How Netflix manages petabyte scale Apache Cassandra in the cloud

Joey Lynch, Vinay Chella
Netflix's Distributed Database Engineers

Who are we?

Vinay Chella

Distributed Systems Engineer

Focusing on Apache Cassandra and Data Abstractions

Cloud Data Engineering
Netflix

Joey Lynch

Distributed Systems Engineer

Distributed system addict and data wrangler

Cloud Data Engineering
Netflix

Agenda

Why use Cassandra?

Scale of Cassandra

Life of Cassandra Cluster

- Where does it start?
- Provisioning
- Keep it running
- Migration / Retiring

Murphy’s law applied

Why Apache Cassandra

— Millions of operations per sec

— Global data replication

— Failure isolation at rack level

— Chaos ready database

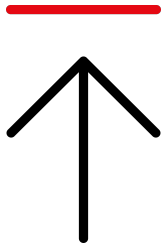
— Tunable consistency

— Log structured storage engine

Scale

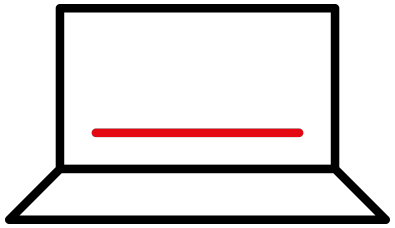
- 10’s of thousands instances
- 100’s of global C* clusters
- >6 PB of data
- Millions of requests / second
- Replicating several GiB/sec data across the globe

Story of Apache Cassandra and Netflix

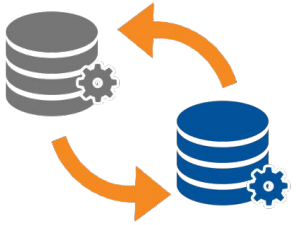


Inception

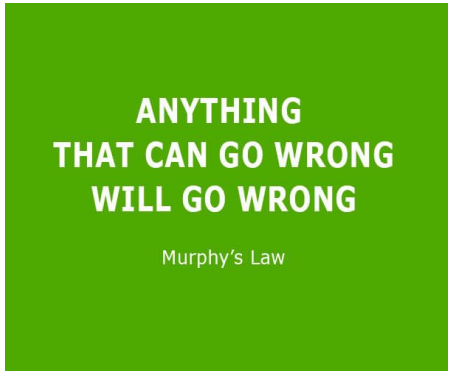
Provision



Keep it running

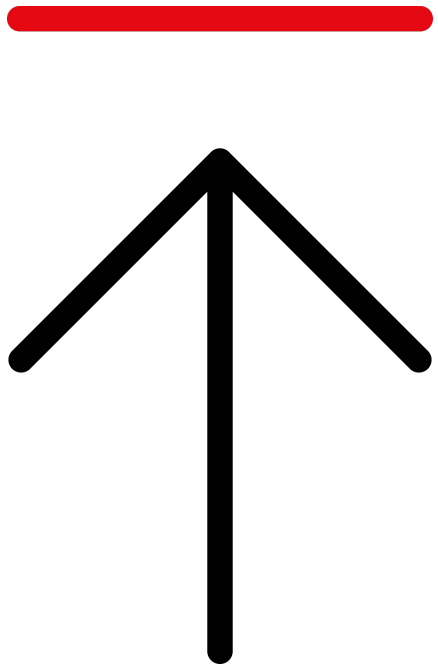


Migrations



Inception

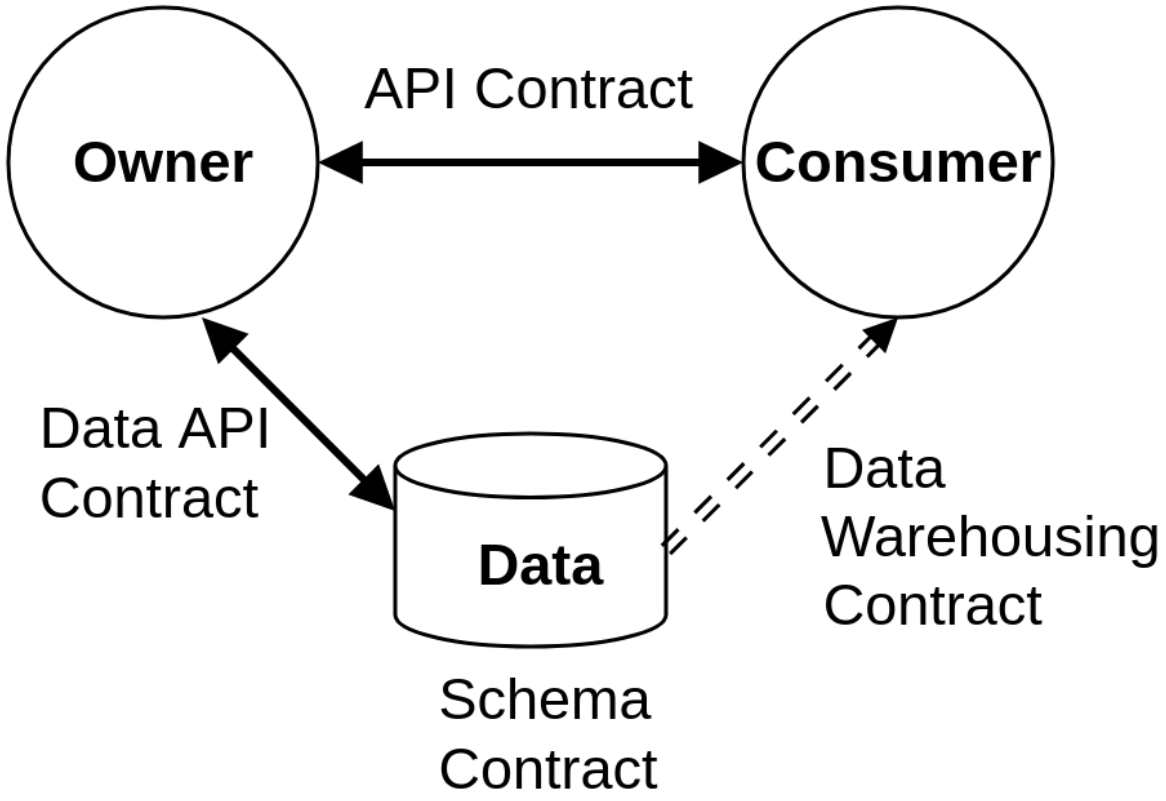
Where does it all start?



Inception

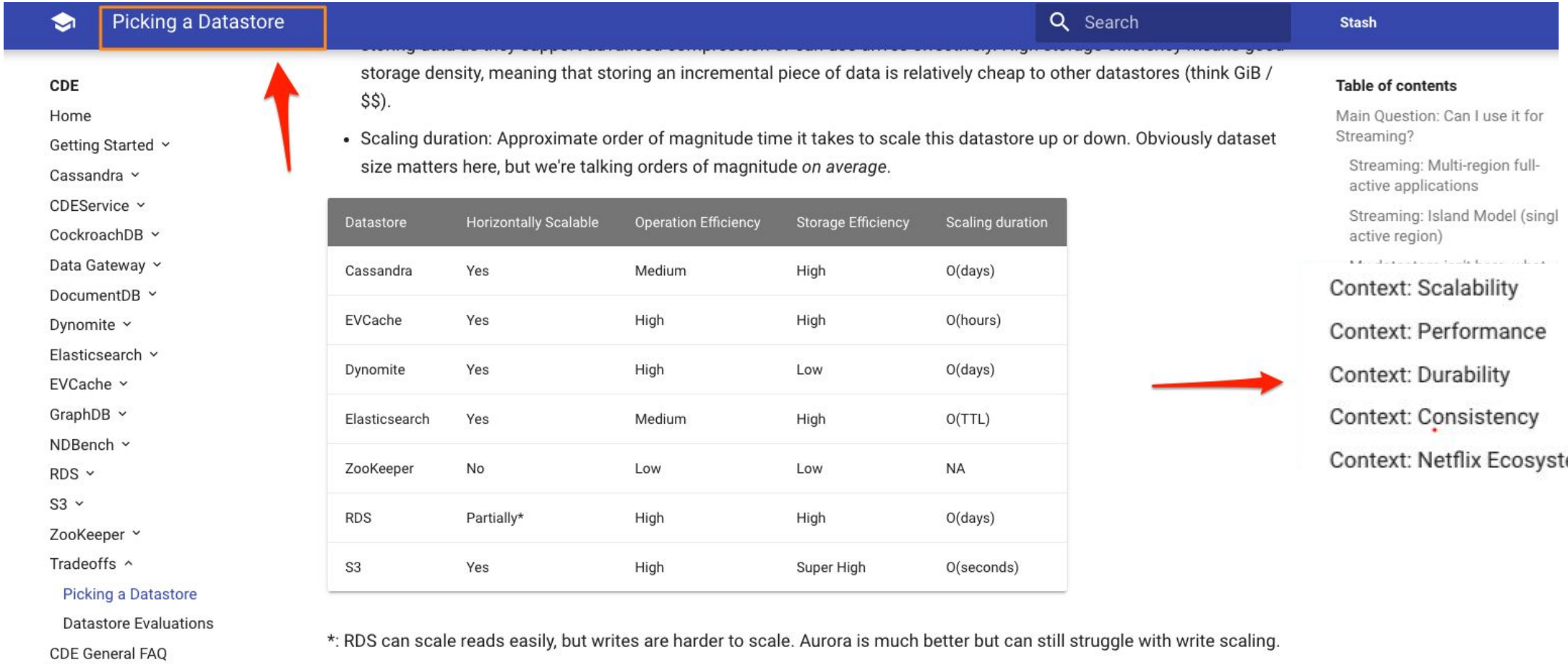
Service philosophy

- Context not *control*
  - Education
  - Tooling
- SLOs are key
  - Size, rate, latency, availability
- Every party must be responsible



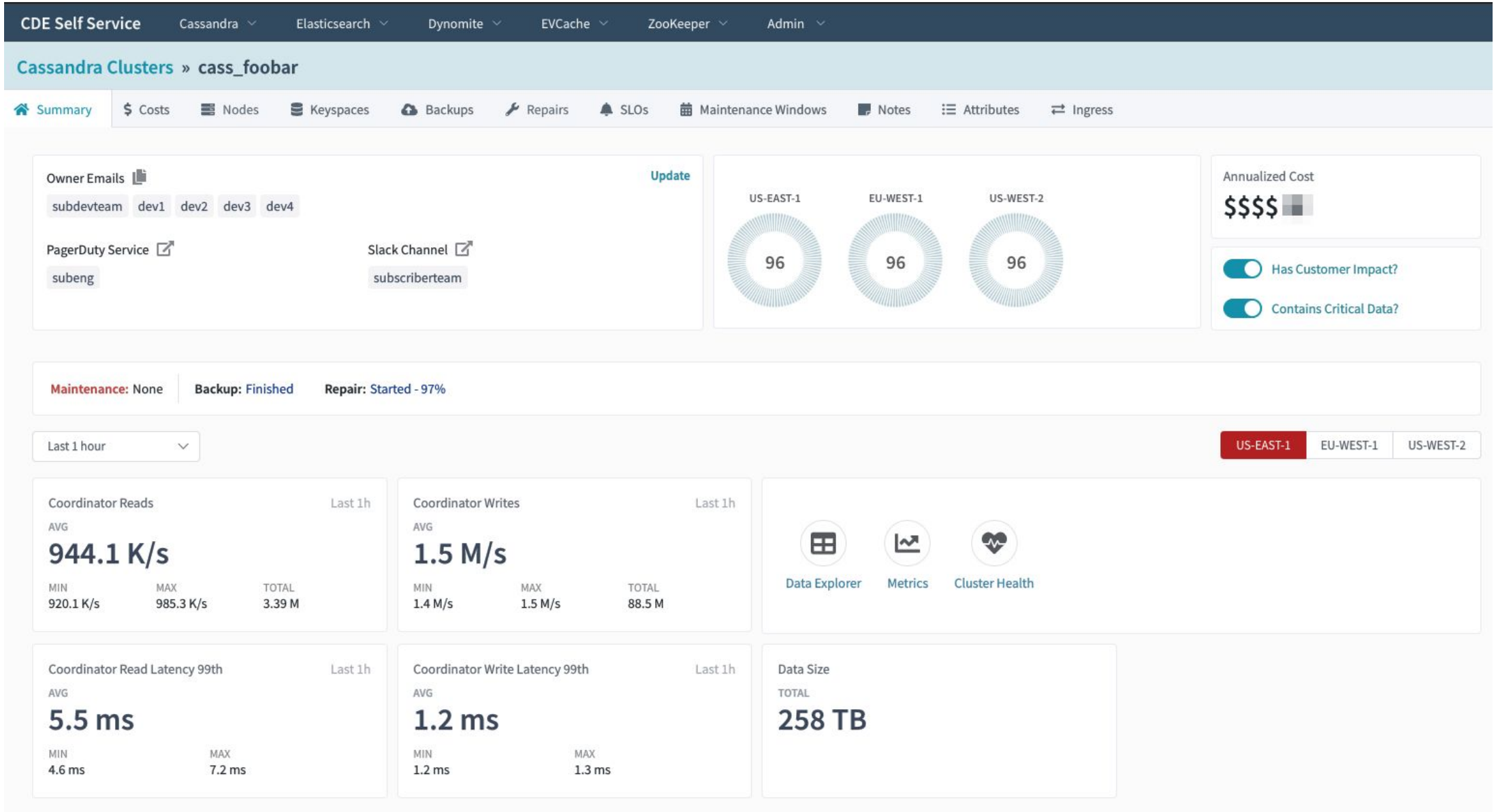
Inception

Invest in DevEd

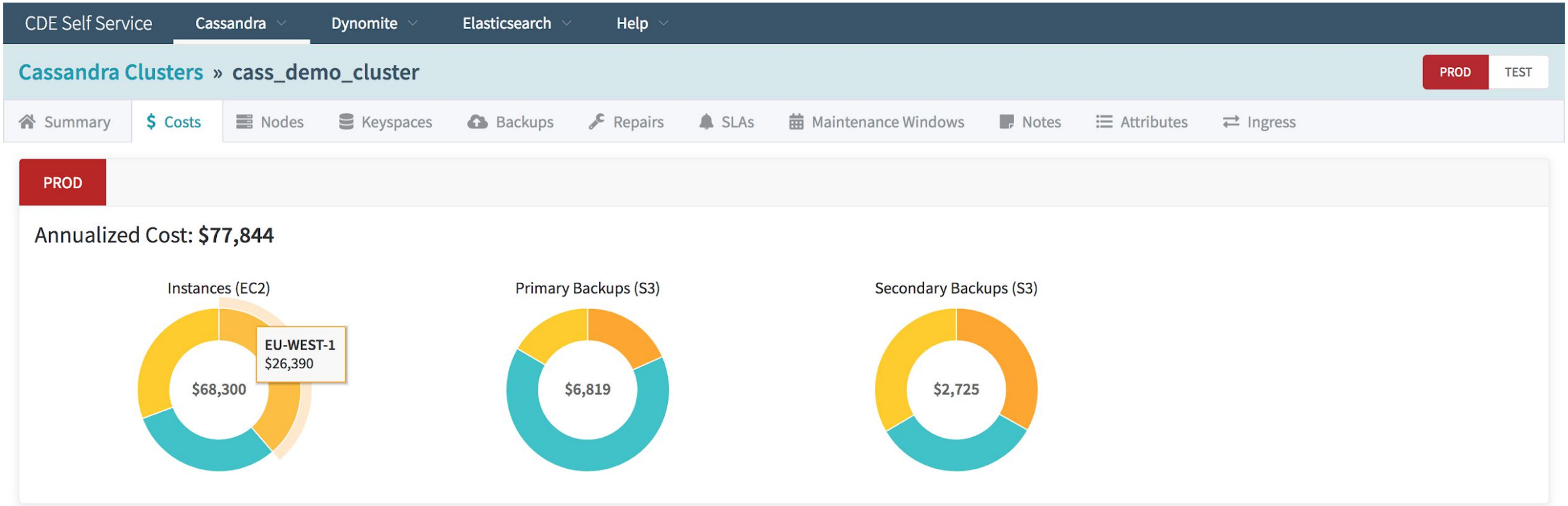


Inception

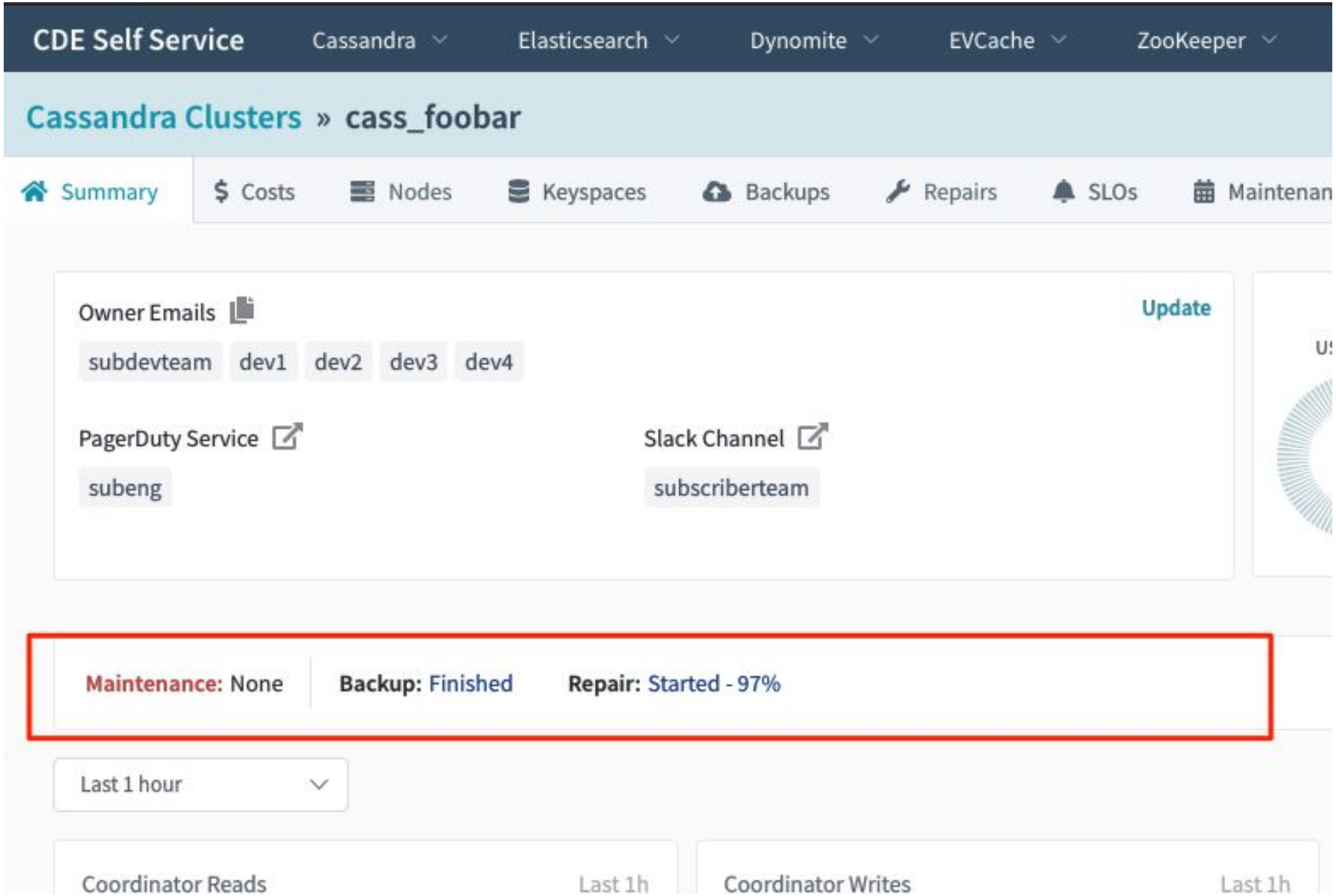
Better Tooling



Cost insights



Maintenance



Maintenance - Repair Insights

CDE Self Service | Cassandra | Dynomite | Elasticsearch | Help

Cassandra Clusters » cass_demo_cluster

PROD TEST

Summary | Costs | Nodes | Keyspaces | Backups | Repairs | SLAs | Maintenance Windows | Notes | Attributes | Ingress

History | Settings

Access to this section is restricted to CDE team members.

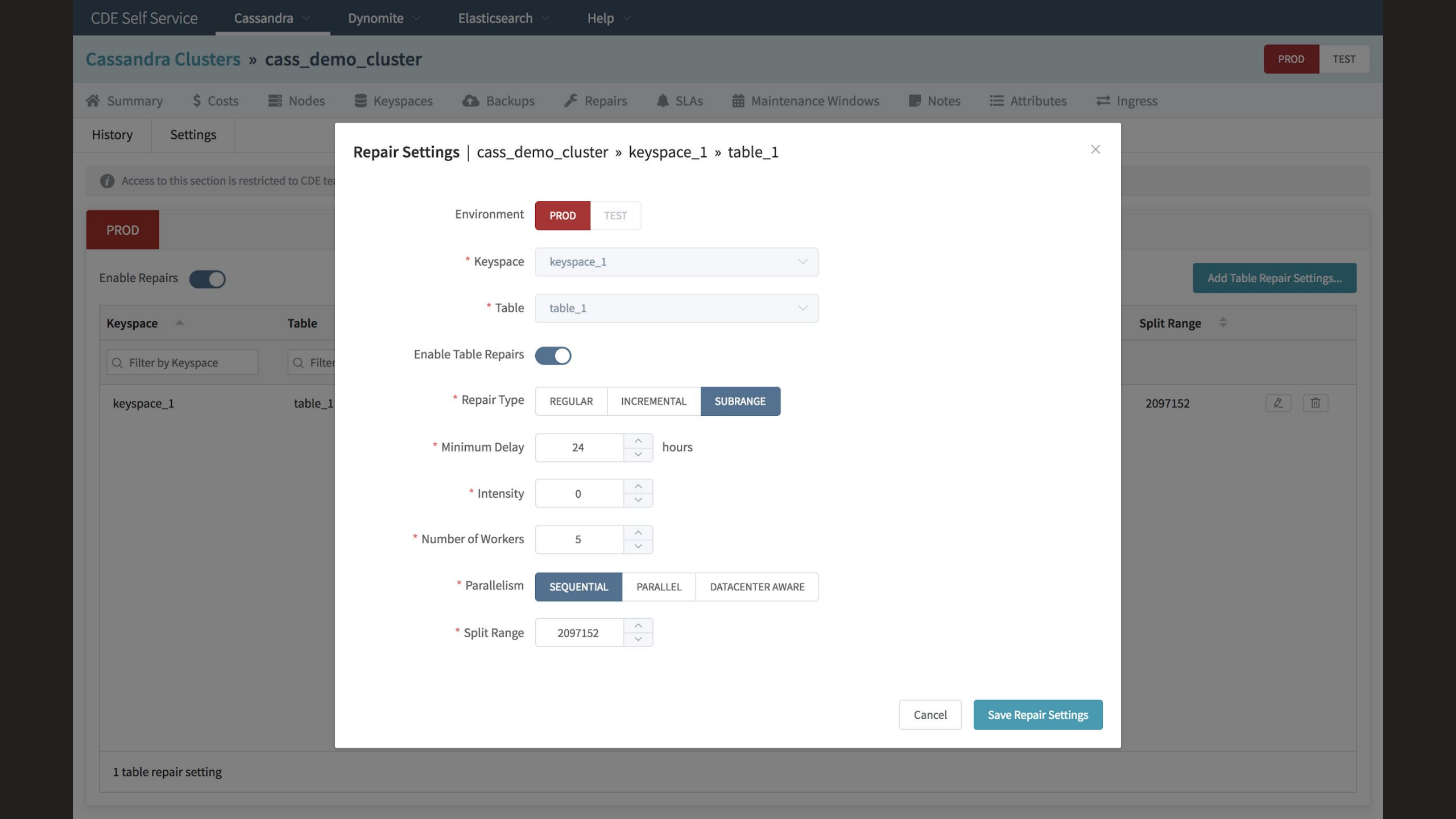
PROD

Enable Repairs

Add Table Repair Settings...

Keyspace	Table	Enabled?	Repair Type	Intensity	Min Delay	# Workers	Parallelism	Split Range
Filter by Keyspace	Filter by Table							
keyspace_1	table_1	✓	SUBRANGE	0	24	5	SEQUENTIAL	2097152

1 table repair setting


CDE Self Service
Cassandra
Dynomite
Elasticsearch
Help
Cassandra Clusters » cass_demo_cluster
PROD
TEST
Summary
Costs
Nodes
Keyspaces
Backups
Repairs
SLAs
Maintenance Windows
Notes
Attributes
Ingress
History
Settings
Access to this section is restricted to CDE te
PROD
Enable Repairs
Add Table Repair Settings...
Keyspace
Table
Split Range
Filter by Keyspace
keyspace_1
table_1
2097152
1 table repair setting
Repair Settings | cass_demo_cluster » keyspace_1 » table_1
Environment
PROD
TEST
* Keyspace
keyspace_1
* Table
table_1
Enable Table Repairs
* Repair Type
REGULAR
INCREMENTAL
SUBRANGE
* Minimum Delay
24
hours
* Intensity
0
* Number of Workers
5
* Parallelism
SEQUENTIAL
PARALLEL
DATACENTER AWARE
* Split Range
2097152
Cancel
Save Repair Settings

Maintenance - Backup Insights

CDE Self Service | Cassandra | Dynomite | Elasticsearch | Help

Cassandra Clusters » cass_demo_cluster

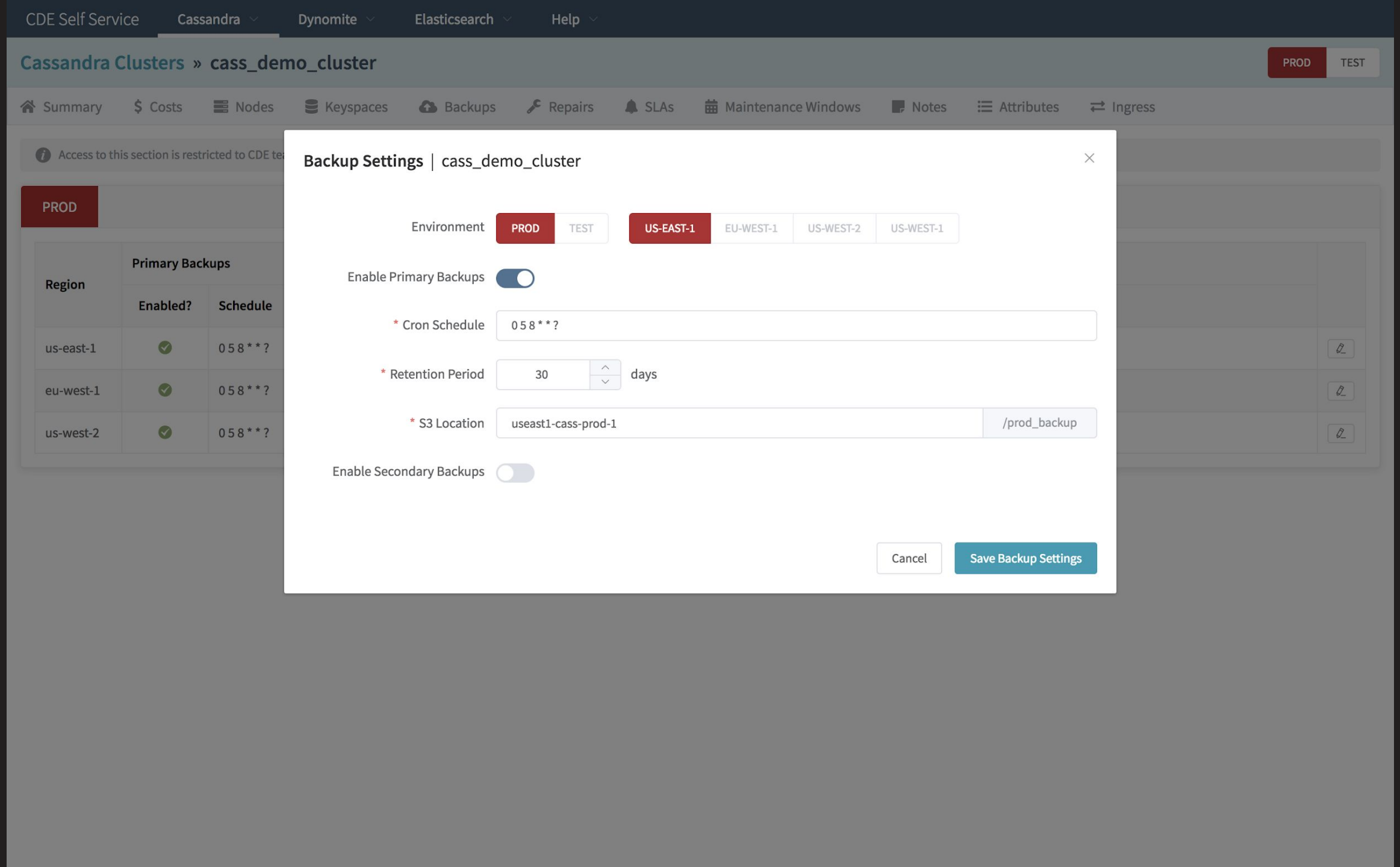
PROD TEST

Summary | Costs | Nodes | Keyspaces | Backups | Repairs | SLAs | Maintenance Windows | Notes | Attributes | Ingress

Access to this section is restricted to CDE team members.

PROD

Region	Primary Backups				Secondary Backups		
	Enabled?	Schedule	Retention	S3 Location	Enabled?	Location	
us-east-1	✓	0 5 8 * * ?	30 days	useast1-cass-prod-1/prod_backup	✗		
eu-west-1	✓	0 5 8 * * ?	7 days	euwest1-cass-prod-1/prod_backup	✗		
us-west-2	✓	0 5 8 * * ?	7 days	uswest2-cass-prod-1/prod_backup	✗		


CDE Self Service
Cassandra
Dynomite
Elasticsearch
Help
Cassandra Clusters » cass_demo_cluster
PROD
TEST
Summary
Costs
Nodes
Keyspaces
Backups
Repairs
SLAs
Maintenance Windows
Notes
Attributes
Ingress
Access to this section is restricted to CDE te
PROD
Region
Primary Backups
Enabled?
Schedule
us-east-1
0 5 8 * * ?
eu-west-1
0 5 8 * * ?
us-west-2
0 5 8 * * ?
Backup Settings | cass_demo_cluster
Environment
PROD
TEST
US-EAST-1
EU-WEST-1
US-WEST-2
US-WEST-1
Enable Primary Backups
* Cron Schedule
0 5 8 * * ?
* Retention Period
30
days
* S3 Location
useast1-cass-prod-1
/prod_backup
Enable Secondary Backups
Cancel
Save Backup Settings

Maintenance - Node Insights

PROD

Last cached: **Sep 2, 2019, 12:12:14 PM**

Node ID	Zone	Token	C* Version	Sidecar Version	OS	Instance	C* Data Size	Disk Used	Disk Size	Disk Used %
i-0c[redacted]f5	us-east-1e					i3.4xlarge		0.0 KB	0.0 KB	
i-0b[redacted]d0	us-east-1c	-1056844710747700824	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-09[redacted]3a	us-east-1e	-1152921502798271405	3.0.19.1	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-06[redacted]f8	us-east-1d	-1248998294848841986	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0a[redacted]51	us-east-1c	-1345075086899412567	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-05[redacted]57	us-east-1e	-1441151878949983148	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0f[redacted]72	us-east-1d	-1537228671000553729	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-08[redacted]f6	us-east-1c	-1633305463051124310	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-01[redacted]073	us-east-1e	-1729382255101694891	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-05[redacted]f6	us-east-1d	-1825459047152265472	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.4 TB	3.8 TB	37.00%
i-04[redacted]a8	us-east-1c	-192153582292565595	3.0.19.1	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-04[redacted]16	us-east-1c	-1921535839202836053	3.0.19.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.4 TB	3.8 TB	35.00%
i-02[redacted]76	us-east-1e	-2017612631253406634	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.4 TB	3.8 TB	37.00%
i-04[redacted]141	us-east-1d	-2113689423303977215	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-04[redacted]40	us-east-1c	-2209766215354547796	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0d[redacted]1d7	us-east-1e	-2305843007405118377	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-02[redacted]8d	us-east-1d	-2401919799455688958	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0e[redacted]788	us-east-1c	-2497996591506259539	3.0.19.1	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0e[redacted]42	us-east-1e	-2594073383556830120	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.4 TB	3.8 TB	35.00%
i-0b[redacted]38c	us-east-1d	-2690150175607400701	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-00[redacted]e58	us-east-1c	-2786226967657971282	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-05[redacted]3b	us-east-1e	-288230374343136176	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.4 TB	3.8 TB	36.00%
i-0e[redacted]79	us-east-1e	-2882303759708541863	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0d[redacted]5df	us-east-1d	-2978380551759112444	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-08[redacted]1e	us-east-1c	-3074457343809683025	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.4 TB	3.8 TB	37.00%
i-06[redacted]c7	us-east-1e	-3170534135860253606	3.0.19.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.3 TB	3.8 TB	35.00%
i-08[redacted]c1	us-east-1d	-3266610927910824187	3.0.17.2	7.0.60-h74.4ef95a1	xenial	i3.4xlarge	1.22 TB	1.4 TB	3.8 TB	36.00%

SLOs are Key

CDE Self Service | Cassandra | Elasticsearch | Dynomite | EVCache | ZooKeeper | Admin | Help

Cassandra Clusters » cass_foobar

PROD TEST

Summary | Costs | Nodes | Keyspaces | Backups | Repairs | SLOs | Maintenance Windows | Notes | Attributes | Ingress

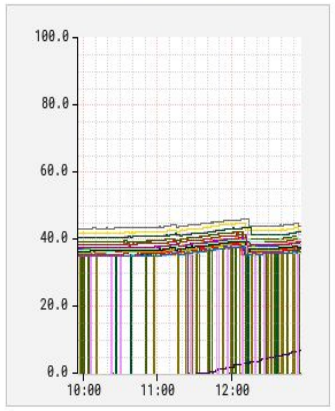
PROD

SLOs are visible to all users, but can be modified by CDE team members only.

Enable Monitoring

Actions | Update SLOs

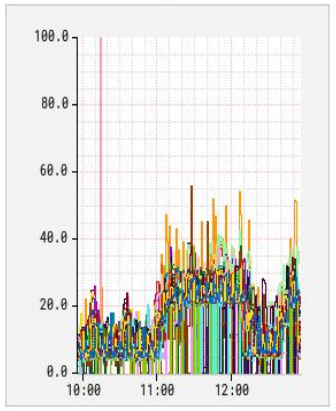
Disk Used (%)



Email Pager SQS Winston

critical	95
threshold_in_percent	50
warn	60

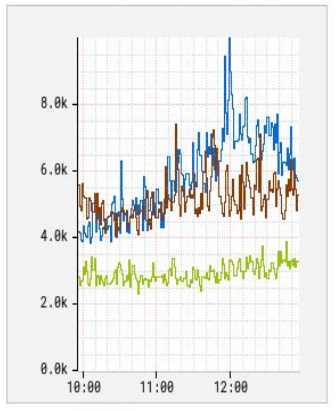
CPU Used (%)



Email Pager SQS Winston

period_in_minutes	5
threshold_in_percentage	80

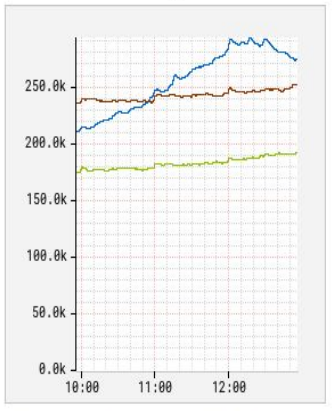
Read Latency 99th (µs)



Email Pager SQS Winston

period_in_minutes	10
threshold_in_microseconds	1000000

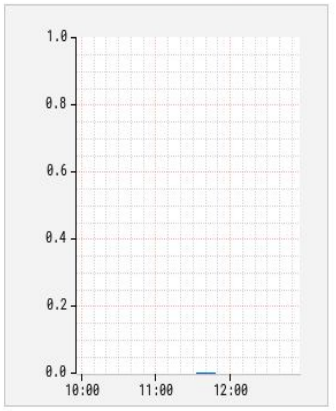
Read Ops/sec



Email Pager SQS Winston

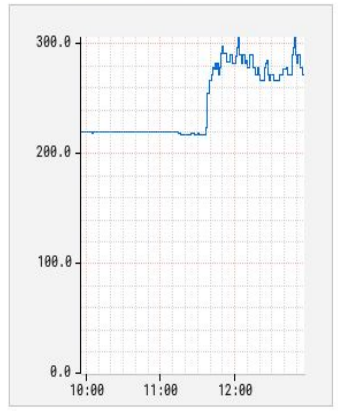
threshold	3000

Backup v2 Failures



Email Pager SQS Winston

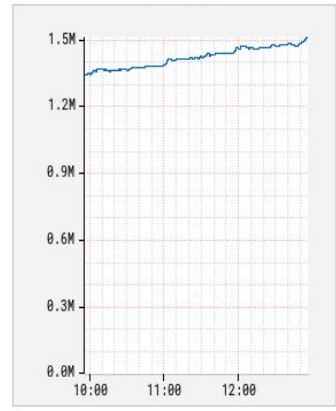
Write Latency 99th (µs)



Email Pager SQS Winston

period_in_minutes	10
threshold_in_microseconds	1000000

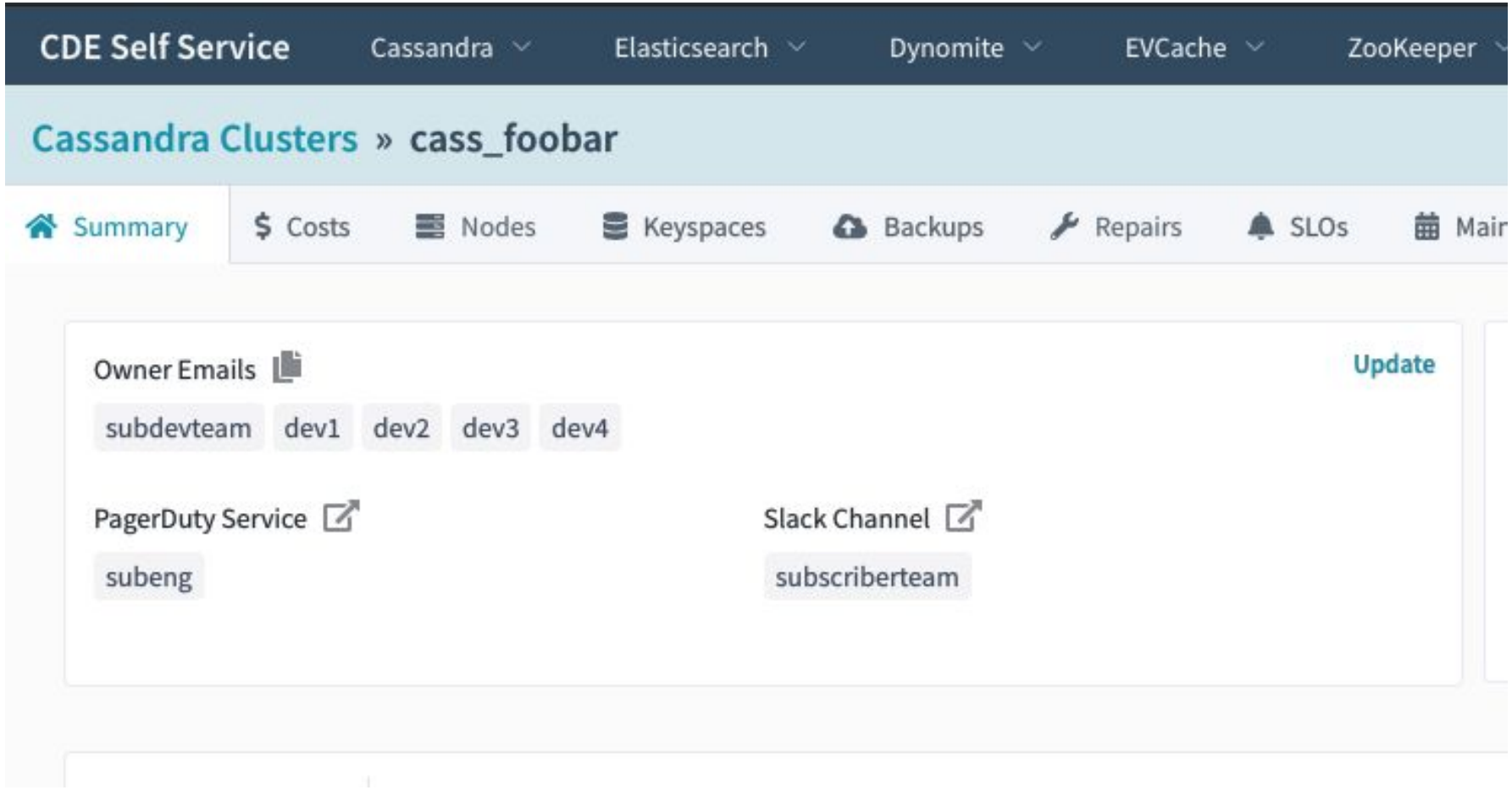
Write Ops/sec



Email Pager SQS Winston

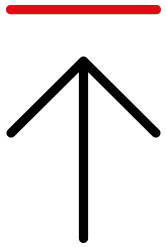
threshold	1000

Whom to page?


CDE Self Service
Cassandra
Elasticsearch
Dynomite
EVCache
ZooKeeper
Cassandra Clusters » cass_foobar
Summary
Costs
Nodes
Keyspaces
Backups
Repairs
SLOs
Owner Emails
Update
subdevteam
dev1
dev2
dev3
dev4
PagerDuty Service
subeng
Slack Channel
subscriberteam

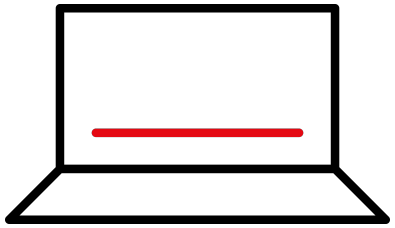
Good contracts make good partners!

Story of C*

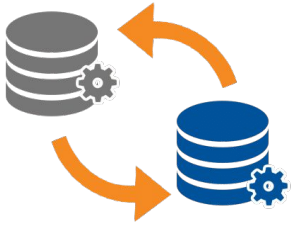


Inception

Provision



Keep it running



Migrations

ANYTHING
THAT CAN GO WRONG
WILL GO WRONG

Murphy's Law

Provision

Get up and running fast!

CDE Self Service | Cassandra | Elasticsearch | Dynomite | EVCache | ZooKeeper | Admin | Hel

Cassandra Clusters » Create new Cluster

* Cluster Name

cass_

* Owner Email(s)

Type to search

PagerDuty Service

Type to filter

Slack Channel

Type to filter

* Cluster Topology

Island Cluster(s) Global cluster

PROD

Enable in PROD?

* Cluster regions

us-east-1 eu-west-1 us-west-2

* Estimated Data Size per region

GB

TEST

Enable in TEST?

* Cluster regions

us-east-1 eu-west-1 us-west-2

* Estimated Data Size per region

GB

Contains Critical Data?

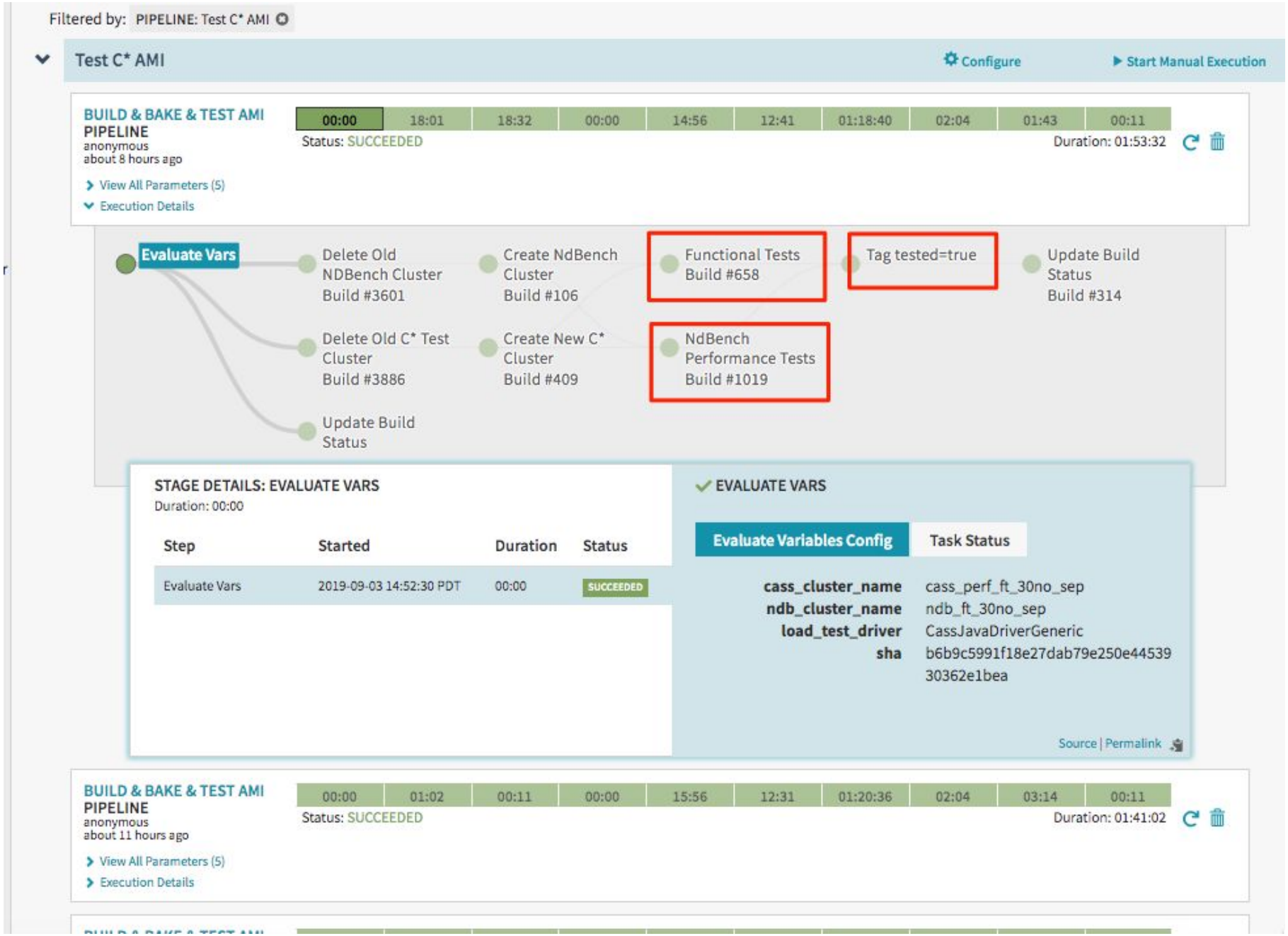
Create Cluster Cancel

Prove it before serving it

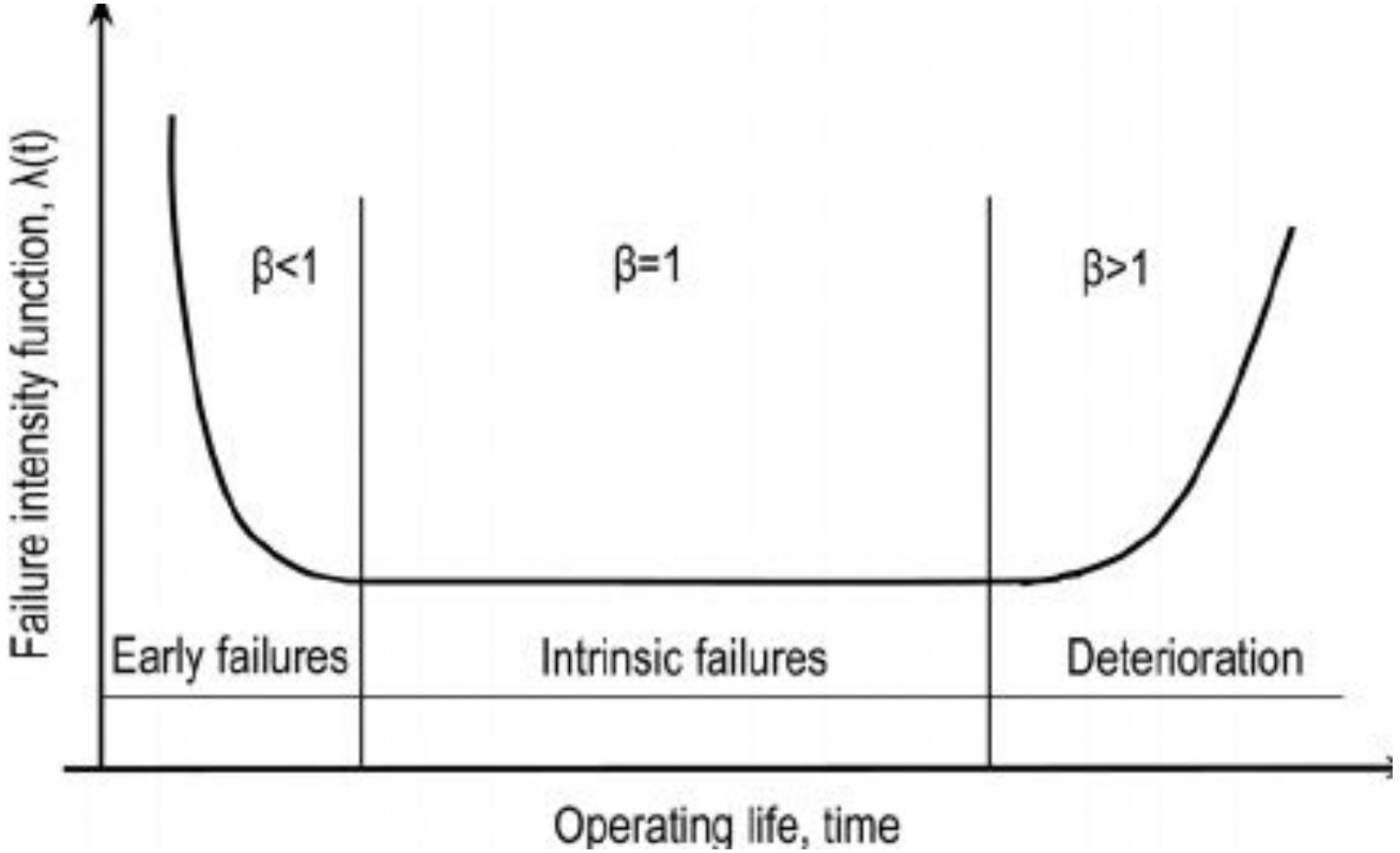
Robust AMI build process

Pre-flight check

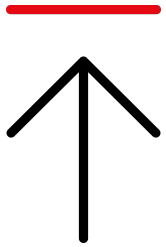
AMI Build process



Pre-flight check EC2 Instance lifecycle

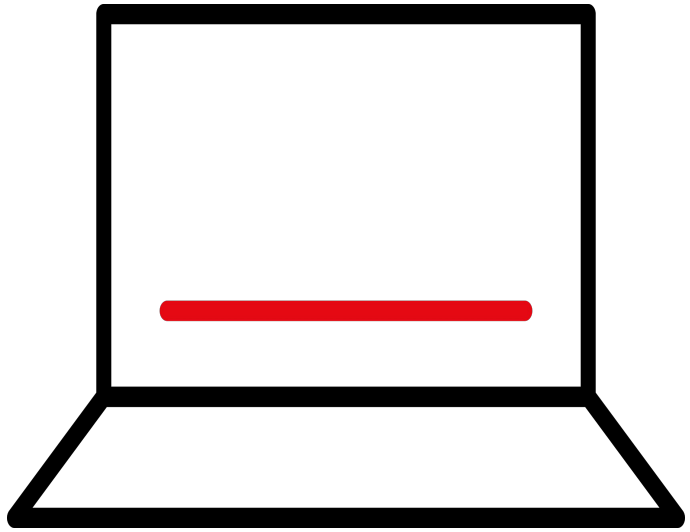


Story of C*

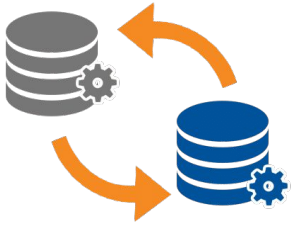


Inception

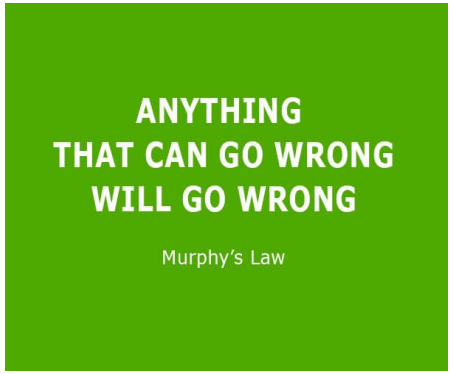
Provision



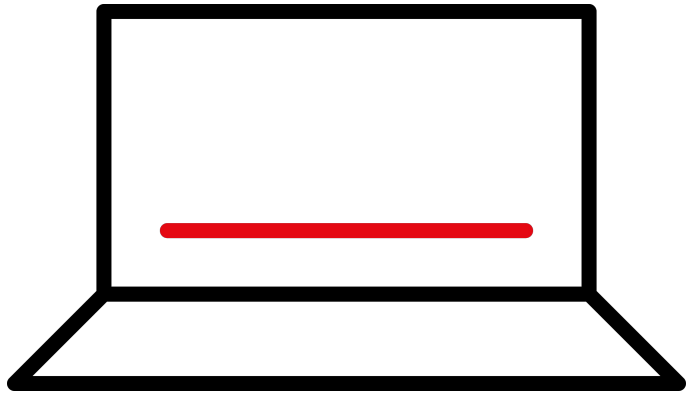
Keep it running



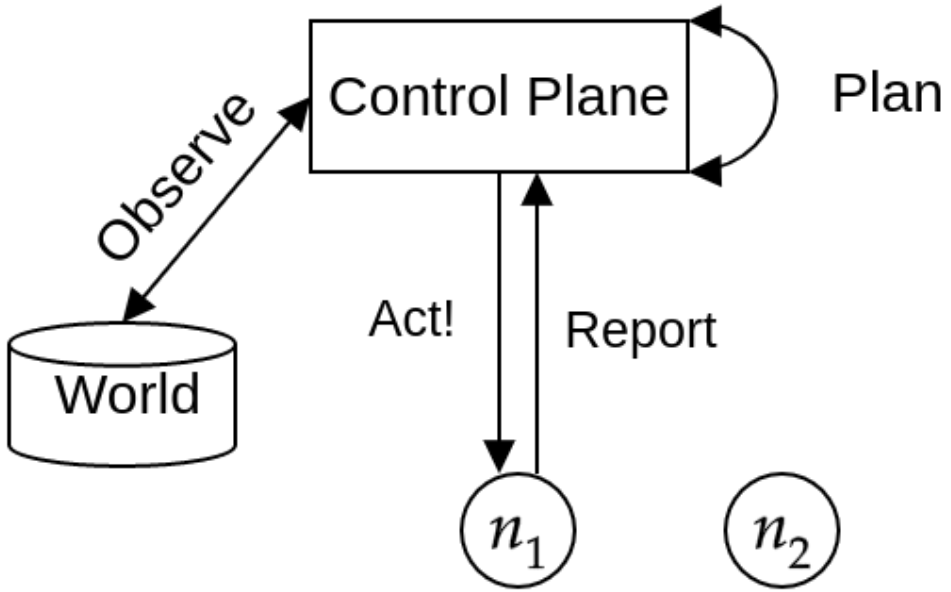
Migrations



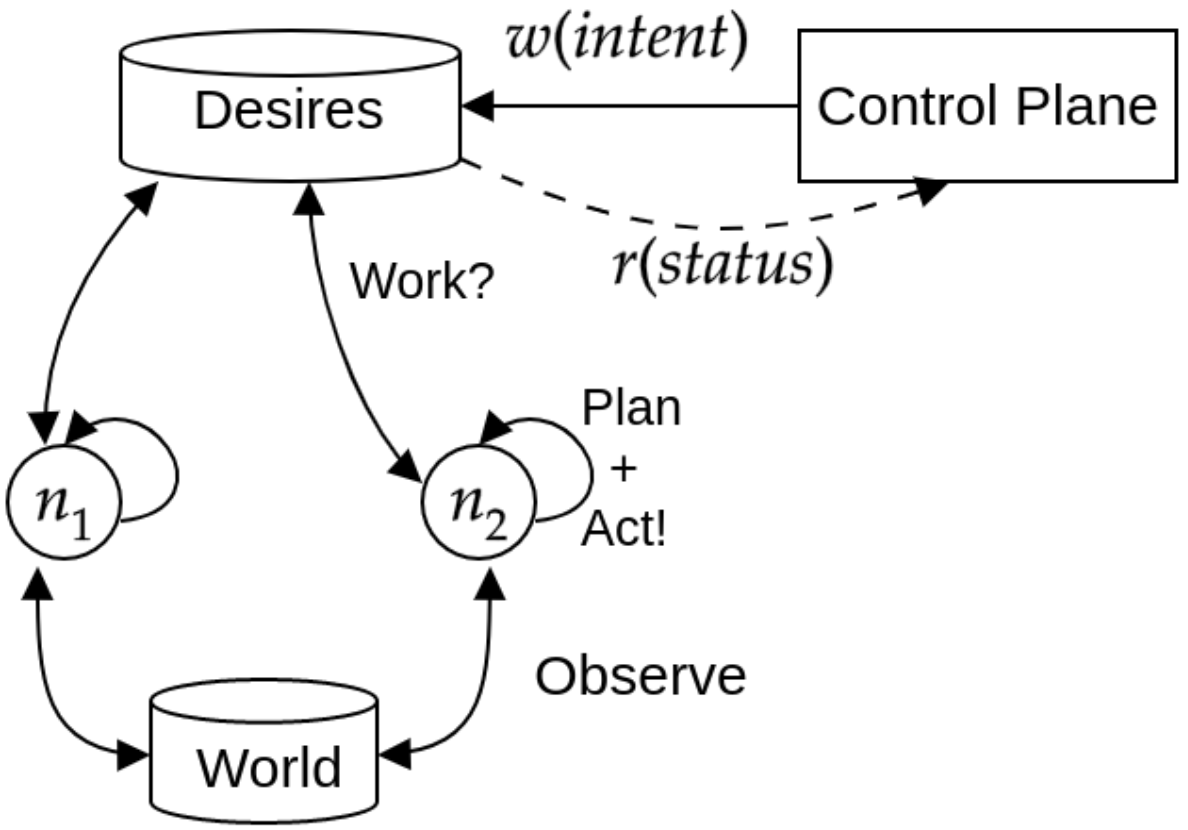
Keep it Running



Imperative Control Plane



Declarative Control Plane



Imperative Control Plane

Pros:

- Easy to implement
- Easy to understand

Cons:

- Fragile to failure
- Scaling issues
- Have to implement parallelism

Declarative Control Plane

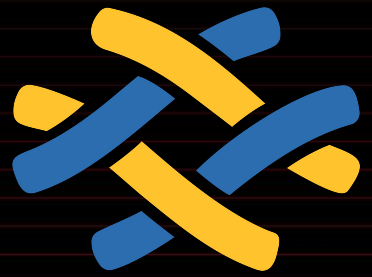
Pros:

- Lower total cost of ownership
- Parallel out of the box

Cons

- Requires different engineering mindset
- Hard to implement
- Have to stop parallelism

Imperative Tools



fabfile.org/

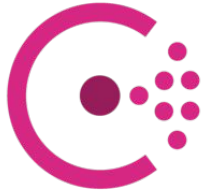
ansible.com/

jenkins.io/

- "Orchestrators"
  - Jenkins
  - Fabric
  - Ansible
- Imperative operation

Declarative Tools

APACHE ZooKeeper™



consul.io/

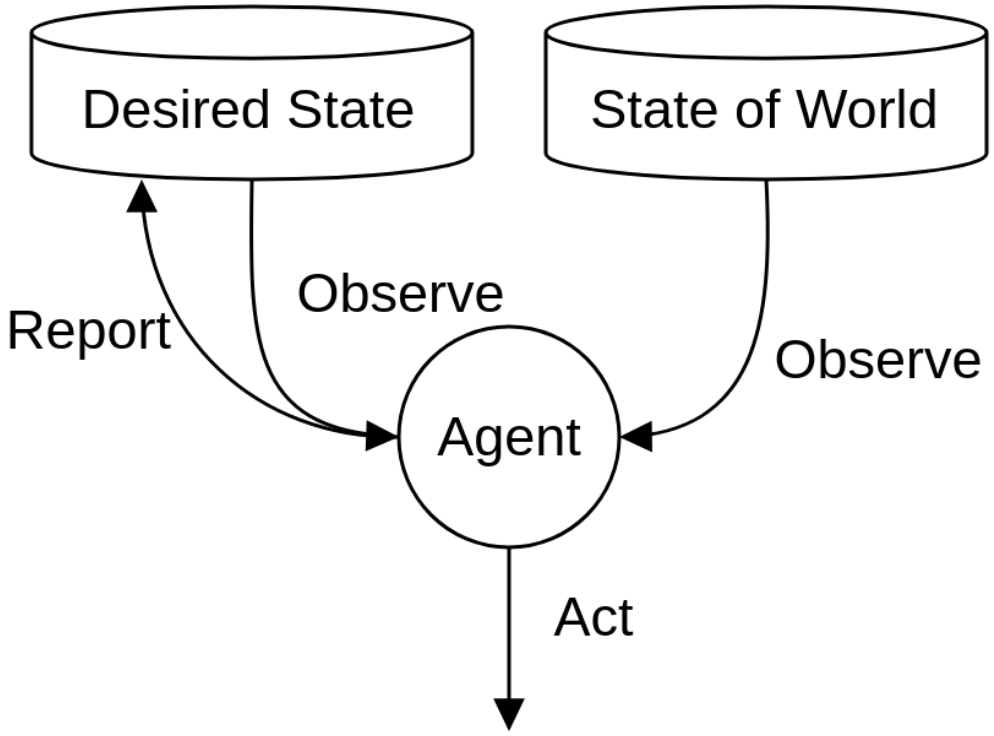
puppet

puppet.com

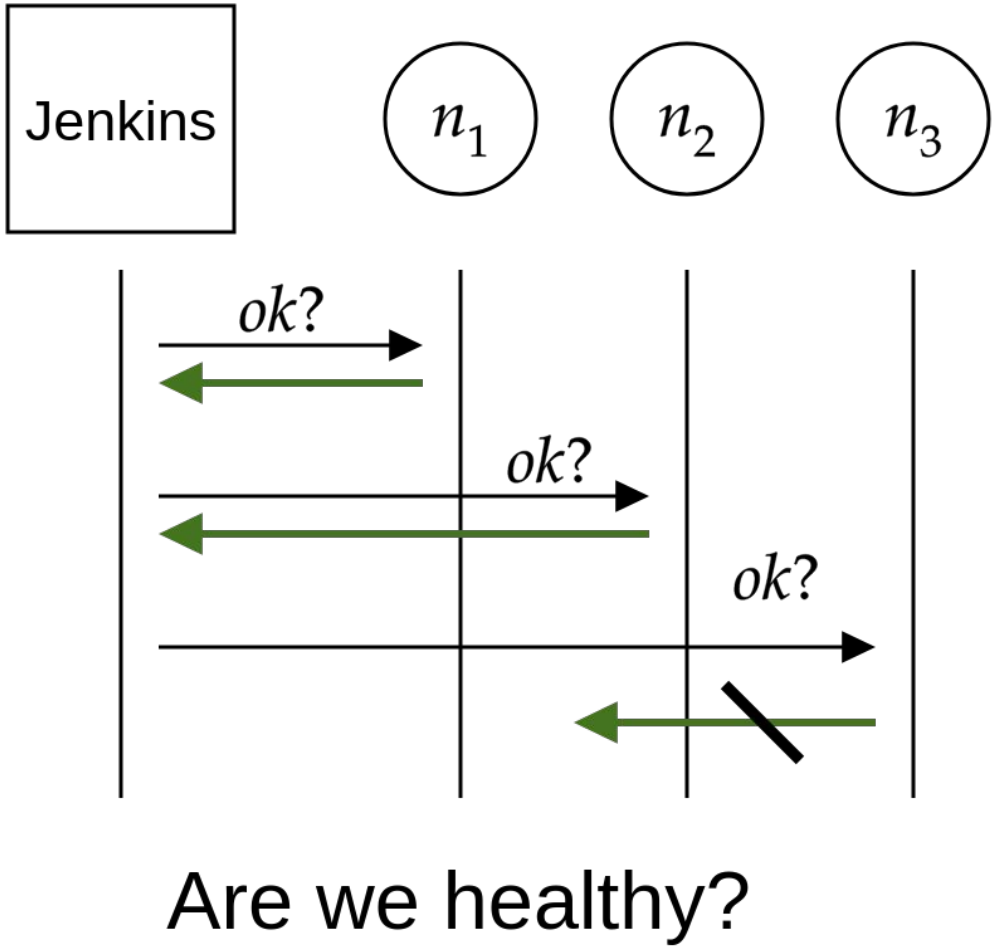
- Distributed Databases
- Distributed operation
- Similar to Actor model

Declarative Control Plane

1. Obtain **desires** from datastore
2. Obtain **state** of world
3. If state != desire, act!



Ring Health *Imperative* Solution

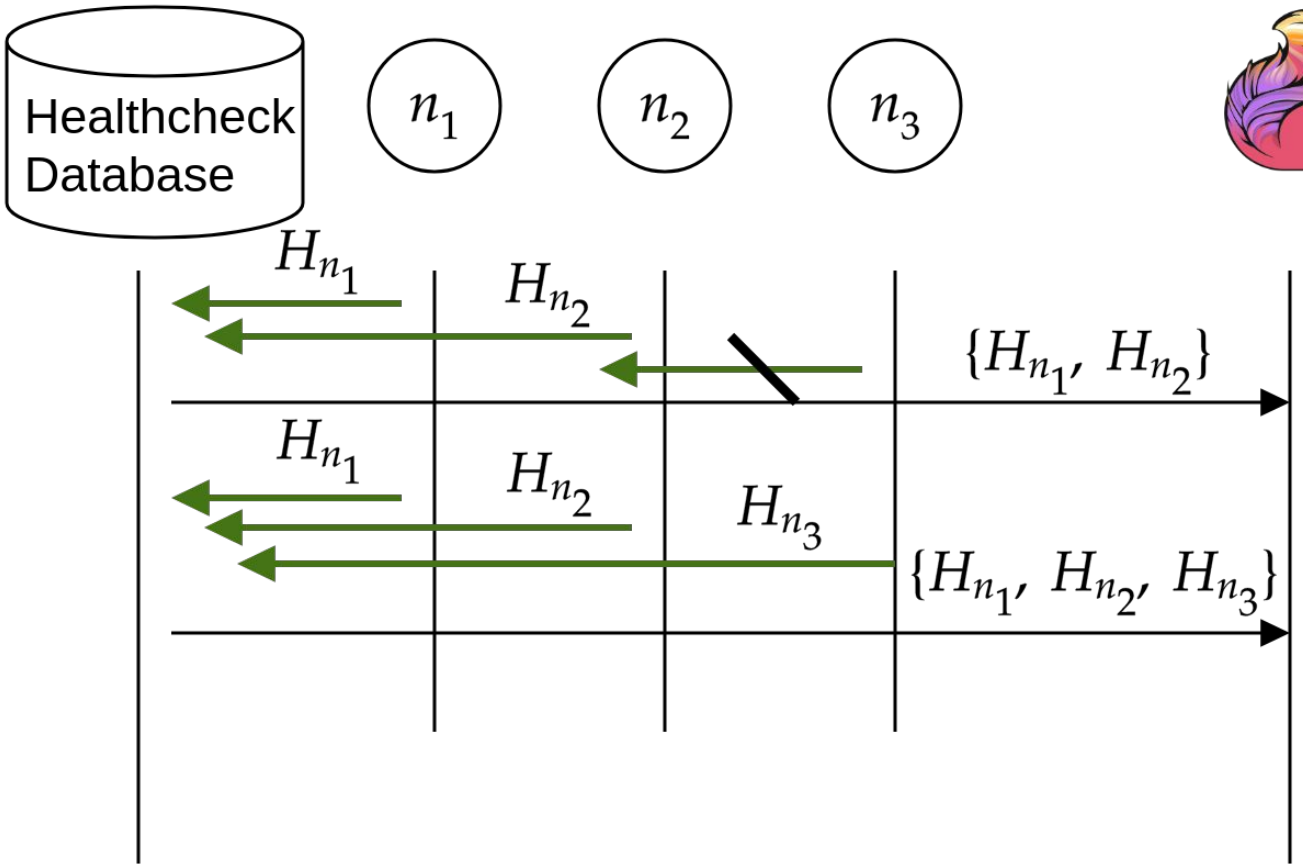


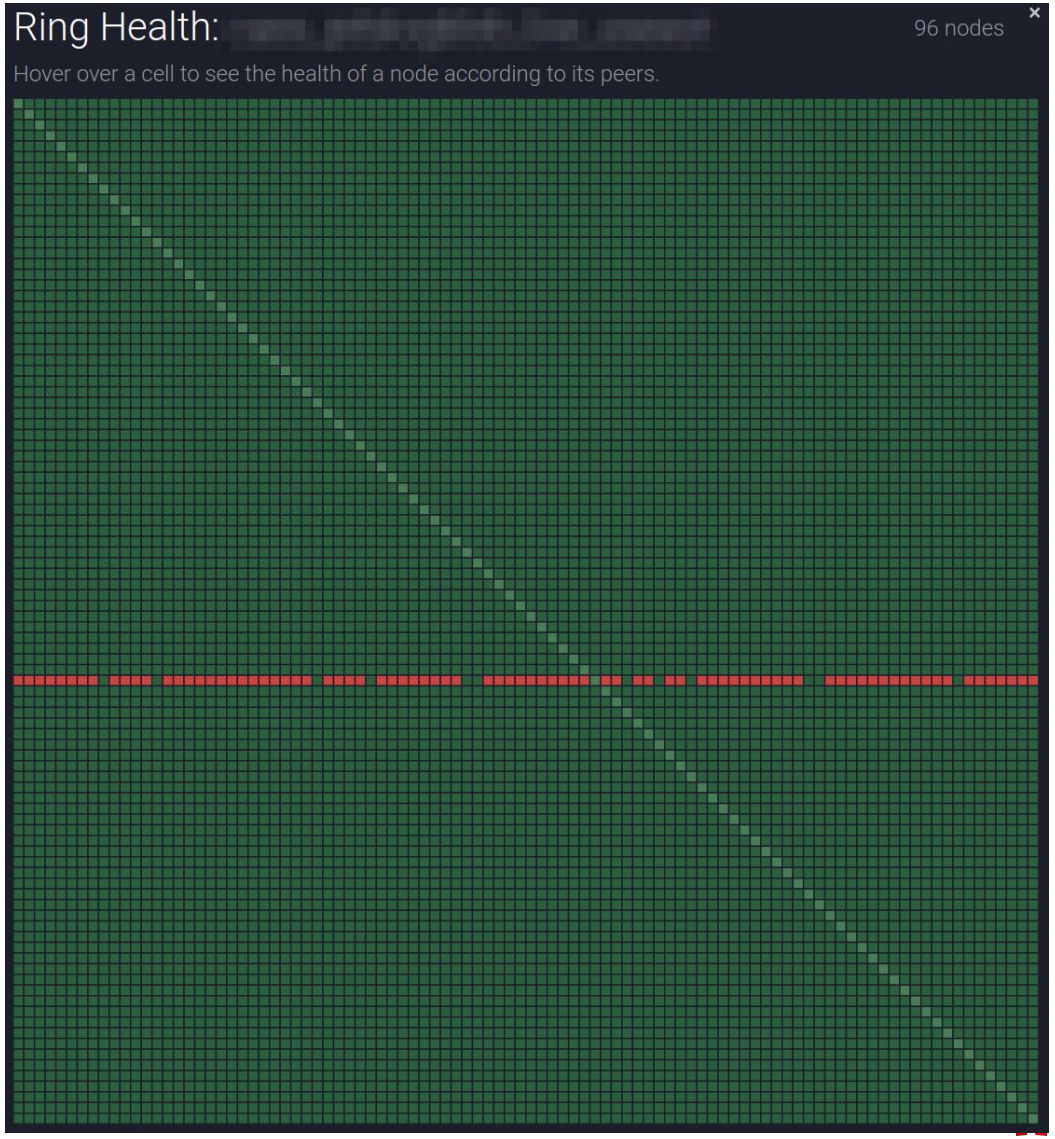
— O(#nodes) commands

— Single Network Path

— Hard to synthesize global view

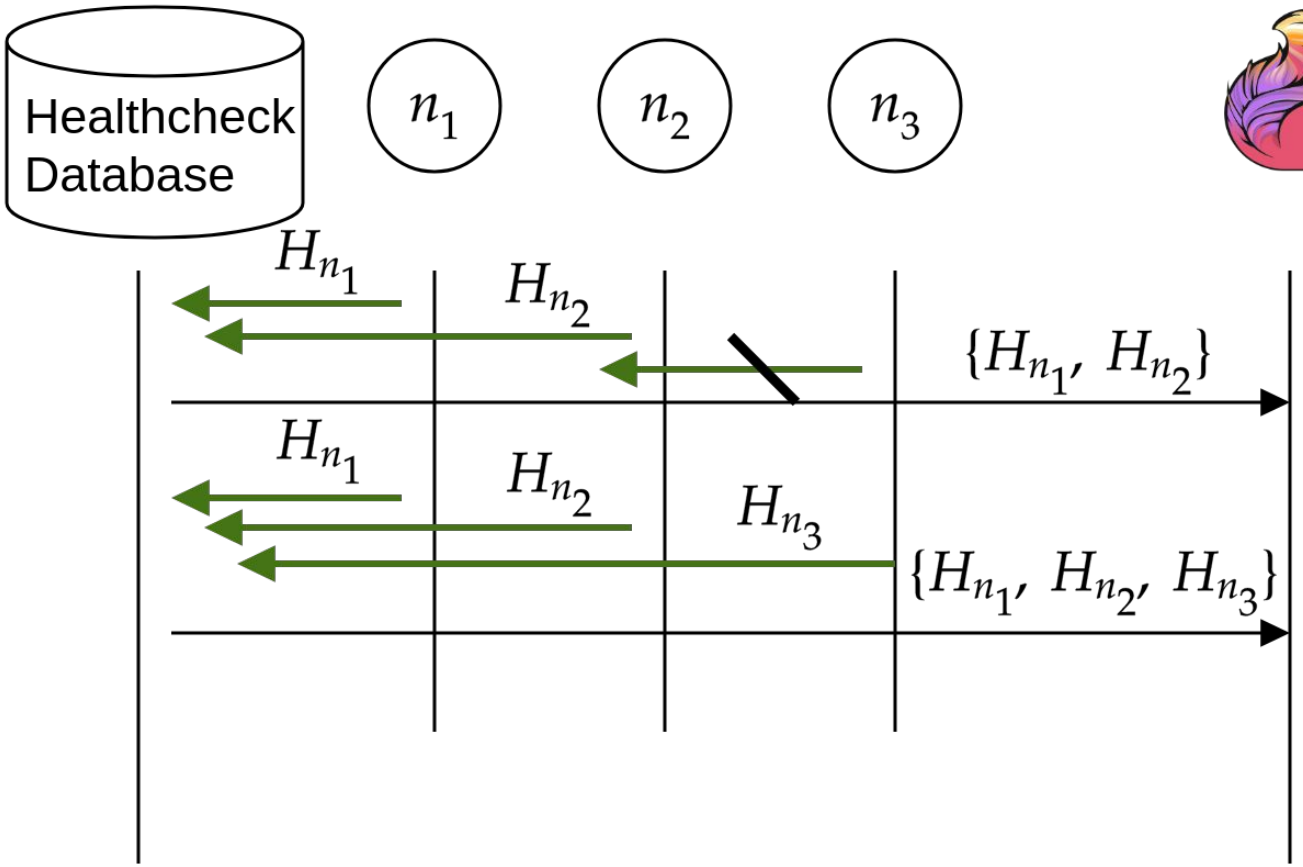
Ring Health *Declarative* Solution





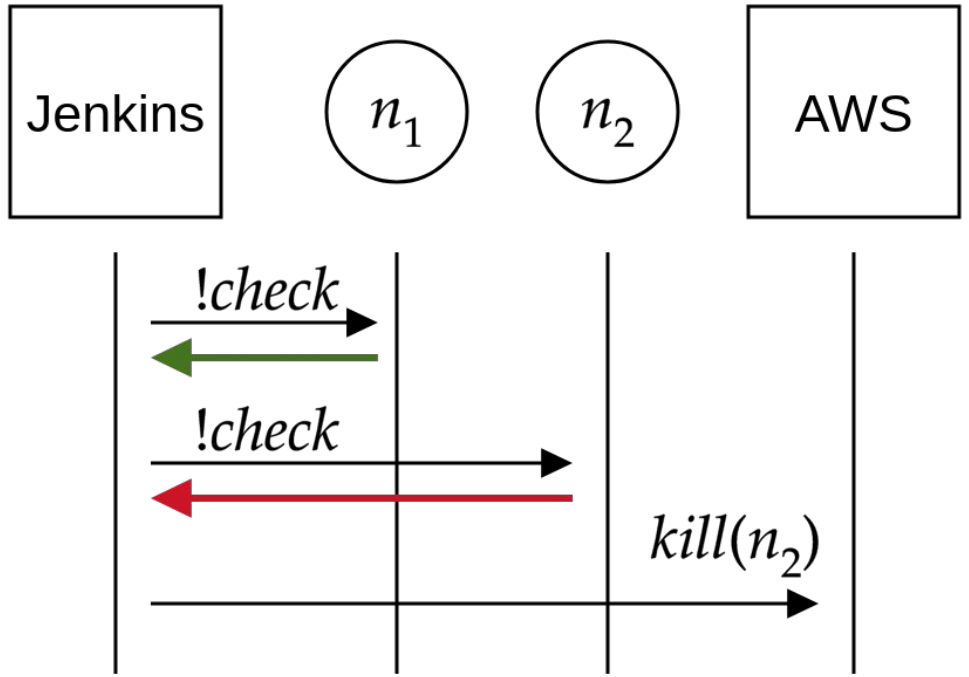
Stream Processing!

Ring Health *Declarative* Solution



- Desire: Healthy <30m
- Observe: Full cluster state
- Action: Remediate or page

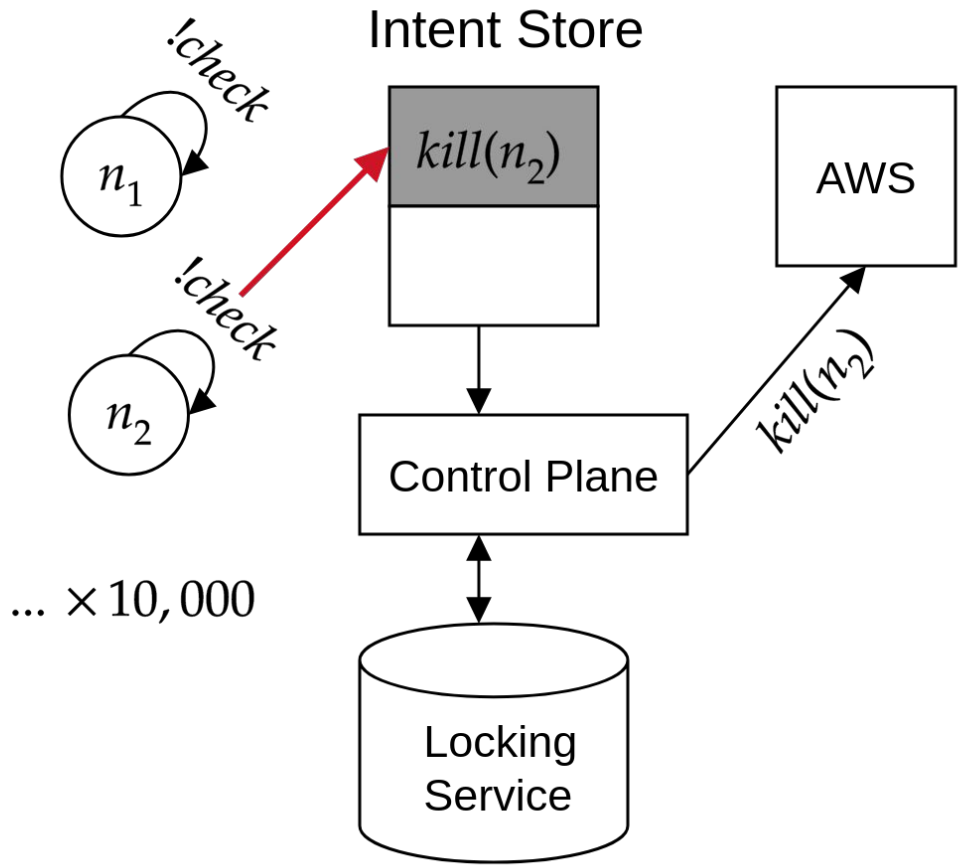
Hardware Health *Imperative* Solution



$\ldots \times 10,000$

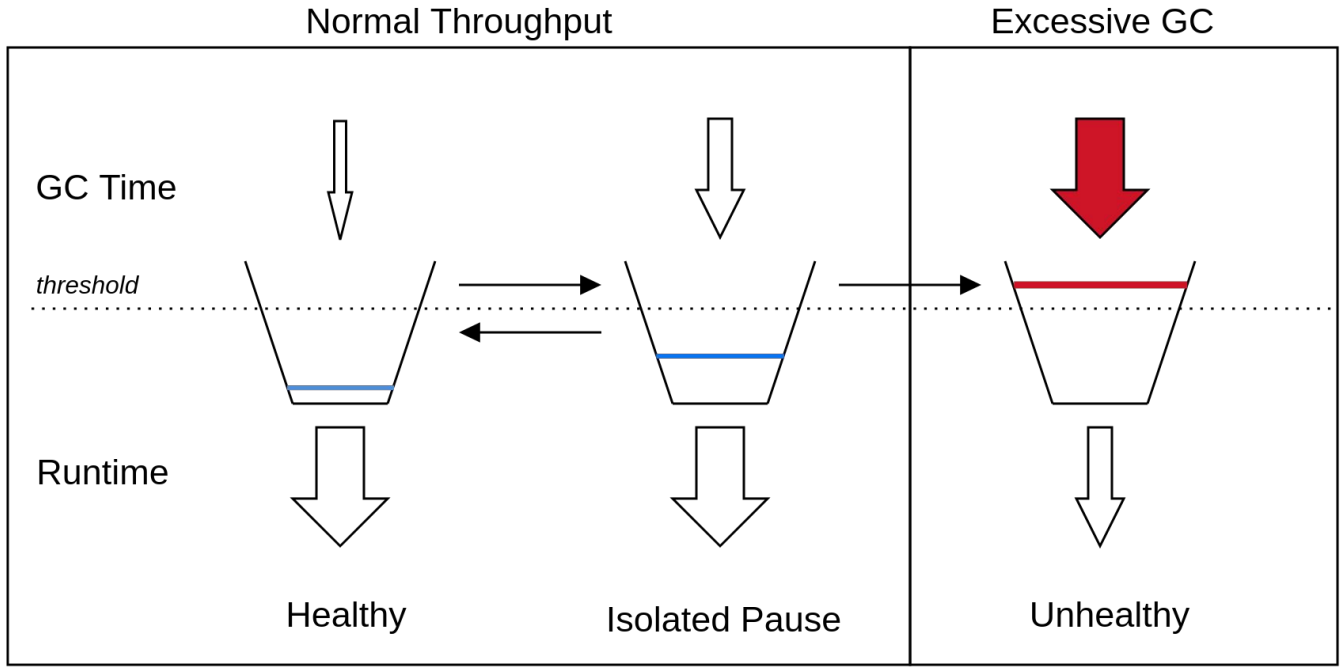
- O(#nodes) messages
- Degraded hardware is hard to SSH into
  - Failed ephemerals
  - Failed network

Hardware Health *Declarative* Solution



- Desire: Degraded hardware should die <10m
- Observe: Local disks, creds, filesystems
- Action: Request termination

JVM Health *Declarative* Solution

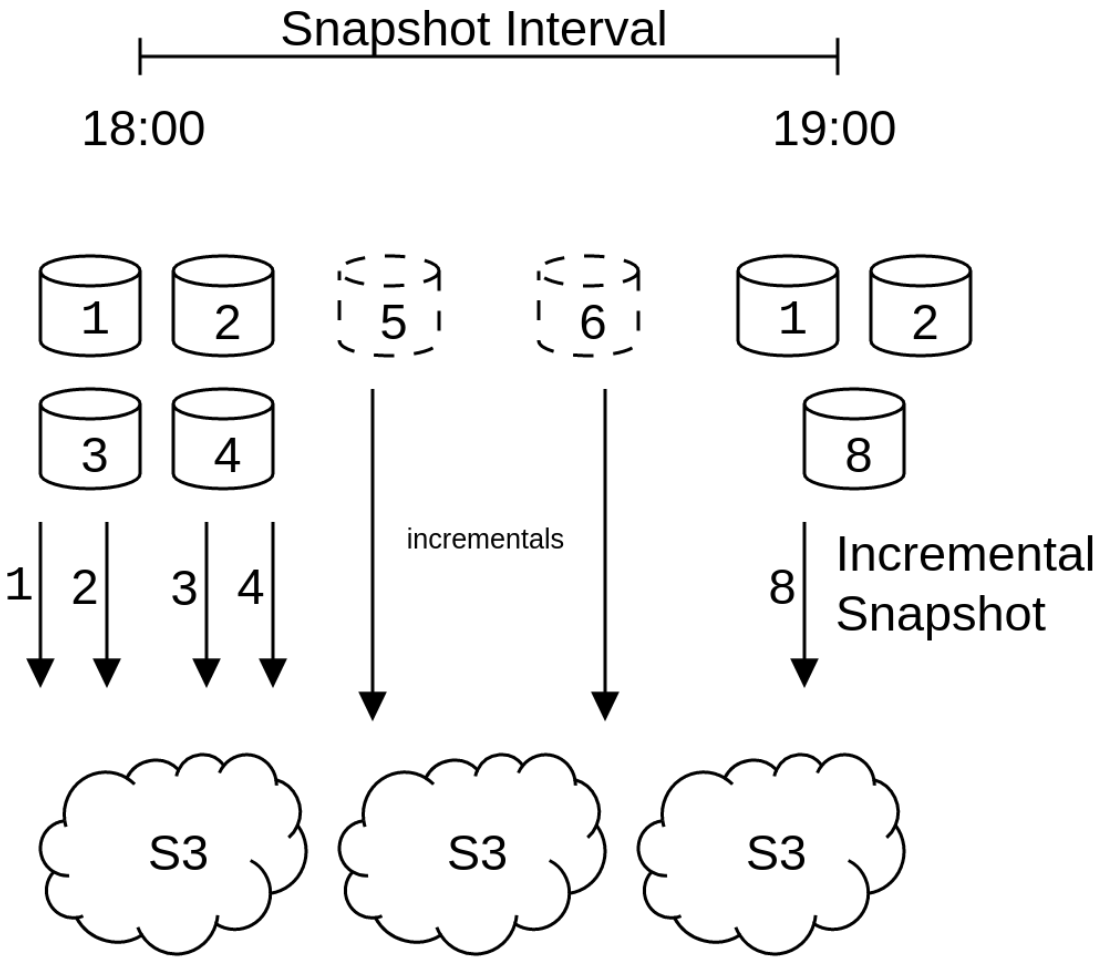


https://github.com/Netflix-Skunkworks/jvmquake

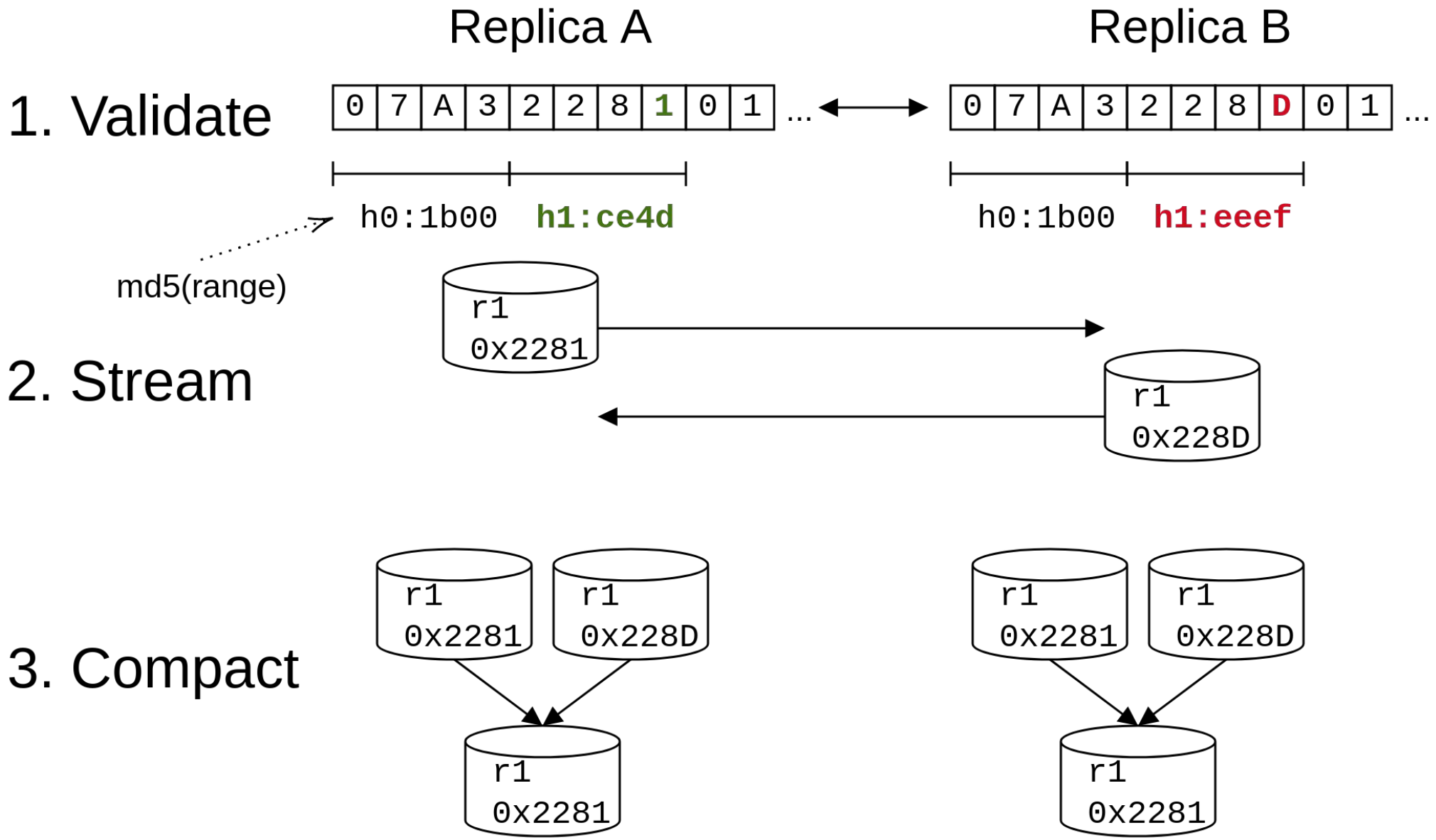
- Desire: JVM Throughput > 90%
- Observe: GC vs Runtime
- Action: core dump + terminate

Backup *Declarative* Solution

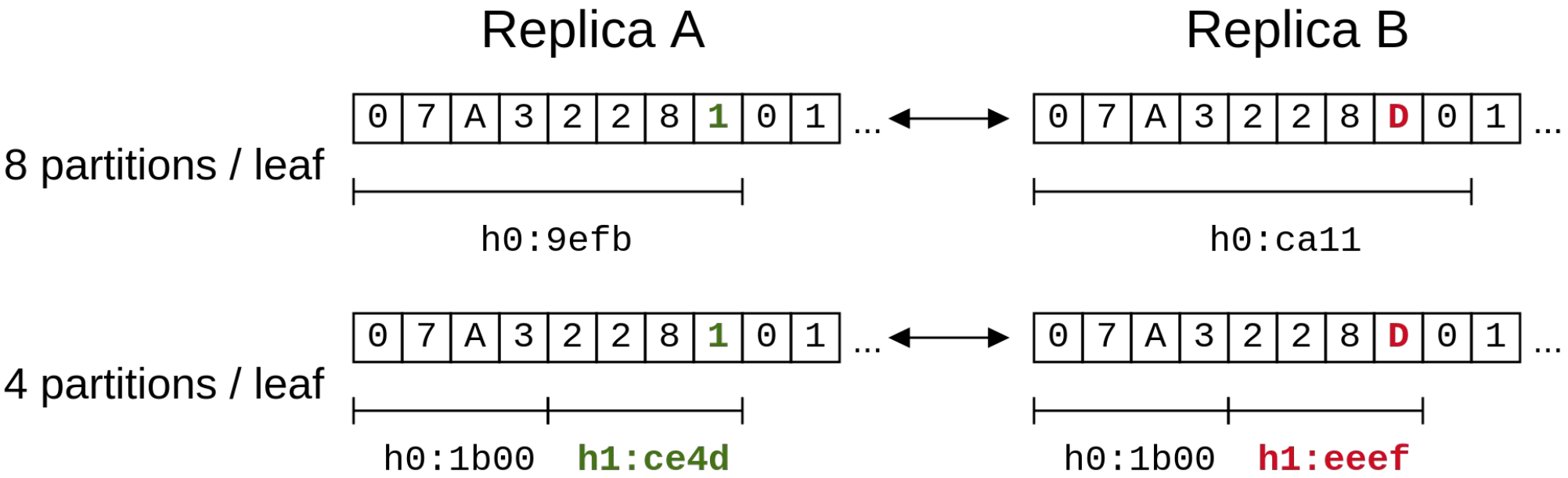
- Desire: Full snapshot every interval
- Observe:
  - State of data
  - State of S3
- Action:
  - Upload diff



Case Study: Cassandra Repair



Case Study: Cassandra Repair



$$resolution = \frac{|partitions|}{|leaf|}$$

$$cost_{memory} \approx 192\ bytes * |leaf|$$

Due to this Cassandra must limit single tree $|leaf|$ to $\lessapprox 2^{18}$

At Scale Subrange is Required

$$192 \frac{bytes}{partition} * 2^{27} \frac{partition}{replica} \approx 24 \frac{GiB}{replica}$$

$$\rightarrow \frac{24\ GiB}{replica} * 9\ replica \approx 216\ GiB$$

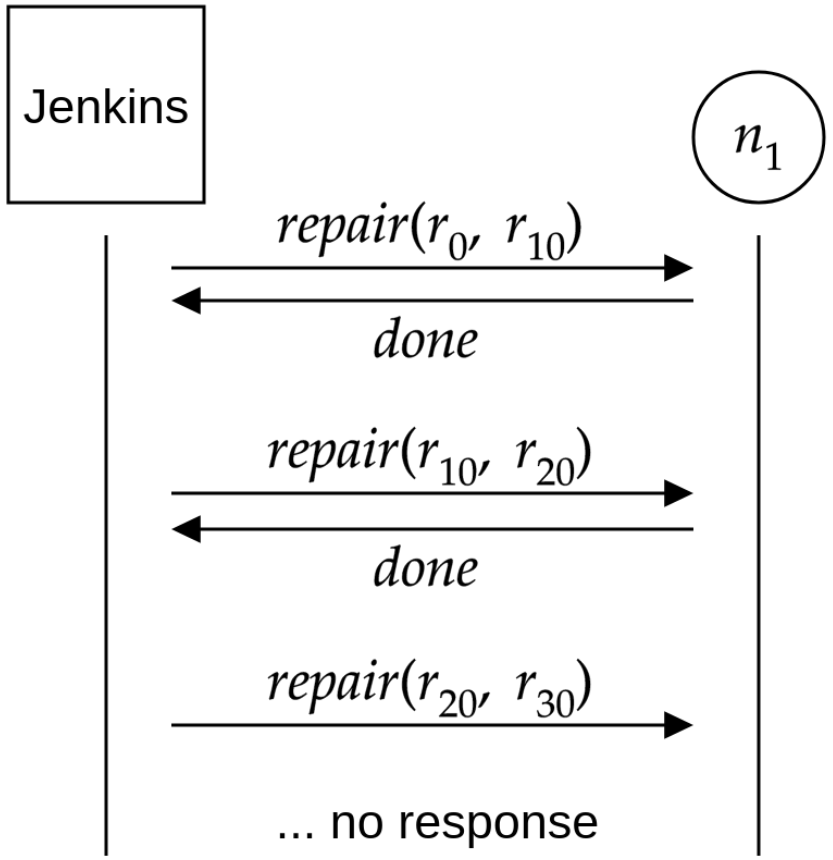
We can OOM

OR

$$resolution \approx \frac{2^{27} partitions}{2^{18}\ leaves} \approx 512$$

Do 512x work ...

Repair *Imperative* Solution



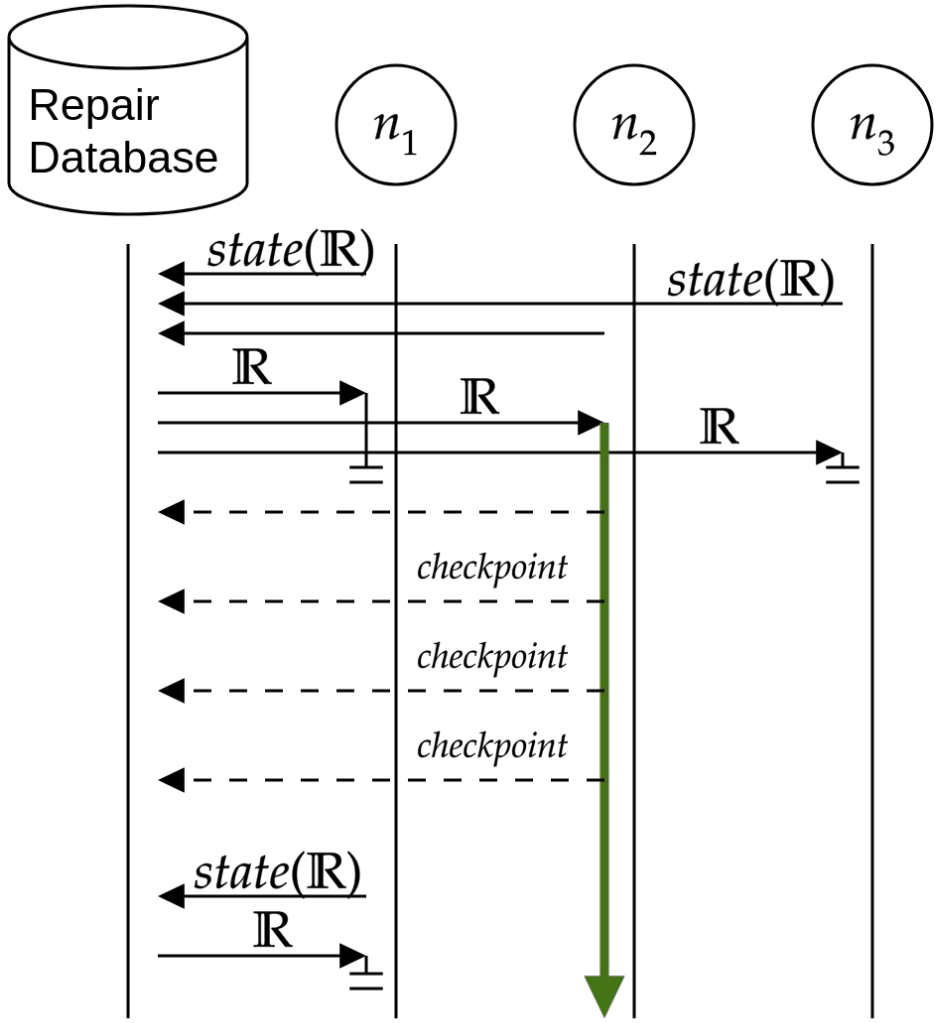
— O(lots) commands

— Remote JMX = :(

— **S**ingle **p**oint **o**f **f**ailure = :(

— 1 month job duration = D:

Repair *Declarative* Solution

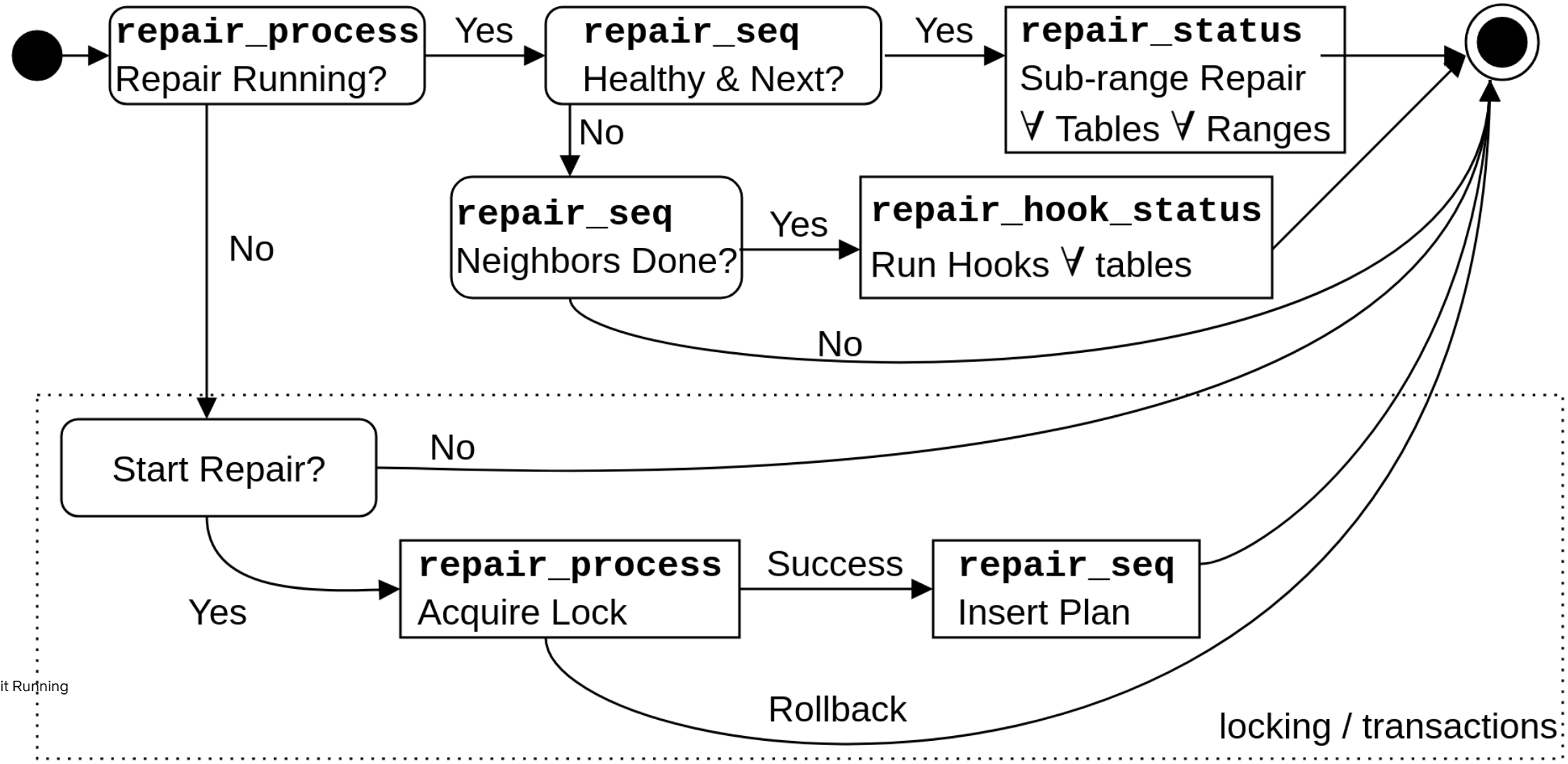


- Distributed Control Plane
- Makes progress when database is available
- Every node follows **repair state machine**

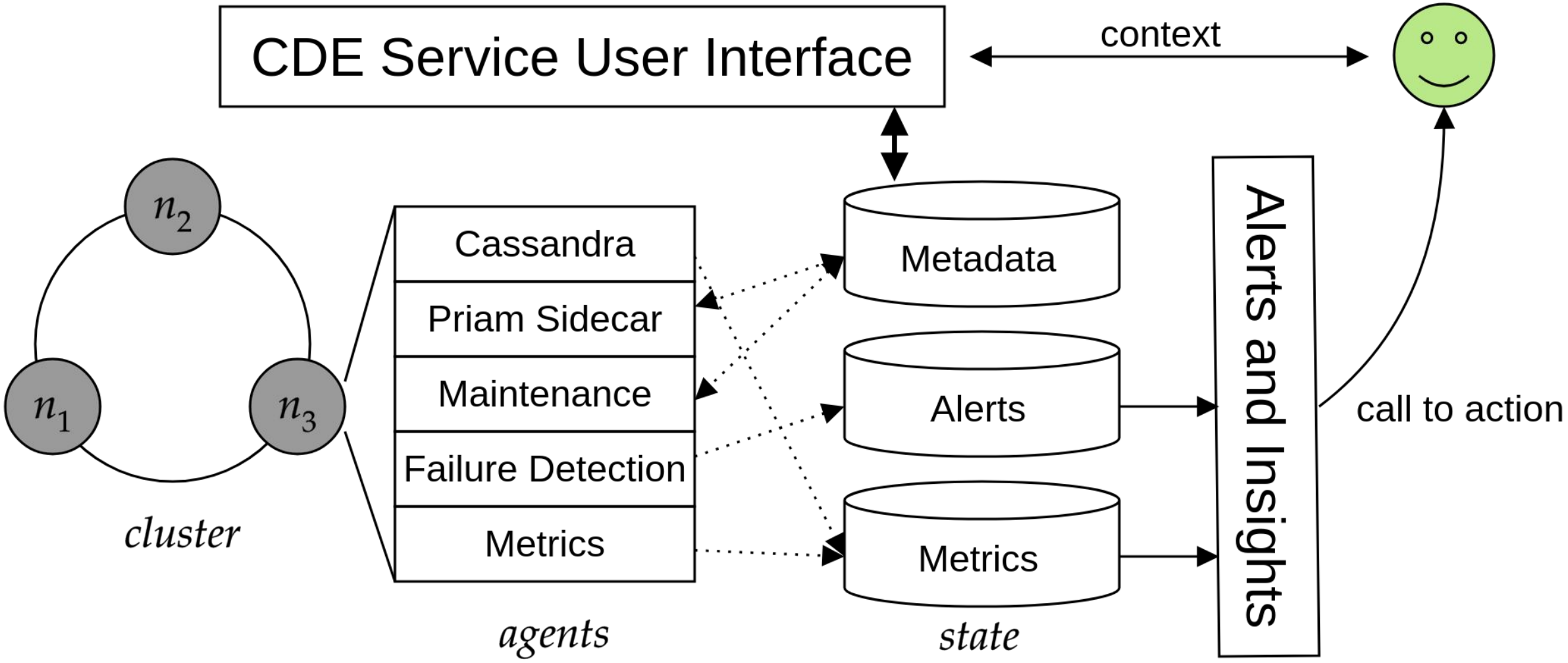
https://issues.apache.org/jira/browse/CASSANDRA-14346

Repair *Declarative* Solution (#14346)

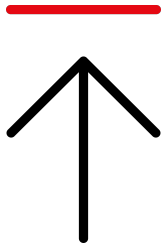
Repair State Machine



Cassandra Ecosystem

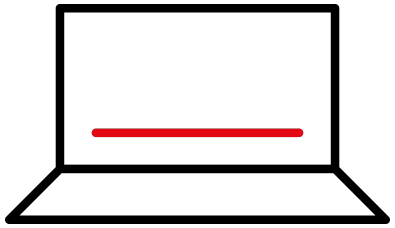


Story of C*

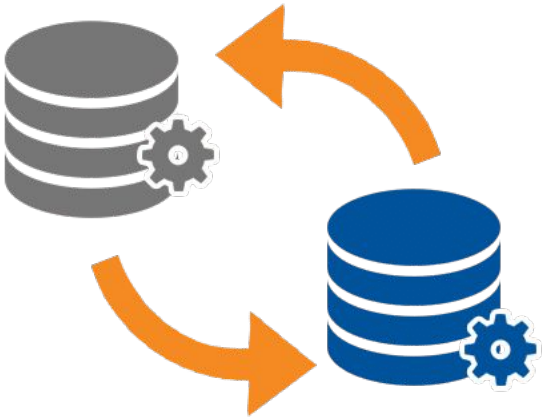


Inception

Provision



Keep it running

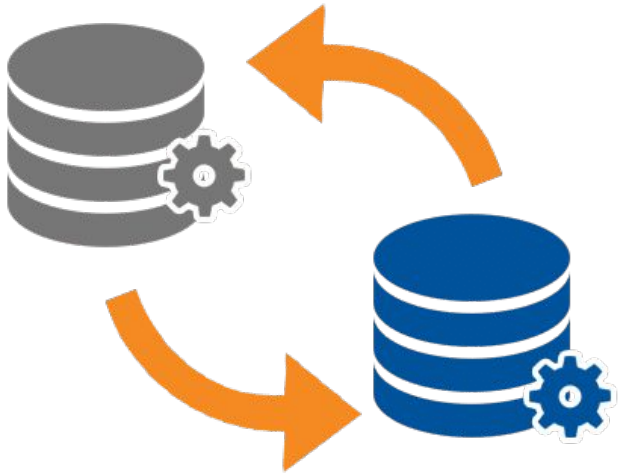


Migration

ANYTHING
THAT CAN GO WRONG
WILL GO WRONG

Murphy's Law

Migration

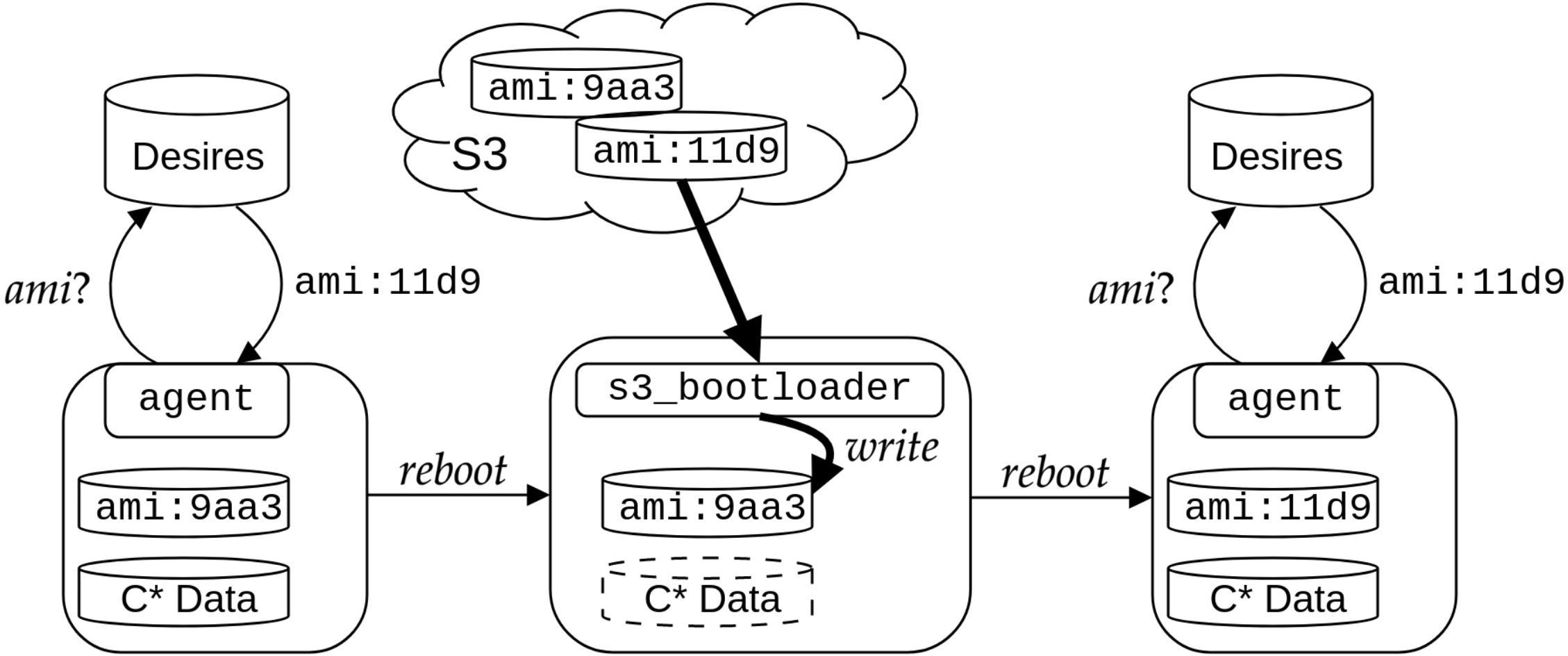


Migration Philosophy

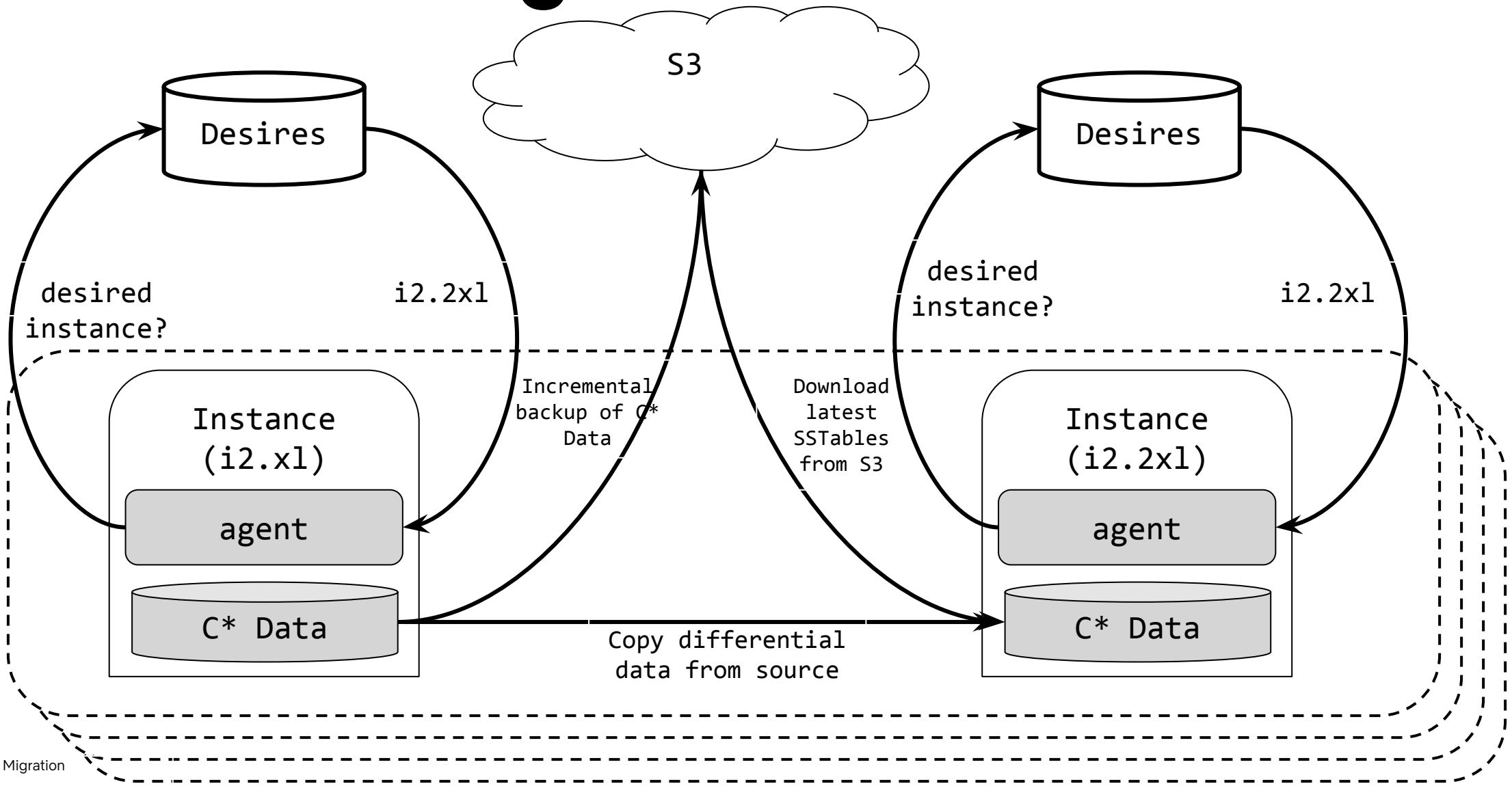
— Immutable infrastructure

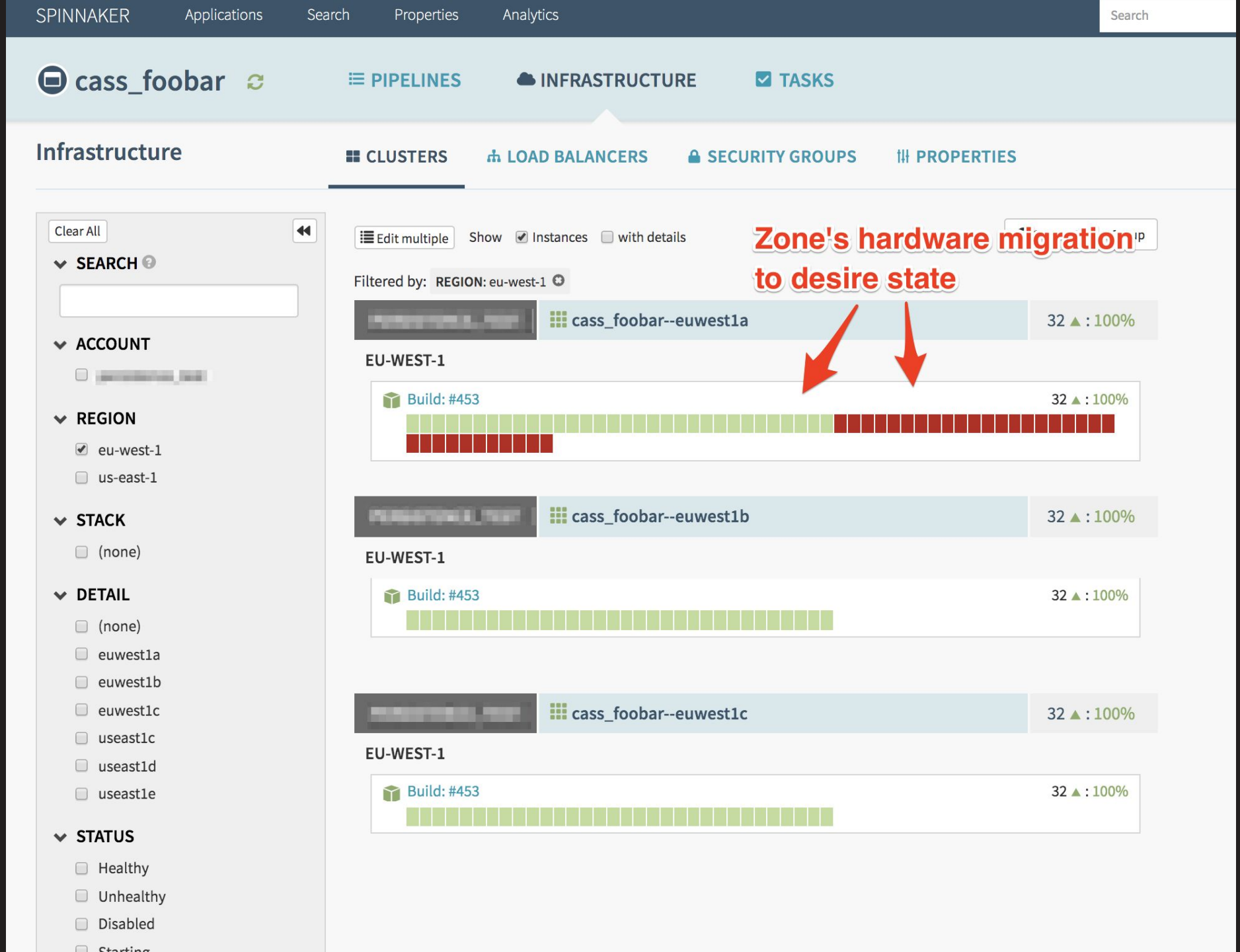
— Seamless and transparent migrations

Software Migration



Hardware Migration




SPINNAKER
Applications
Search
Properties
Analytics
Search
cass_foobar
PIPELINES
INFRASTRUCTURE
TASKS
Infrastructure
CLUSTERS
LOAD BALANCERS
SECURITY GROUPS
PROPERTIES
Clear All
SEARCH
ACCOUNT
REGION
eu-west-1
us-east-1
STACK
(none)
DETAIL
(none)
euwest1a
euwest1b
euwest1c
useast1c
useast1d
useast1e
STATUS
Healthy
Unhealthy
Disabled
Edit multiple
Show
Instances
with details
Filtered by:
REGION: eu-west-1
Zone's hardware migration
to desire state
cass_foobar--euwest1a
32 ▲ : 100%
EU-WEST-1
Build: #453
32 ▲ : 100%
cass_foobar--euwest1b
32 ▲ : 100%
EU-WEST-1
Build: #453
32 ▲ : 100%
cass_foobar--euwest1c
32 ▲ : 100%
EU-WEST-1
Build: #453
32 ▲ : 100%

Hardware Migration: The Numbers

$$500 \frac{\text{GiB}}{node} \;/\; 200 \frac{\text{mb}}{second} = 5.96 \frac{hours}{node}$$

$$5.96 \frac{hours}{node} * 288\ node = \mathbf{2.35\ months}$$

$$500\ \text{GiB} \;/\; 2000 \frac{\text{mb}}{second} = 0.6\ hours$$

$$0.6\ hours + 10 \frac{minutes}{node} * 288\ node = \mathbf{2.02\ days}$$

$$500\ \text{GiB} \;/\; 2000 \frac{\text{mb}}{second} = 0.6\ hours$$

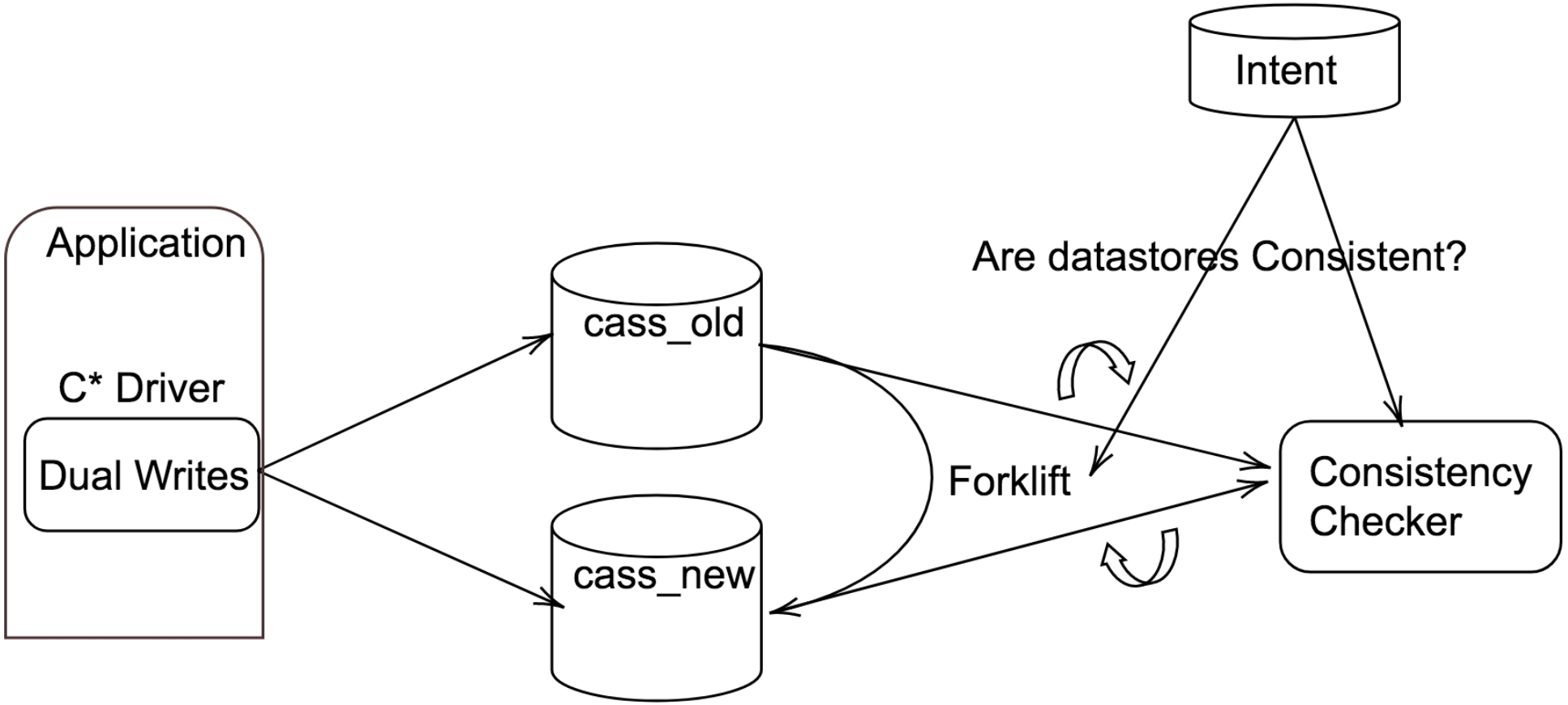
$$0.6\ hours + 20 \frac{minutes}{zone} * 3\ zone = \mathbf{1.596\ hours}$$

Naive solution, sequential transfer

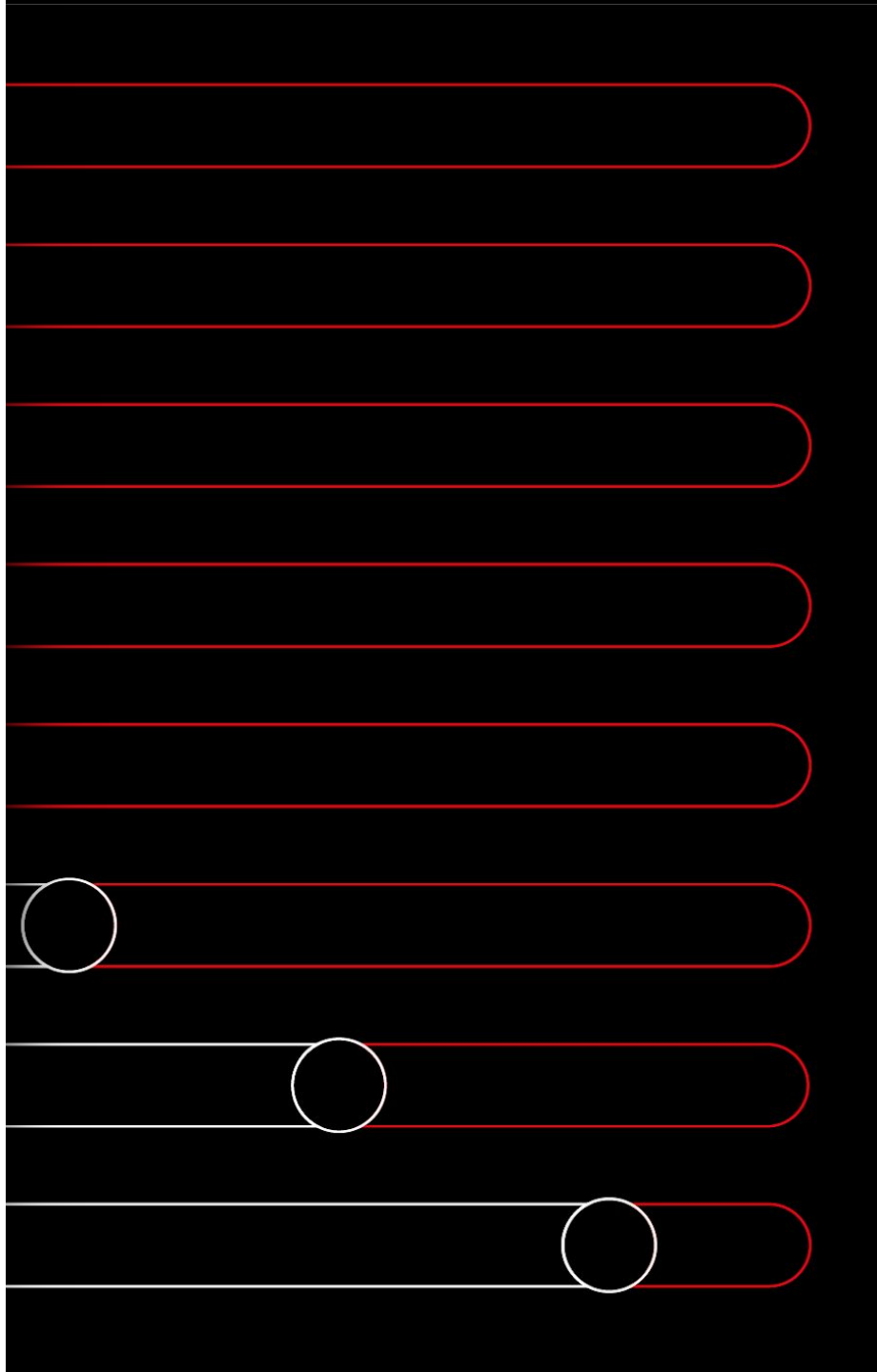
Parallel S3 download sequential fixup

Parallel download and zone wide fixup

Data Migration



Murphy’s law



Cautionary Tale: Rarely run automation

- Auto remediation to clean up stale backups, repair snapshots
  - Triggers only above FATAL disk threshold
  - Assumption mismatch

Cautionary Tale: Rarely run automation

- Auto remediation to clean up stale backups, repair snapshots
  - Triggers only above FATAL disk threshold
  - Assumption mismatch

- Result:
  - Automation ended up deleting real backup data

Do you restore your backups constantly?

Take Away

Invest in context sharing, tooling and declarative(desire) infrastructure to run Apache Cassandra at petabyte scale

Thank You.