Scaling and operating Kafka in Kubernetes

Balthazar Rouberol - Jamie Alquiza
Datadog - Data Reliability Engineering team

NYC Kafka meetup - 2018/10/30

Who are we?

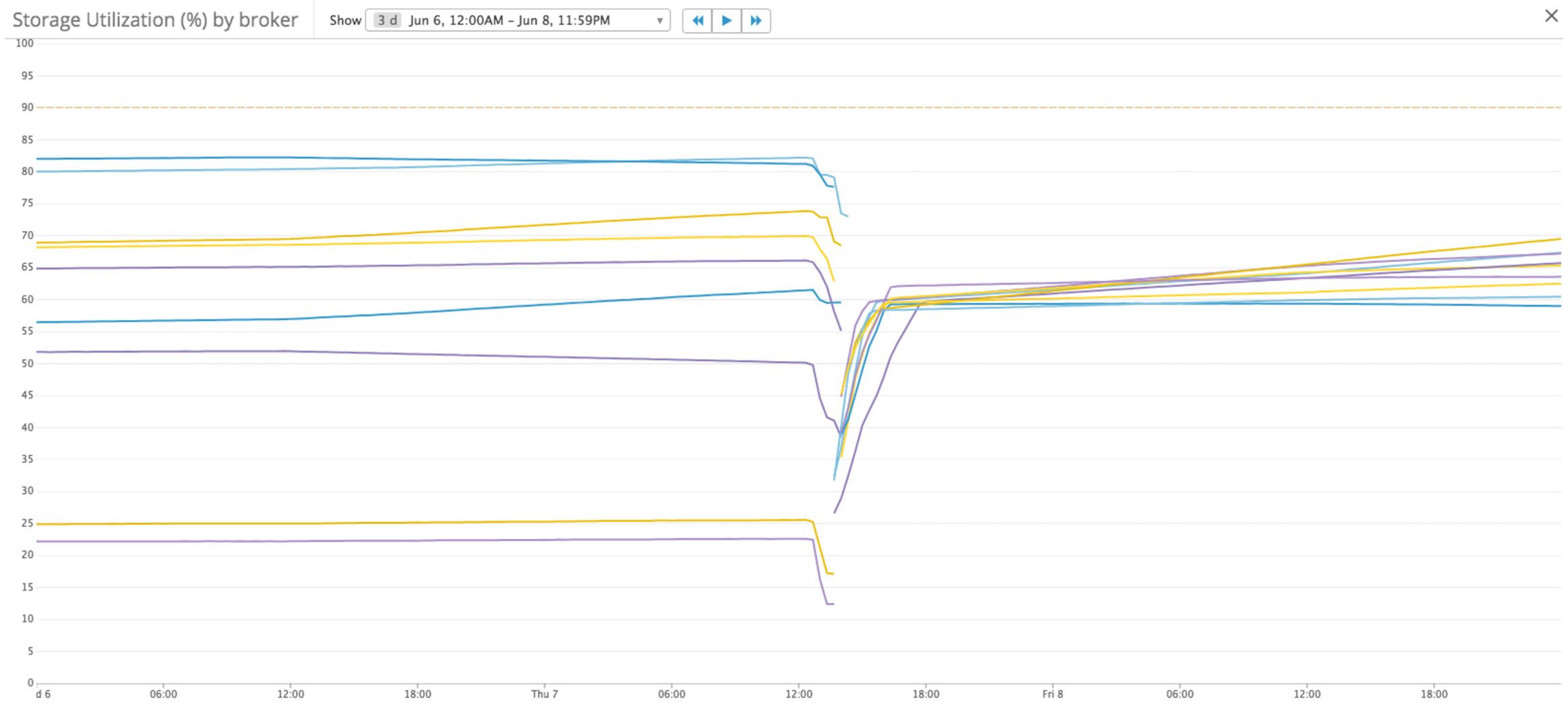
- Data Reliability Engineering: datastore reliability and availability, data security, data modeling, scaling, cost-control and tooling
- In charge of PostgreSQL, Kafka, ZooKeeper, Cassandra and Elasticsearch
- Team of 4 SREs
- @brouberol, @jamiealquiza
- We are hiring! https://www.datadoghq.com/careers

Our Kafka infrastructure

– Multiple regions
– 40+ Kafka/ZooKeeper clusters
– PB of data on local storage
– Trillions of messages per day
– Double-digit GB/s bandwidth
– 2 dedicated SREs

Kafka-Kit: scaling operations

- `topicmappr`:
  - partition to broker mapping
  - failed broker replacement
  - storage-based cluster rebalancing


Storage Utilization (%) by broker
Show
3 d
Jun 6, 12:00AM – Jun 8, 11:59PM
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
d 6
06:00
12:00
18:00
Thu 7
06:00
12:00
18:00
Fri 8
06:00
12:00
18:00

Kafka-Kit: scaling operations

- `topicmappr`:
  - partition to broker mapping
  - failed broker replacement
  - storage-based cluster rebalancing
- `autothrottle`: replication auto-throttling

tags:name:kafka-autothrottle,cluster:[redacted]

Show 17m Feb 22, 8:11AM – Feb 22, 8:28AM

Events

[kafka-autothrottle] Broker replication throttle removed ▪▪▪

Replication throttle removed on the following brokers: [1003 1004 1002 1007 1005 1006]

Thu Feb 22 2018 08:26:36 GMT-0700 (MST) · Add comment · Lower priority

[kafka-autothrottle] Topics done reassigning ▪▪▪

Topics done reassigning: [redacted]

Thu Feb 22 2018 08:26:29 GMT-0700 (MST) · Add comment · Lower priority

[kafka-autothrottle] Broker replication throttle set ▪▪▪

Replication throttle of 225.00MB/s set on the following brokers: [1004 1005 1007]

Topics currently undergoing replication: [redacted]

Thu Feb 22 2018 08:23:28 GMT-0700 (MST) · Add comment · Lower priority

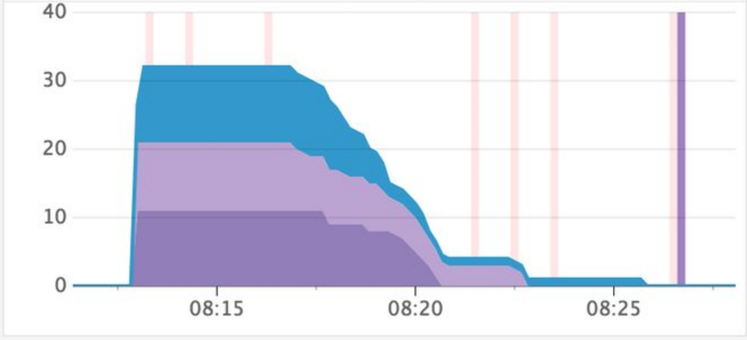
[kafka-autothrottle] Broker replication throttle set ▪▪▪

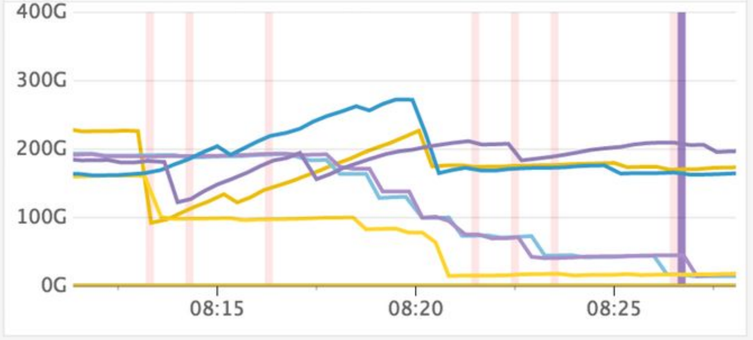
Replication throttle of 174.11MB/s set on the following brokers: [1003 1004 1005 1007]

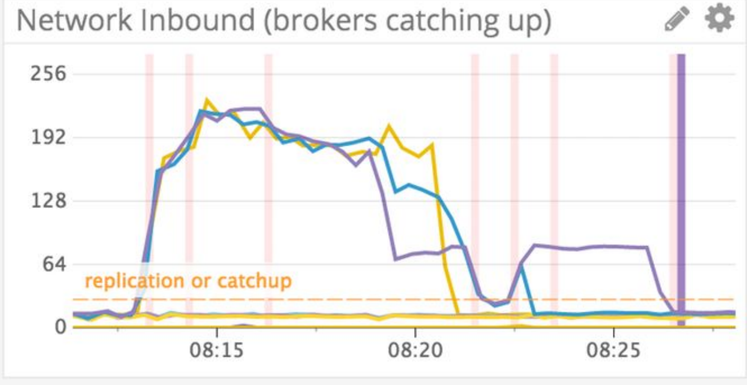
Topics currently undergoing replication: [redacted]

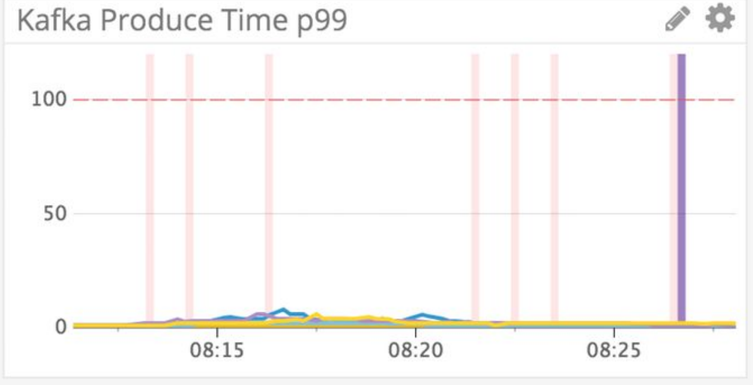
Thu Feb 22 2018 08:22:26 GMT-0700 (MST) · Add comment · Lower priority

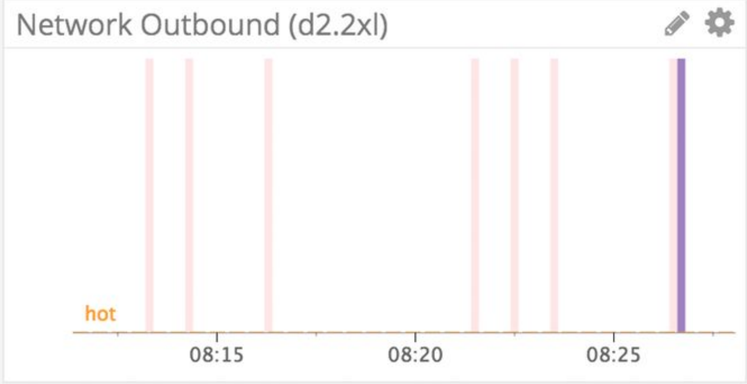
[kafka-autothrottle] Broker replication throttle set ▪▪▪

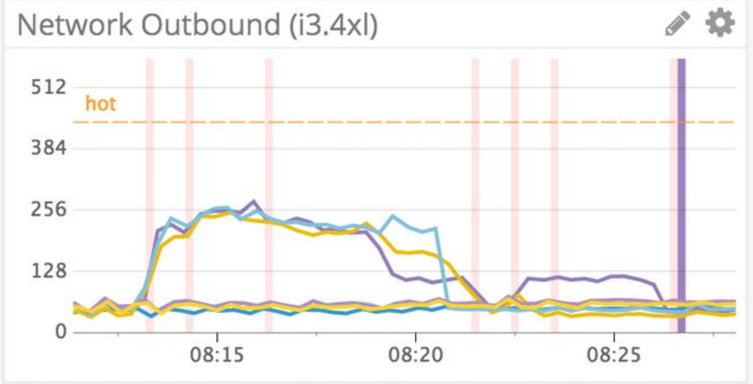












Kafka-Kit: scaling operations

- `topicmappr`:
  - partition to broker mapping
  - failed broker replacement
  - storage-based cluster rebalancing
- `autothrottle`: replication auto-throttling
- untied to Datadog

Topic mapping

"Map": assignment of a set of topics to Kafka brokers

```
map1: "test_topic.*" => [1001,1002,1003,1004,1005,1006]

map2: "load_testing|latency_testing" => [1007,1008,1009]
```

Topic mapping

Heterogeneous broker specification within a cluster

```
map1: "test_topic.*" => 6x i3.4xlarge

map2: "load_testing|latency_testing" => 3x i3.8xlarge
```

Kafka in k8s

Background

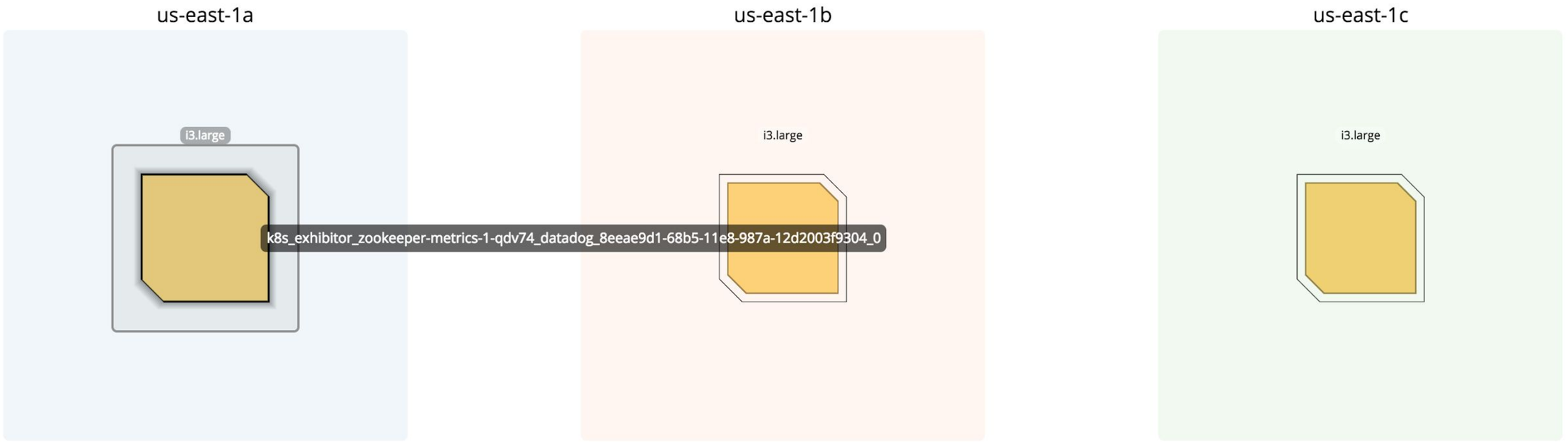
– New instance of Datadog

– Completely independant and isolated

– Leave legacy behind and start fresh

– Have everyone use it

Broker deployment



– NodeGroup: kubernetes CRD provisioning an ASG
– One broker pod per node

Data persistence and locality

– Instance store drives
– Data is persisted between pod restarts
– Data replicated on new nodes
– Rack-awareness

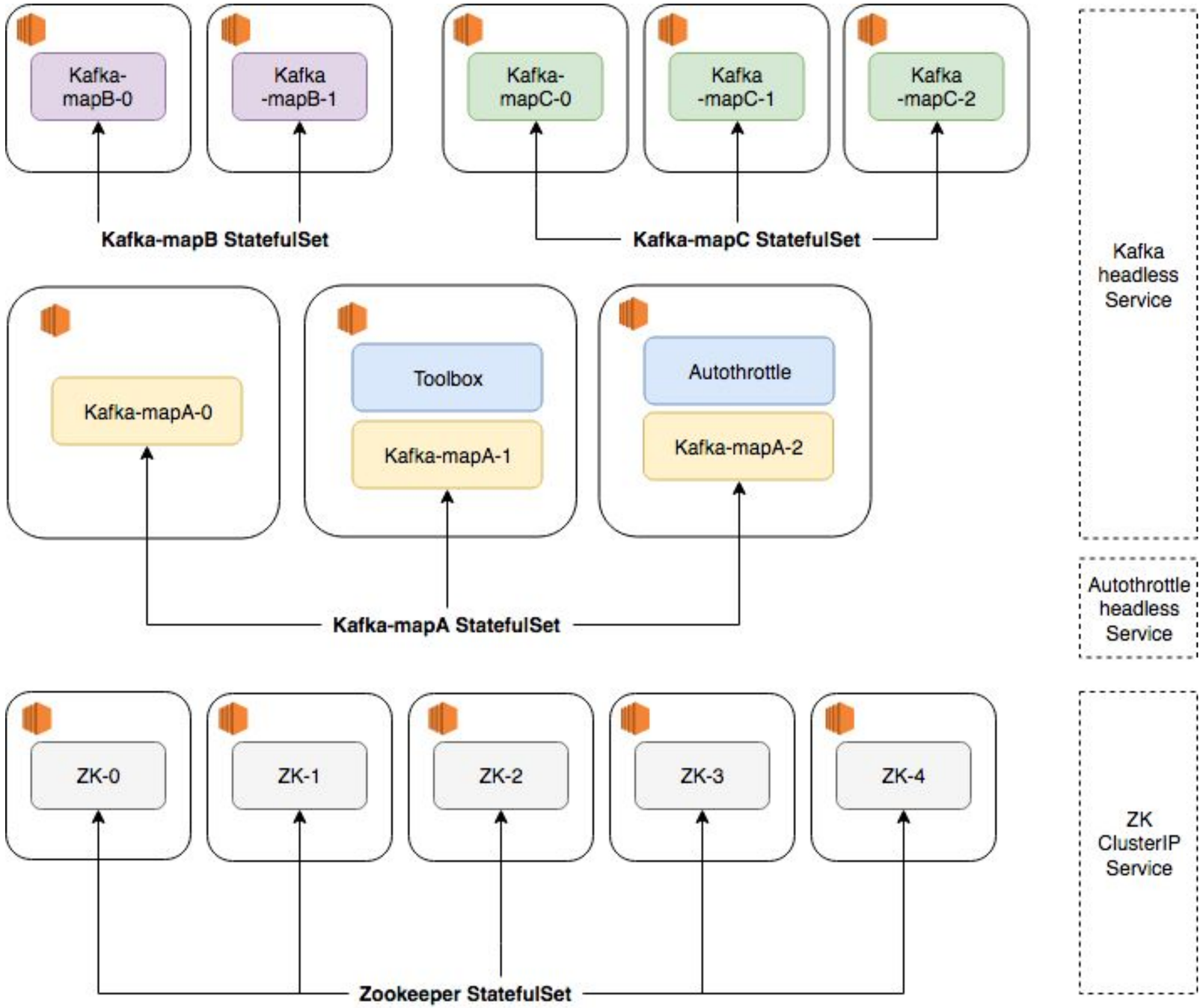
Kubernetes primitives

– NodeGroups
– Persistent Volume (PV) and Persistent Volume Claim (PVC)
– Headless service for Kafka
– ClusterIP service for ZooKeeper
– Host network
– Deployments
– ConfigMaps
– CronJob
– StatefulSet

One NodeGroup/StatefulSet per map

– A map has a dedicated StatefulSet
– Each StatefulSet runs on a dedicated NodeGroup
– Scale map independently

A Kafka cluster

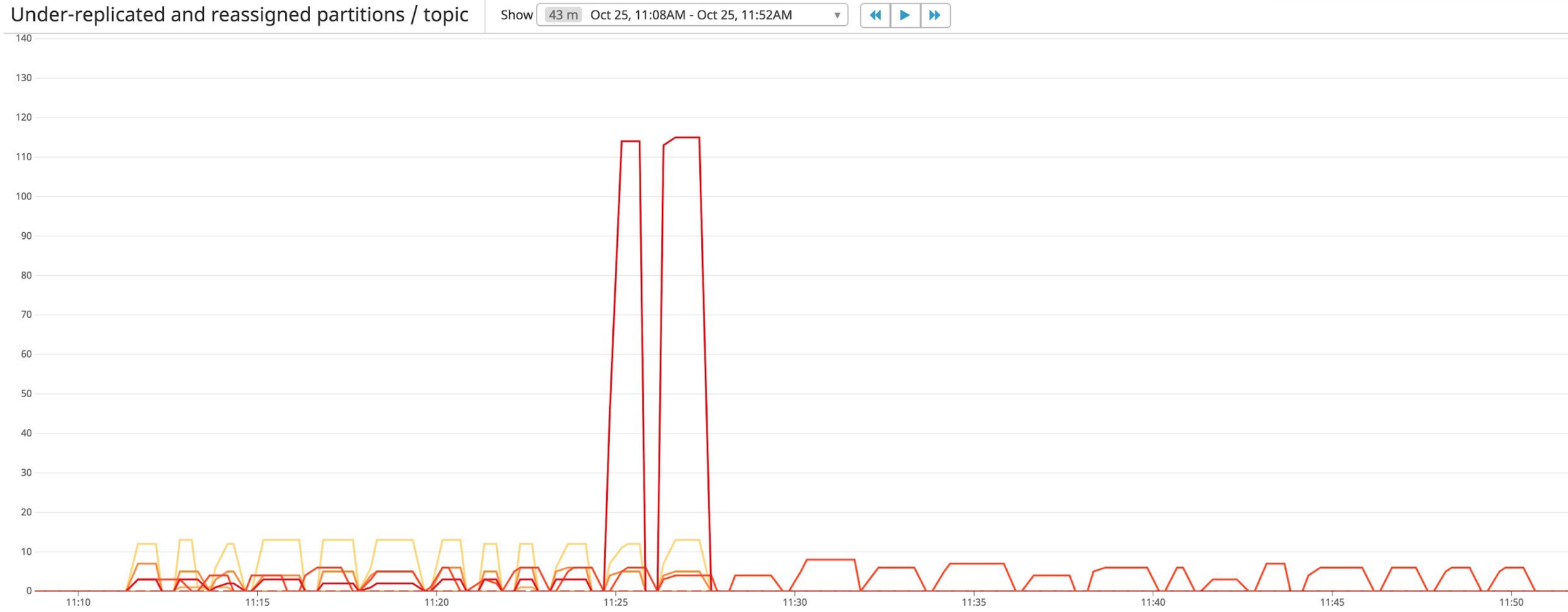


Pod health and readiness

ZooKeeper:
– Liveness: port 2181 open?
– Readiness: leader/follower?

Kafka:
– Liveness: port 9092 open?
– Readiness: broker 100% in-sync?

Safe rolling-restarts



Broker identity

– broker ID assigned when first deployed
– Pod/node labeled with broker ID
– broker ID kept between restarts
– Similar strategy for ZK, with ConfigMap annotations

Topic management

– Topic definition in a ConfigMap
– Regularly applied via a CronJob

Toolbox pod

– partition mapping
– topic management
– offset management
– load testing
– config management
– replication automatic throttler
– ZooKeeper dynamic configuration management
– Side effect stored in datadog as events

ZooKeeper

– Coordination of ensemble membership
– ZooKeeper 3.5: dynamic reconfiguration
– No longer requires Exhibitor

Monitoring: under-replication

– One alert / under-replicated topic
– > 5 topics : one cluster-wide alert
– Exports tagged partition metrics
– Automatically muted during statefulset 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

Monitoring: brokers/config

Resource usage

– Storage over/under-utilization
– Storage utilization forecast
– Unused brokers
– Sustained elevated traffic

Configuration

– Topic replication factor == 1
– Incoherent ZooKeeper ensemble configuration

Membership:

– Unsafe ZK ensemble number

What’s next?

– Management API
– Kubernetes operator
– Retention controller

Oh and one more thing...

– In-depth kafka-kit blog post: https://dtdg.co/2w7vLgL

– Kafka-kit is open source! https://github.com/datadog/kafka-kit

Thank you!

@brouberol

We're hiring!
https://www.datadoghq.com/careers