ZERO-DOWNTIME DATACENTER FAILOVERS (SWITCHING HOSTING PROVIDERS FOR DUMMIES)

WHO?

Luka Kladaric

formerly a web developer for >10 years

now: freelancing, consulting, architecting, securing

MIGRATING AN ENTIRE COMPANY'S INFRASTRUCTURE

FROM RACKSPACE TO AMAZON AWS

60 virtual machines

3 baremetal boxes (db)

assorted networking equipment

THE MIGRATION TOOK 2 MONTHS TO EXECUTE

BUT A YEAR AND A HALF TO PREPARE

FOUND STATE

hand-crafted build server, unreproducible

half the servers are not deployable from scratch

or their deployability is unknown

same mysql account used by everyone everywhere

that mysql account is "root"

that mysql db is 1.5 TB big

no access to LB config

has a bunch of magic in it

changes often result in issues and outages

no server metrics / perfdata

no idea if overprovisioned and by how much

no access to disaster recovery instance
in case the primary DC went down

(access goes through primary DC)

RACKSPACE WAS REALLY TERRIBLE

a constant pain to deal with

unexpected outages of never explained causes

unresponsive support team

zero flexibility

HOW LONG WOULD IT TAKE TO MIGRATE THIS?

optimistically: 3 months

conservatively: 6-9 months

realistically: a year

NO LEADERSHIP BUY-IN

2 failed attempts to get approval

Infrastructure team makes a pact
"Do Things The Right Way From Now On"

mask cleanup work with ongoing maintenance

A YEAR AND A HALF LATER...

MAJORITY OF THE ISSUES WERE FIXED

OR AT LEAST SIGNIFICANTLY IMPROVED

PLOT TWIST

RACKSPACE STARTS FALLING APART

New estimate: 19 man-days

(after final push for preparation)

SAVINGS ESTIMATE

$18k -> $6k

that's -66%

GOT APPROVAL!

Actually executed in 25-30 man-days

over 2 months

HOW?

"upgrading the fleet to Ubuntu 16.04"

all servers rebuilt and redeployed with Ansible

build server rebuilt from scratch

deployed from Ansible

all build jobs defined in code

no more tweaking jobs through UI

CloudFlare implemented for faster DNS failover

all LB logic slowly moved to our own haproxies

haproxy configuration auto-generated from Ansible

makes it easy to shuffle things around

all apps slowly migrated to be served through haproxies

avoiding Rackspace LB magic

VPN bridge between DCs
~20 MB/s, ~20ms ping

good enough to treat as a "local" connection
for shorter periods of time

mysql master-master replication between DCs

app servers in both DCs

haproxies in both DCs

aware of app servers in both DCs
but preferring local ones

"no request left behind"

failover with DNS at CloudFlare near-instantly

but even stray requests get handled

metrics, metrics, metrics

(Datadog ftw)

RESULTS

core production migrated in days

internal tools migrated within a week or two

developer tools migrated within a month
(git hosting, build server, etc)

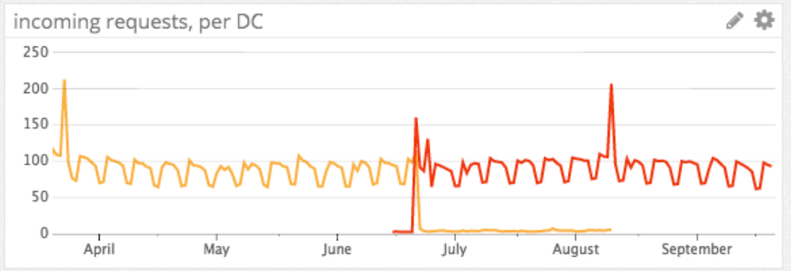
obscure legacy services migrated within 2 months

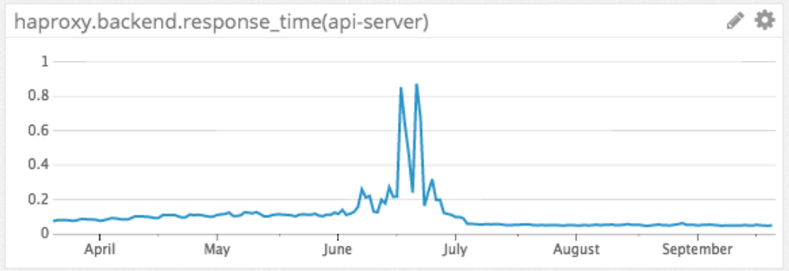
all hardware at Rackspace
decomissioned within 3 months

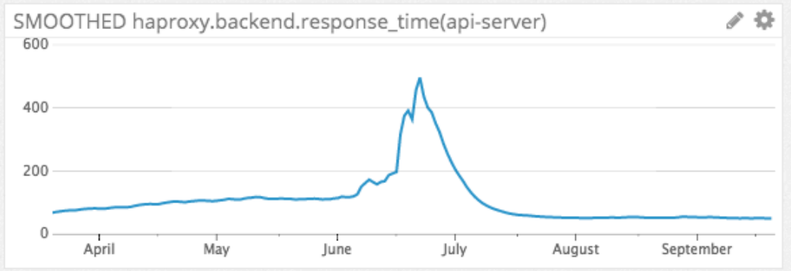
sideffect: actual HA instead of fake HA

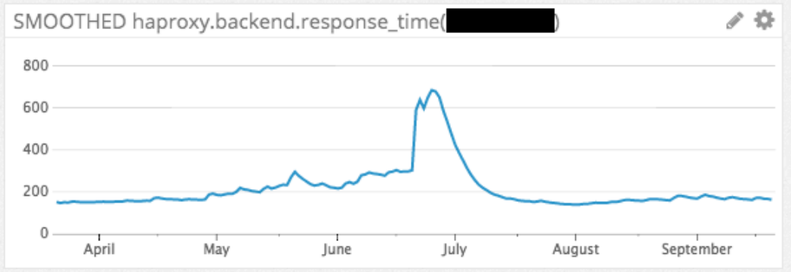
old "two or more of everything" approach translated well into Availability Zones

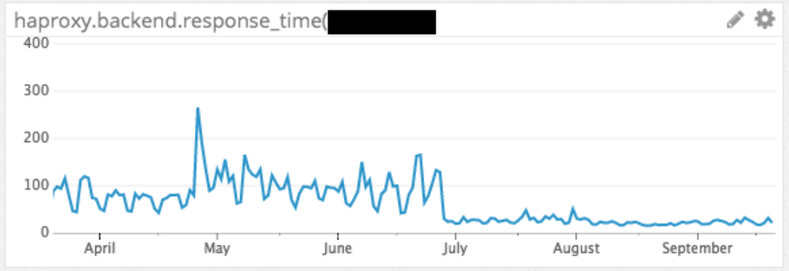
AND IT WAS GOOD

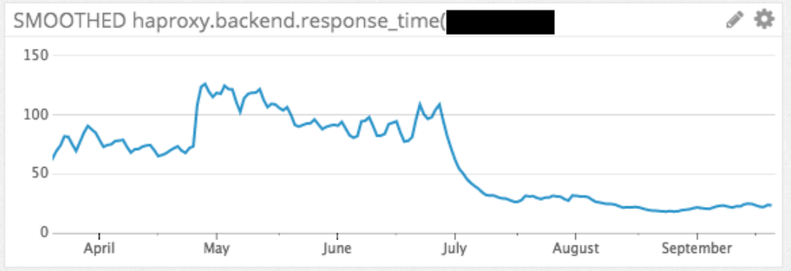


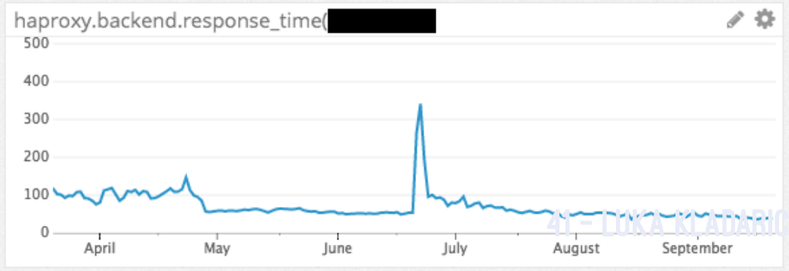












41 — LUKA KLADARIĆ @ AWS ADRIA 2017.

QUESTIONS?

THANK YOU!

LUKA KLADARIC
HELLO@LUKA.IO
@KLL