>_ ENDOCODE

THE DEATH OF THE SYSADMINS

… and their resurrection as Resilience Engineers, Application Managers, Chaos Engineers

Thomas Fricke thomas@endocode.com

Secure Linux Administration Conference (SLAC), May 7, 2018

HI!

Thomas Fricke

thomas@endocode.com

CTO Endocode

- System Automation
- DevOps
- Cloud, Database and Software Architect
- K8S since September 2015

MY FIRST COMPUTERS

By Rama, CC BY-SA 2.0 fr,
https://commons.wikimedia.org/w/index.php?curid=11276454

By Bill Bertram, CC-BY-2.5,
https://commons.wikimedia.org/wiki/File:Atari_1040STf.jpg

digital

By Stephen Edmonds (http://computers.popcorn.cx)
[CC BY-SA 2.5 au (https://creativecommons.org/licenses/by-sa/2.5/au/deed.en)],
via Wikimedia Commons

SITUATION TODAY

- 95% still on premises (Urs Hölzle)
- 5% in the cloud
- We will see 50%-50% in a few years
- Market Leaders
  - AWS
  - Azure
  - Google

THE DATACENTER IN THE AGE OF ABUNDANCE

History

- Disks
  - disks of rotating rust: perform 200 disk seeks (I/O Operations per second, IOPS),
  - Five years ago we converted all customer databases to SSD with about 20.000 to 50.000 IOPS
  - Future: millions of IOPS. Fundamentally, IOPS are not a limited resource any more
- Network
  - Five years ago, we converted the first systems to 10 GBit/s at scale,
  - Today: 400 MBit/s to 1 GBit/s per Thread (so a 50 core system gets a dual-25 GBit/s network cards
  - Mellanox: with a large two digit number of 100 GBit/s Interfaces.
  - leaf-and-spine architectures: getting the 1 GBit/s per thread on the entire path between *any* thread and *any* disk in our data center, concurrently
- Latency
  - In the past 500 µs (1/2000 of a second) and more likely in the low milliseconds
  - Today: below 200 µs.
  - Add scary stuff such as RDMA/RoCE to the mix, and we may be able to routinely crack the 100 µs barrier. That makes writes to the data center sized fabric as fast or faster than writes to a slow local SSD

THE DATACENTER IN THE AGE OF ABUNDANCE
STATEMENT

- Today we are at an inflection point,
- Each of the three limiters, IOPS, bandwidth and latency, have been throughly vanquished

“We can now build a system where the data center sized fabric at scale provides bandwidth and latency comparable to a system bus of a slow home computer (and is consecutively faster the smaller the domain gets). We can build machines the size of a data center, up and past one million cores, that provide essentially enough coupling to be able to act as a single machine.”

THE DATACENTER IN THE AGE OF ABUNDANCE
HARDWARE

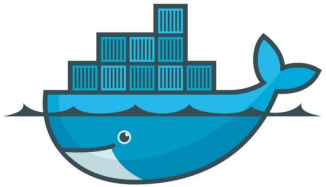
The building blocks are Open Compute Racks at 12 kW a piece.

*The **Open Compute Project** (**OCP**) is an organization that shares designs of data center products among companies, including Facebook, Intel, Nokia, Google, Microsoft, Seagate Technology, Dell, Rackspace, Cisco, Goldman Sachs, Fidelity, Lenovo and Alibaba Group.*
The Open Compute Project's mission is to design and enable the delivery of the most efficient server, storage and data center hardware designs for scalable computing. "We believe that openly sharing ideas and specifications is the key to maximizing innovation and reducing operational complexity in the scalable computing space

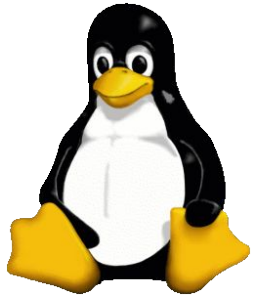
https://en.wikipedia.org/wiki/Open_Compute_Project

SOFTWARE STACK

- **The operating system of the machine is Kubernetes.**
- **The units of work are container images.**
- **The local API is the Linux Kernel API.**



cri-o



http://blog.koehntopp.info/index.php/2088-the-data-center-in-the-age-of-abundance/

GOOGLE

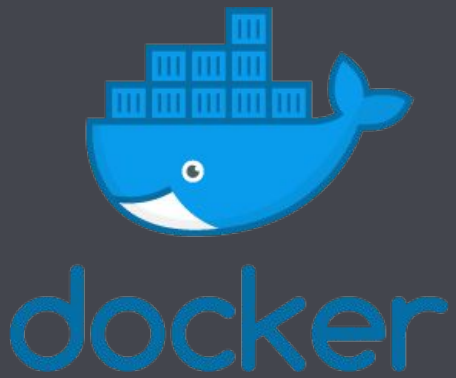
Everything at Google runs in containers:

- Gmail, Web Search, Maps, ...
- MapReduce, batch, ...
- GFS, Colossus, ...
- Even **Google's Cloud Platform:** our VMs run in containers!

We launch over **2 billion** containers **per week**

Shipping Containers At Clyde, by Steve Gibson

CONTAINERS



- Isolation based on Linux
- No Hypervisor necessary
- 30% more efficient than virtual machines (jd.com)
- Efficient distribution format
- Docker made it popular
- OCI (Open Container Initiative) is a standard
- Available for more than a decade
- Google runs everything in a container since the mid 2000s

cri-o

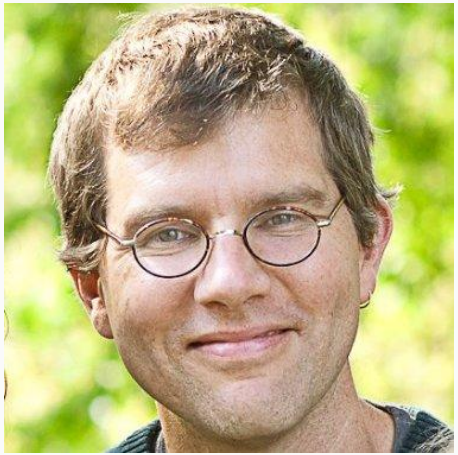
Greek for *“Helmsman”*; also the root of the words *“governor”* and *“cybernetic”*

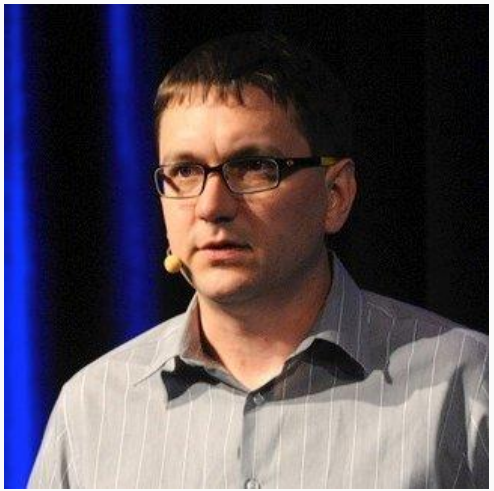
- Runs and manages containers
- Inspired and informed by Google’s experiences and internal systems
- Supports multiple cloud and bare-metal environments
- Supports multiple container runtimes
- **100% Open source**, written in Go

Manage applications, not machines

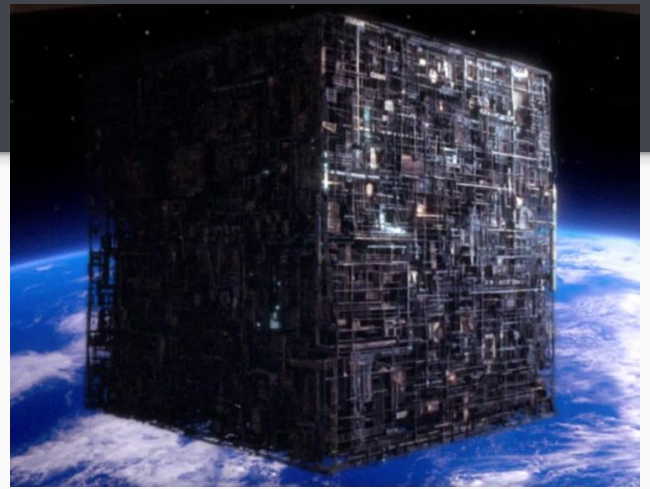
HISTORY

Brendan Burns (Microsoft), Joe Beda and Craig McLuckie (Heptio)



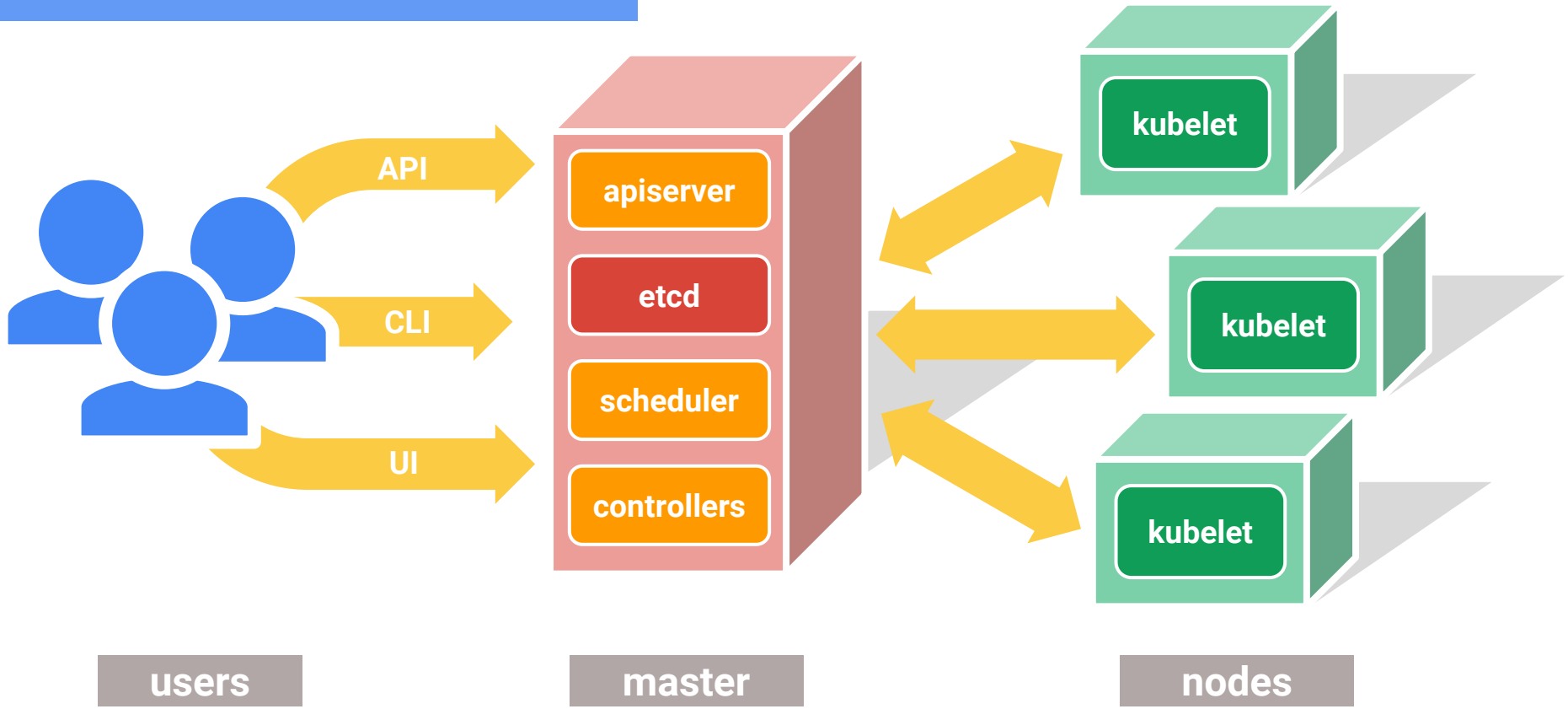


HISTORY

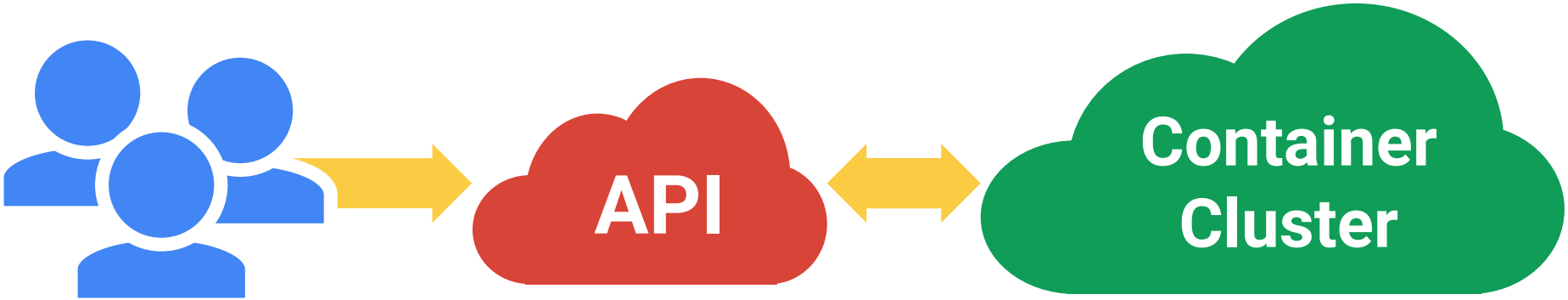


- Announced by Google in mid-2014.
- Successor of Google's Borg system
- Many Borg Contributors
- Project Seven
- Seven spokes
  on the wheel

The 10000 foot view



All you really care about

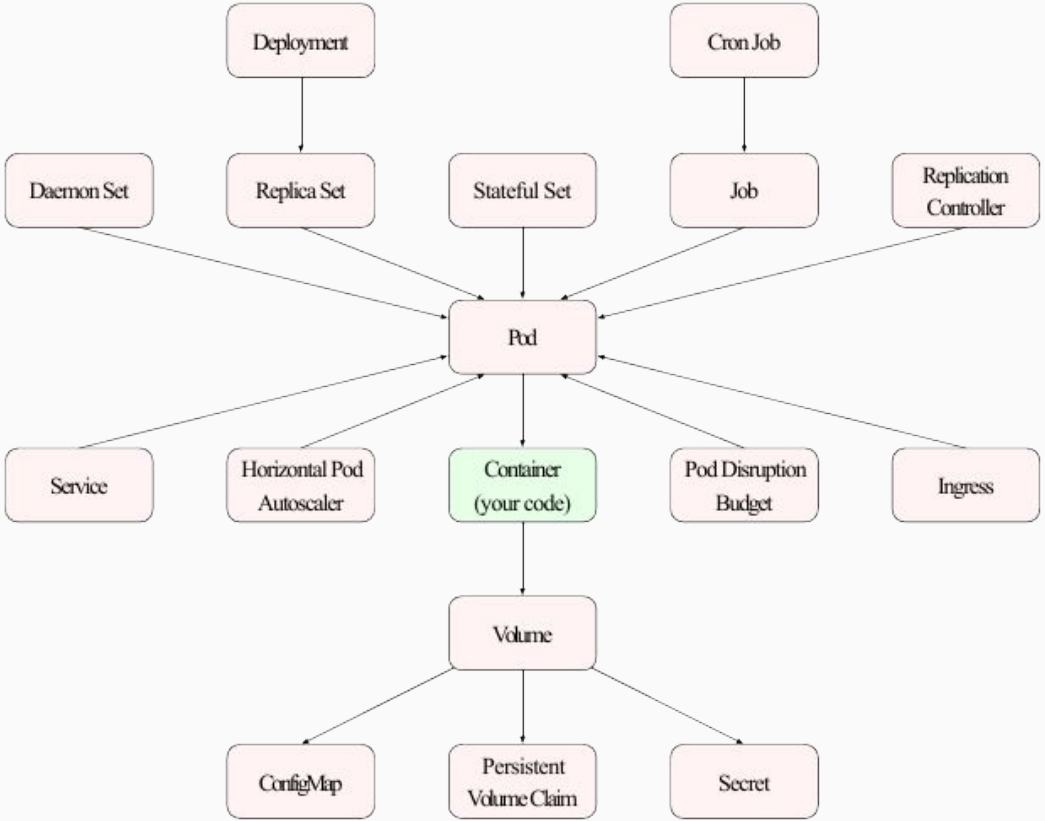


WHY KUBERNETES?

- #GIFEE
- Open Source
- Google Governance
- Release Cycle: three months
- Contributions from lot of parties
  Google, CoreOS, Red Hat, IBM, Huawei
- Available in all clouds
- Available on premises
- Ubiquitous
  GKE, Azure, AWS
- Will be the Operating System of the Datacenter
- 3000+ projects on top of Kubernetes

POD in K8S

Pod Centric View



from Roland Huss
https://github.com/ro14nd-talks/kubernetes-patterns

DEMO

GONE WITH THE WIND

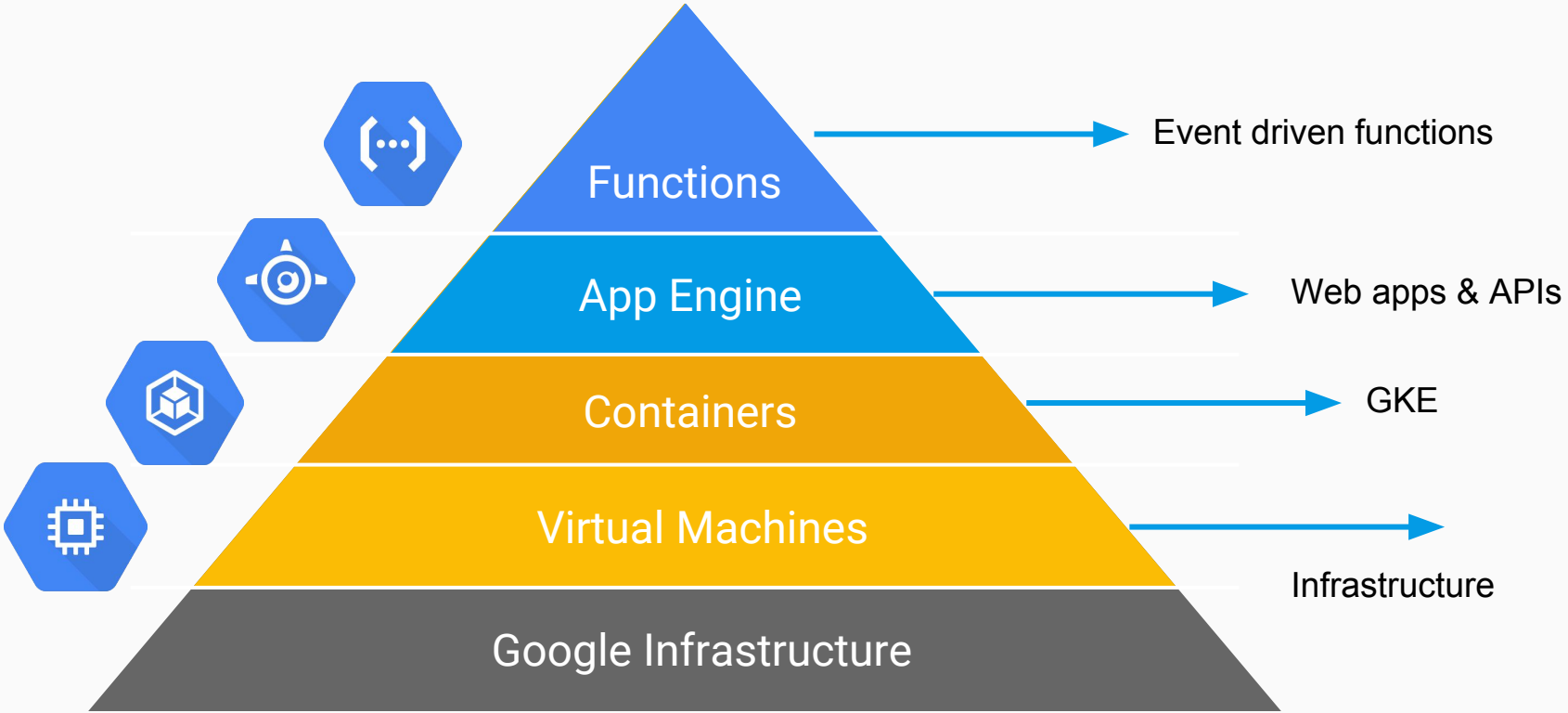
Say Good Bye to

- Package Management
- Config Management
- Network Setup

INSTEAD NOW

- The Microservices Paradigm
  - What is different
  - Organisational and technical impact
  - Stateless Applications: Cattle vs pets
  - Pods
  - Services
  - Ingress
- 12 factor https://12factor.net/
- Patterns
- Implementing Patterns
  - Helm Charts
  - Service Discovery

GOOGLE *AAS Pyramide



DO I HAVE TO RUN IT IN THE CLOUD?

WHAT ABOUT OUR FREEDOM?

WHY NOT RUNNING YOUR OWN PaaS CLOUD?

FREE VERSION

Red Hat's Largest Deals Now Coming from OpenShift Containers

The largest deal was virtually entirely OpenShift, actually two of the top four were primarily OpenShift," Red Hat CEO Jim Whitehurst said during his company's earnings call. "Two of the others were virtually entirely OpenStack".

https://www.serverwatch.com/server-news/red-hats-largest-deals-now-coming-from-openshift-containers.htm

THERE IS NOTHING LEFT TO DO?

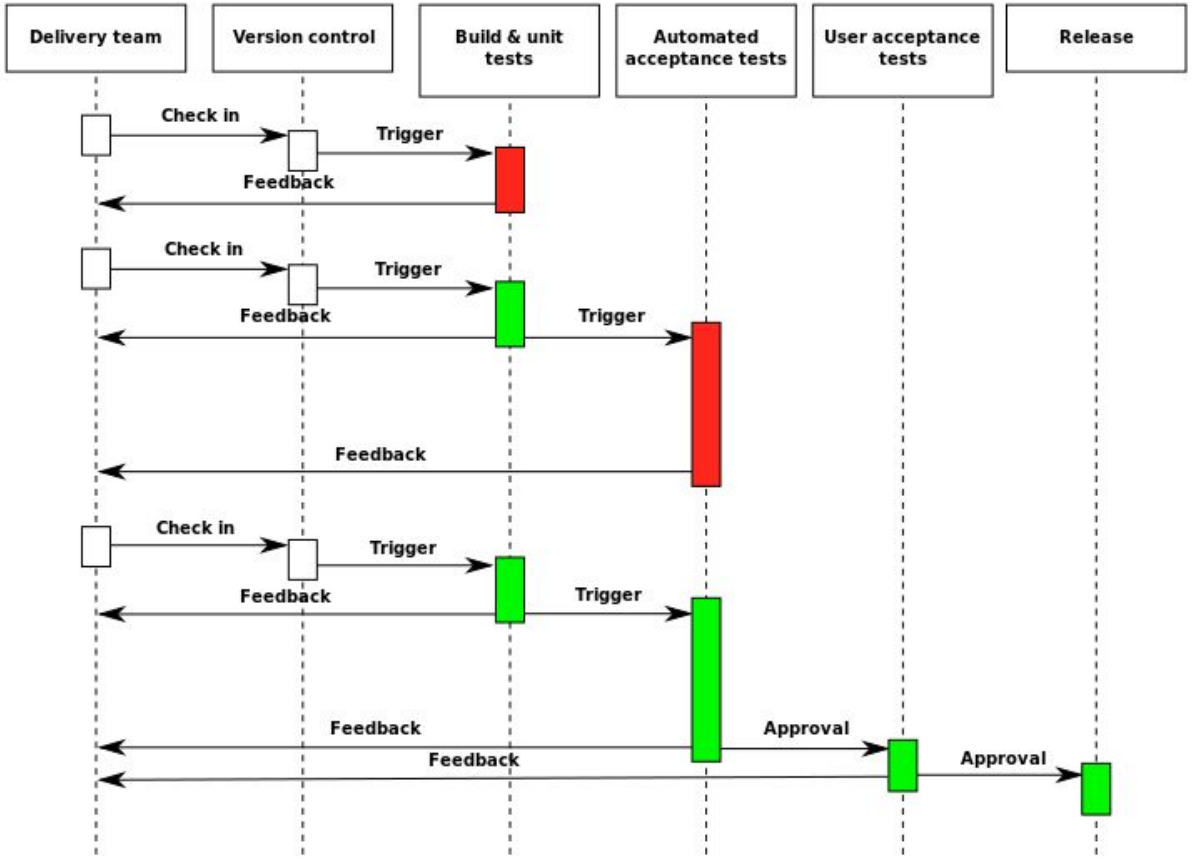
My Datacenter is automated, what now? I feel useless?

Wait, wait wait...

- Automating the Platform is a MUST
- Chaos Engineering
- Distributed Data Center
- Better Testing
- More Sophisticated Distributed Applications

CLD
Continuous Live Delivery and Deployment

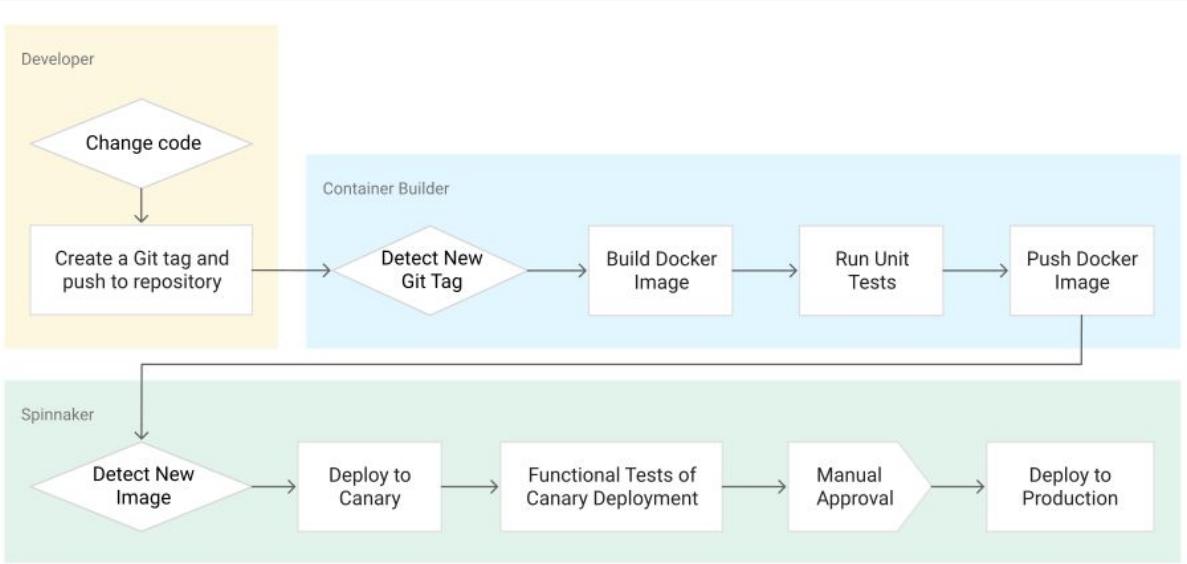
- (Nearly) Automated
- Quality Gates
- Tests on different levels
- Continuous Live
  - Delivery (ready)
    One manual step
  - Deployment (done)
    Fully Automated



Google Spinnaker

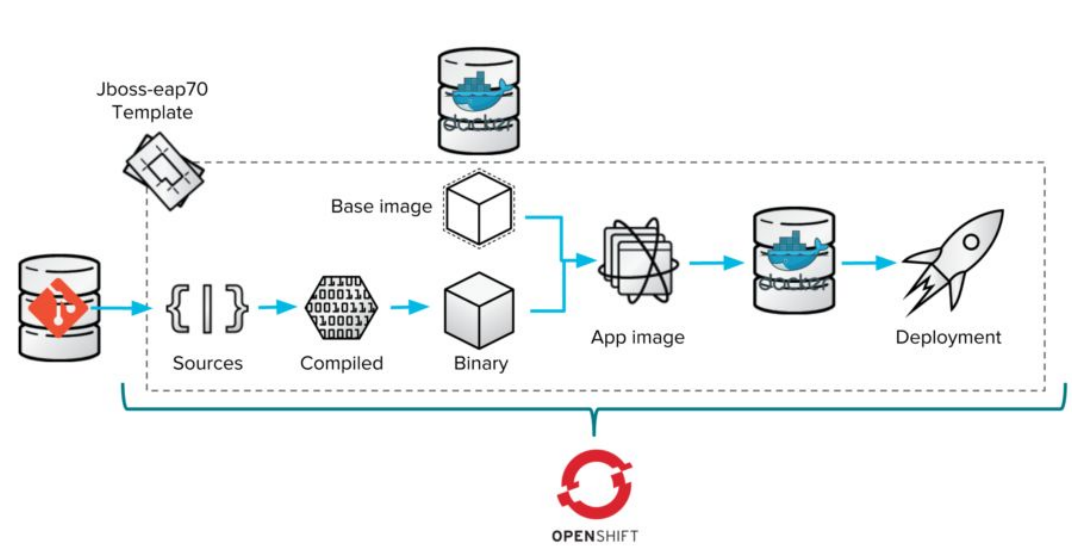
Like

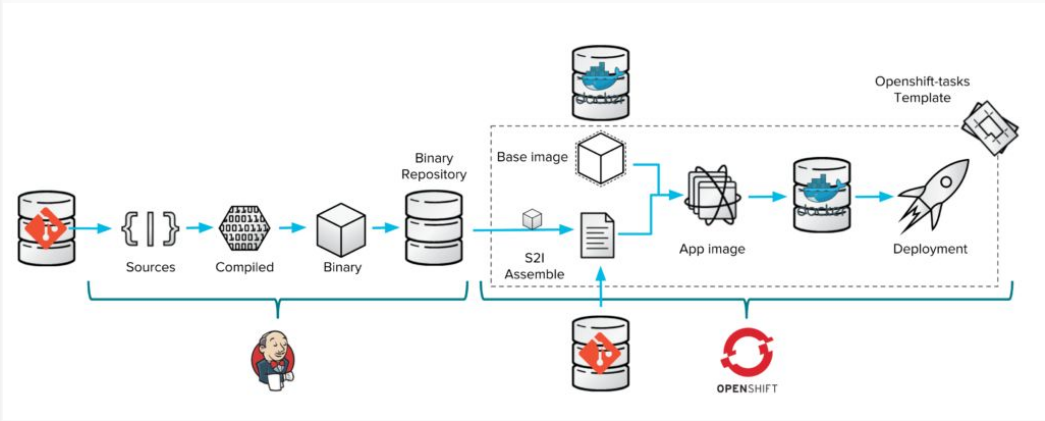
- Jenkins
- Travis
- Teamcity
- Gitlab CI
- Amazon Pipeline
- Concourse
- ...

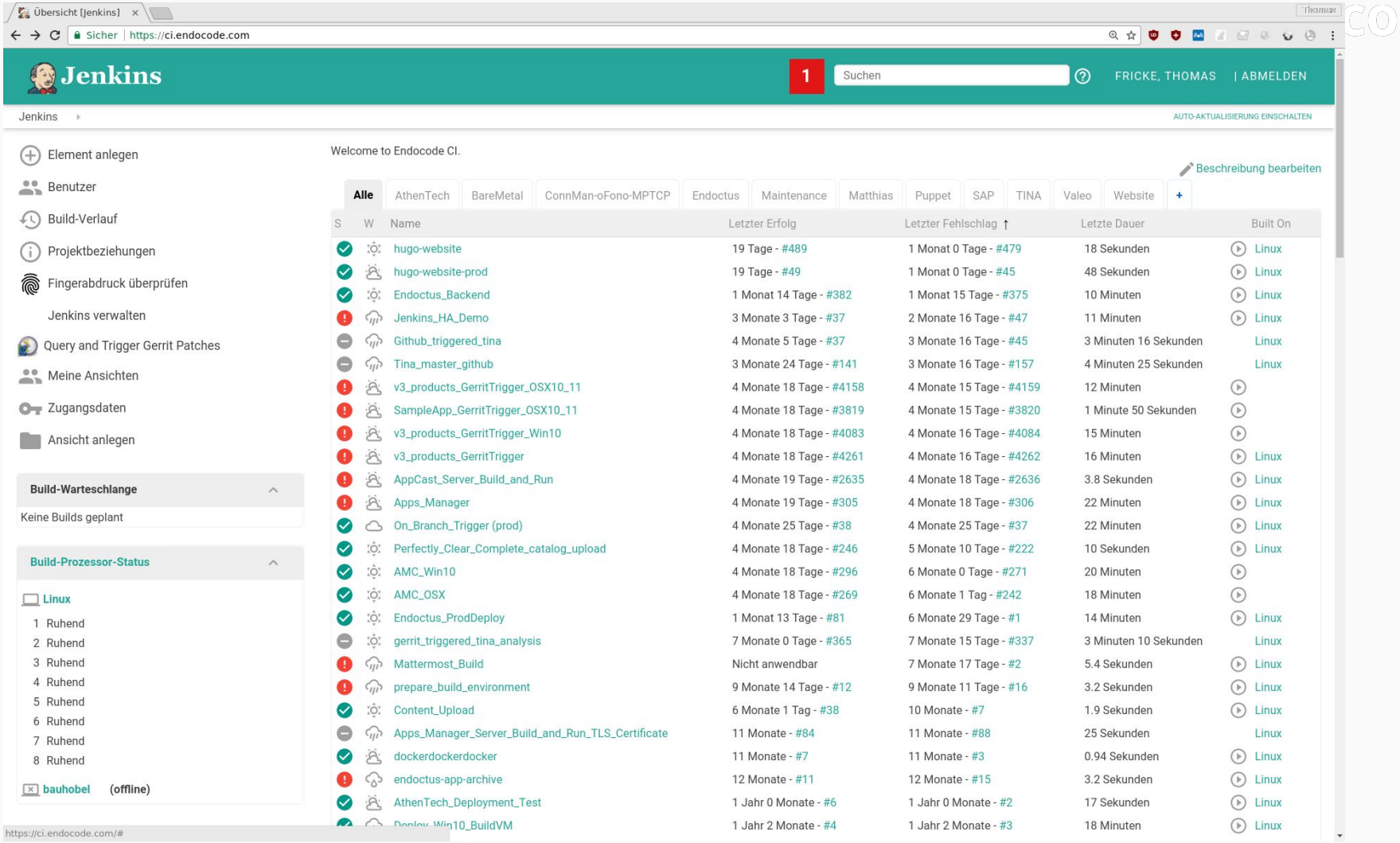


OpenShift Deployment Pipeline

S2I: Source to Image





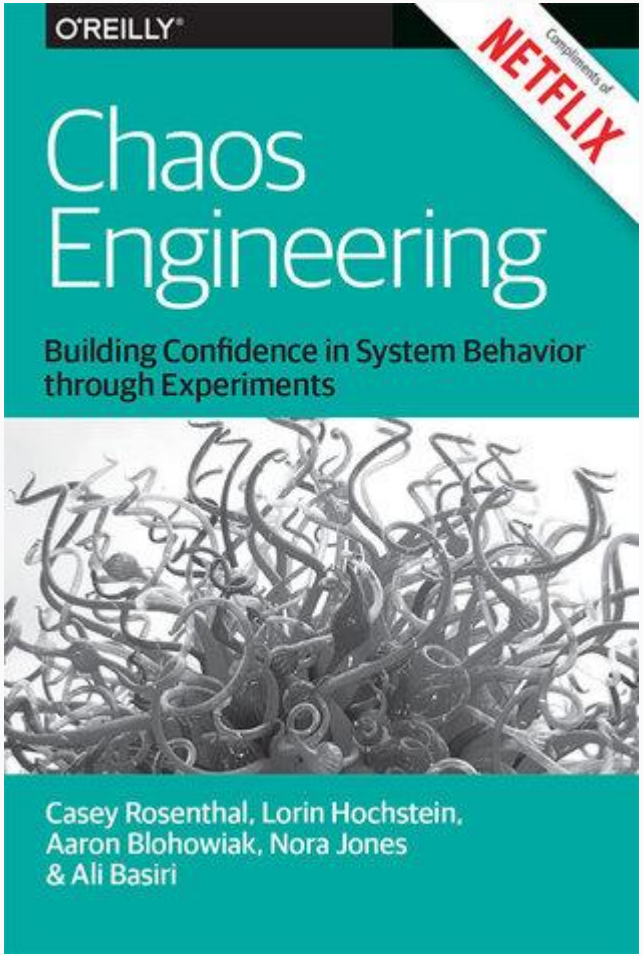

Übersicht [Jenkins]
Sicher | https://ci.endocode.com
Jenkins
1
Suchen
FRICKE, THOMAS | ABMELDEN
Jenkins
AUTO-AKTUALISIERUNG EINSCHALTEN
Element anlegen
Benutzer
Build-Verlauf
Projektbeziehungen
Fingerabdruck überprüfen
Jenkins verwalten
Query and Trigger Gerrit Patches
Meine Ansichten
Zugangsdaten
Ansicht anlegen
Build-Warteschlange
Keine Builds geplant
Build-Prozessor-Status
Linux
1 Ruhend
2 Ruhend
3 Ruhend
4 Ruhend
5 Ruhend
6 Ruhend
7 Ruhend
8 Ruhend
bauhobel (offline)
Welcome to Endocode CI.
Beschreibung bearbeiten
Alle
AthenTech
BareMetal
ConnMan-oFono-MPTCP
Endoctus
Maintenance
Matthias
Puppet
SAP
TINA
Valeo
Website
+
S W Name Letzter Erfolg Letzter Fehlschlag ↑ Letzte Dauer Built On
hugo-website 19 Tage - #489 1 Monat 0 Tage - #479 18 Sekunden Linux
hugo-website-prod 19 Tage - #49 1 Monat 0 Tage - #45 48 Sekunden Linux
Endoctus_Backend 1 Monat 14 Tage - #382 1 Monat 15 Tage - #375 10 Minuten Linux
Jenkins_HA_Demo 3 Monate 3 Tage - #37 2 Monate 16 Tage - #47 11 Minuten Linux
Github_triggered_tina 4 Monate 5 Tage - #37 3 Monate 16 Tage - #45 3 Minuten 16 Sekunden Linux
Tina_master_github 3 Monate 24 Tage - #141 3 Monate 16 Tage - #157 4 Minuten 25 Sekunden Linux
v3_products_GerritTrigger_OSX10_11 4 Monate 18 Tage - #4158 4 Monate 15 Tage - #4159 12 Minuten
SampleApp_GerritTrigger_OSX10_11 4 Monate 18 Tage - #3819 4 Monate 15 Tage - #3820 1 Minute 50 Sekunden
v3_products_GerritTrigger_Win10 4 Monate 18 Tage - #4083 4 Monate 16 Tage - #4084 15 Minuten
v3_products_GerritTrigger 4 Monate 18 Tage - #4261 4 Monate 16 Tage - #4262 16 Minuten Linux
AppCast_Server_Build_and_Run 4 Monate 19 Tage - #2635 4 Monate 18 Tage - #2636 3.8 Sekunden Linux
Apps_Manager 4 Monate 19 Tage - #305 4 Monate 18 Tage - #306 22 Minuten Linux
On_Branch_Trigger (prod) 4 Monate 25 Tage - #38 4 Monate 25 Tage - #37 22 Minuten Linux
Perfectly_Clear_Complete_catalog_upload 4 Monate 18 Tage - #246 5 Monate 10 Tage - #222 10 Sekunden Linux
AMC_Win10 4 Monate 18 Tage - #296 6 Monate 0 Tage - #271 20 Minuten
AMC_OSX 4 Monate 18 Tage - #269 6 Monate 1 Tag - #242 18 Minuten
Endoctus_ProdDeploy 1 Monat 13 Tage - #81 6 Monate 29 Tage - #1 14 Minuten Linux
gerrit_triggered_tina_analysis 7 Monate 0 Tage - #365 7 Monate 15 Tage - #337 3 Minuten 10 Sekunden Linux
Mattermost_Build Nicht anwendbar 7 Monate 17 Tage - #2 5.4 Sekunden Linux
prepare_build_environment 9 Monate 14 Tage - #12 9 Monate 11 Tage - #16 3.2 Sekunden Linux
Content_Upload 6 Monate 1 Tag - #38 10 Monate - #7 1.9 Sekunden Linux
Apps_Manager_Server_Build_and_Run_TLS_Certificate 11 Monate - #84 11 Monate - #88 25 Sekunden Linux
dockerdockerdocker 11 Monate - #7 11 Monate - #3 0.94 Sekunden Linux
endoctus-app-archive 12 Monate - #11 12 Monate - #15 3.2 Sekunden Linux
AthenTech_Deployment_Test 1 Jahr 0 Monate - #6 1 Jahr 0 Monate - #2 17 Sekunden Linux
Deploy_Win10_BuildVM 1 Jahr 2 Monate - #4 1 Jahr 2 Monate - #3 18 Minuten Linux
https://ci.endocode.com/#

~~Chaos~~ Resilience Engineering

Netflix Simian Army

From Chaos Monkey to

Chaos Kong



The Netflix Simian Army

Chaos Monkey randomly disables our production instances to make sure we can survive this common type of failure without any customer impact. The name comes from the idea of unleashing a wild monkey with a weapon in your data center (or cloud region) to randomly shoot down instances and chew through cables – all the while we continue serving our customers without interruption. By running Chaos Monkey in the middle of a business day, in a carefully monitored environment with engineers standing by to address any problems, we can still learn the lessons about the weaknesses of our system, and build automatic recovery mechanisms to deal with them. So next time an instance fails at 3 am on a Sunday, we won't even notice.

Latency Monkey induces artificial delays in our RESTful client-server communication layer to simulate service degradation and measures if upstream services respond appropriately. In addition, by making very large delays, we can simulate a node or even an entire service downtime (and test our ability to survive it) without physically bringing these instances down. This can be particularly useful when testing the fault-tolerance of a new service by simulating the failure of its dependencies, without making these dependencies unavailable to the rest of the system.

Conformity Monkey finds instances that don't adhere to best-practices and shuts them down. For example, we know that if we find instances that don't belong to an auto-scaling group, that's trouble waiting to happen. We shut them down to give the service owner the opportunity to re-launch them properly.

Doctor Monkey taps into health checks that run on each instance as well as monitors other external signs of health (e.g. CPU load) to detect unhealthy instances. Once unhealthy instances are detected, they are removed from service and after giving the service owners time to root-cause the problem, are eventually terminated.

Janitor Monkey ensures that our cloud environment is running free of clutter and waste. It searches for unused resources and disposes of them.

Security Monkey is an extension of Conformity Monkey. It finds security violations or vulnerabilities, such as improperly configured AWS security groups, and terminates the offending instances. It also ensures that all our SSL and DRM certificates are valid and are not coming up for renewal.

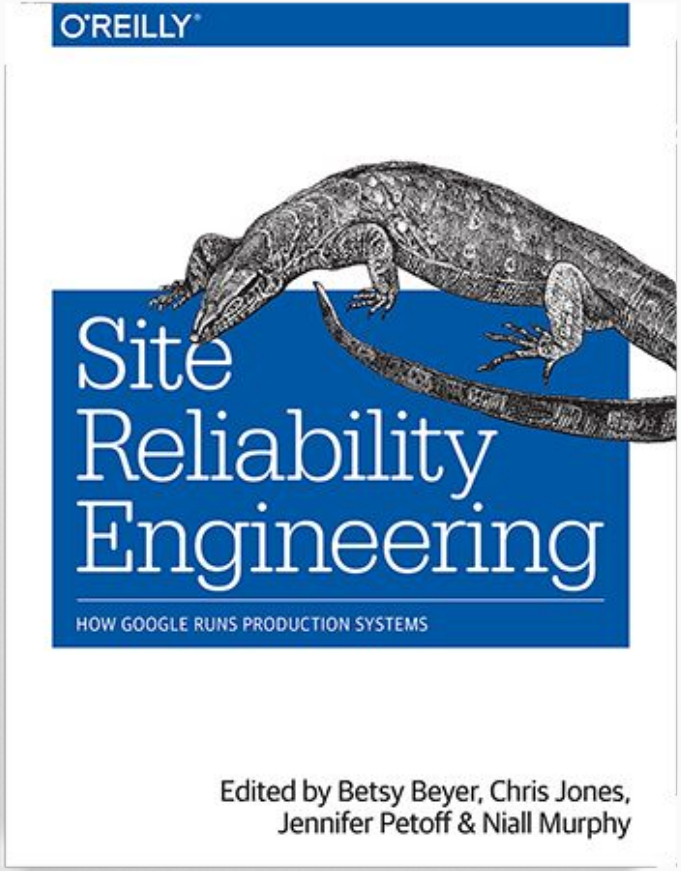
10-18 Monkey (short for Localization-Internationalization, or l10n-i18n) detects configuration and run time problems in instances serving customers in multiple geographic regions, using different languages and character sets.

Chaos Gorilla is similar to Chaos Monkey, but simulates an outage of an entire Amazon availability zone. We want to verify that our services automatically re-balance to the functional availability zones without user-visible impact or manual intervention.

Google SRE

- Every application can be stopped any time
- Memory overcommit
- Effective 10%
- 800M$ savings at 8G$ / year Data Center costs

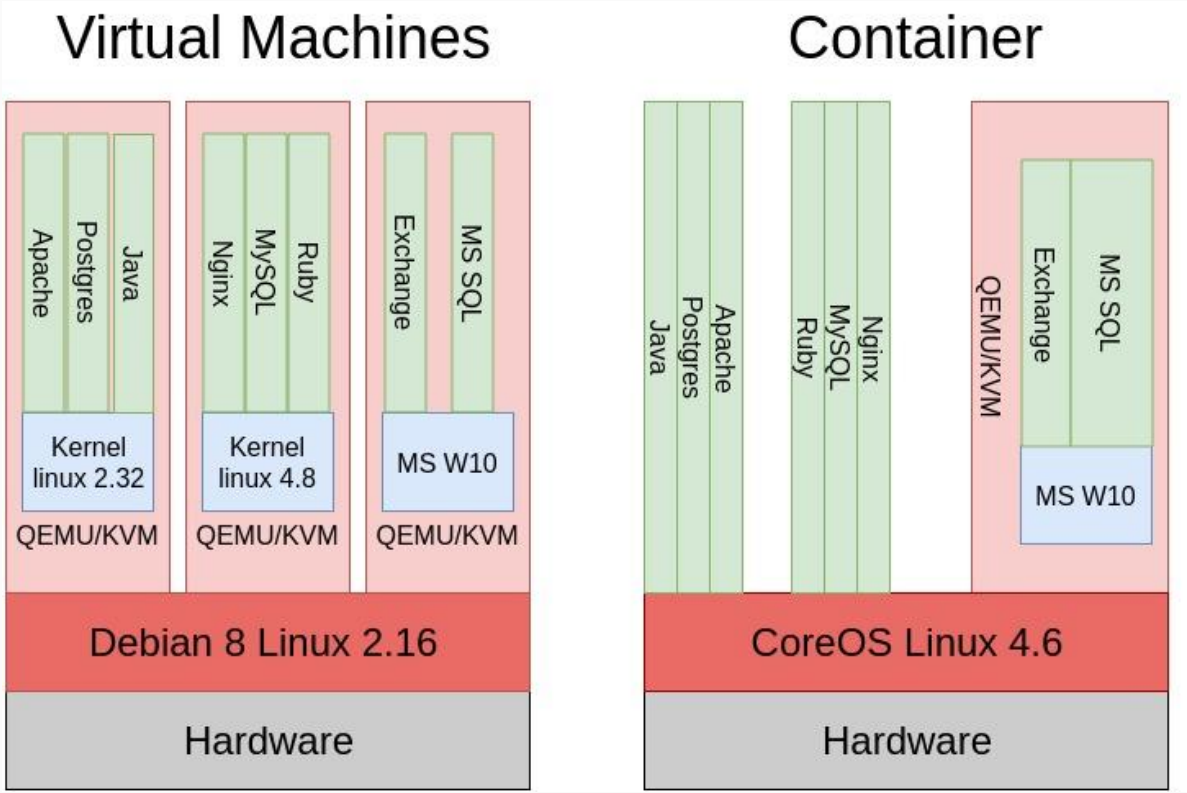
Mandy Waite




TRY TO STAY ABOVE
THE CLOUDS!
QUESTIONS?

AND WHAT ABOUT SECURITY?

LAYOUT


Virtual Machines
Container
Apache
Postgres
Java
Nginx
MySQL
Ruby
Exchange
MS SQL
Kernel
linux 2.32
Kernel
linux 4.8
MS W10
QEMU/KVM
QEMU/KVM
QEMU/KVM
Debian 8 Linux 2.16
Hardware
Apache
Postgres
Java
Nginx
MySQL
Ruby
QEMU/KVM
Exchange
MS SQL
MS W10
CoreOS Linux 4.6
Hardware

YOU CAN HARDEN YOUR CONTAINERS

Intel: Clear Containers

Google: gVisor

https://techcrunch.com/2018/05/02/google-open-sources-gvisor-a-sandboxed-container-runtime/

CONTAINERS OR VIRTUALIZATION

>_ ENDOCODE

Topic	**Container**	**Virtualization**	
Isolation	OS Level, OS namespaces	CPU Level: Ring 0/Ring 3	
foreign CPU	no	yes, with emulation	
foreign kernels, OS	no	yes	kernel is common
emulated devices	no	yes	security
host devices	direct	virtio driver	security
CPU performance	100%	95%	
IO performance	100%	<<100%	
root isolation	yes	yes	USER directive
CPU cache attacks	easy	possible	PoC ?

CONTAINERS vs VMs

Keen on updating your entire Infrastucture?

https://www.heise.de/security/meldung/Spectre-NG-Intel-Prozessoren-von-neuen-hochriskanten-Sicherheitslueck en-betroffen-4039302.html

2013 Side Channel Attacks Predicted

By GAL DISKIN

https://events.ccc.de/congress/2013/Fahrplan/system/attachments/2266/original/Gal_Diskin_-_Virtually_Impossib le_-_30C3__release_version_.pdf

QEMU is CRAP

KVM is fine

https://cloudplatform.googleblog.com/2017/01/7-ways-we-harden-our-KVM-hypervisor-at-Google-Cloud-security-in-plaintext.html