Adapting brownfield technology to cloud-native technology

One year of Prometheus

Richard Hartmann,
RichiH@{freenode,OFTC,IRCnet},
richih@{fosdem,debian,richih}.org

2016-11-09

‘whoami‘

- Richard ”RichiH” Hartmann
- System architect at SpaceNet AG
- FOSDEM, DebConf, DENOGx, PromCon staff
- Debian Developer
- Author of `https://github.com/RichiH/vcsh`
- Always looking for nice co-workers in the Munich area

‘whowasi‘

- 2009 - 2015: Solely responsible for a Germany-wide backbone’s
  - Architecture
  - Purchasing
  - Operations
  - ...and On-Call for 24 * 365 * 7 hours
- Literally, my sanity depended on aggressive, yet correct, monitoring & alerting
- Love monitoring, but despise (almost) all monitoring tools
- Used Zabbix exclusively

SpaceNet

- SpaceNet is the oldest commercial ISP in Germany; operating since 1993
- Legacy, in-house solutions which predate everything else
- One company-wide monitoring solution: watchdog & watchcat
- Powerful and efficient, but alerting done through B52-style email carpet bombing
- Every team has its own custom tools on top
- Islands of data: no APIs, no machine-readable export

Seeing the light

- Ran DebConf15 on LibreNMS, wanted to do the same for FOSDEM 2016
- 2015-10-01: Inform FOSDEM team of planned migration
- 2015-10-02: Murali Suriar suggests Prometheus instead
- 2015-10-03: PoC at SpaceNet and submit first patch

Prometheus 101

- Incredibly efficient
- No hierarchical model, n-dimensional label set
- PromQL: for graphing, alerting, and export
- Delta-of-delta data storage
- One binary; up and running in minutes

Here, there be networks

- We are an ISP, remember?
- Roughly 1000 devices polled via SNMP
- Might be the world's largest snmp_exporter installation
- Python implementation at pathologic system load
  - It goes up to eleven...
  - About 60/300 devices flapping
  - Set of affected devices stable
  - Never found root cause

Solution

- Contracted Brian Brazil to reimplement in Go
- Go implementation hit some unexpected pitfalls of real life SNMP
  - Some data structures returned repeatedly
  - Duplicate identifiers
  - Table indices emtpy
- Go errors out completely for those
- Still using Python for affected devices

Caveats

- InetAddress broken in Python
- IOS XR non-standard layout not fully supported yet
- Some devices die when polled too often

That was easy; let's have lunch!

The biggest challenge

The hardest problems to solve are the social ones.

Resistance to change

- Incentives often run counter to change
- Change is hard
- Unless processes embrace and automate change
- Trade-off between delayed/disputed payoff during transition
- Due diligence: Critical systems run in parallel for some time

Toil

- If teams are busy firefighting, they don't have time to engineer
- Keep extra effort on the team low, if possible
- Strive for immediate benefits
- Focus on removing repeated, manual tasks of no lasting benefit
- Show that you free up time and reduce toil

Sanity & sleep

- If it's not actionable, it's not an alert
- If it's not urgent, it's not an alert
- Important, but not urgent, stuff is handled during business hours
- Predict your usage so you add capacity during business hours
- If there's no playbook, it does not go into production
- If a service does not have proper SLOs and alerts, it does not go into production

That one mailserver incident...

- Wrong flag in config
- One server accepting outside mail
- Spammers do a clean, staggered ramp-up
- Once they go all-in the mail gateways come under heavy load
- Quote from On-Call ”It took me less than 30 seconds to figure out the problem; with our old system it would have taken at least 15 minutes”
- ...and all of a sudden, you have buy-in from a few more people

Perspective & Incentives

”An engineer can talk for hours about a function; try that with the CEO”

- Managers: revenue, process execution
- Architects: clean design, process defintion
- Product/Service owners: Powerful dashboards
- Team leads: morale, quick execution
- Operators: reduce toil, increase sleep
- Tell everyone what they need to hear (but never lie)

Big Picture

- Put a big picture on the (proverbial) wall
- Show everyone the pieces they care about
- Make sure to play to their intrinsic motivation
- Get buy-in
- Going forward, align steps with that picture
- Distributed alignment with goals across teams

Leverage

- One combined system allows for correlation and combination
- Power usage against service load
- Optical networks against outside temperature
- Datacenter power feed load against new deployments
- ...and lots more

Leverage

- One source of truth for
  - Tactical overview for current state
  - Dashboards for drill-down
  - Auto-generated PDFs for customers
  - Global SLO statements for sales
  - Usage exports for accounting
- If all you have is a hammer... choose your hammer well

TODO

- Merge config management across teams
- Adapt machines and services to modern orchestration
  - Highly fractured and specific customer setups
  - Revenue comes from those brownfield installation
  - Finding the correct balance will be tricky
- Adopt error budgets
- Hire more people. Munich is beautiful!

Thanks!

Thanks for listening!

Questions?

See slide footer for contact information.