Scaling for the Known Unknown

Suhail Patel

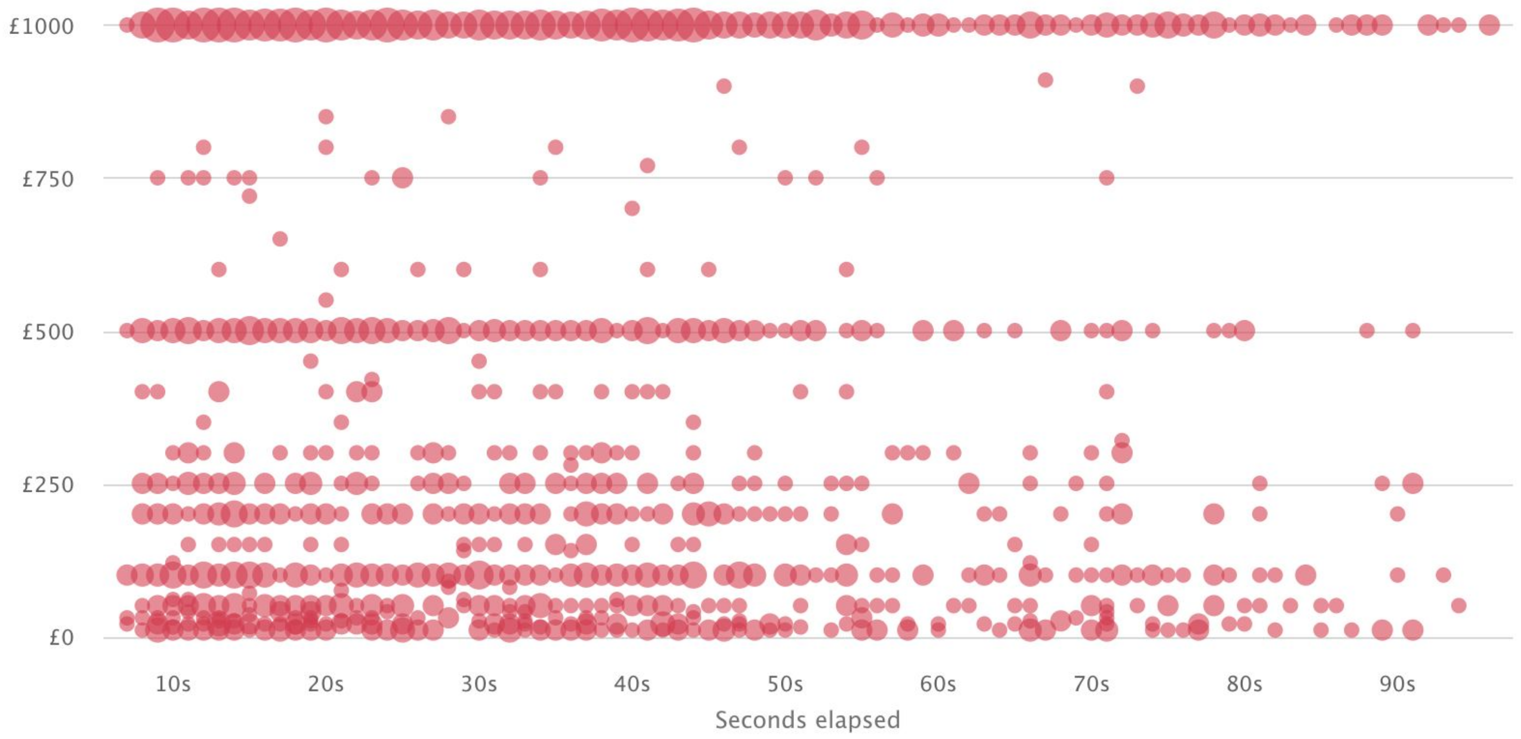
March 2016

1,861
Investors

£1,000,000
Raised

96
Seconds

March 2016



£1000
£750
£500
£250
£0
10s
20s
30s
40s
50s
60s
70s
80s
90s
Seconds elapsed

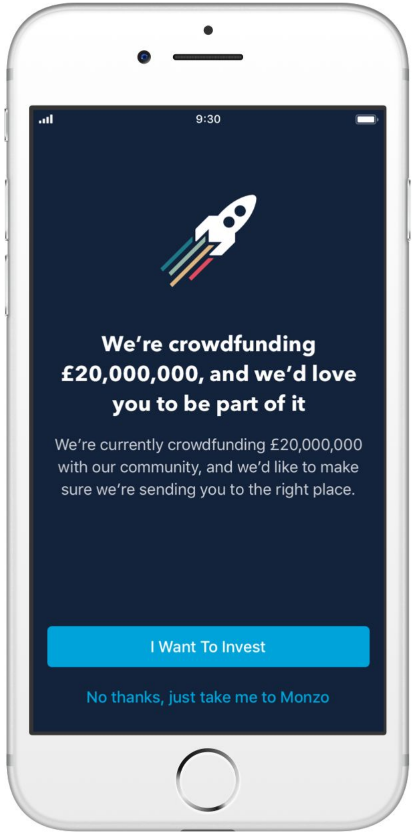
February 2017

41,267
Pledges to invest

£2,500,000
Raised

Late 2018

Monzo is raising £20,000,000 and all our customers will be eligible to participate



We're crowdfunding £20,000,000, and we'd love you to be part of it

We're currently crowdfunding £20,000,000 with our community, and we'd like to make sure we're sending you to the right place.

👋

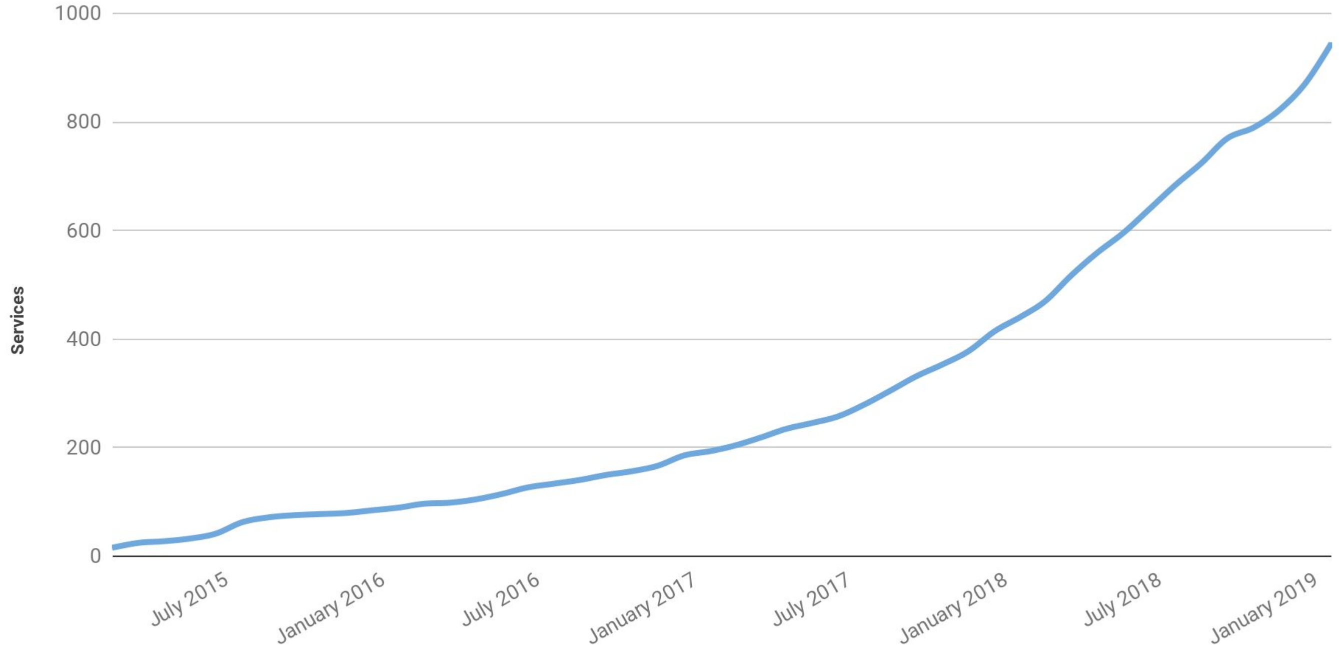
Hi, i’m Suhail

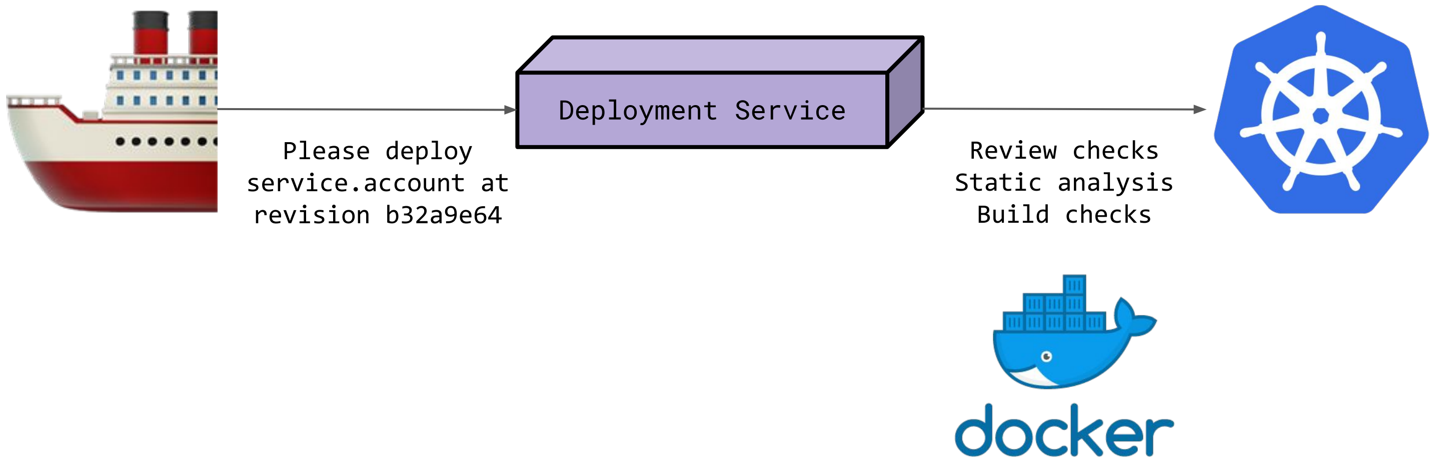
I’m a Platform Engineer at **Monzo**. I work on the Infrastructure and Reliability squad. We help build the base so other engineers can ship their services and applications.

- Email: hi@suhailpatel.com
- Twitter: @suhailpatel

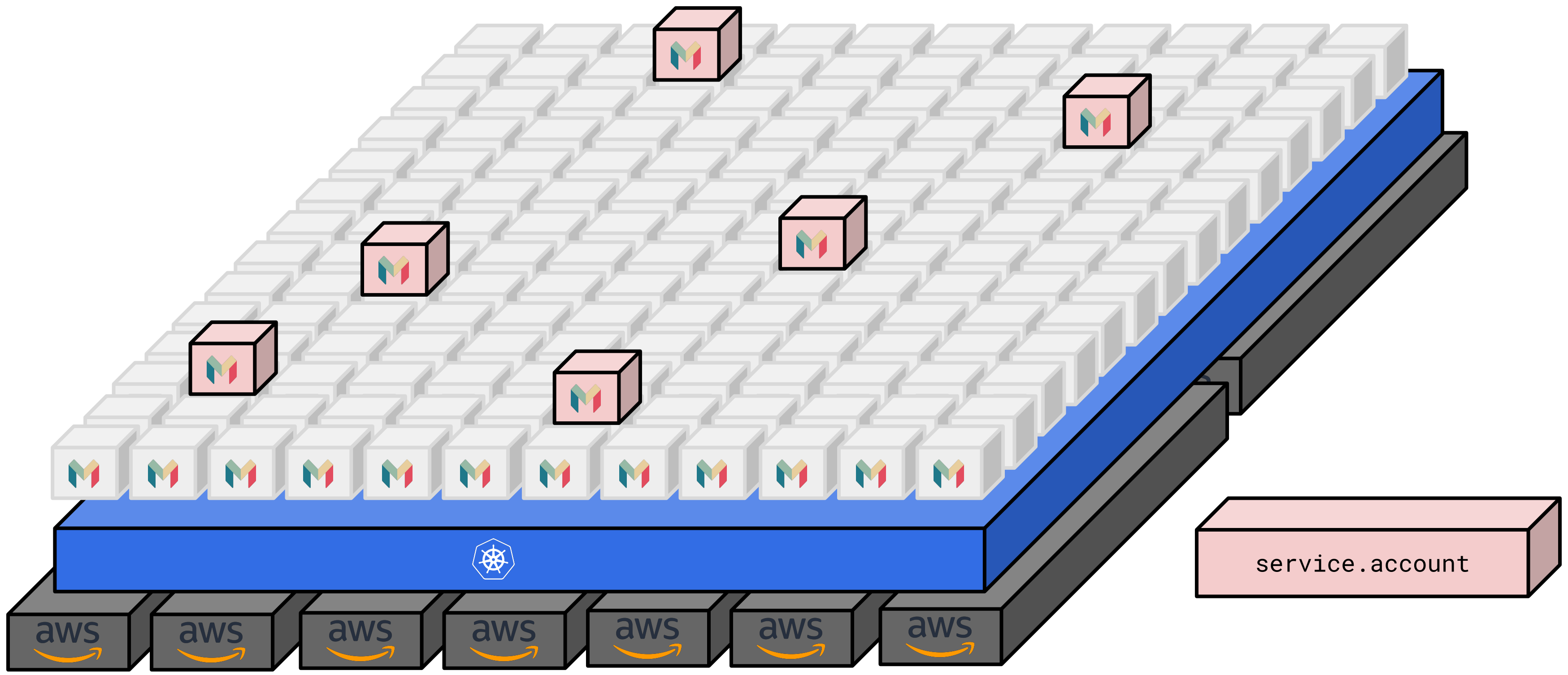
monzo
5354 2000 0000 0000
EXP END 08/1
469-624-882

Number of services




Deployment Service
Please deploy service.account at revision b32a9e64
Review checks
Static analysis
Build checks
docker

Running services

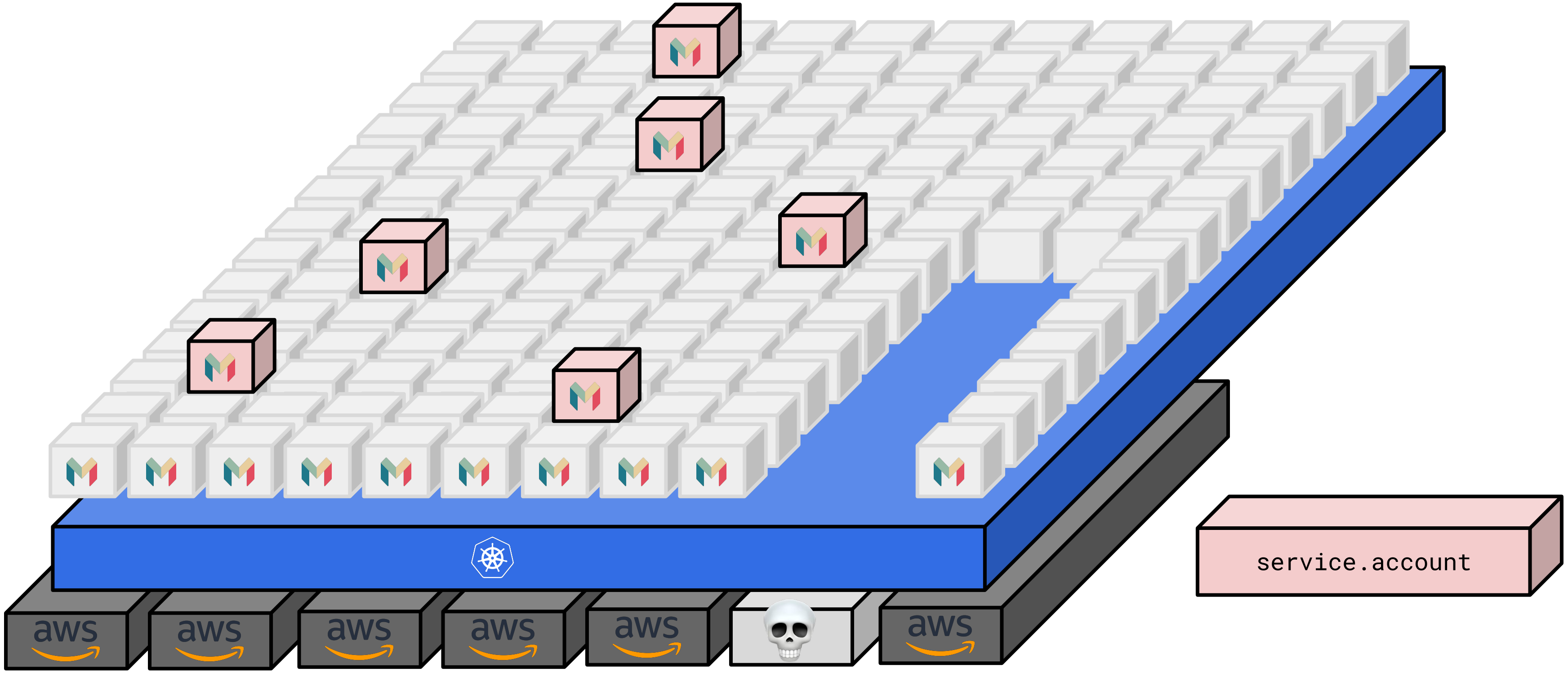


Running services

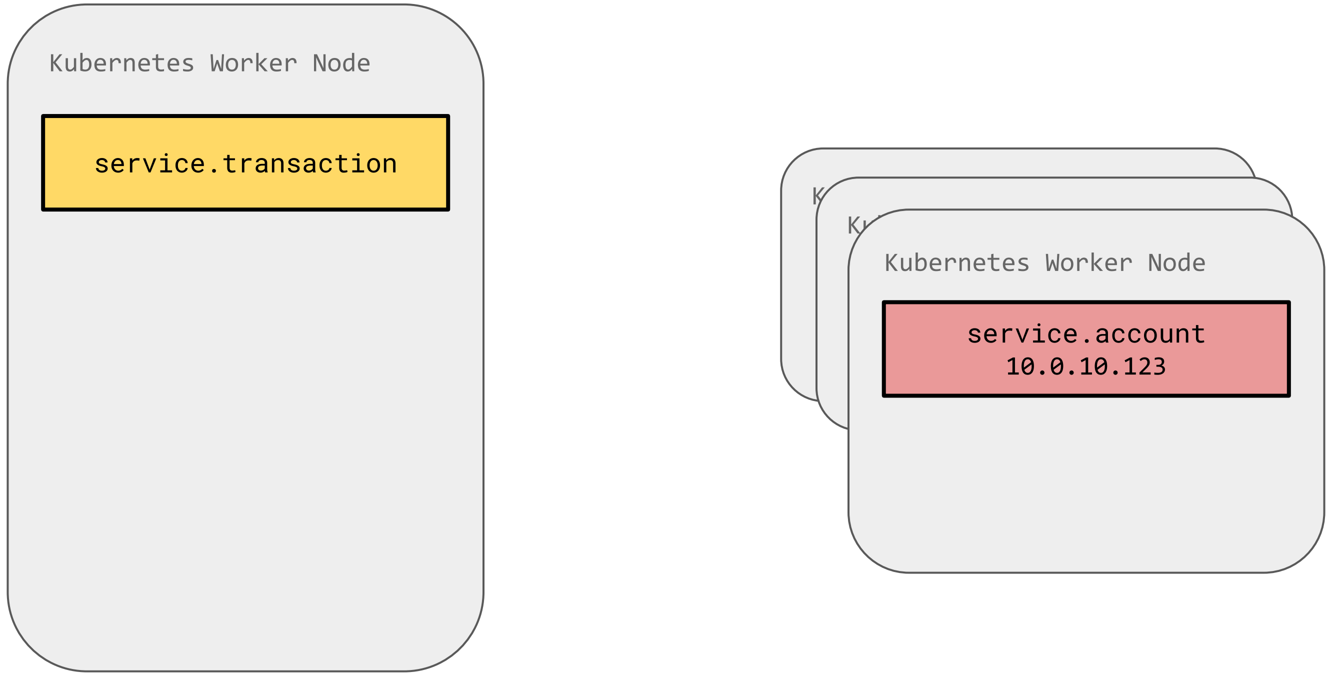
What we want from services:

- Self-contained
- Scalable
- Stateless
- Fault tolerance

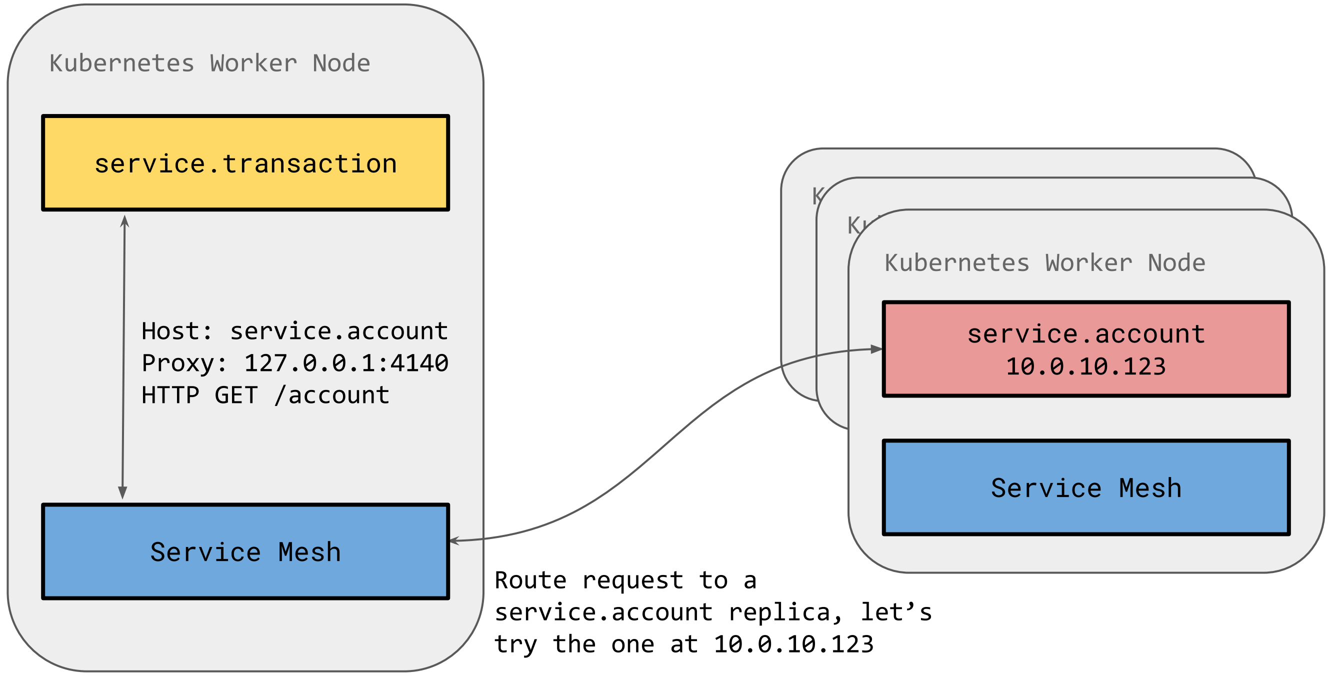
Running services



Running services



Running services



Route request to a
service.account replica, let's
try the one at 10.0.10.123

Service Mesh

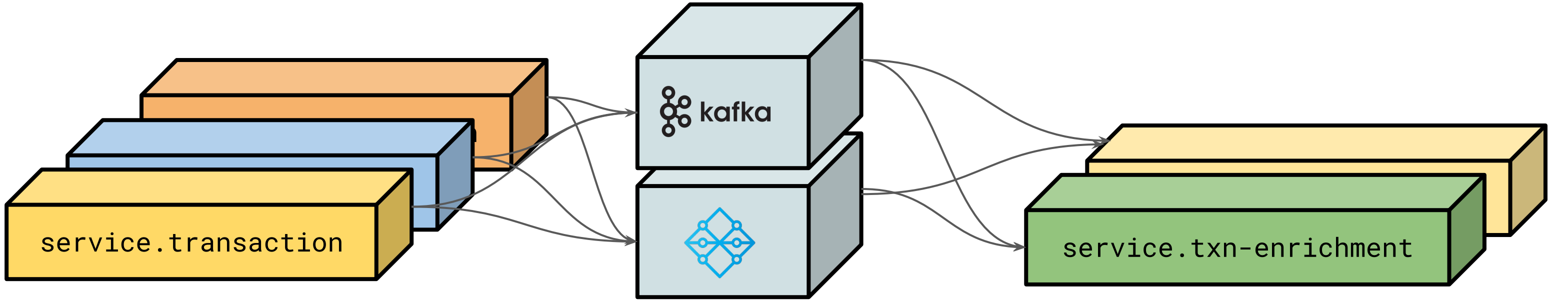
The Service Mesh ties the microservices together.
It acts as the RPC proxy.

- Handles service discovery and routing
- Retries / Timeouts / Circuit Breaking
- Observability

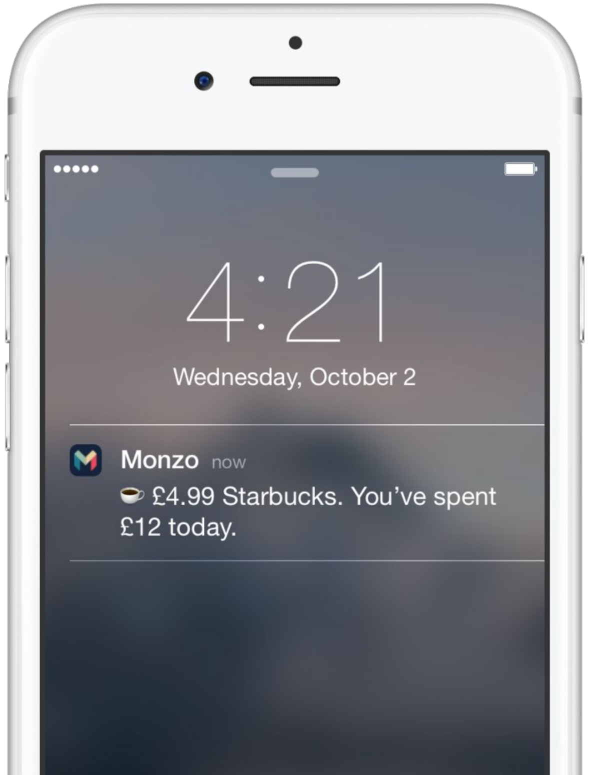
Asynchronous messaging

Many things can occur asynchronously rather than a direct blocking RPC.

Message queues like NSQ and Kafka provide asynchronous flows with at least once message delivery semantics.

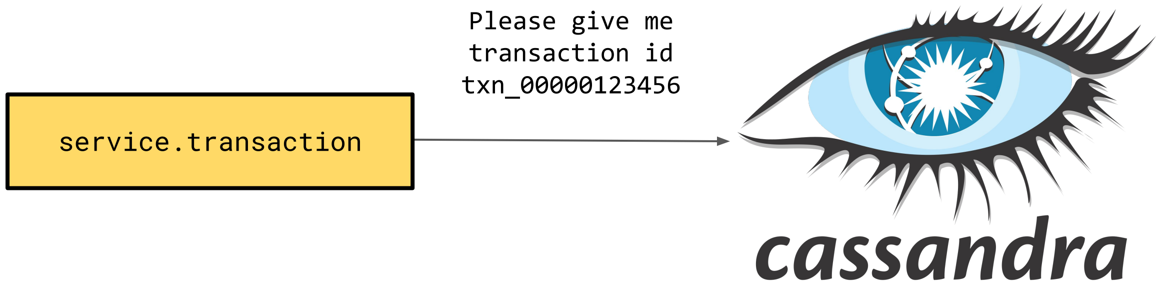


Asynchronous messaging

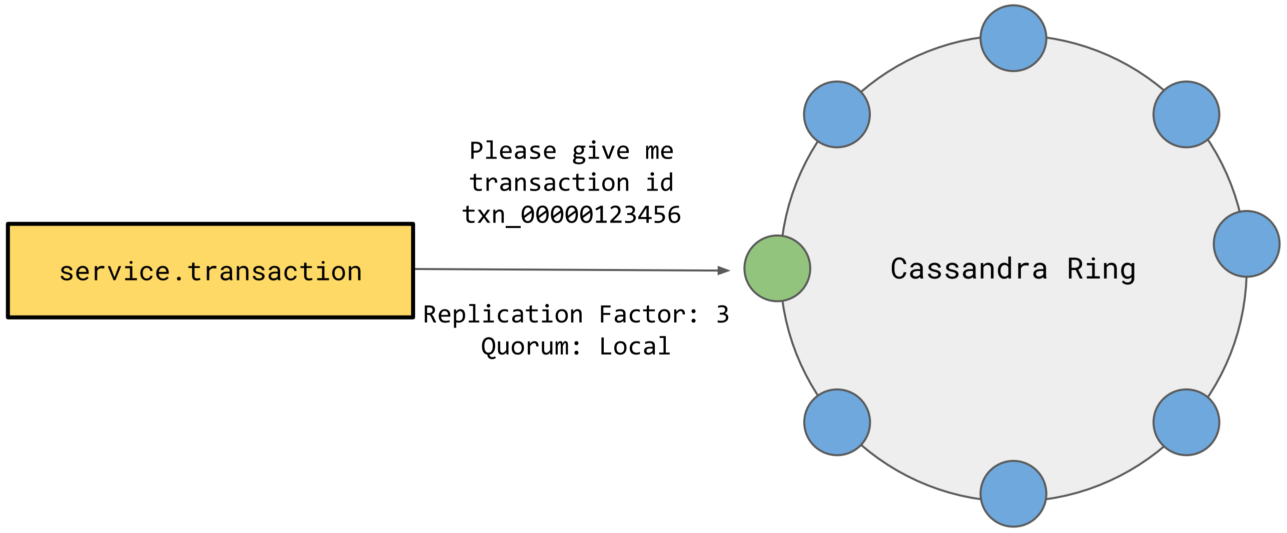


☕ £4.99 Starbucks. You've spent £12 today.

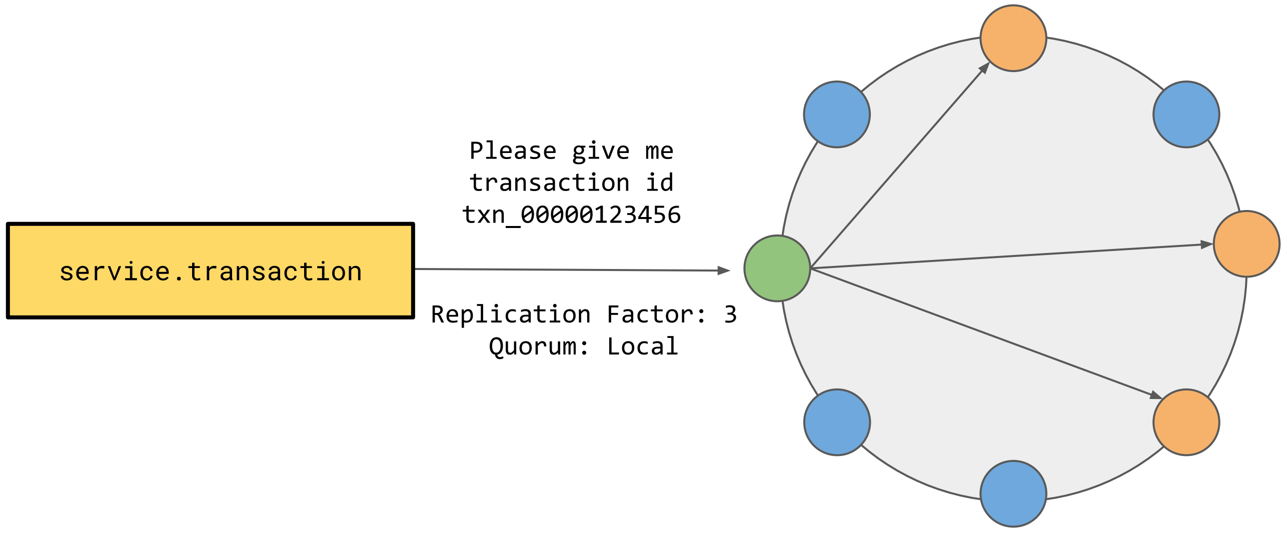
Storing data with Cassandra



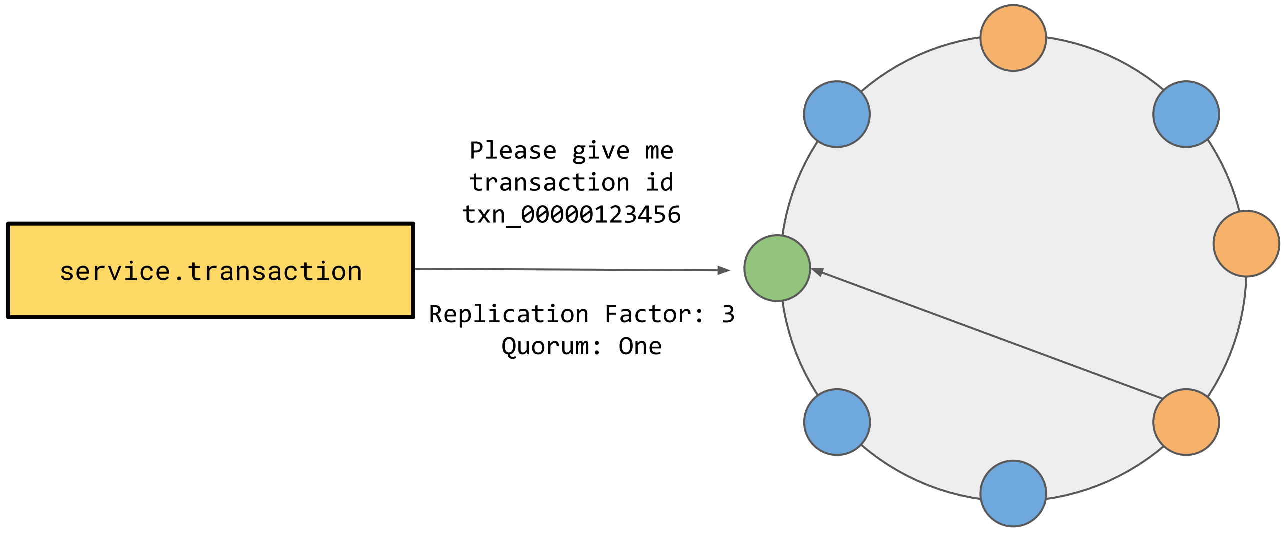
Storing data with Cassandra



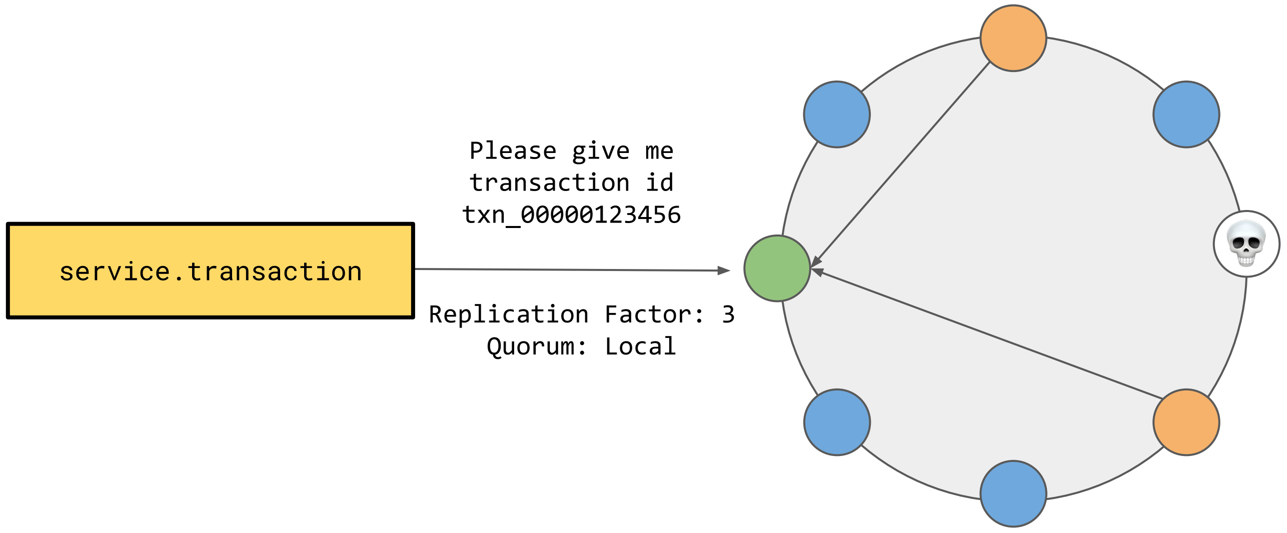
Storing data with Cassandra



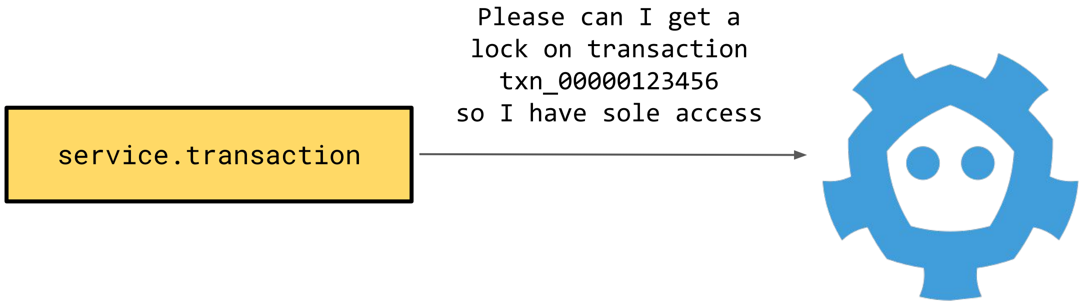
Storing data with Cassandra



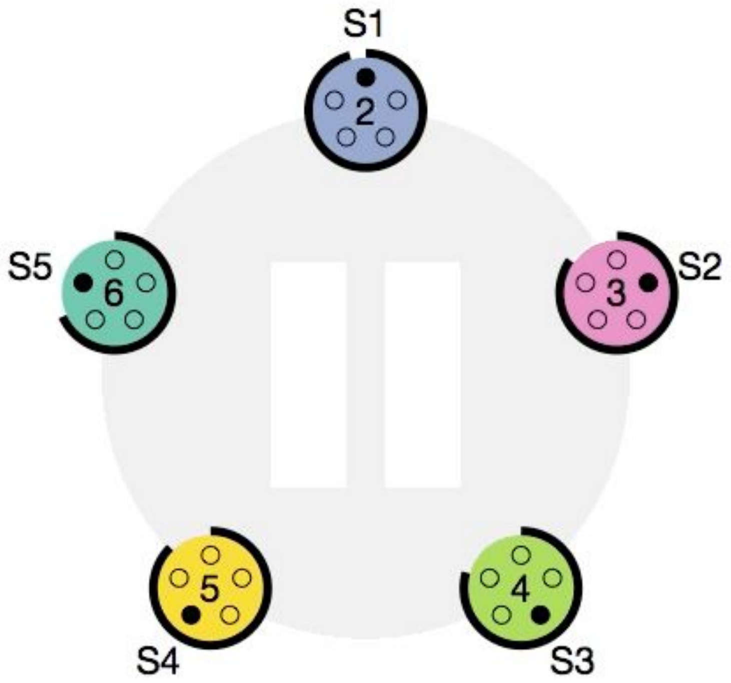
Storing data with Cassandra



Distributed Locking with etcd


service.transaction
Please can I get a lock on transaction txn_00000123456 so I have sole access

Distributed Locking with etcd



Source: https://raft.github.io/

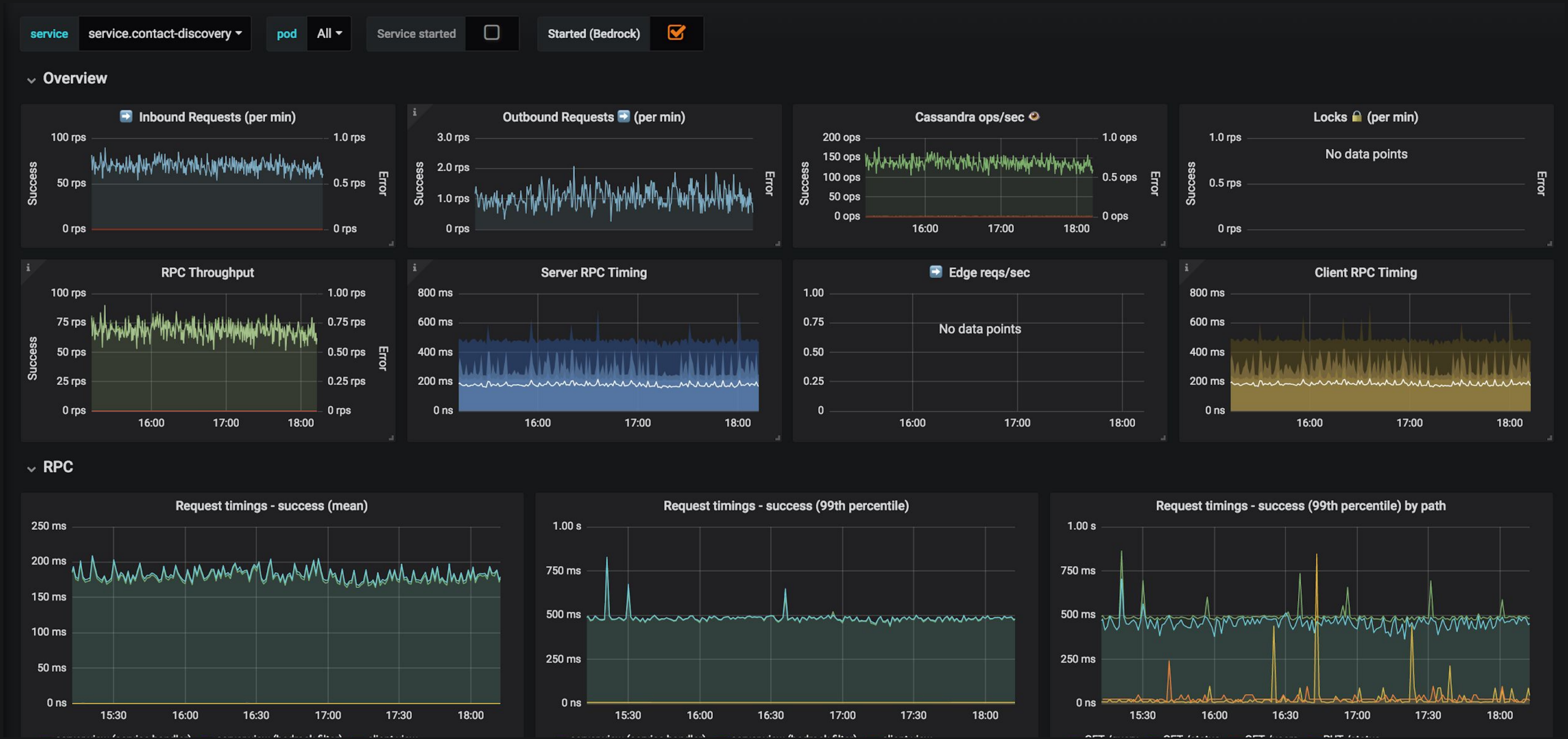
Monitoring with Prometheus

Prometheus is a flexible time-series data store and query engine

Each of our services expose metrics in Prometheus format at `/metrics`

Monitor all the things

- RPC Request/Response cycles
- CPU / Memory / Network use
- Asynchronous processing
- C* and Distributed Locking


service service.contact-discovery
pod All
Service started ☐
Started (Bedrock) ✓

Overview
Inbound Requests (per min)
Outbound Requests (per min)
Cassandra ops/sec
Locks (per min)
No data points
RPC Throughput
Server RPC Timing
Edge reqs/sec
No data points
Client RPC Timing
Success
Error
RPC
Request timings - success (mean)
Request timings - success (99th percentile)
Request timings - success (99th percentile) by path
15:30
16:00
16:30
17:00
17:30
18:00

Introduction

A brief overview of our Platform

Building a Crowdfunding Backend

Load testing + Finding bottlenecks

Requirements

1. Raise at most £20,000,000
We'd agreed with institutional investors leading the funding round that £20M was the cap

2. Ensure users have enough money
Users should have the money they are pledging. We need to verify this before accepting the investment.

3. Handle lots of traffic
It was first-come-first-serve so we expected a lot of interest at the start of the crowdfunding round

4. Don't bring down the bank
All banking functions should continue to work whilst we're running the crowdfunding

Requirements

1. Raise at most £20,000,000
We'd agreed with institutional investors leading the funding round that £20M was the cap

2. Ensure users have enough money
Users should have the money they are pledging. We need to verify this before accepting the investment.

3. Handle lots of traffic
It was first-come-first-serve so we expected a lot of interest at the start of the crowdfunding round

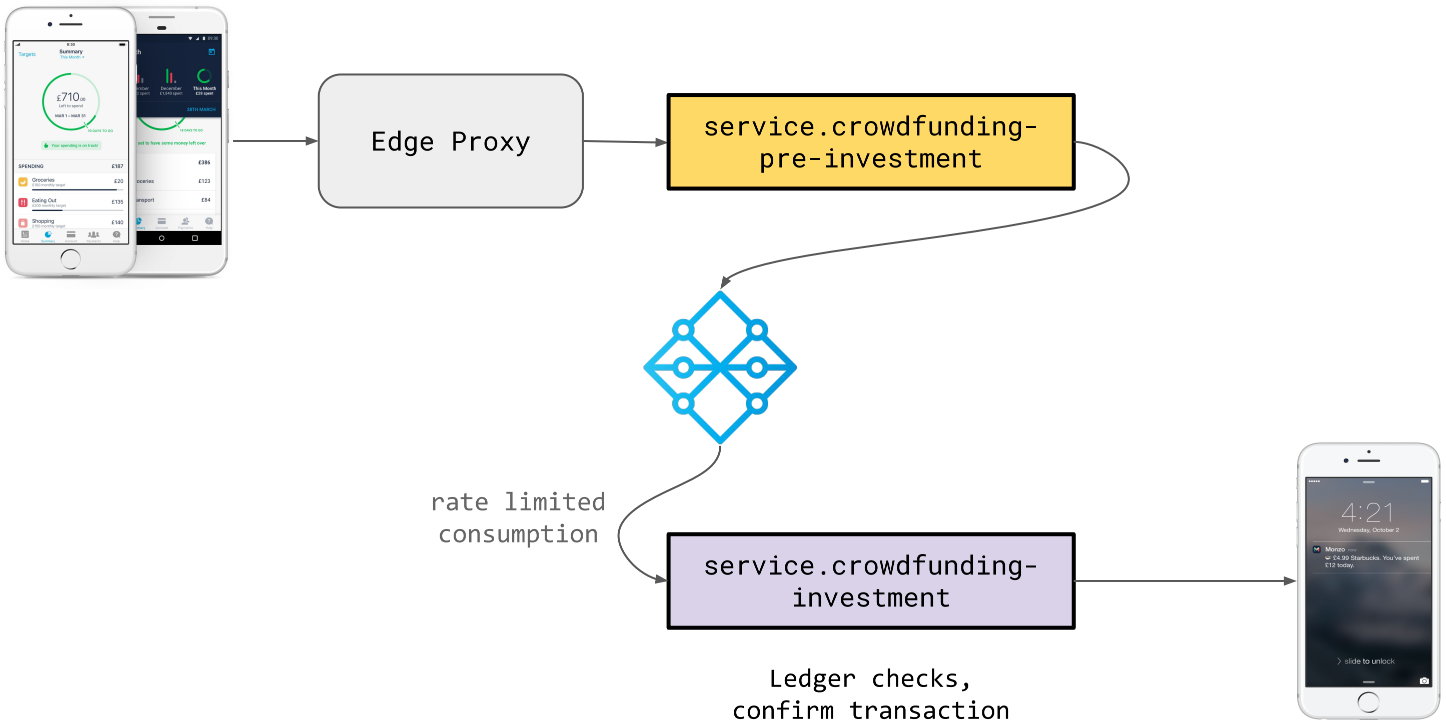
4. Don't bring down the bank
All banking functions should continue to work whilst we're running the crowdfunding

Counters / Transactions

What if we used as Cassandra counter?

"In Cassandra, at any given moment, the counter value may be stored in the Memtable, commit log, and/or one or more SSTables. Replication between nodes can cause consistency issues in certain edge cases"

Source: https://docs.datastax.com/en/cql/3.3/cql/cql_using/useCountersConcept.html


Edge Proxy
service.crowdfunding-pre-investment
rate limited consumption
service.crowdfunding-investment
Ledger checks, confirm transaction

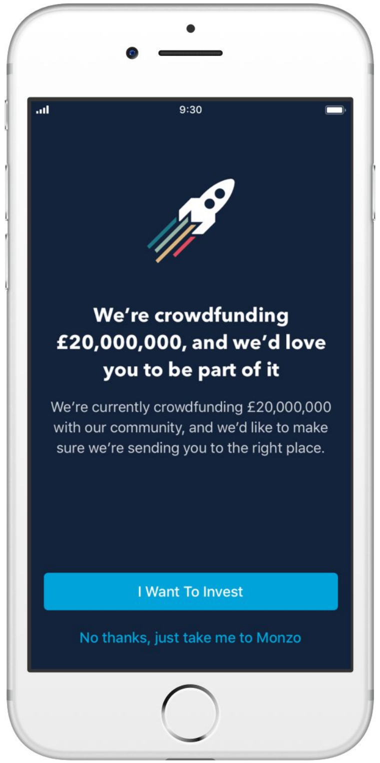
Requirements

1. Raise at most £20,000,000
We'd agreed with institutional investors leading the funding round that £20M was the cap

2. Ensure users have enough money
Users should have the money they are pledging. We need to verify this before accepting the investment.

3. Handle lots of traffic
It was first-come-first-serve so we expected a lot of interest at the start of the crowdfunding round

4. Don't bring down the bank
All banking functions should continue to work whilst we're running the crowdfunding



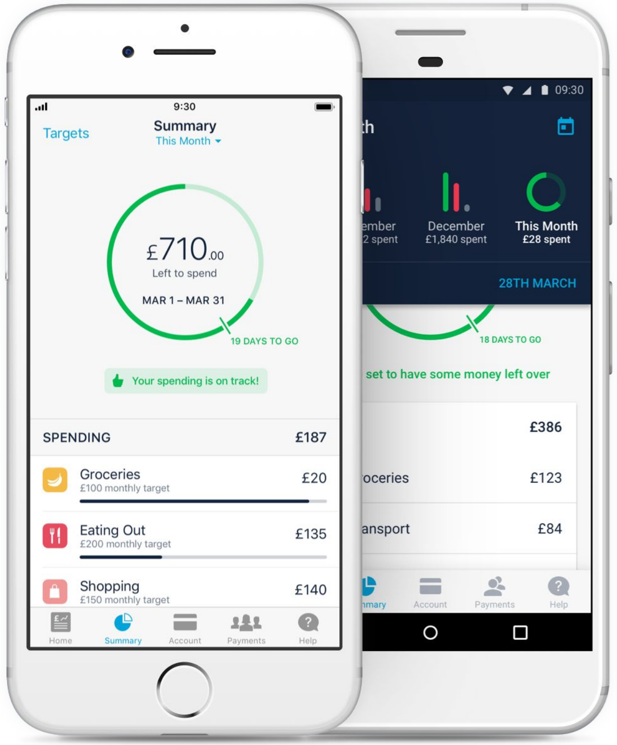
9:30

We're crowdfunding £20,000,000, and we'd love you to be part of it

We're currently crowdfunding £20,000,000 with our community, and we'd like to make sure we're sending you to the right place.

I Want To Invest

No thanks, just take me to Monzo



9:30

Targets | Summary | This Month

£710.00
Left to spend
MAR 1 – MAR 31
19 DAYS TO GO

Your spending is on track!

SPENDING	£187
Groceries £100 monthly target	£20
Eating Out £200 monthly target	£135
Shopping £150 monthly target	£140

Home · Summary · Account · Payments · Help

09:30

th

ember 2 spent	December £1,840 spent	This Month £28 spent

28TH MARCH

18 DAYS TO GO

set to have some money left over

	£386
oceries	£123
ansport	£84

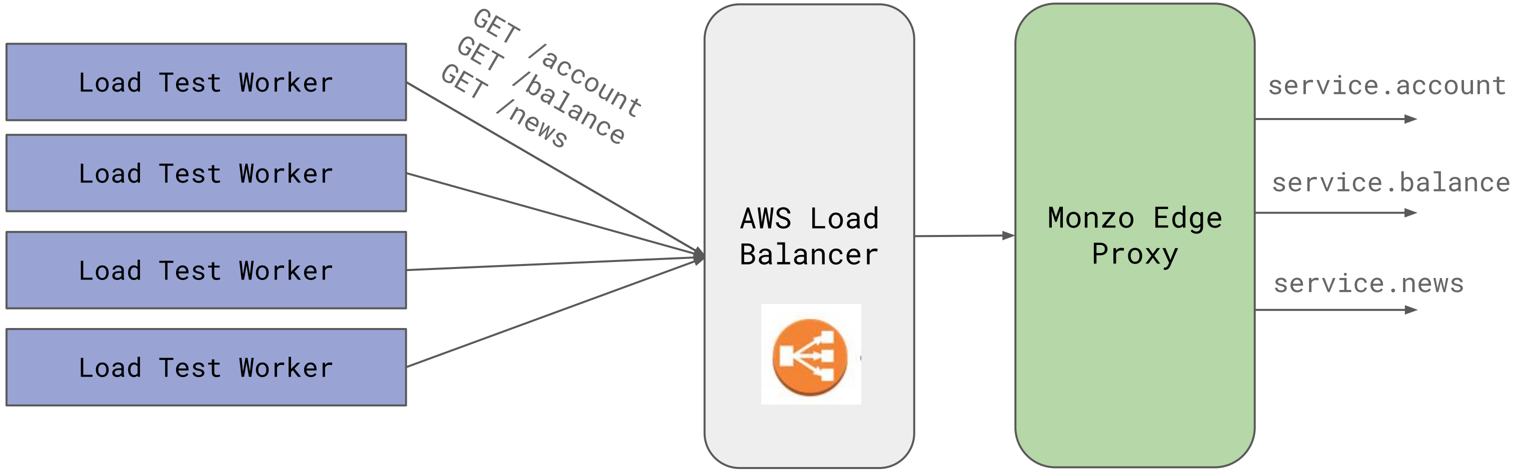
mmary · Account · Payments · Help

Building our own load tester

There's some great off-the-shelf solutions for load testing

- Bees with Machine Guns
- Locust
- ApacheBench (ab)
- Gatling

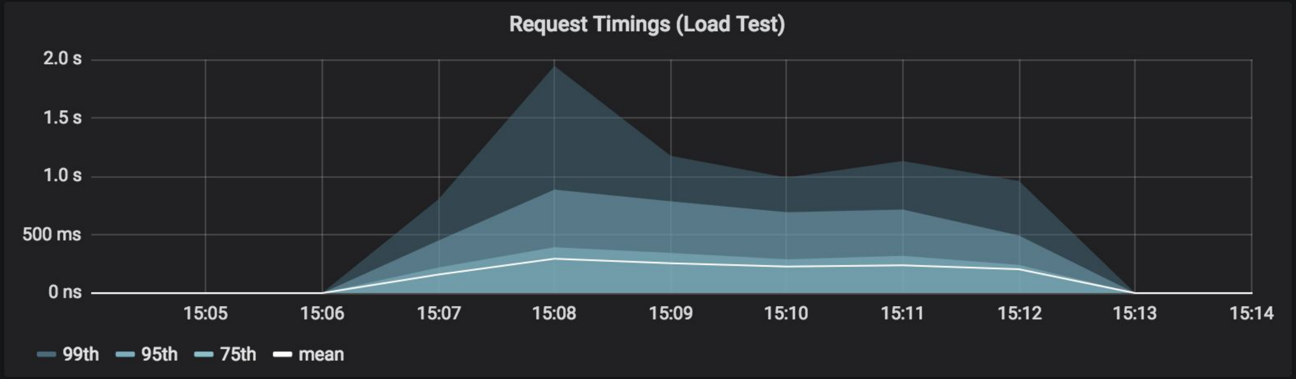
Building our own load tester

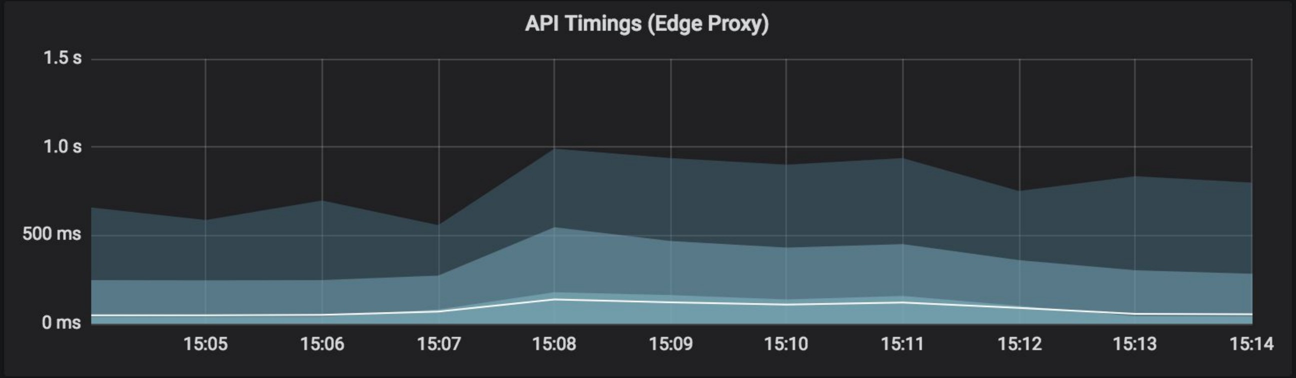


😅

At one point, we saw really high error rates in the load testing metrics. We didn’t see load test requests make it to our our AWS Load Balancer.

The load test nodes were using internal DNS provided by Amazon Route 53. We were constantly resolving *.monzo.com subdomains.


Request Timings (Load Test)
2.0 s
1.5 s
1.0 s
500 ms
0 ns
15:05
15:06
15:07
15:08
15:09
15:10
15:11
15:12
15:13
15:14
99th
95th
75th
mean


API Timings (Edge Proxy)
1.5 s
1.0 s
500 ms
0 ms
15:05
15:06
15:07
15:08
15:09
15:10
15:11
15:12
15:13
15:14

Load testing in production

For our testing to create realistic load and give us useful results, we needed to test against our production systems – the real bank.

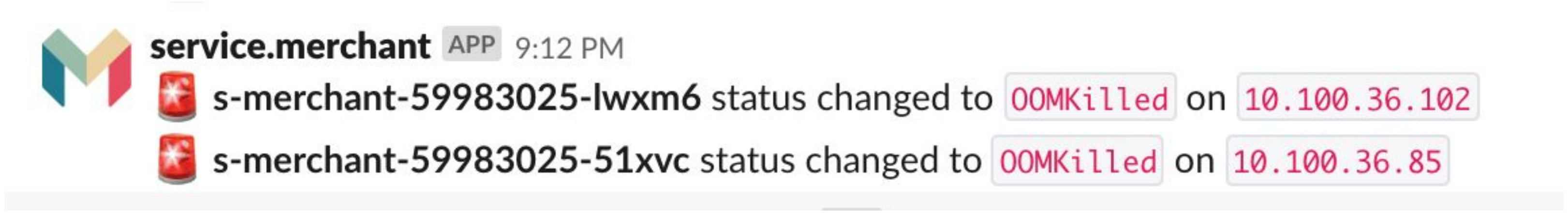
Load testing in production

We set up our load testing system as a third “app” alongside our iOS and Android apps, and we gave it read-only access to the data we needed to test.

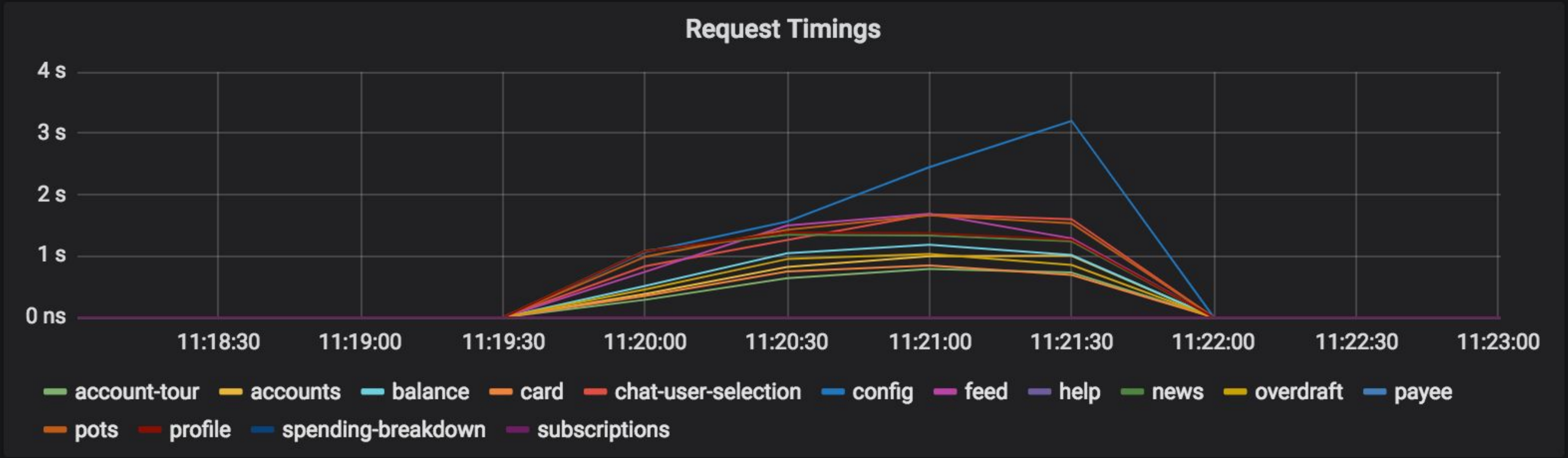
Target: Reach 1,000 app launches per second

Scaling services

Target: Reach 1,000 app launches per second



service.merchant APP 9:12 PM
🚨 s-merchant-59983025-lwxm6 status changed to `OOMKilled` on `10.100.36.102`
🚨 s-merchant-59983025-51xvc status changed to `OOMKilled` on `10.100.36.85`



Request Timings
4 s
3 s
2 s
1 s
0 ns
11:18:30
11:19:00
11:19:30
11:20:00
11:20:30
11:21:00
11:21:30
11:22:00
11:22:30
11:23:00
account-tour
accounts
balance
card
chat-user-selection
config
feed
help
news
overdraft
payee
pots
profile
spending-breakdown
subscriptions

Scaling services

Target: Reach 1,000 app launches per second

```
replicas: 9
template:
  spec:
    containers:
      resources:
        limits:
          cpu: 30m
          memory: 40Mi
        requests:
          cpu: 10m
          memory: 20Mi
```

Scaling services

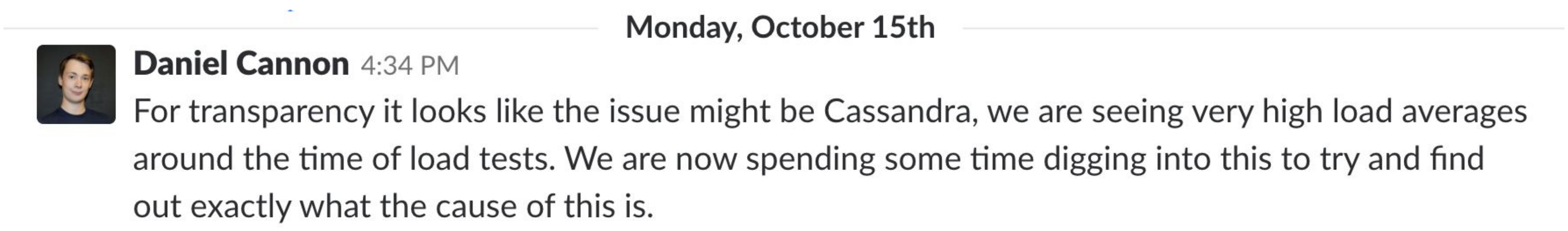
Target: Reach 1,000 app launches per second

```
replicas: 9
template:
  spec:
    containers:
      resources:
        limits:
          cpu: 100m
          memory: 40Mi
        requests:
          cpu: 50m
          memory: 20Mi
```

“But wait, you are re-inventing autoscaling, manually?”

Cassandra Bottlenecks

We got to around 500-600 app launches before we found a major Platform bottleneck



Monday, October 15th

Daniel Cannon 4:34 PM

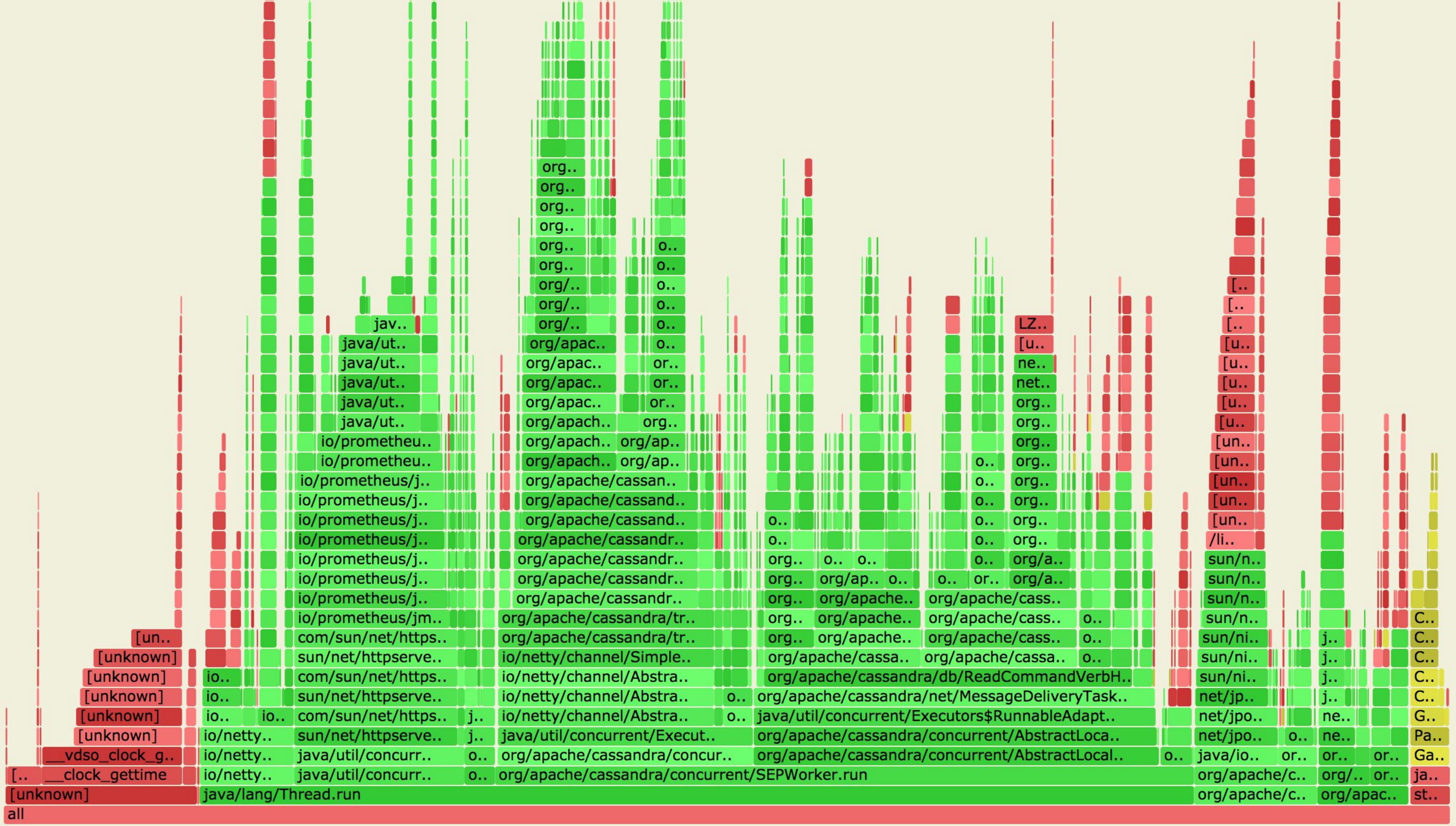
For transparency it looks like the issue might be Cassandra, we are seeing very high load averages around the time of load tests. We are now spending some time digging into this to try and find out exactly what the cause of this is.

The numbers

21 x i3.4xlarge EC2 machines

- 16 cores
- 122GiB memory
- 2 * 1.9TiB of NVMe disks

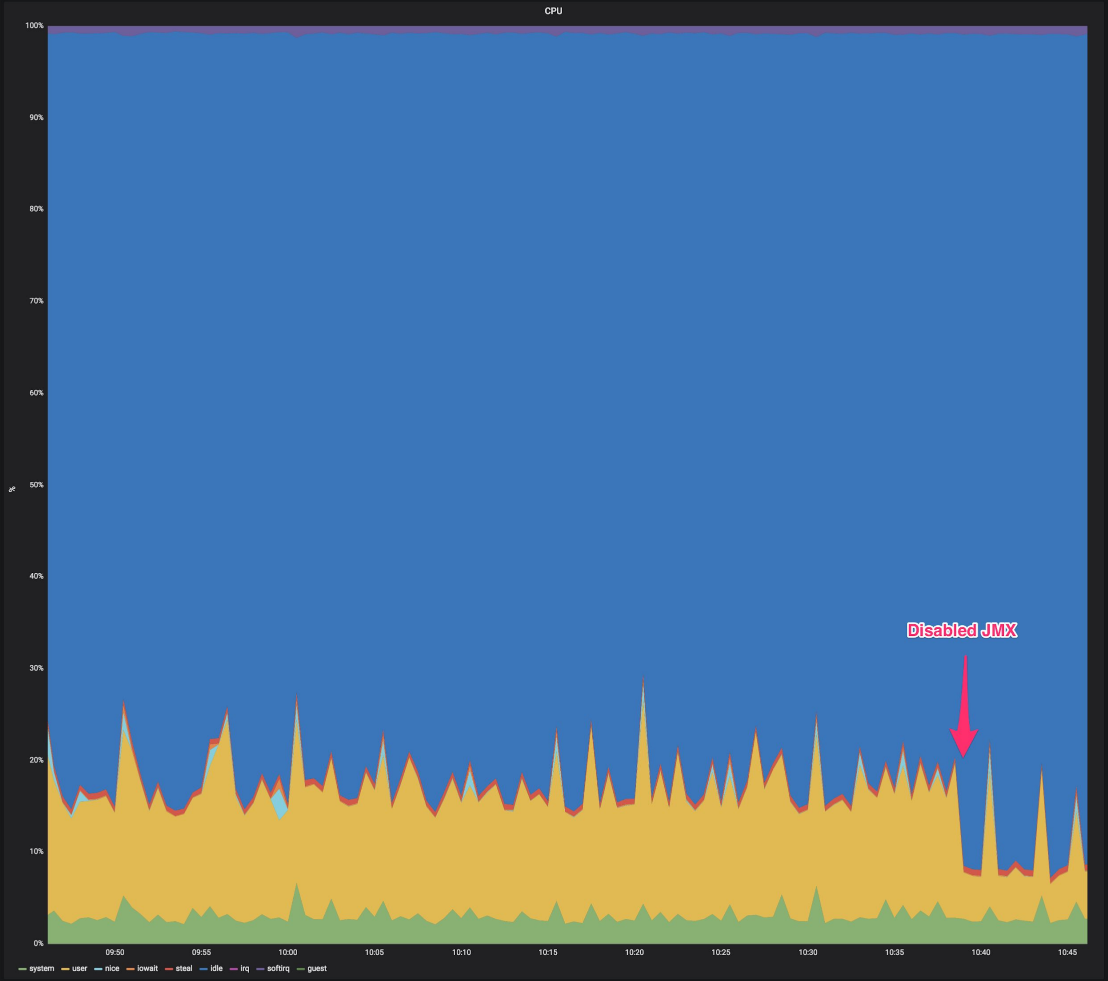
Each node holds about 500GB of data


all
[unknown]
java/lang/Thread.run
org/apache/c..
org/apac..
st..
[..
__clock_gettime
io/netty..
java/util/concurr..
o..
org/apache/cassandra/concurrent/SEPWorker.run
org/apache/c..
org/..
or..
ja..
__vdso_clock_g..
org/apache/cassandra/concur..
org/apache/cassandra/concurrent/AbstractLocal..
java/io..
Ga..
[unknown]
sun/net/httpserve..
j..
java/util/concurrent/Execut..
org/apache/cassandra/concurrent/AbstractLoca..
net/jpo..
ne..
Pa..
com/sun/net/https..
io/netty/channel/Abstra..
java/util/concurrent/Executors$RunnableAdapt..
G..
org/apache/cassandra/net/MessageDeliveryTask..
net/jp..
org/apache/cassandra/db/ReadCommandVerbH..
sun/ni..
io/netty/channel/Simple..
org/apache/cassa..
org/apache/cassandra/tr..
org/apache/cass..
io/prometheus/jm..
io/prometheus/j..
org/apache/cassandr..
sun/n..
io/prometheu..
org/apache/cassand..
org/apache/cassan..
org/apach..
org/ap..
java/ut..
org/apac..
jav..
org/..
LZ..
net..
ne..
[un..
/li..
C..

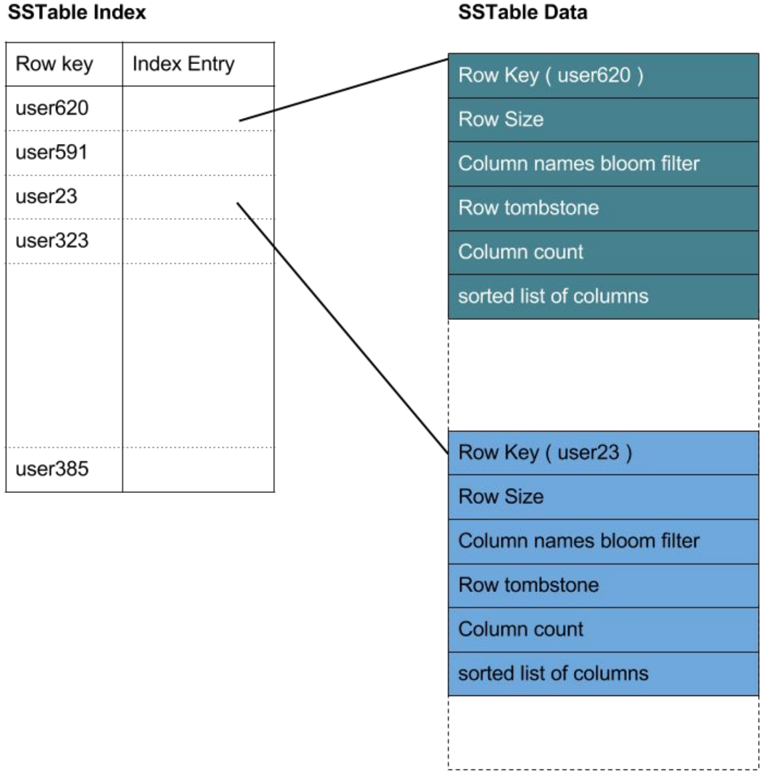
Cassandra Bottlenecks

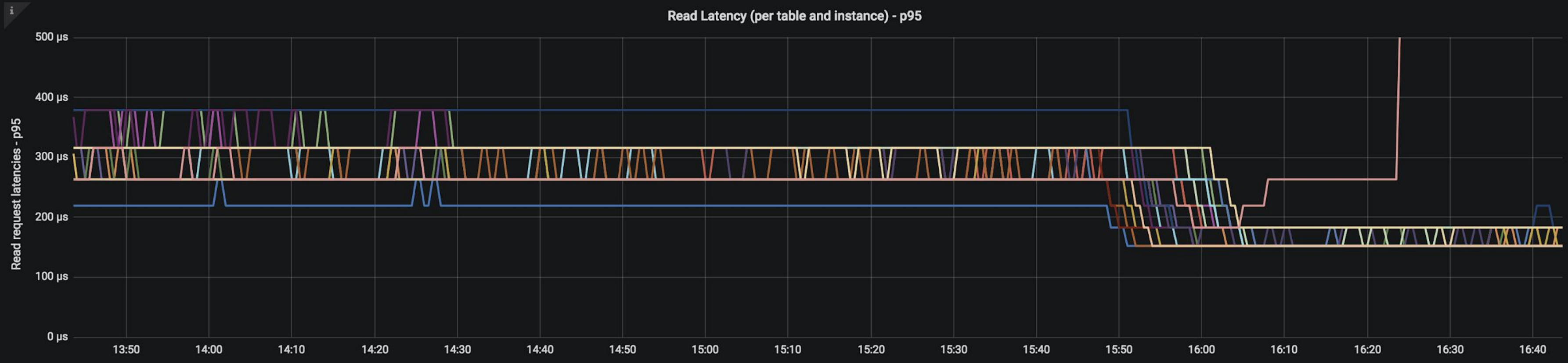
Our profiling identified three key areas

- Generating Prometheus metrics
- LZ4 Decompression
- CQL Statement Processing


CPU
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
%
09:50
09:55
10:00
10:05
10:10
10:15
10:20
10:25
10:30
10:35
10:40
10:45
Disabled JMX
system
user
nice
iowait
steal
idle
irq
softirq
guest

LZ4 Decompression




Read Latency (per table and instance) - p95
Read request latencies - p95
500 µs
400 µs
300 µs
200 µs
100 µs
0 µs
13:50
14:00
14:10
14:20
14:30
14:40
14:50
15:00
15:10
15:20
15:30
15:40
15:50
16:00
16:10
16:20
16:30
16:40

CQL Statement Parsing

We saw a significant amount of time being spent in parsing CQL statements.

The majority of our applications had a fixed model during the service pod lifetime so we would’ve been processing the same statement over and over again.

Prepared Statements

Cassandra supports prepared statements! Our gocql library which runs Cassandra queries was actively using them too for the majority of queries.

Prepared Statements

```
SELECT id, accountid, userid, amount, currency
FROM transaction.transaction_map_Id
WHERE id = ?
```

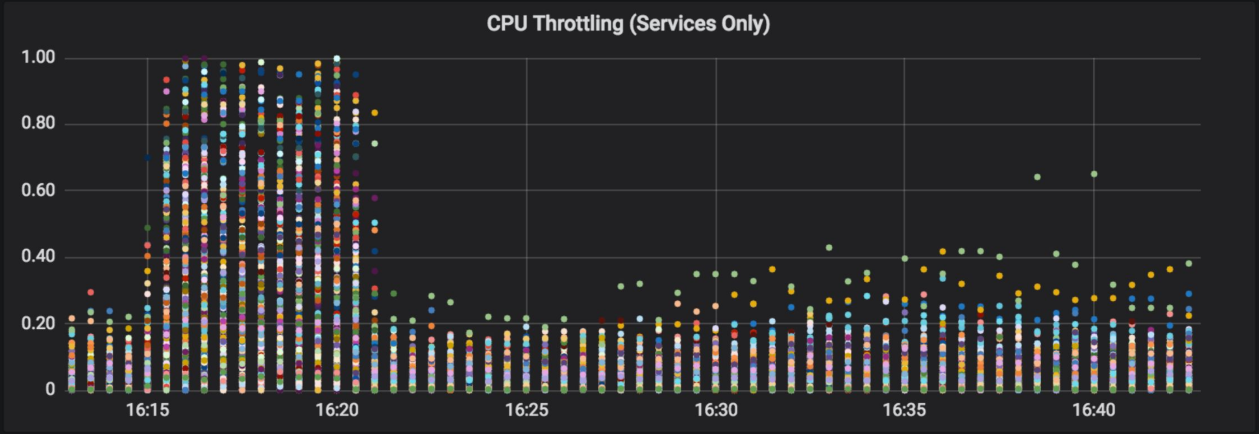
```
SELECT currency, accountid, userid, id, amount
FROM transaction.transaction_map_Id
WHERE id = ?
```

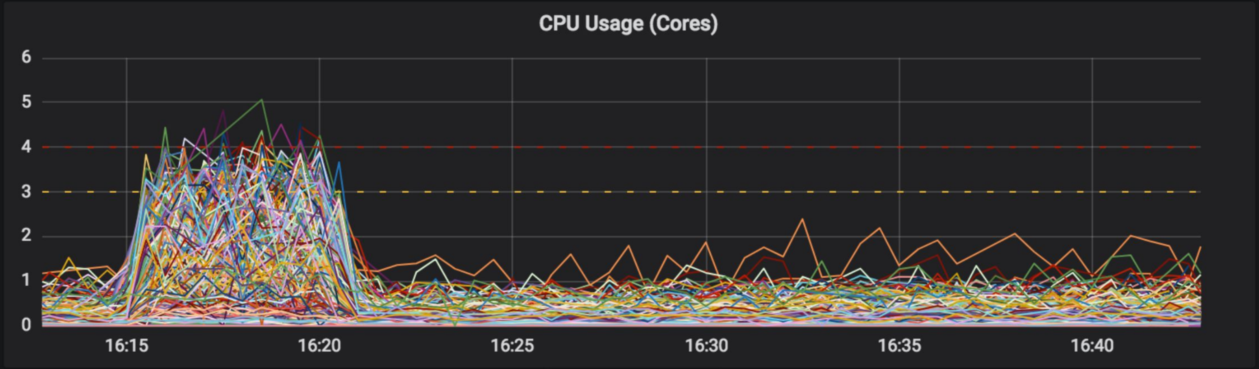
Service Mesh Bottlenecks

Target: Reach 1,000 app launches per second

At around 800 app launches per second, we saw our RPCs take a really long time across our Platform.

linkerd-m4-2xlarge





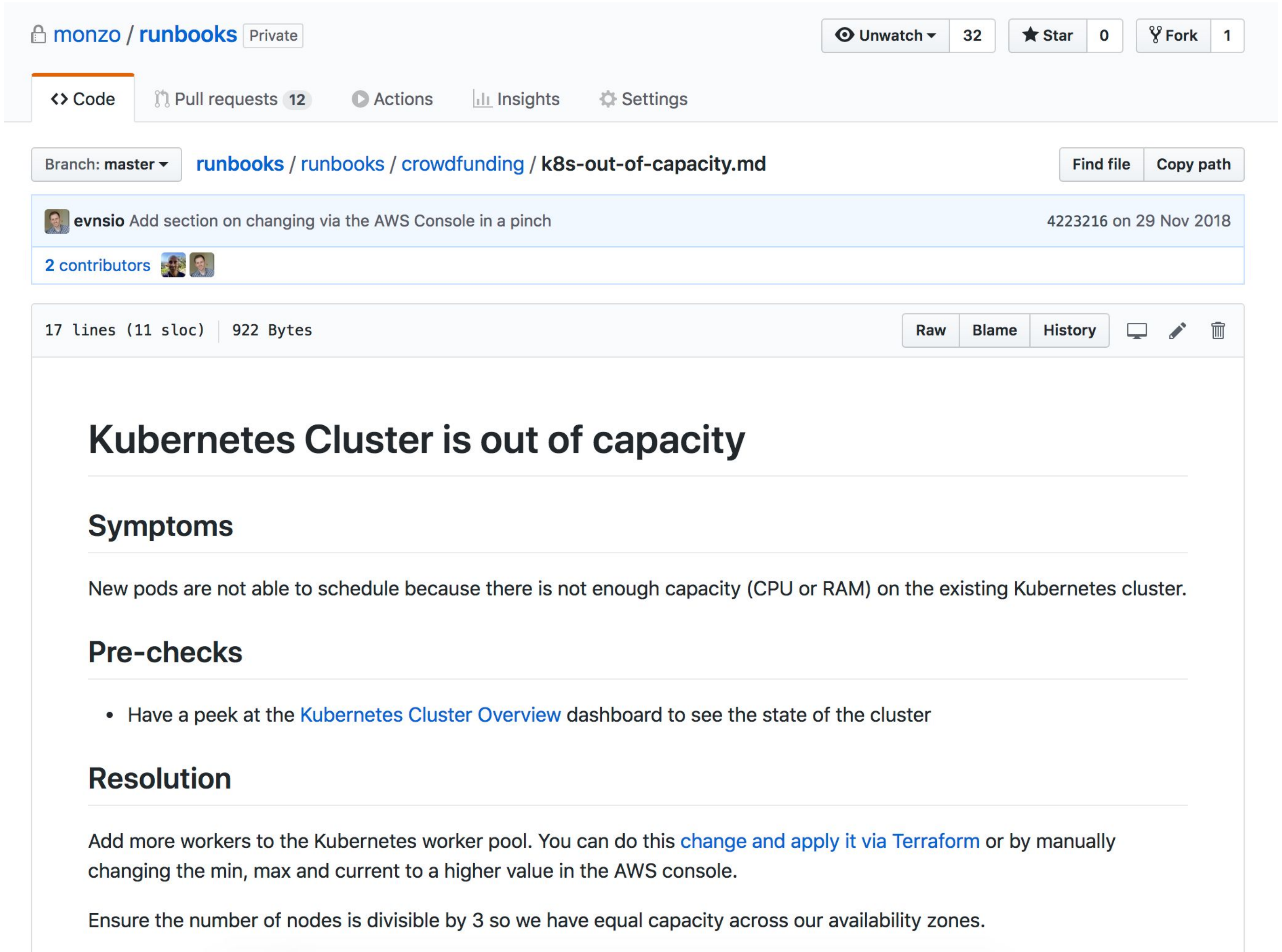
What we ended up with

- A comprehensive spreadsheet of all the services involved and how much we'd need to scale them (replicas/resource requests/limits)
- An idea of how many EC2 Kubernetes Worker Nodes we need, so we could provision them before it started
- Much more knowledge of where things can fail at this scale
- Confidence!
    - Knowing what levers you can pull when things go wrong

Levers

No matter how much preparation we did beforehand, we wanted to ensure we could recover the Platform if anything went wrong

- Feature Toggles
  - Gracefully degrading the less critical app features
- Shedding traffic
  - Stopping the traffic before it even enters our edge



Kubernetes Cluster is out of capacity

Symptoms

New pods are not able to schedule because there is not enough capacity (CPU or RAM) on the existing Kubernetes cluster.

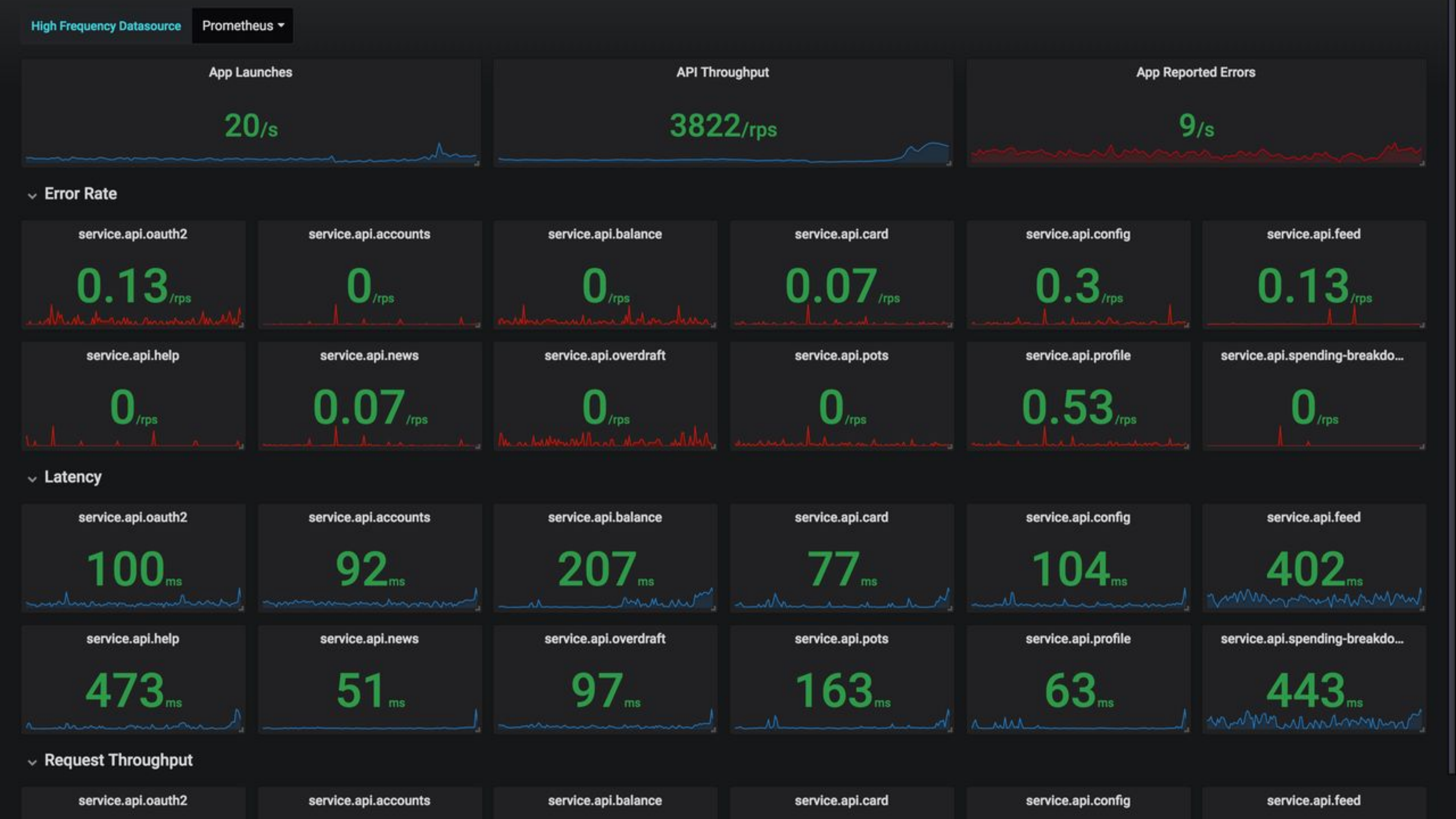
Pre-checks

- Have a peek at the Kubernetes Cluster Overview dashboard to see the state of the cluster

Resolution

Add more workers to the Kubernetes worker pool. You can do this change and apply it via Terraform or by manually changing the min, max and current to a higher value in the AWS console.

Ensure the number of nodes is divisible by 3 so we have equal capacity across our availability zones.



High Frequency Datasource   Prometheus

App Launches 20/s

API Throughput 3822/rps

App Reported Errors 9/s

Error Rate

service.api.oauth2 0.13/rps

service.api.accounts 0/rps

service.api.balance 0/rps

service.api.card 0.07/rps

service.api.config 0.3/rps

service.api.feed 0.13/rps

service.api.help 0/rps

service.api.news 0.07/rps

service.api.overdraft 0/rps

service.api.pots 0/rps

service.api.profile 0.53/rps

service.api.spending-breakdo... 0/rps

Latency

service.api.oauth2 100ms

service.api.accounts 92ms

service.api.balance 207ms

service.api.card 77ms

service.api.config 104ms

service.api.feed 402ms

service.api.help 473ms

service.api.news 51ms

service.api.overdraft 97ms

service.api.pots 163ms

service.api.profile 63ms

service.api.spending-breakdo... 443ms

Request Throughput

service.api.oauth2

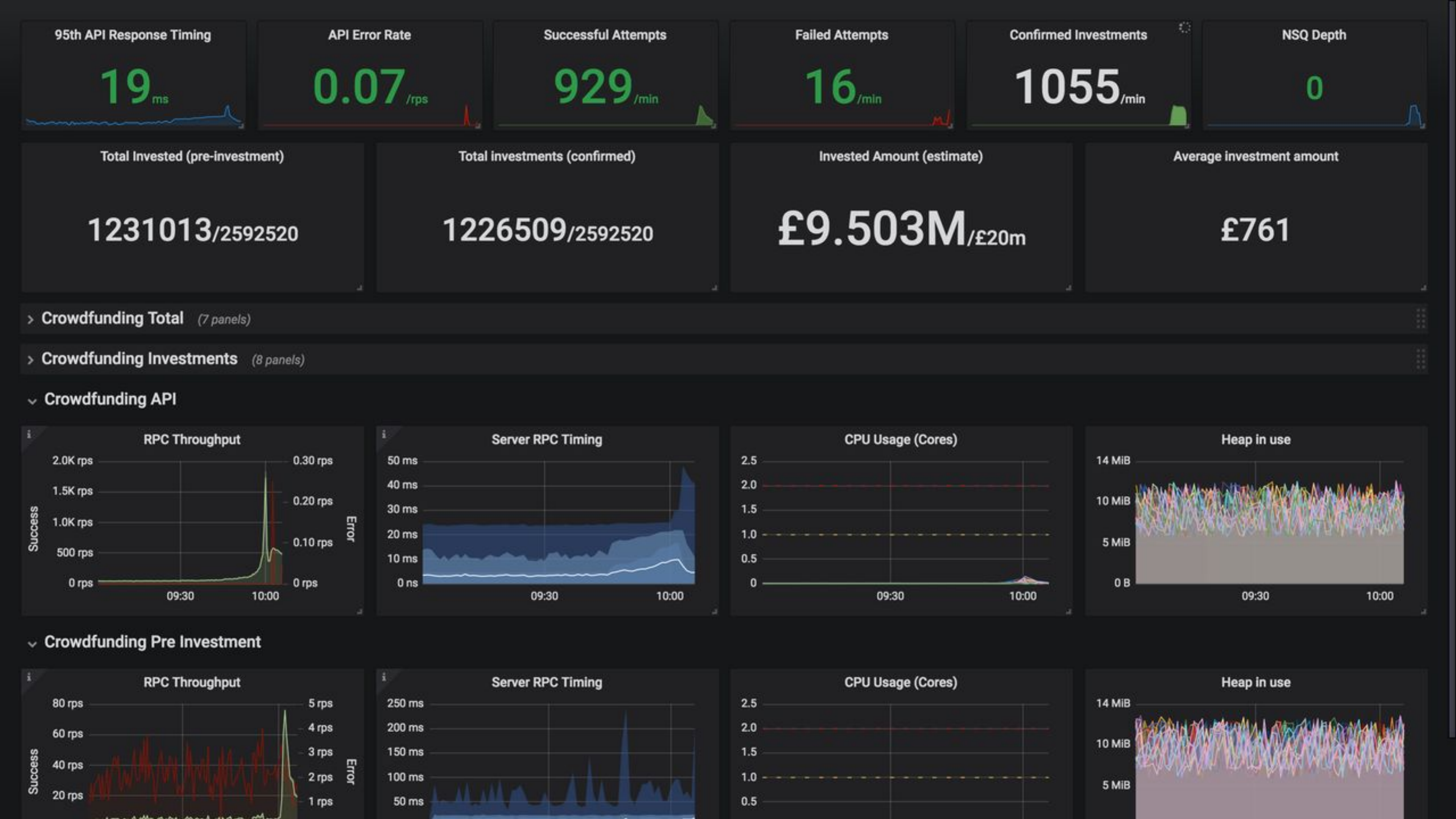
service.api.accounts

service.api.balance

service.api.card

service.api.config

service.api.feed


95th API Response Timing
19ms
API Error Rate
0.07/rps
Successful Attempts
929/min
Failed Attempts
16/min
Confirmed Investments
1055/min
NSQ Depth
0
Total Invested (pre-investment)
1231013/2592520
Total investments (confirmed)
1226509/2592520
Invested Amount (estimate)
£9.503M/£20m
Average investment amount
£761
Crowdfunding Total (7 panels)
Crowdfunding Investments (8 panels)
Crowdfunding API
RPC Throughput
Server RPC Timing
CPU Usage (Cores)
Heap in use
Crowdfunding Pre Investment
RPC Throughput
Server RPC Timing
CPU Usage (Cores)
Heap in use

Things went well

36,006
Investors

£20M
Raised

£6.8M
first 5 minutes

What we learned

Here are the key takeaways and what we learnt as a result of this exercise

- Horizontal scaling has limits
- Treat software as just that, software
- Continuously load test

Thanks!

Email: hi@suhailpatel.com
Twitter: @suhailpatel / @monzo

monzo