ZERO-DOWNTIME DATACENTER FAILOVERS (Switching Hosting Providers for Dummies)

WHO?

Luka Kladaric

formerly: web developer for 10+ years now: architecture, infrastructure & security consultant

also a startup founder and remote work evangelist

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...

hand-crafted build server, unreproducible

jobs for 3 Android apps... ...each completely different

massive monolthic 10 GB git repository touching anything triggers a rollout of everything no concept of "stable"

half the servers are not deployable from scratch or their deployability is unknown

no local dev environments

half the company has to VPN into production to get any work done everyone works directly on production systems no db schema migration system == no db versioning

horrible code review tool (Rietveld)

same mysql account used by everyone everywhere

>

>



same mysql account used by everyone everywhere that mysql account is "root"

>



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

optimistically:

conservatively:

realistically:

optimistically: 3 months

conservatively:

realistically:

optimistically: 3 months

conservatively: 6-9 months (of dedicated work)

realistically:

optimistically: 3 months

conservatively: 6-9 months (of dedicated work)

realistically: a year (with interruptions)

NO LEADERSHP BUY-N

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



RACKSPACE STARTS FALLING APART



NEW ESTIMATE

19 man-days

(after final push for preparation)

HOSTING COST ESTIMATE

before: \$18k

after: \$6k

savings: \$12k (-66%!)



Actually executed in 25-30 man-days

over 2 months

build server rebuilt from scratch

deployed from Ansible

all build jobs defined in code with inheritance and templating

tweaking jobs through UI disabled

monolithic git repository split up into 40 smaller repositories

changes trigger rollout only on affected project

all servers rebuilt and redeployed with Ansible "upgrading the fleet to Ubuntu 16.04" ;)



better code review tool (Phabricator)

allows code ownership rules of engagement per repository

don't ask me about Phabricator
(it's amazing)

most dev work doesn't require VPN any more but even if it did...

no more shared mysql root account (RIP)

no write access to production database
(to people *or* their software)

local dev environments!
(far from perfect)

db schema migration system based on gh-ost sql scripts -> code review -> git -> web ui to run

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

metrics, metrics, metrics

(Datadog ftw)



AND ALL BEFORE WE HAD APPROVAL TO DO ANYTHING

"yeah yeah yeah...

how did you do the actual migration?"

VPN bridge between AWS and RS

~20 MB/s, ~20ms ping

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

mysql master-master replication between DCs this was a massive pain to achieve with a 1.5TB db

recreate the entire fleet in AWS

app servers in both DCs (java+python)

haproxies in both DCs

aware of app servers in both DCs preferring local ones but falling back to remote if necessary

"no request left behind"

CloudFlare used for near-instant DNS failover but even stray requests will get handled



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

cost estimate? right on the money.

once the dust settled
the \$18k/mo bill from RS

was replaced
with a \$6k/mo bill from AWS





	14
vvv	www
August	September
er)	/ \$
August	September
)	14
August	September
	14
~~~~	
August	September
	14
August	September

### THE MORAL OF THIS STORY IS: Don't wait for permission to do your job right.

### **1. IF YOU SEE SOMETHING BROKEN. FIX IT**

## IF YOU SEE SOMETHING BROKEN, FIX IT IF YOU DON'T HAVE TIME TO FIX IT - WRITE IT DOWN

# IF YOU SEE SOMETHING BROKEN, FIX IT IF YOU DON'T HAVE TIME TO FIX IT - WRITE IT DOWN BUT DO COME BACK TO IT WHEN YOU CAN STEAL A MINUTE

### **1. IF YOU SEE SOMETHING BROKEN. FIX IT** 2. IF YOU DON'T HAVE TIME TO FIX IT - WRITE IT DOWN **3. BUT DO COME BACK TO IT WHEN YOU CAN STEAL A MINUTE 4.** EVEN IF IT TAKES MONTHS TO MAKE PROGRESS

### THE TEAM WAS WELL AWARE OF HOW BROKEN THINGS WERE. IF WE PUSHED FOR IT TO BE A SINGLE MASSIVE **PROJECT. IT WOULD'VE NEVER HAPPENED.**

### 



Luka Kladaric twitter: @kll luka@sekura.io www.sekura.io

