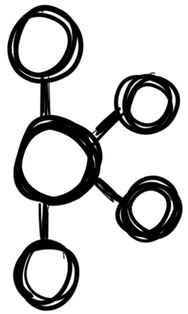
Running large Kafka clusters with minimum toil

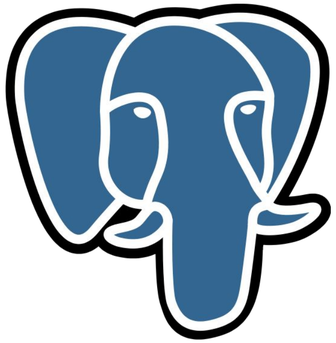
Balthazar Rouberol
DRE - Datadog

Who am I?

Balthazar Rouberol

Senior Data Reliability Engineer, Datadog





Our Kafka infrastructure

– Multiple regions / datacenters / cloud providers

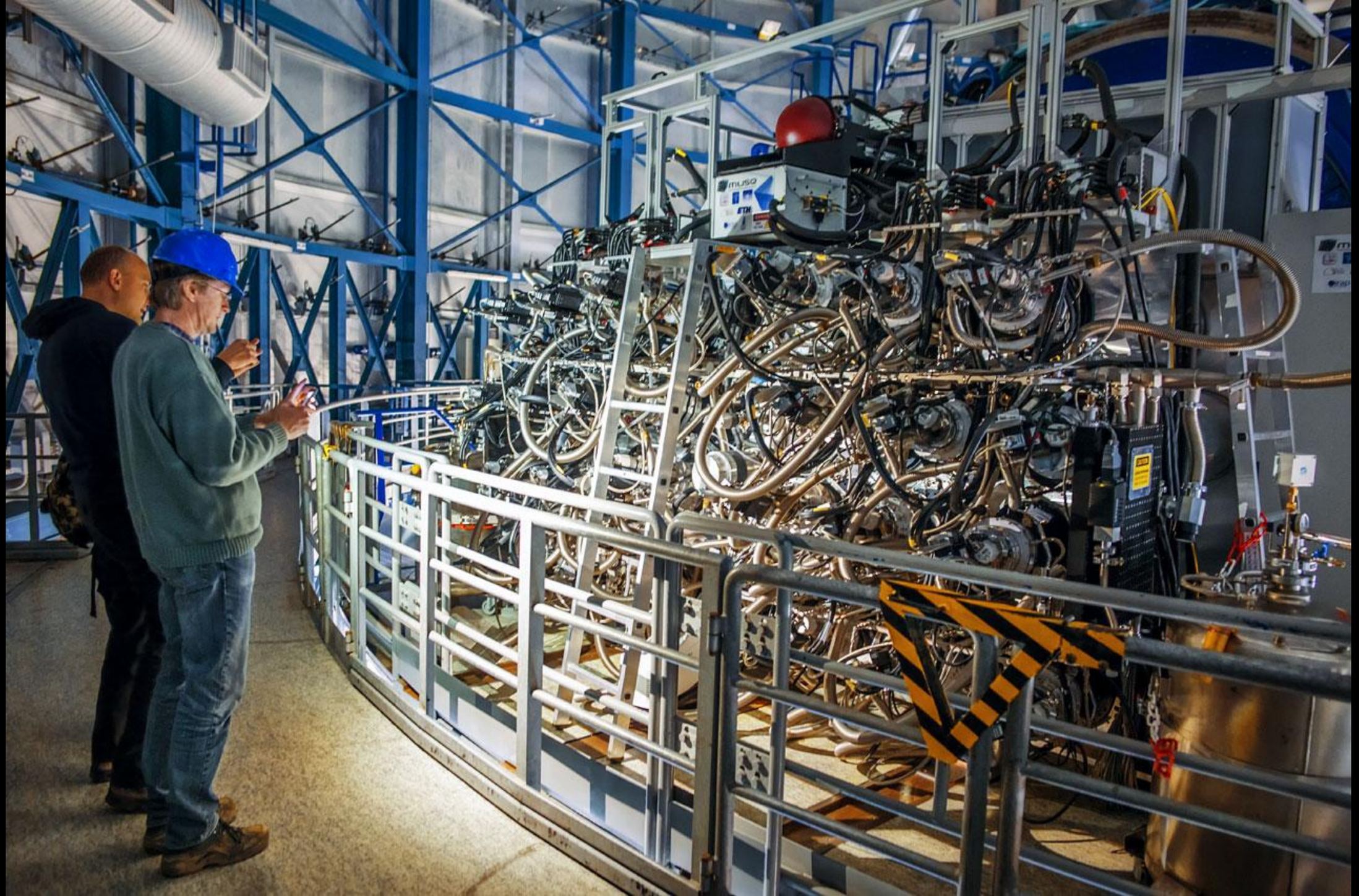
– dozens of Kafka/ZooKeeper clusters

– PB of data on local storage

– Trillions of messages per day

– Double-digit GB/s bandwidth

– 2 (mostly) dedicated SREs


muse

toil

/tɔɪl/

verb

work extremely hard or incessantly.
"we toiled away"

Similar: work hard | labour | work one's fingers to the bone | work like a Trojan

noun

exhausting physical labour.
"a life of toil"

Similar: hard work | toiling | labour | slaving | struggle | effort

What can go wrong?

- Disk full
- Broker dead
- Storage hotspot
- Network hotspot
- Hot reassignment
- Expired SSL certificates
- $$$
- Computers

What can be time consuming?

- Partition assignment calculation
- Investigating under-replication
- Replacing brokers
- Adjusting reassignment throttle
- Scaling up / down
- Computers
- Humans

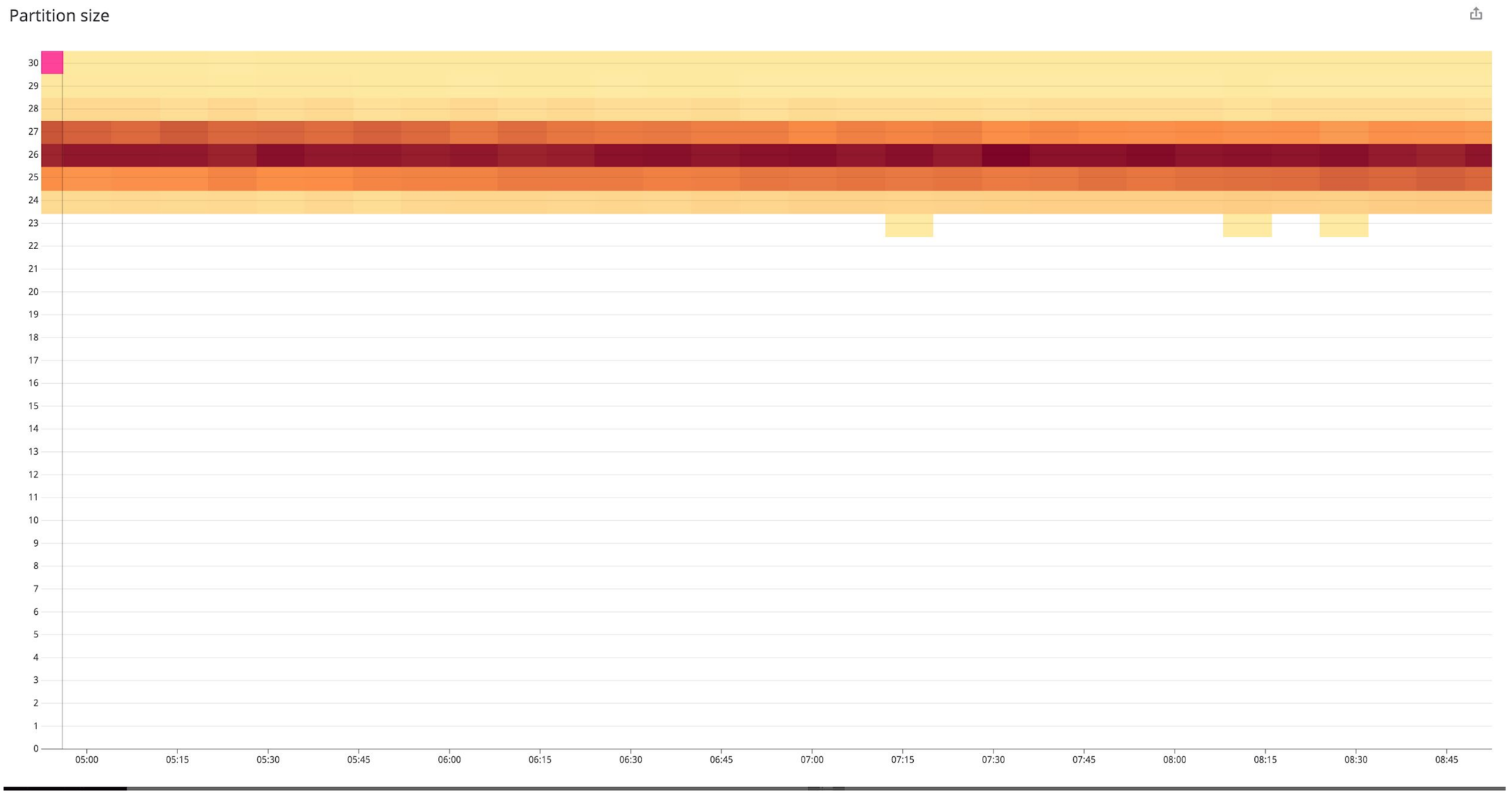
Tooling

Getting partition assignment right

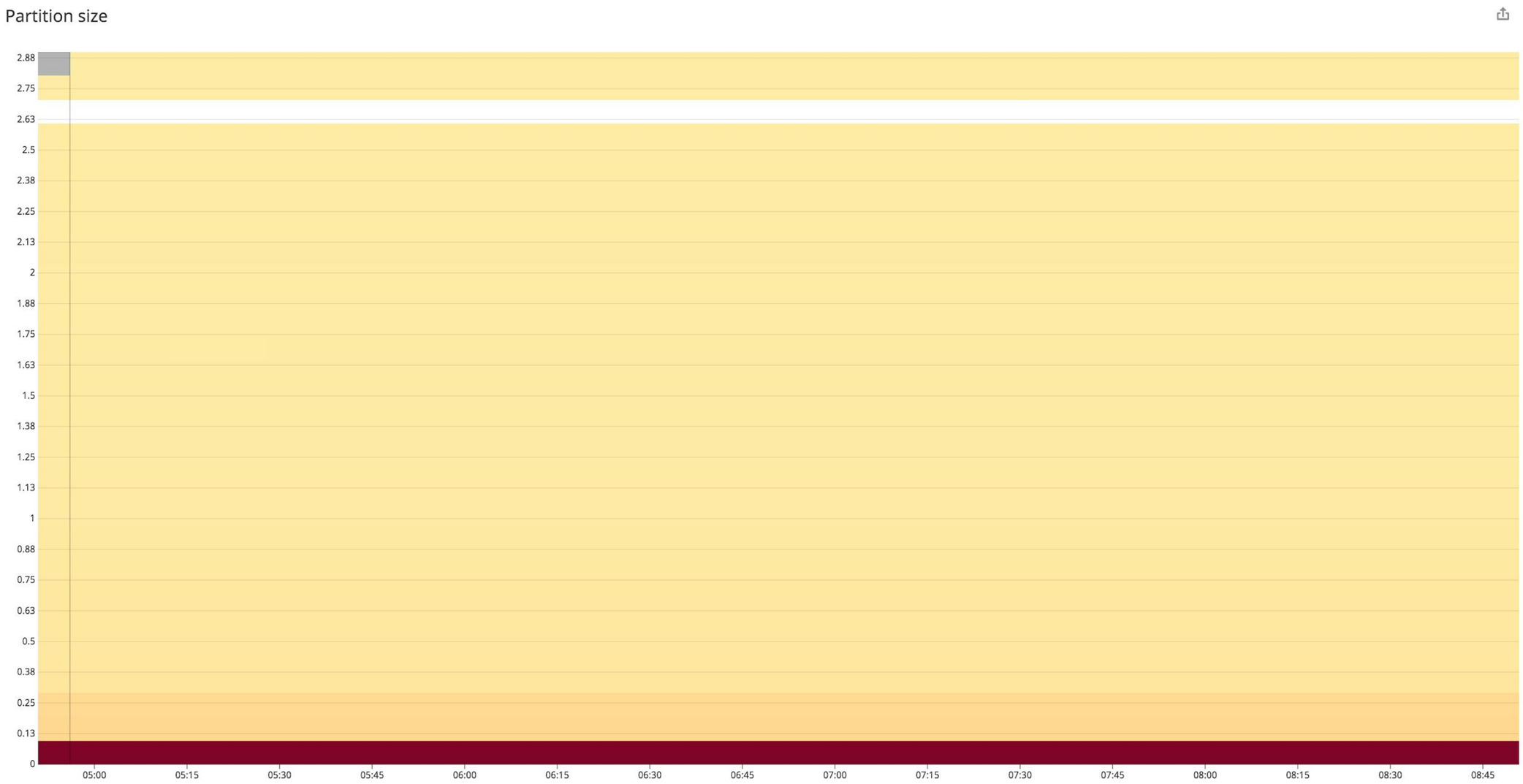
A good partition assignment enforces rack balancing and de-hotspots

- disk usage
- network throughput
- leadership

Homogeneous partition size?



Homogeneous partition size?



Enters **topicmappr**

```
Usage:
  topicmappr [command]

Available Commands:
  help        Help about any command
  rebalance   Rebalance partition allotments among a set of topics and brokers
  rebuild     Rebuild a partition map for one or more topics
  version     Print the version
```

https://github.com/datadog/kafka-kit

`topicmappr rebuild`

```
$ topicmappr rebuild --topics <regex> --brokers <csv>
```

- Assumes homogeneous partition size by default
- Can binpack on partition sizes and disk usage
- Possible optimizations:
  - Partition spread
  - Storage homogeneity
  - Leadership / broker

Broker replacement

```
$ topicmappr rebuild --topics .* --brokers 1,3,4 --sub-affinity

Broker change summary:
  Broker 2 marked for removal
  New broker 4
```

Change replication factor

```
$ topicmappr rebuild --topics test --brokers -1 --replication 2

Topics:
  test

Action:
  Setting replication factor to 2

Partition map changes:
  test p0: [12 11 13] -> [12 11] decreased replication
  test p1: [9 10 8] -> [9 10] decreased replication
```

`topicmappr rebalance`

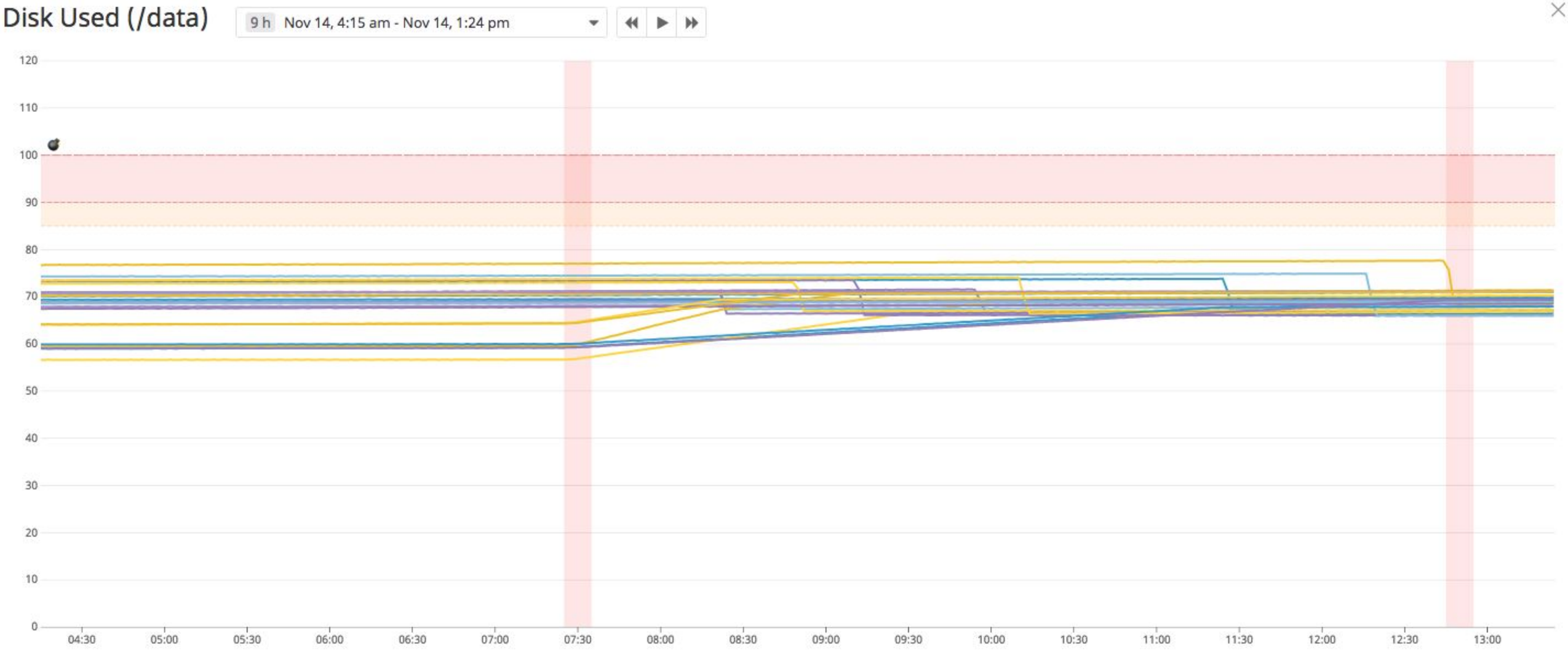
```
$ topicmappr rebalance --topics <regex> --brokers <csv>
```

- targeted broker storage rebalancing (partial moves)
- incremental scaling
- AZ-local traffic (free $$$)

In-place rebalancing

```
$ topicmappr rebalance --topics .* --brokers -1

  Storage free change estimations:
   range: 131.07GB -> 27.85GB
   range spread: 39.90% -> 1.92%
   std. deviation: 40.07GB -> 10.11GB
```

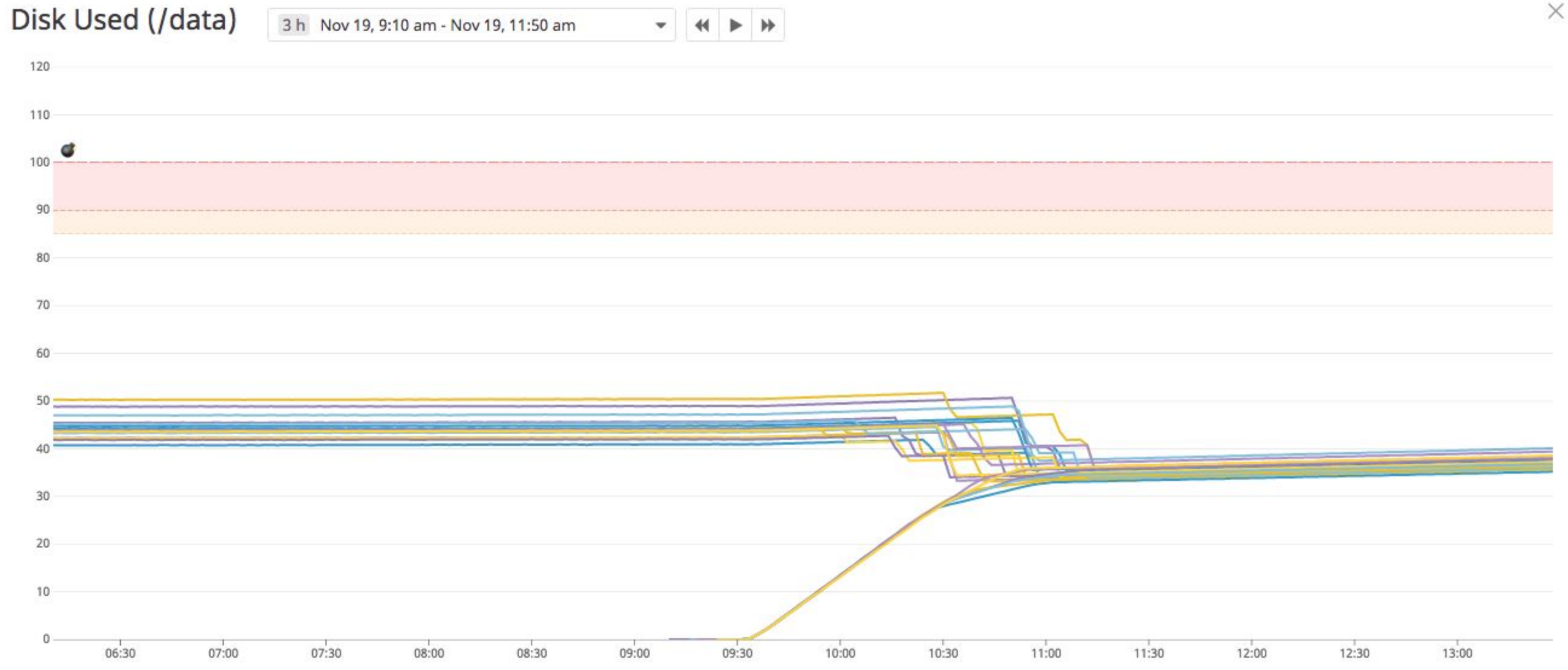
Disk Used (/data)



Scale up + rebalance

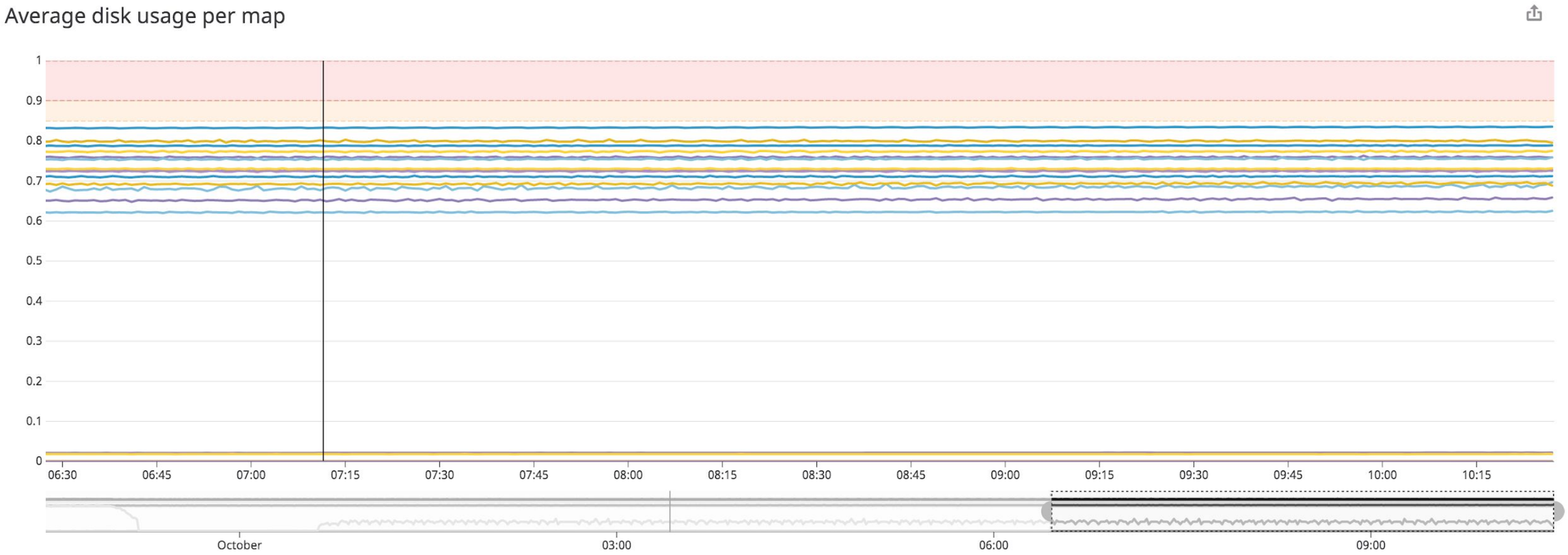
```
$ topicmappr rebalance --topics .* --brokers -1,101,102,103

Storage free change estimations:
  range: 330.33GB -> 149.22GB
  range spread: 19.12% -> 6.70%
  std. deviation: 79.92GB -> 38.49GB
```

Disk Used (/data)

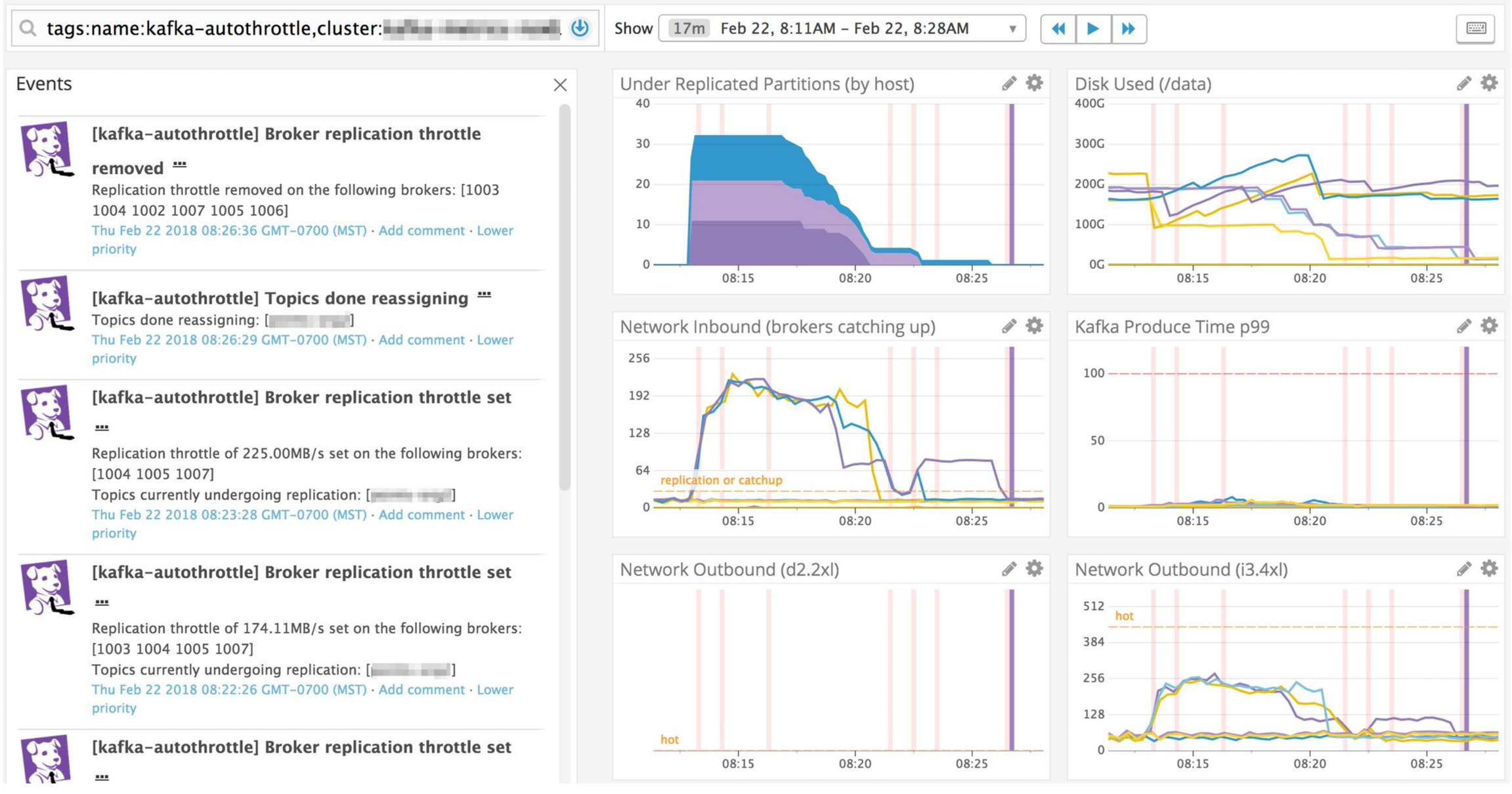


Capacity model

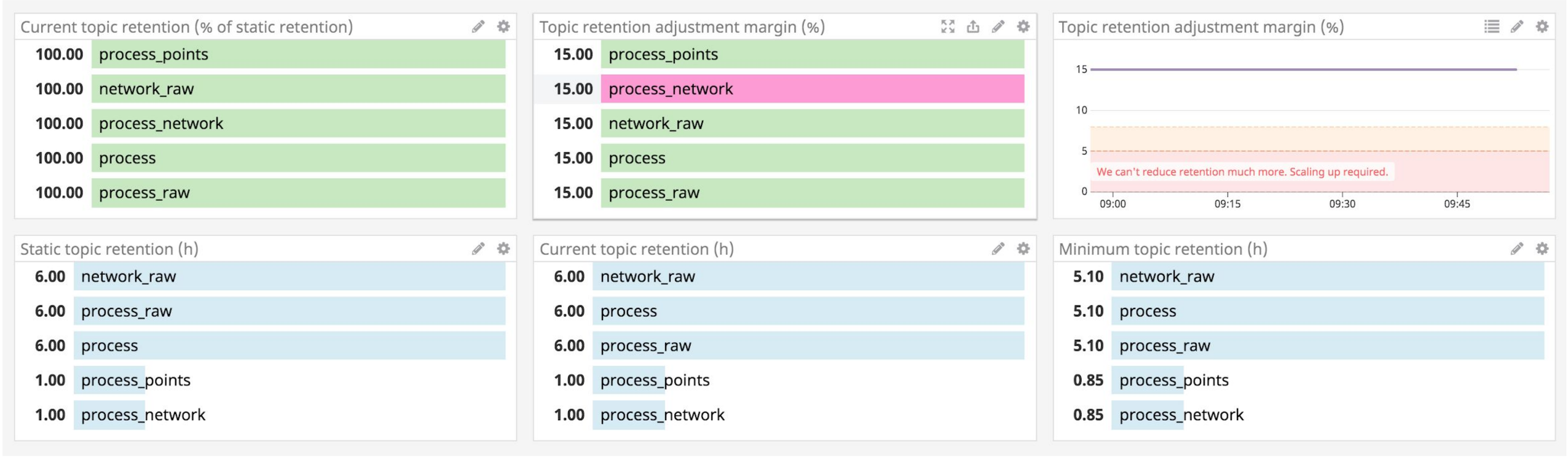
- ~80-85% {disk storage, bandwidth} per broker pool
- Rebalance first, scale up with leeway



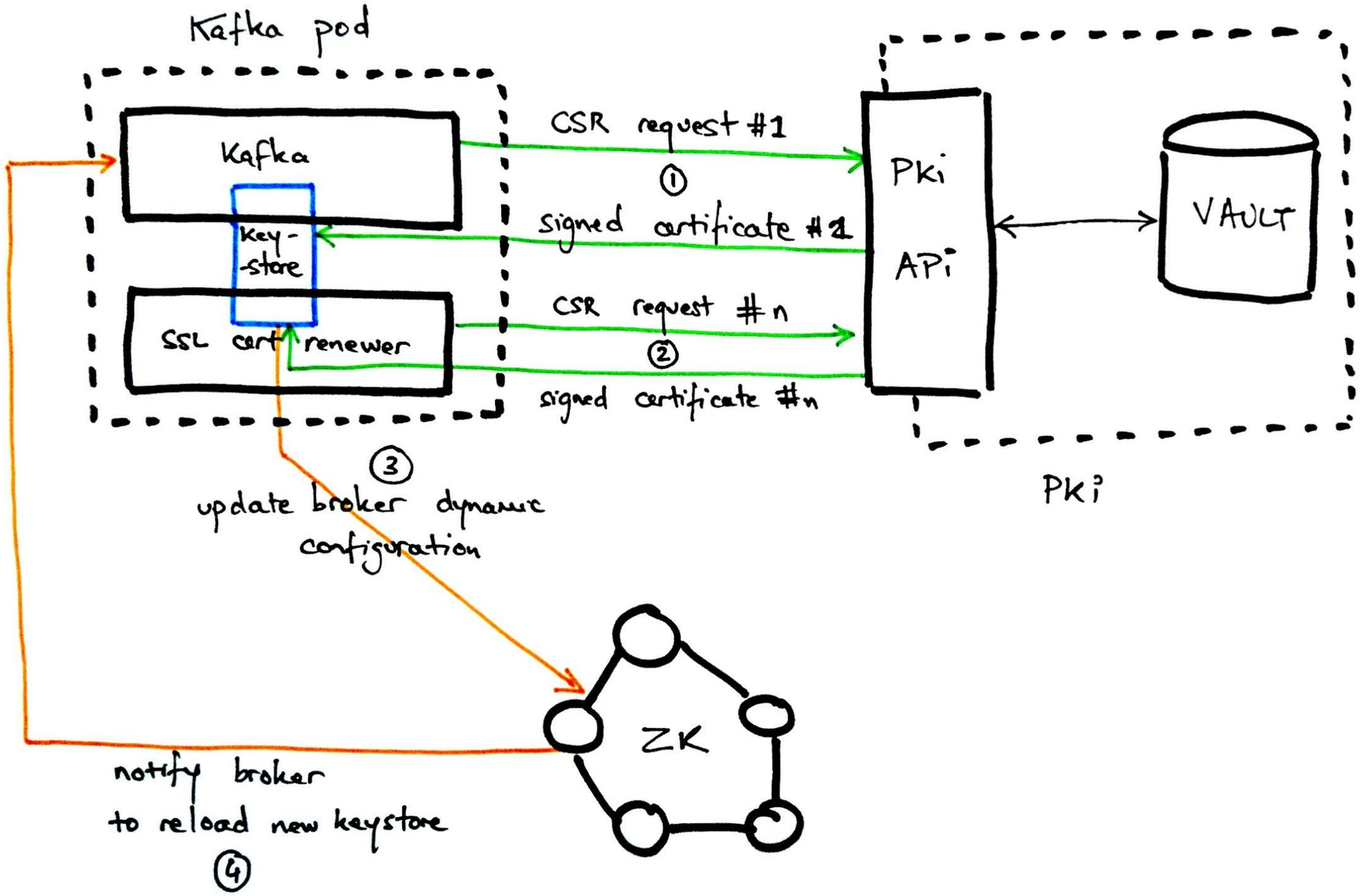
autothrottle: reassign fast enough



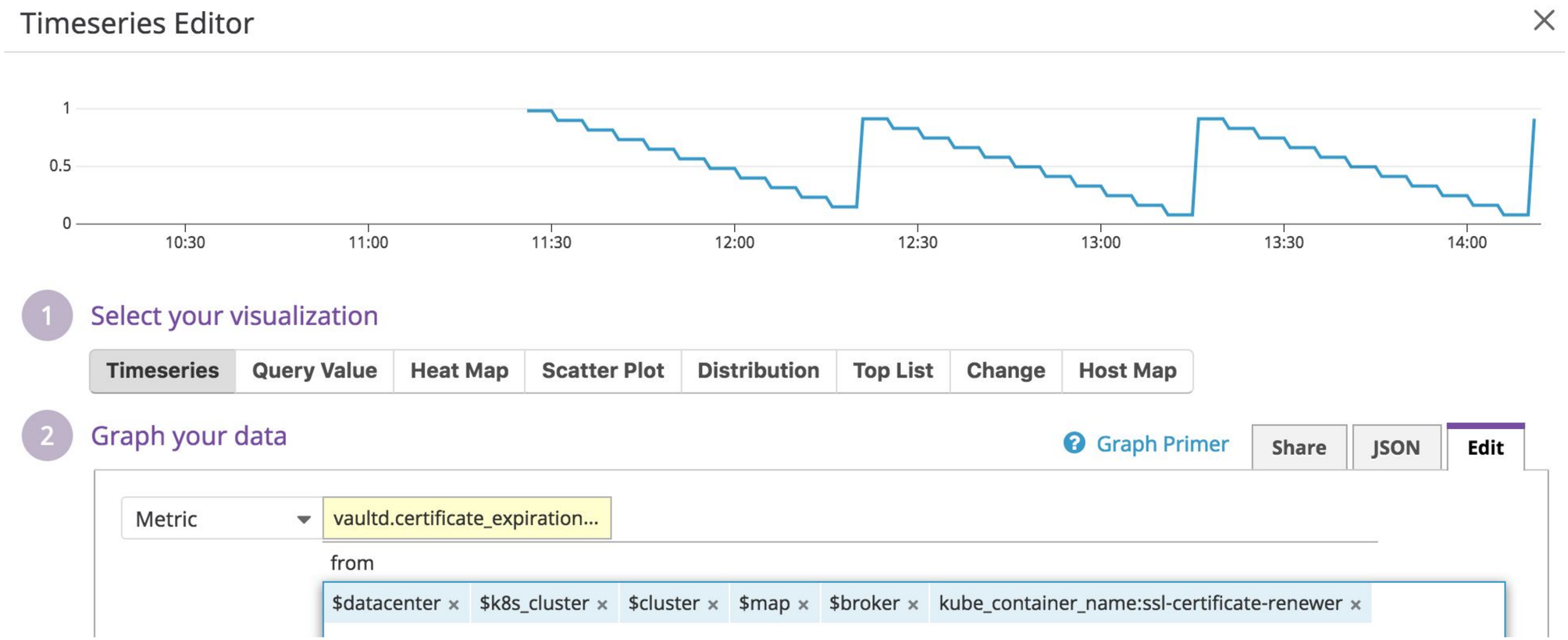
Adjust retention, don’t page



SSL certificates hot reloading



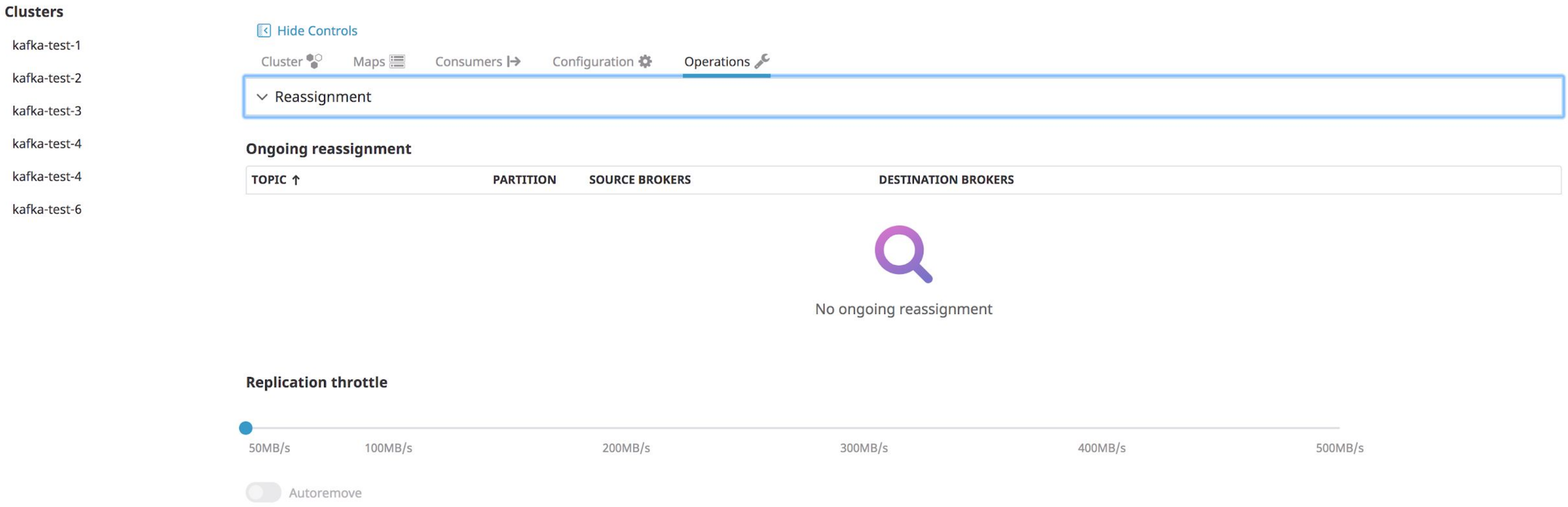
SSL certificates hot reloading



Make everything discoverable

```
$ autothrottle-cli get

no throttle override is set


$ curl localhost:8080/api/kafka/ops/throttle
{
  "throttle": null,
  "autoremove": false
}
```

Build layered tooling



Monitoring

Monitoring

- Storage hotspot (>90%)
- Sustained elevated traffic
- Under replication by topic/cluster
- Long running reassignment
- Replication factor = 1
- Set write success SLI/SLO
- SSL certificate TTL

Monitoring: under-replication

– Alert by topic or even cluster
– Exports tagged partition metrics
– Automatically muted during rolling-restarts

[Triggered on {cluster:kafka-demo,datacenter:demo-dc,kafka_topic:test-topic}] [kafka] Under-replicated topic #cluster:kafka-demo ...

Check the under-replicated topic wiki page.

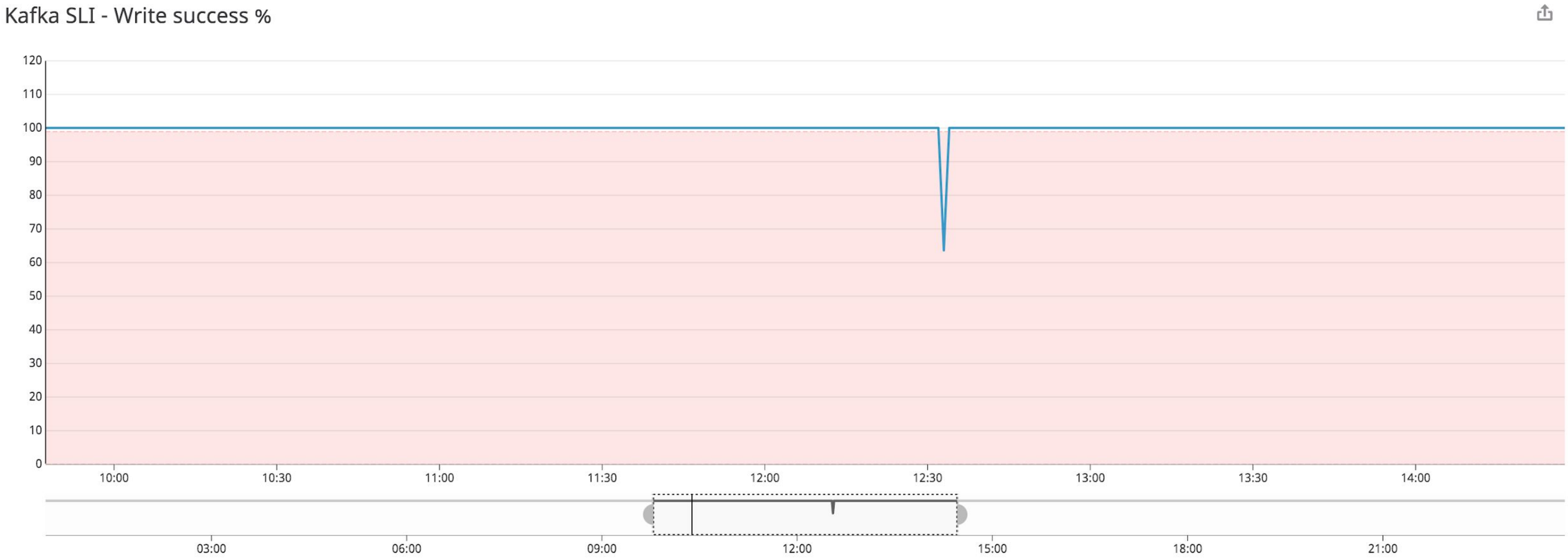
cluster:kafka-demo,datacenter:demo-dc,kafka_topic:test-topic: Topic: test-topic

Partition: 39 Current ISR: [1029] Expected ISR [1029, 1027] Out of sync replicas: 1027: 1.2.3.4

Partition: 38 Current ISR: [1029] Expected ISR [1029, 1027] Out of sync replicas: 1027: 1.2.3.4.

Partition: 59 Current ISR: [1047] Expected ISR [1047, 1027] Out of sync replicas: 1027: 1.2.3.4

Measure write success



Measure write success: poor man’s version

– Write synthetics data to a SLI topic

– Every broker is at least leader of a partition

– Should reflect write success

Conclusion

– Kafka admin tools are not sufficient at scale
– Measure partition volume
– Measure under-replication / topic
– Partition assignment is a machine job
– Know your bottleneck (storage / bandwidth)
– Make everything discoverable
– Monitor unsafe configuration
– Set write success SLO

Thanks!
Questions?