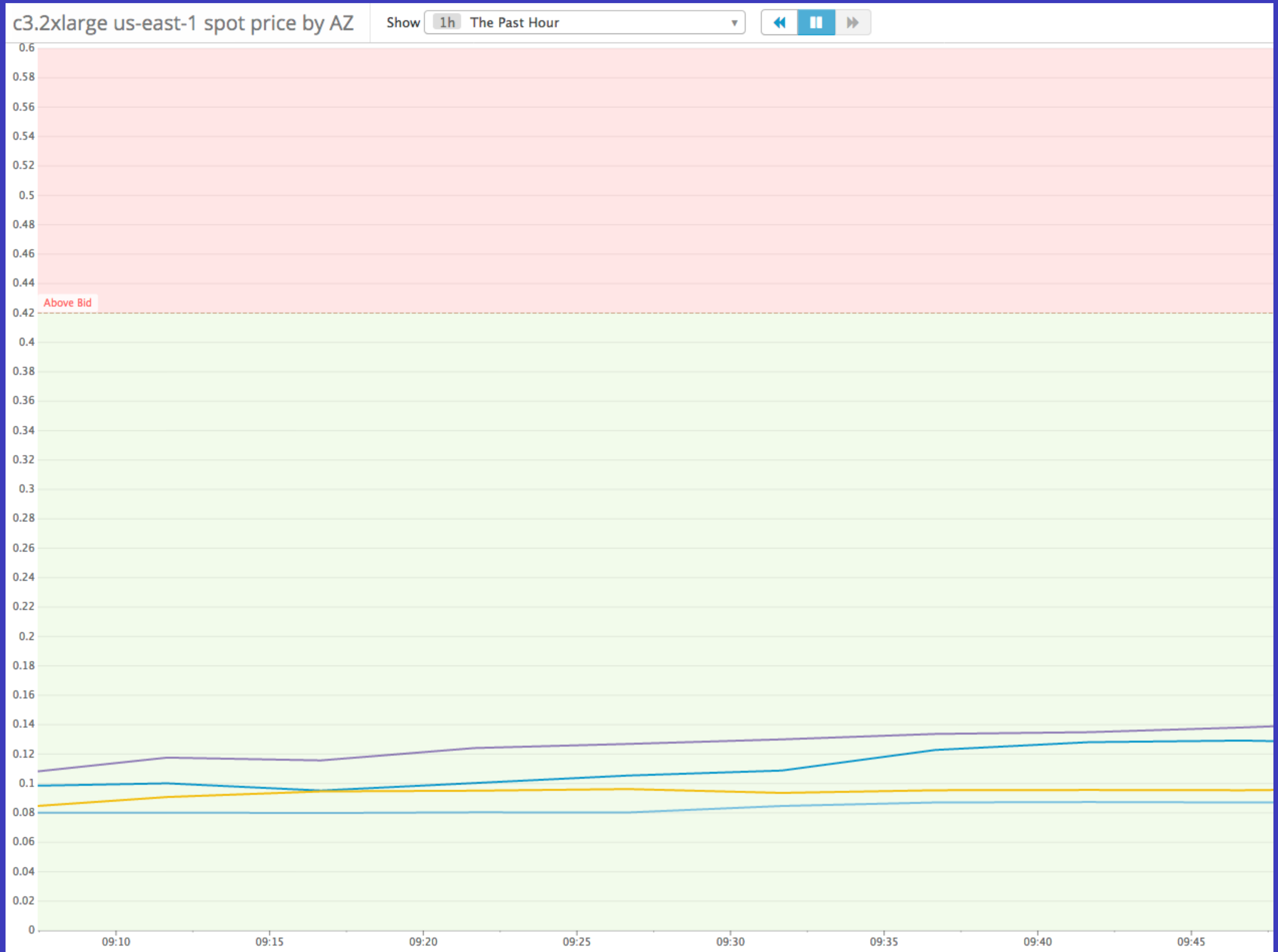
- Bid the on-demand price, pay the spot price
- Fallback to on-demand instances if you can’t get spot



How we do spot clusters

In the big data platform

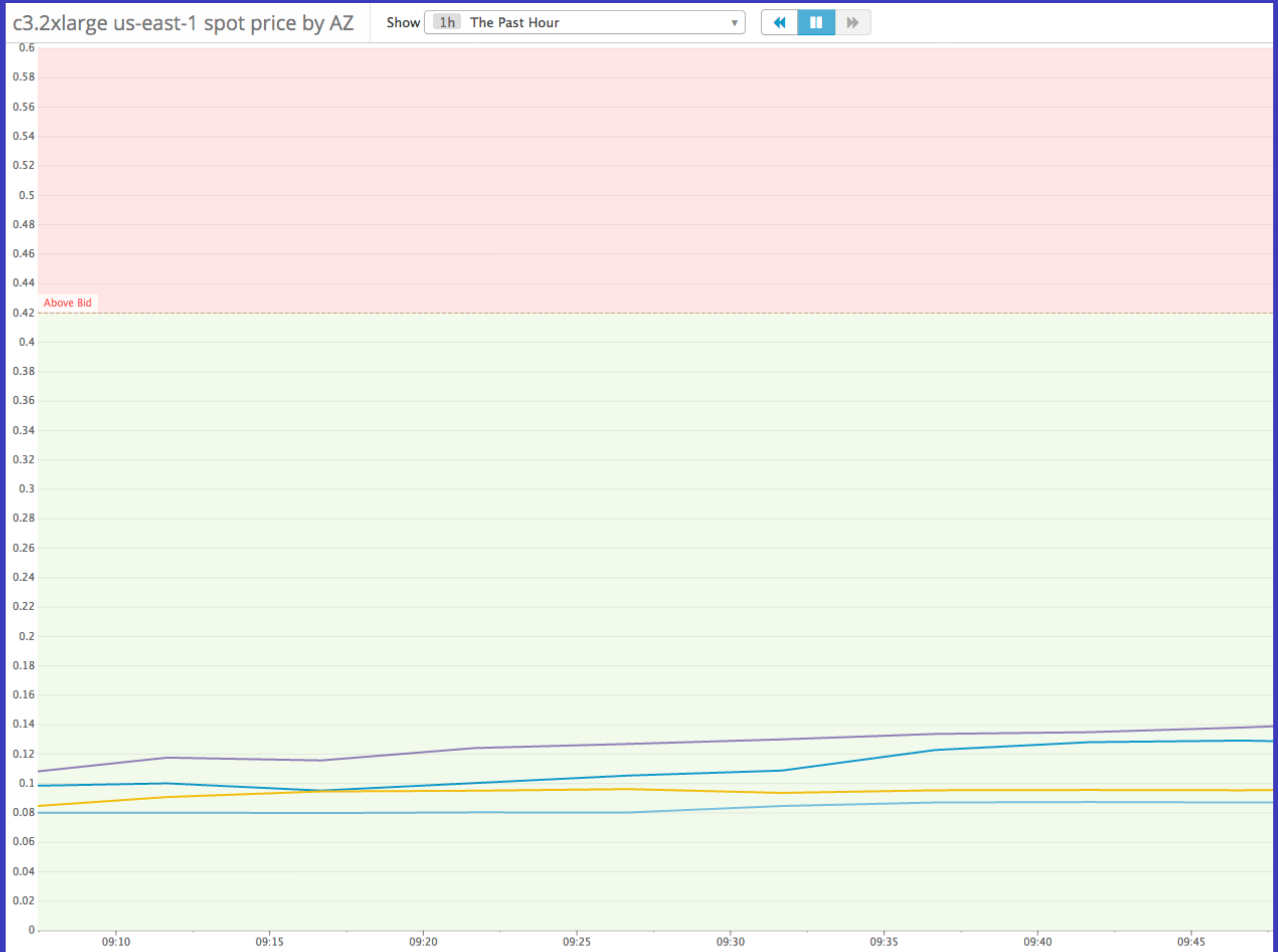
- Bid the on-demand price, pay the spot price
- Fallback to on-demand instances if you can’t get spot
- Monitor everything: jobs, clusters, spot market



How we do spot clusters

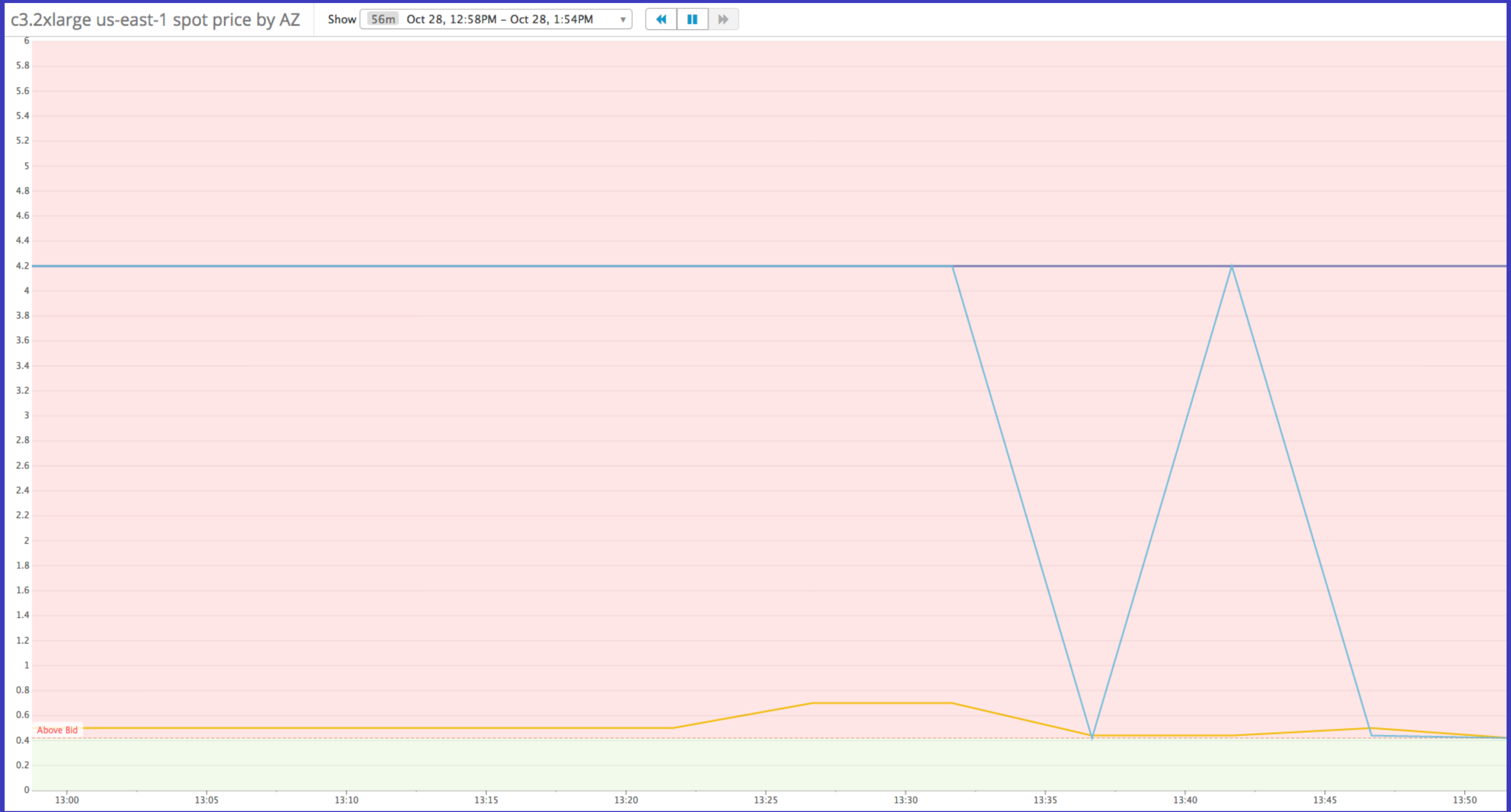
In the big data platform

- 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

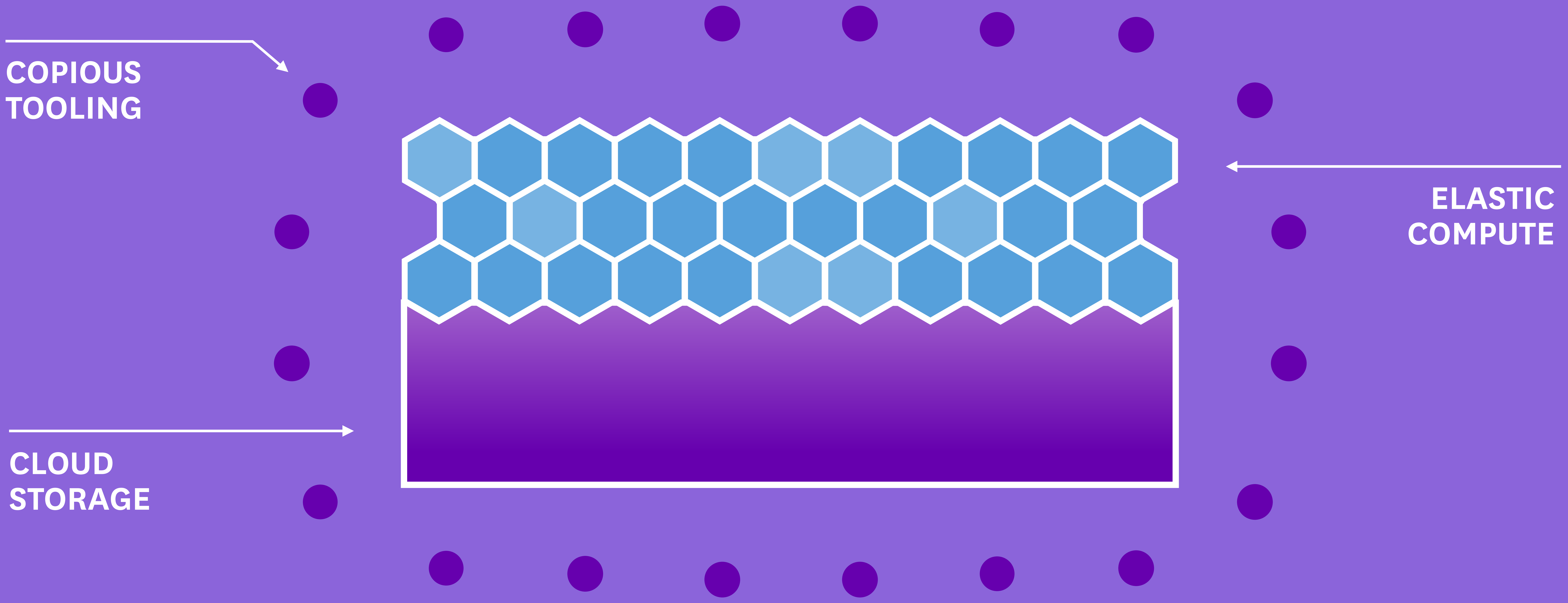


We like this strategy a lot!

- ✓ 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?

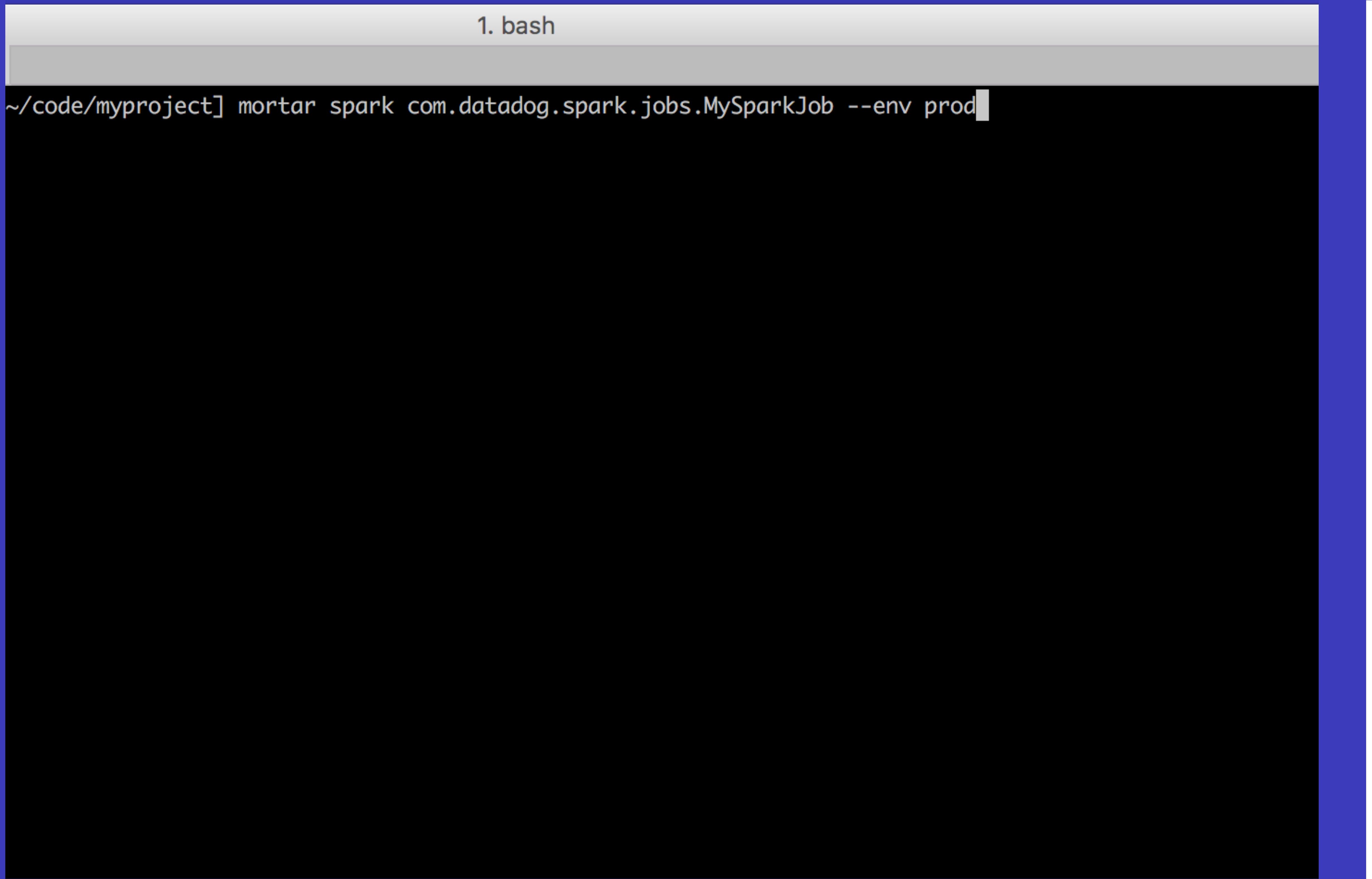
Many things that disappear.


COPIOUS TOOLING
ELASTIC COMPUTE
CLOUD STORAGE

Platform as a service

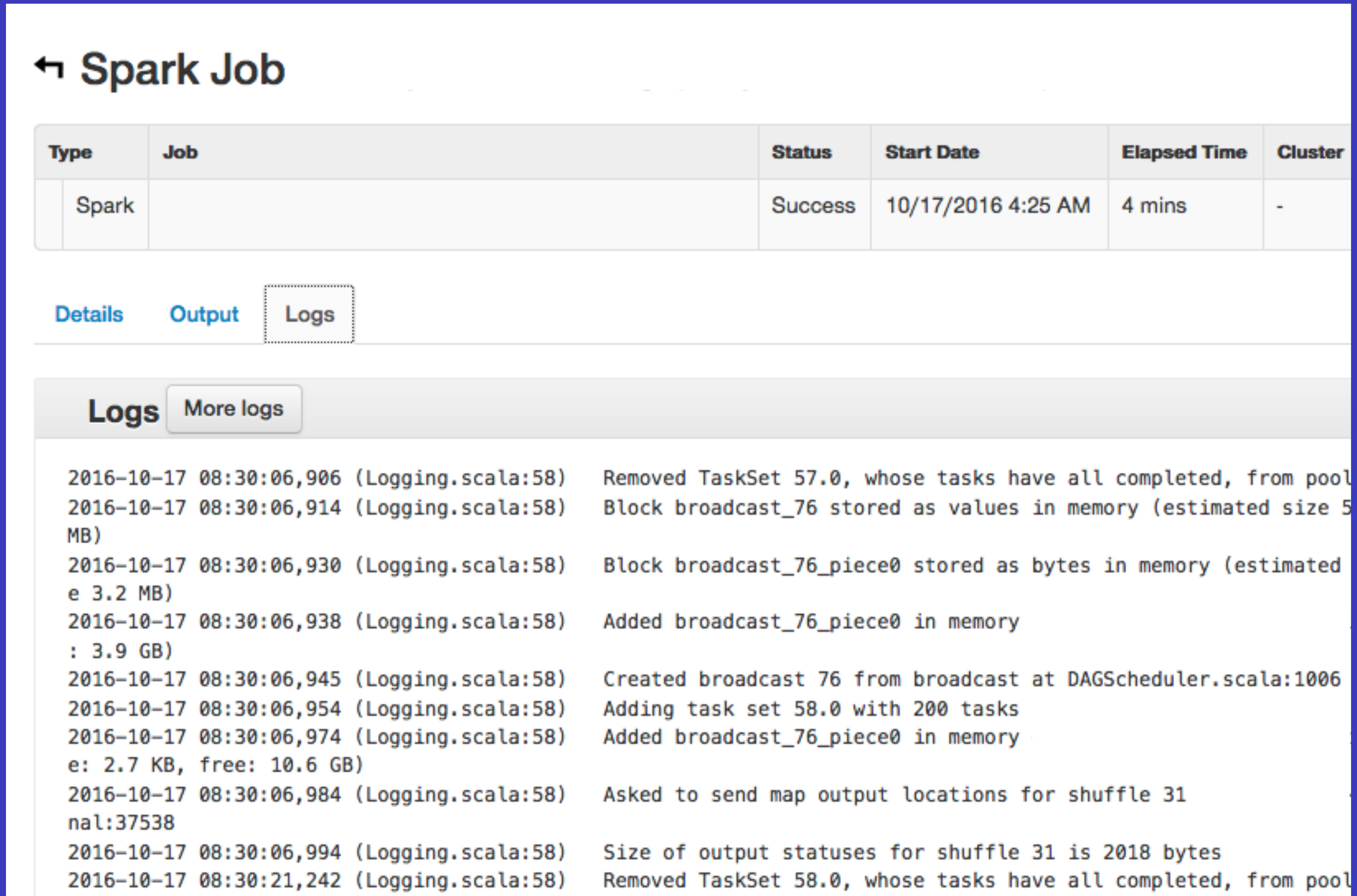
Jobs, Clusters, Schedules, Users, Code, Monitoring, Logs, and more

CLI



```
~/code/myproject] mortar spark com.datadog.spark.jobs.MySparkJob --env prod
```

Web and APIs



Type	Job	Status	Start Date	Elapsed Time	Cluster
Spark		Success	10/17/2016 4:25 AM	4 mins	-

Logs

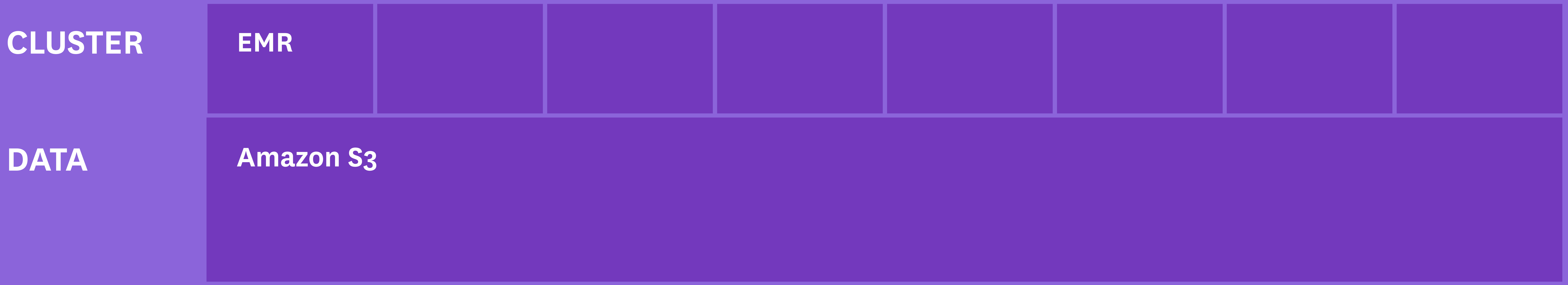
```
2016-10-17 08:30:06,906 (Logging.scala:58)   Removed TaskSet 57.0, whose tasks have all completed, from pool
2016-10-17 08:30:06,914 (Logging.scala:58)   Block broadcast_76 stored as values in memory (estimated size 5
MB)
2016-10-17 08:30:06,930 (Logging.scala:58)   Block broadcast_76_piece0 stored as bytes in memory (estimated
e 3.2 MB)
2016-10-17 08:30:06,938 (Logging.scala:58)   Added broadcast_76_piece0 in memory
: 3.9 GB)
2016-10-17 08:30:06,945 (Logging.scala:58)   Created broadcast 76 from broadcast at DAGScheduler.scala:1006
2016-10-17 08:30:06,954 (Logging.scala:58)   Adding task set 58.0 with 200 tasks
2016-10-17 08:30:06,974 (Logging.scala:58)   Added broadcast_76_piece0 in memory
e: 2.7 KB, free: 10.6 GB)
2016-10-17 08:30:06,984 (Logging.scala:58)   Asked to send map output locations for shuffle 31
nal:37538
2016-10-17 08:30:06,994 (Logging.scala:58)   Size of output statuses for shuffle 31 is 2018 bytes
2016-10-17 08:30:21,242 (Logging.scala:58)   Removed TaskSet 58.0, whose tasks have all completed, from pool
```

Big Data Platform Architecture

DATA

Amazon S3

Big Data Platform Architecture



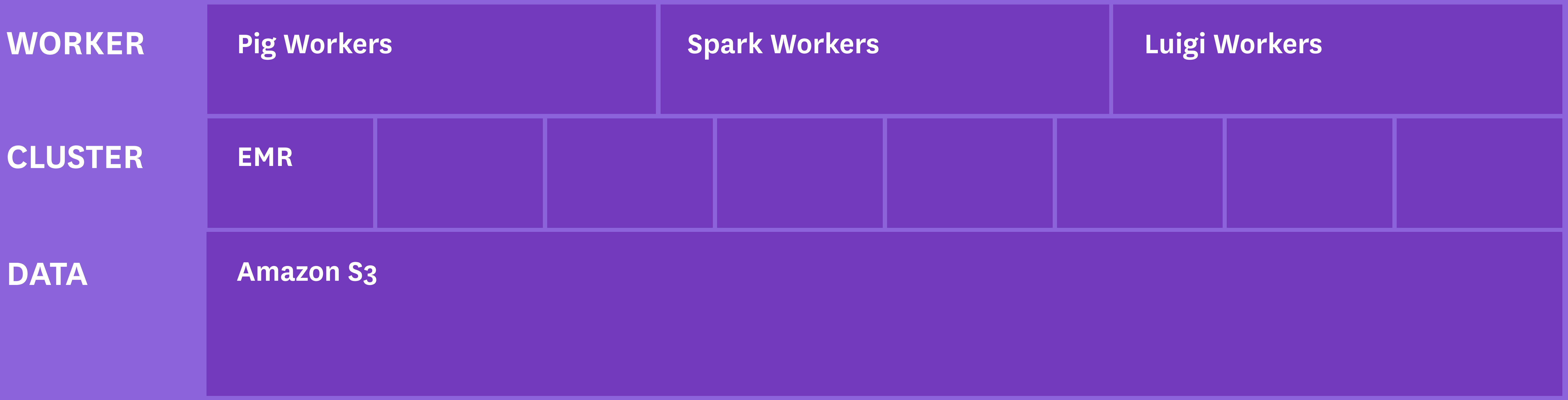
CLUSTER

EMR

DATA

Amazon S3

Big Data Platform Architecture



WORKER

Pig Workers

Spark Workers

Luigi Workers

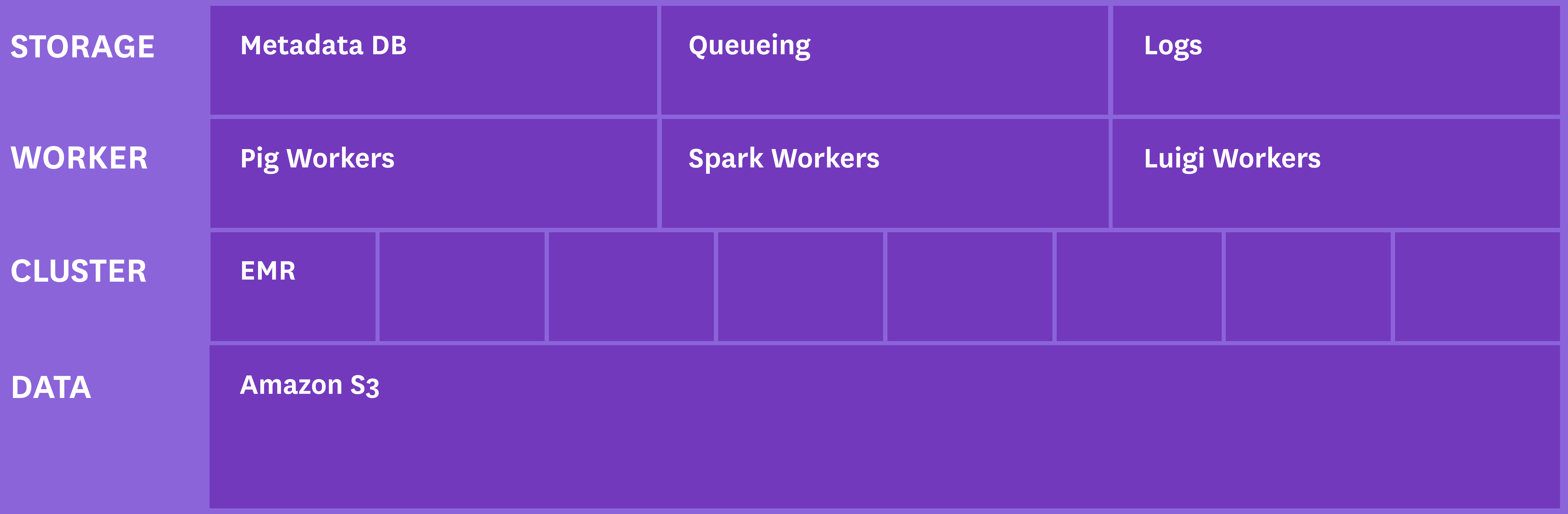
CLUSTER

EMR

DATA

Amazon S3

Big Data Platform Architecture



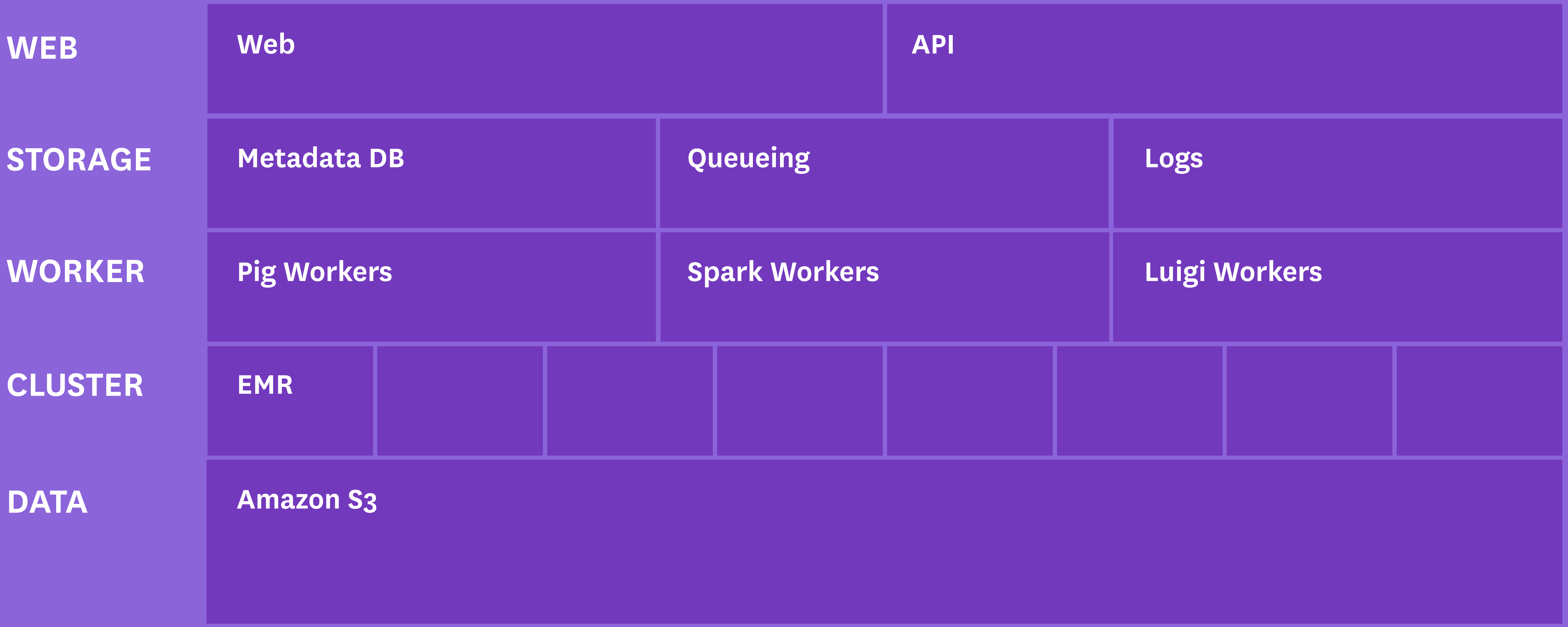
STORAGE: Metadata DB | Queueing | Logs

WORKER: Pig Workers | Spark Workers | Luigi Workers

CLUSTER: EMR

DATA: Amazon S3

Big Data Platform Architecture



WEB: Web, API

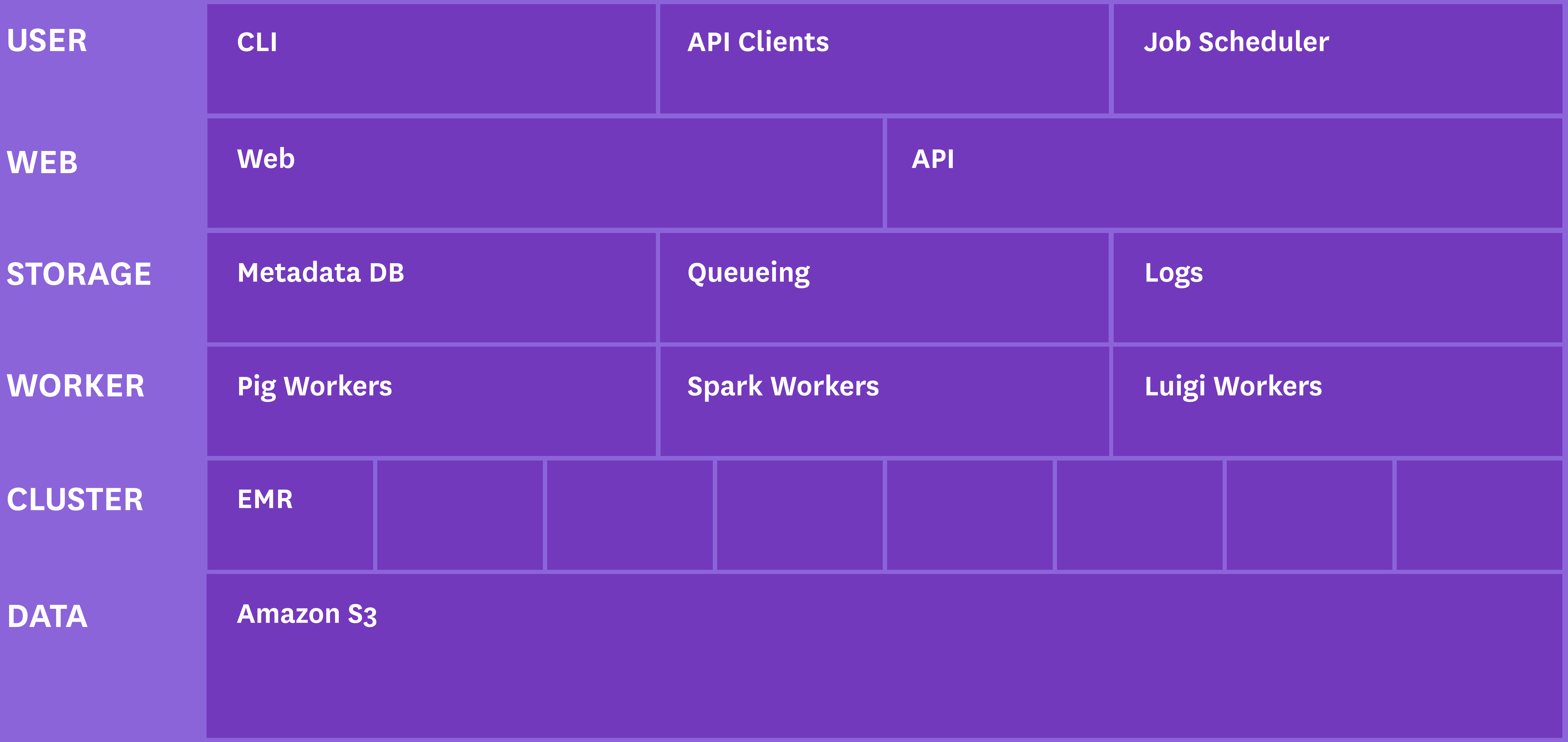
STORAGE: Metadata DB, Queueing, Logs

WORKER: Pig Workers, Spark Workers, Luigi Workers

CLUSTER: EMR

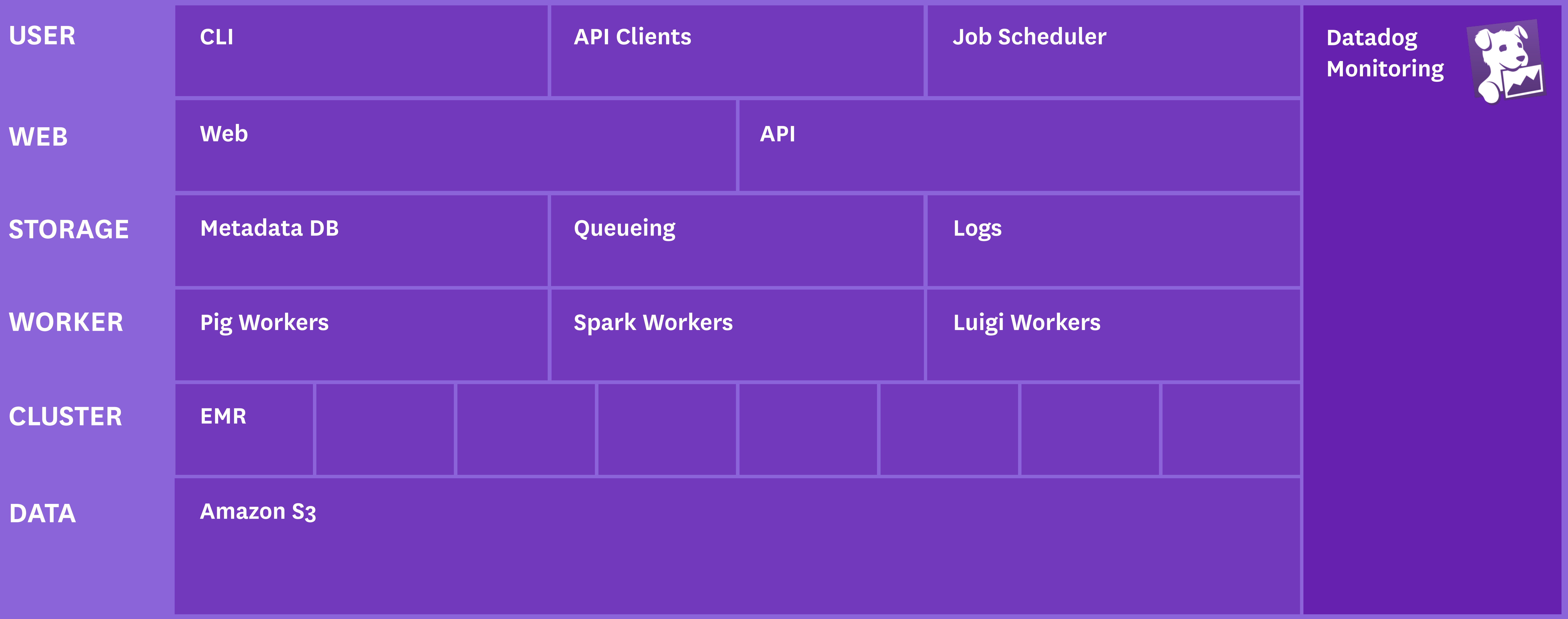
DATA: Amazon S3

Big Data Platform Architecture



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

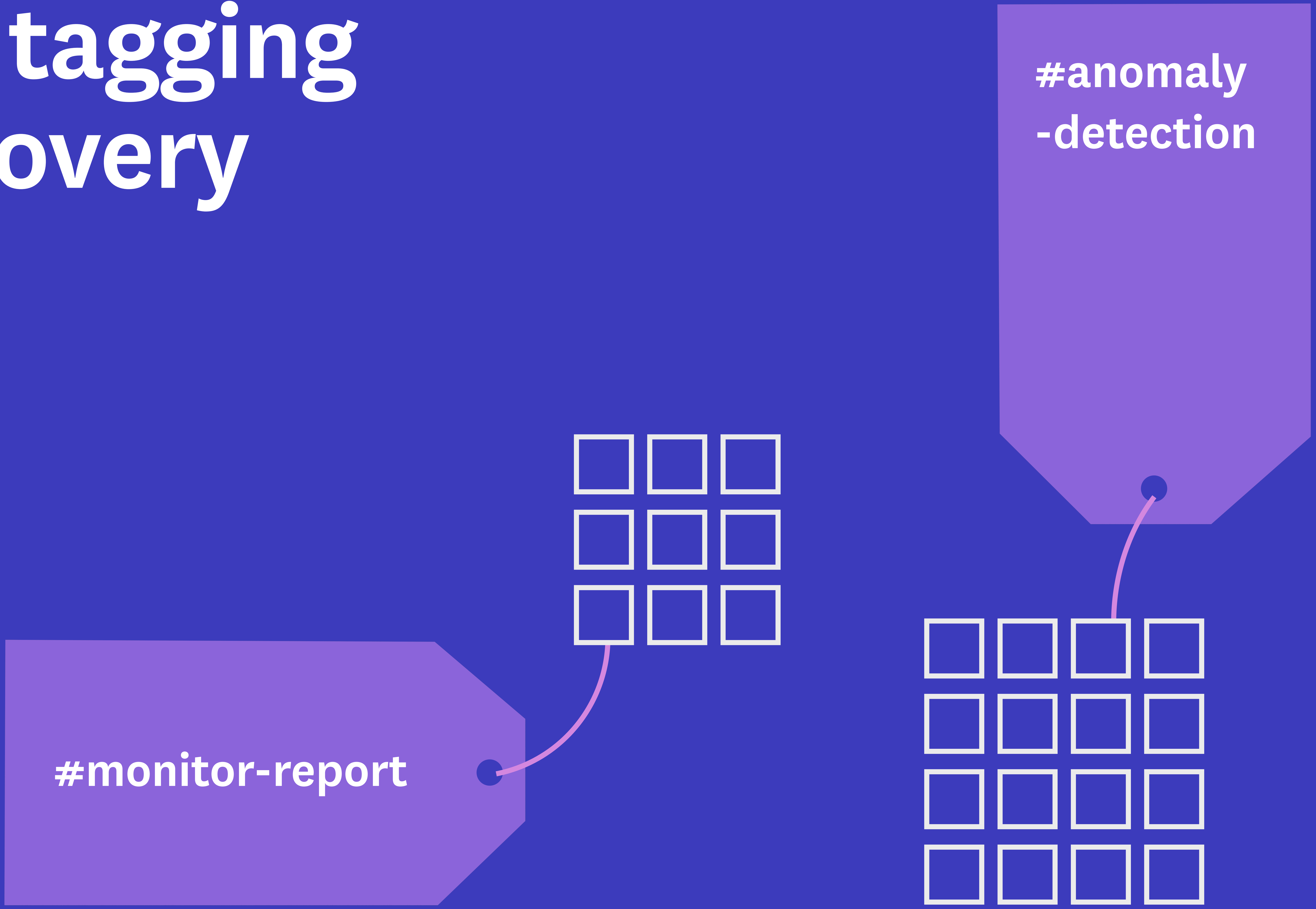
Big Data Platform Architecture



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?

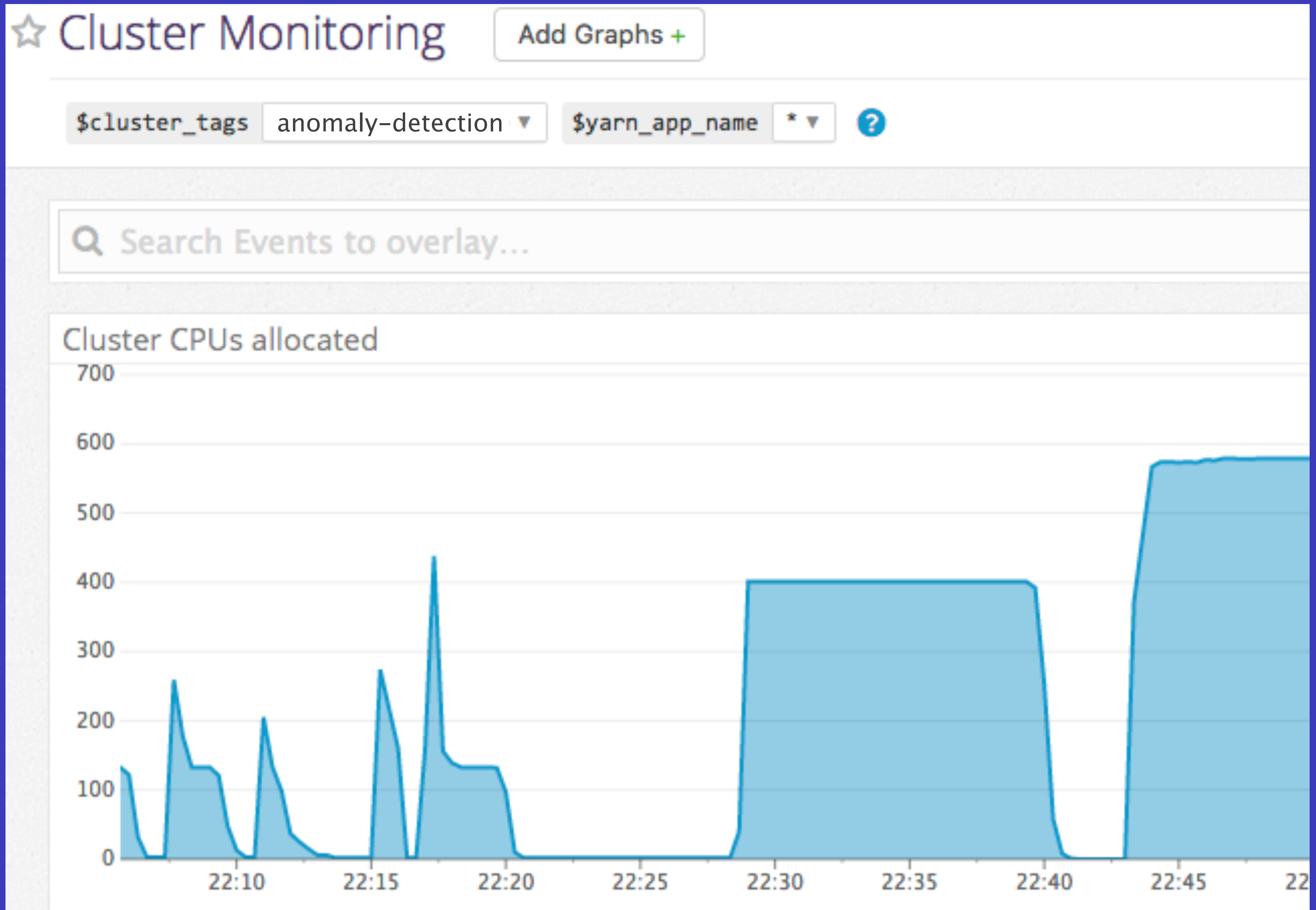
Cluster tagging for discovery



How to monitor many disappearing clusters?

Dynamic Monitoring on Tags

Dashboards

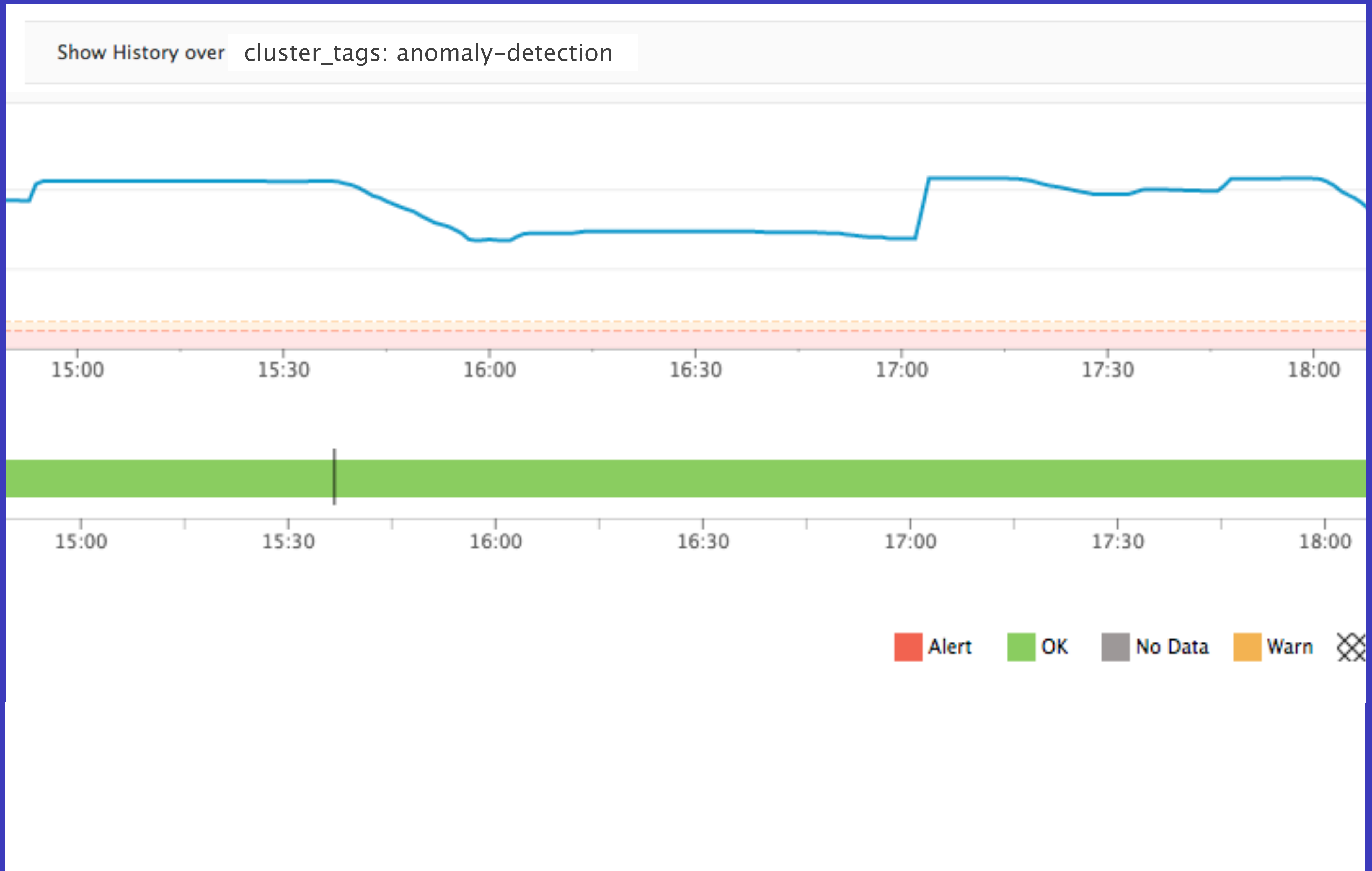


Cluster Monitoring

$cluster_tags anomaly-detection

$yarn_app_name *

Monitors



Show History over cluster_tags: anomaly-detection

How to debug problems when the cluster’s gone?

Debugging In a Post-Cluster World

Debugging In a Post-Cluster World

Send all logs to S3

- HDFS
- YARN
- Pig
- Spark

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

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

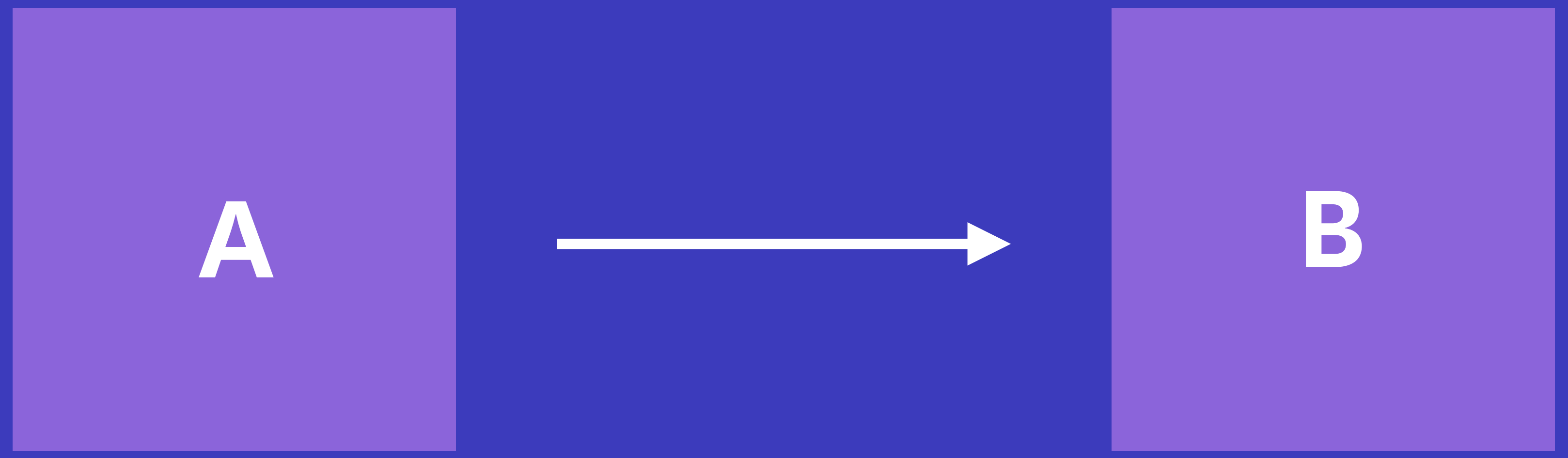
Preserve historical monitoring data

Keep history, by tag, after the cluster disappears

How to handle certain cluster failure in your jobs?

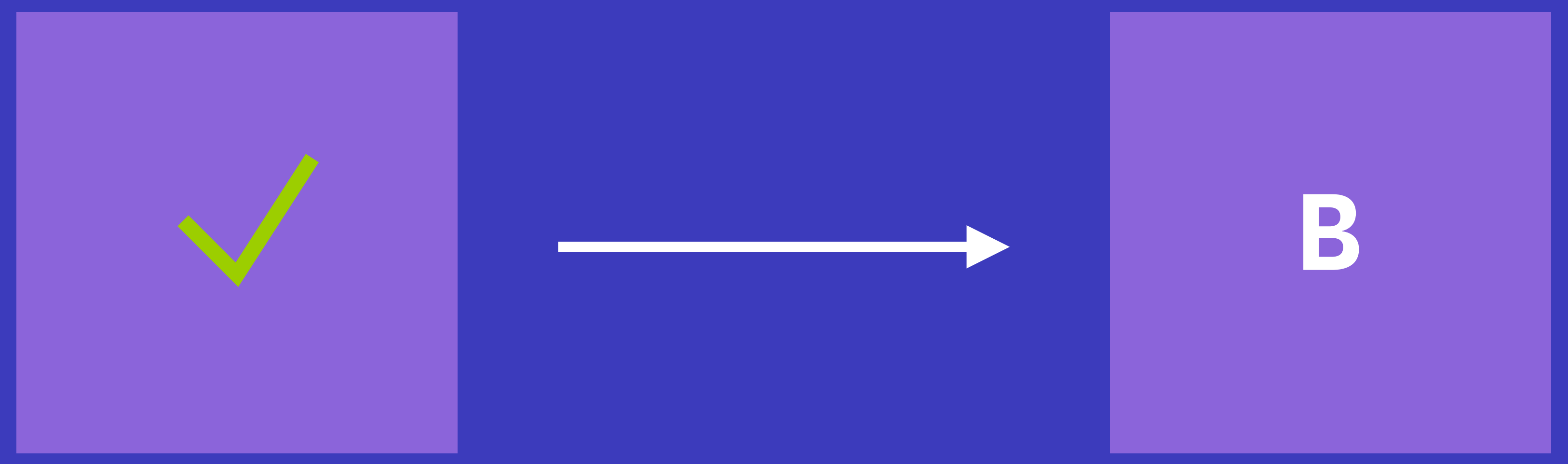
Luigi: design for failure.

Automatic cleanup and restart



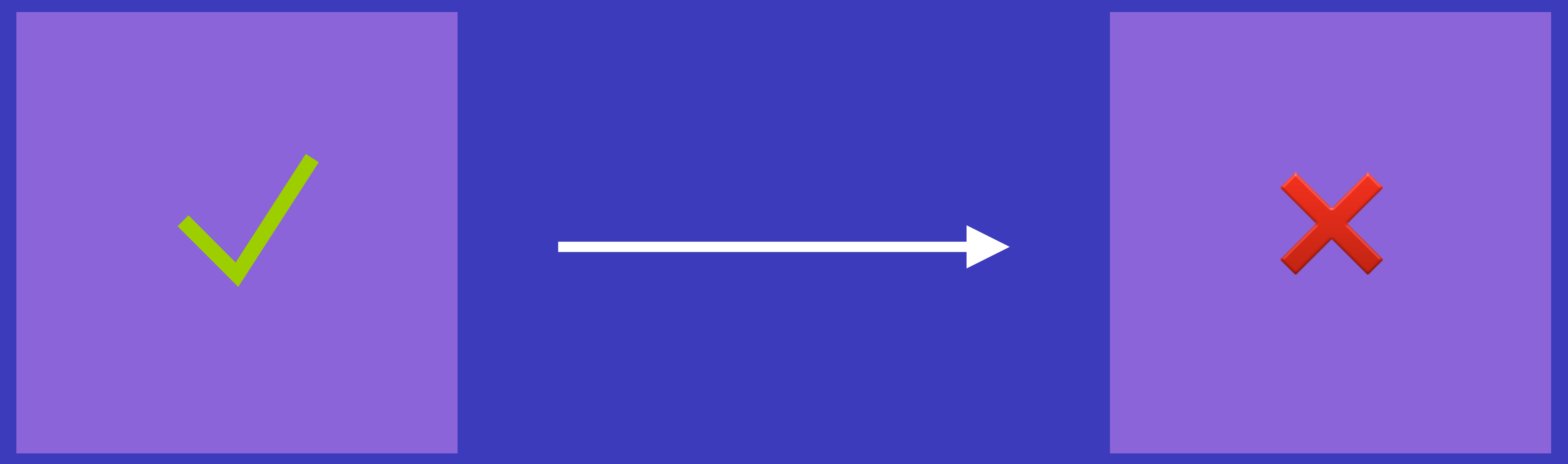
Luigi: design for failure.

Automatic cleanup and restart



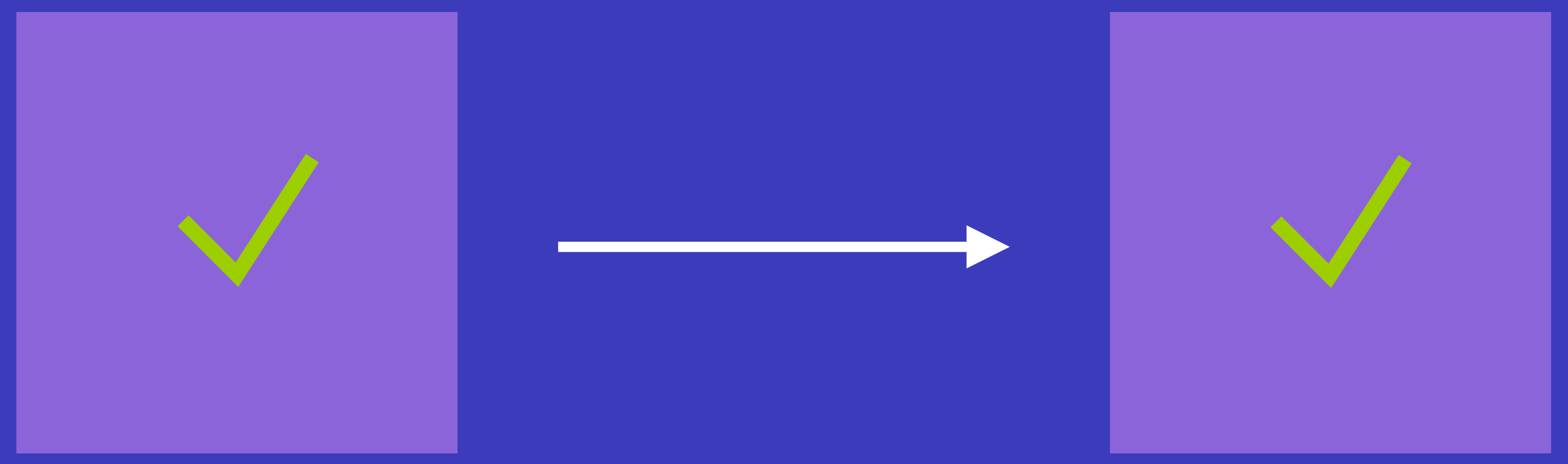
Luigi: design for failure.

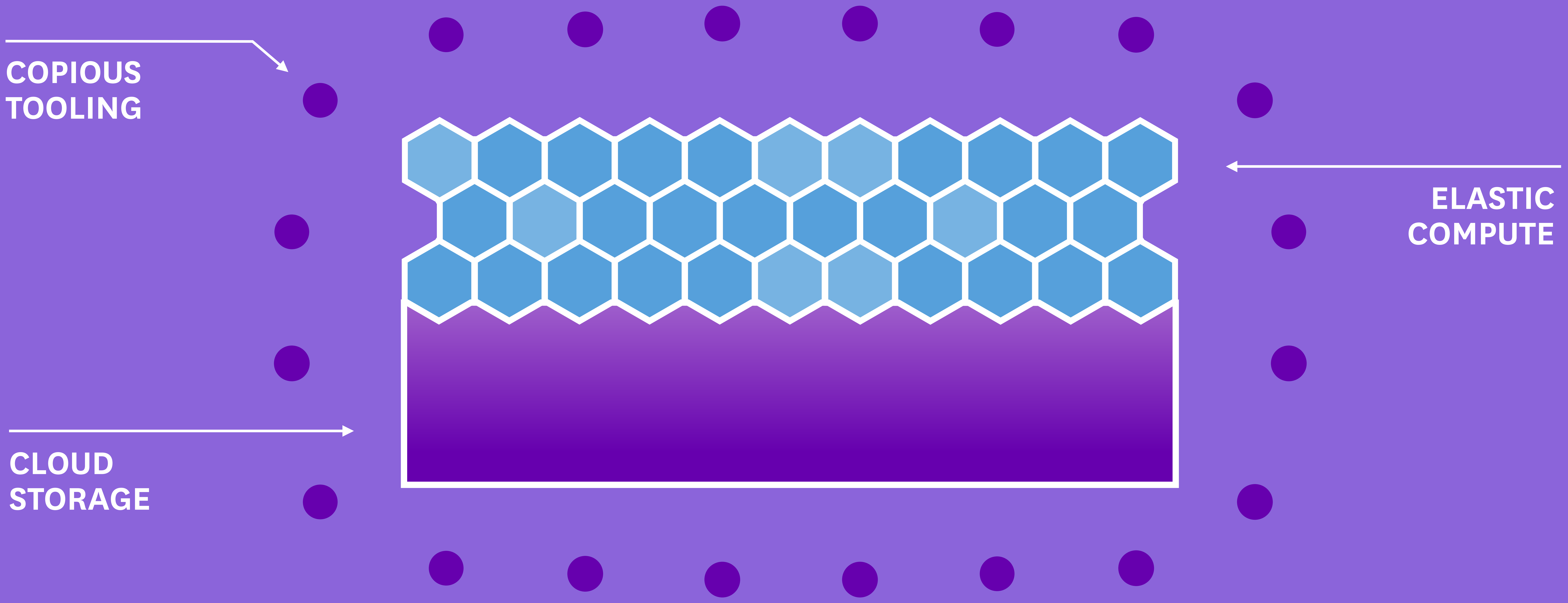
Automatic cleanup and restart



Luigi: design for failure.

Automatic cleanup and restart




COPIOUS TOOLING
ELASTIC COMPUTE
CLOUD STORAGE

Recommendations for Cloud Big Data

Recommendations for Cloud Big Data

- Use S3 for permanent data, not HDFS

Recommendations for Cloud Big Data

- Use S3 for permanent data, not HDFS
- Start from EMR if building yourself

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

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

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