Introduction to Container Security

Are they safe enough?

Aristide Bouix, IT Security Consultant

AWS Security Specialty Certified

June 4th, 2018

devoteam

Innovative technology consulting for business

Table of Contents


1
A Brief
Container
History

A brief container history

From chroot to docker

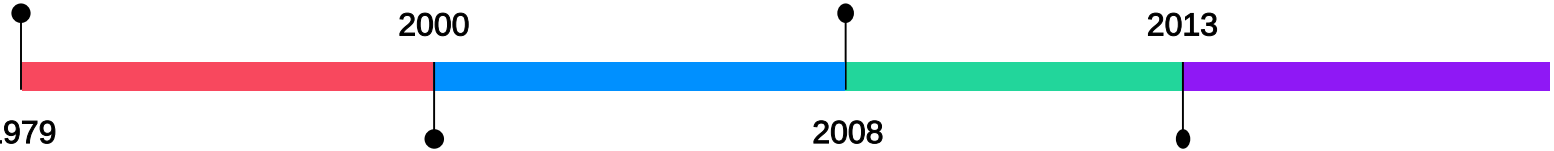
A container is an isolated process packaged with its dependencies

chroot

Firstly defined in Unix V7, it allows to change the root directory of a process and its children to a new location.

LXC

Built on linux, it adopts the container terminology. The use of linux cgroups and namespaces allows a better isolation.



FreeBSD Jails

Flash-forward partitions FreeBSD computers in independant smaller systems called jails, with the ability to configure IP addresses for each jail.

Docker

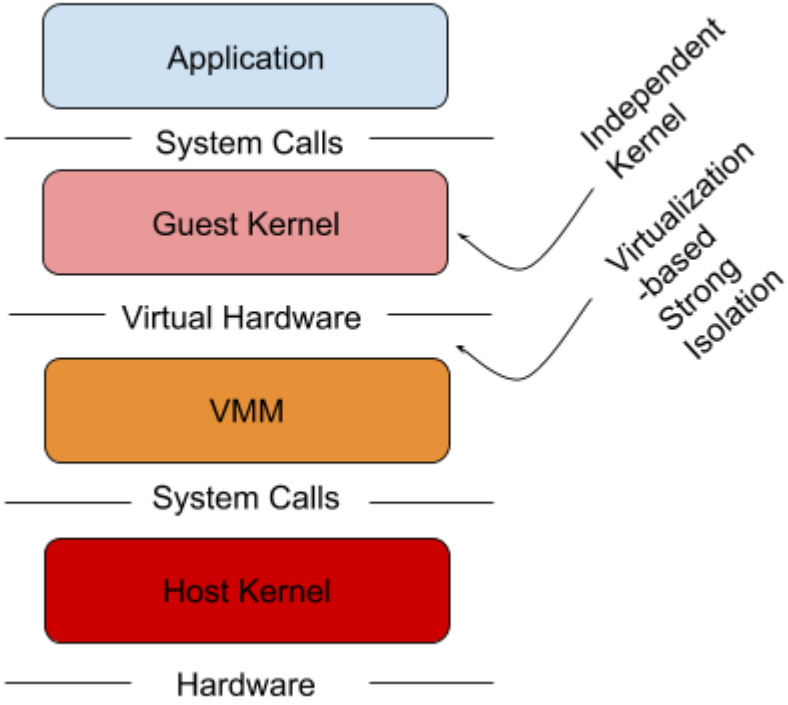
Based on LXC, the inclusion of user namespaces in linux Kernel 3.8 makes the product popularity hit the top. It offers an entire API for container management.

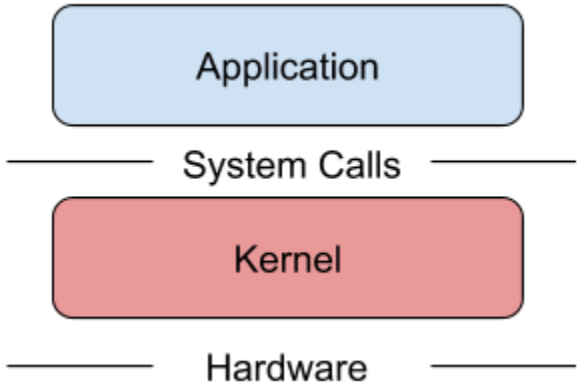
2

Containers vs Virtual Machines

Containers vs Virtual machines

Why choosing Containers?





Smaller: VM ~ Go; Container ~ few Mo

Faster startup: VM ~few mins; Container ~ s

Easier integration: VM configuring network on the host hypervisor and each VM
Containers integration built-in the docker-engine (RAFT)

Containers vs Virtual machines

The 3 rules to enhance isolation

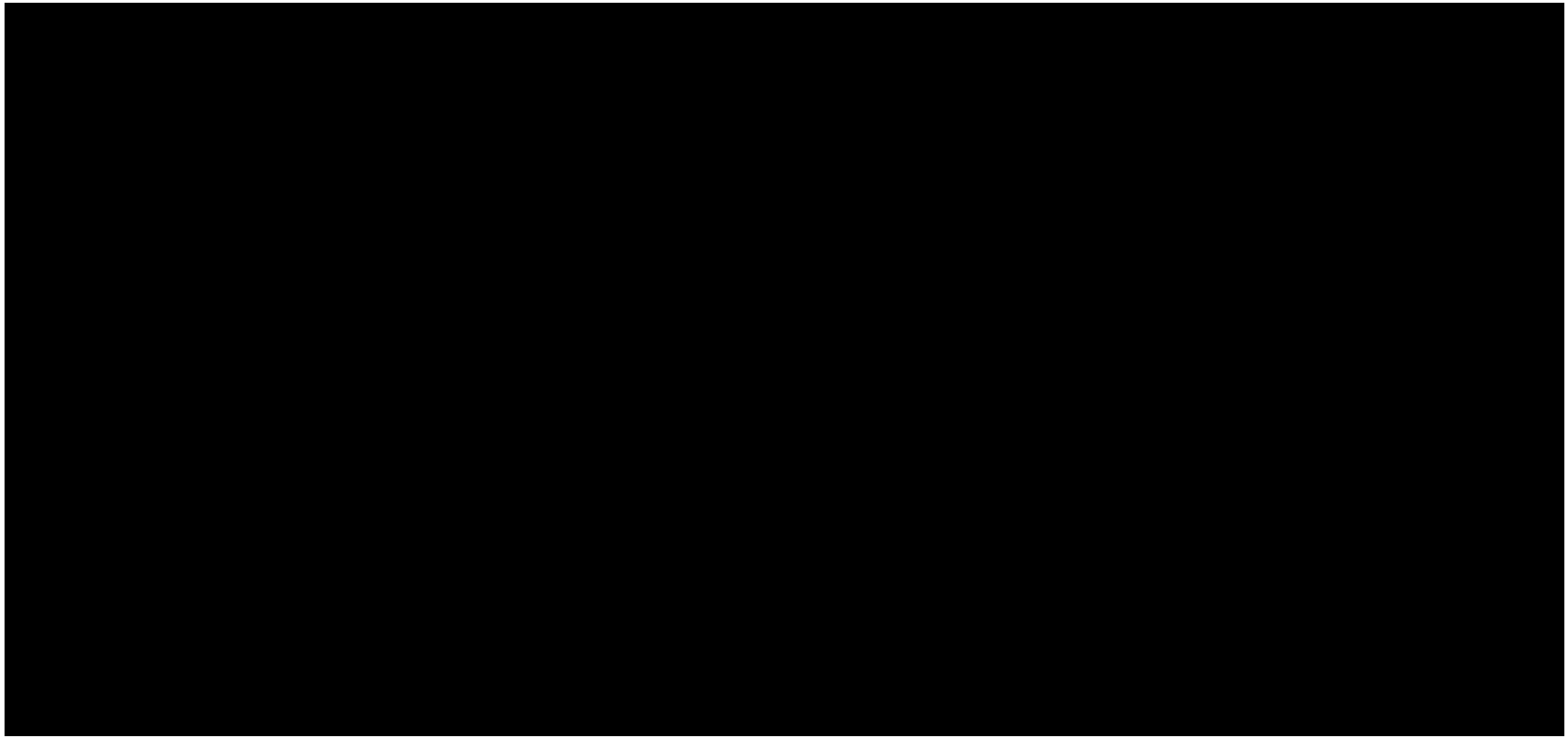
I Careful with share volumes you will be.
Docker use directly the host kernel, so in case of kernel vulnerability restricted permissions won't help

II Control your container ressources, don't let them controlling you.
Use options such as --limit-memory, --limit-cpu, --ulimit …
(more --reserve-cpu, --reserve-memory for sensitive containers as db)

III Limit your container permissions before they limit you.
By default you're root inside of a container, as such your container make syscall as root on the host kernel

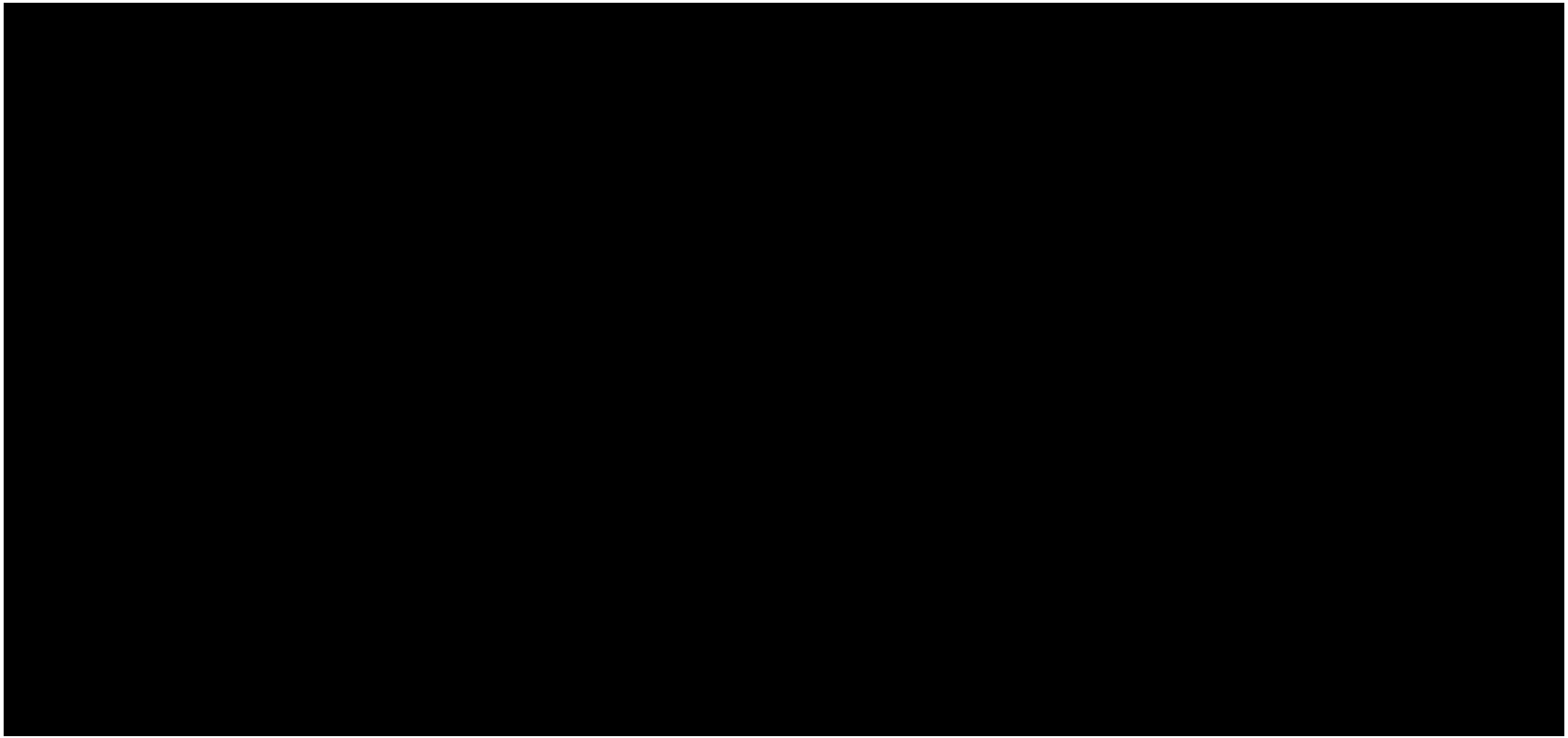
Containers vs Virtual machines

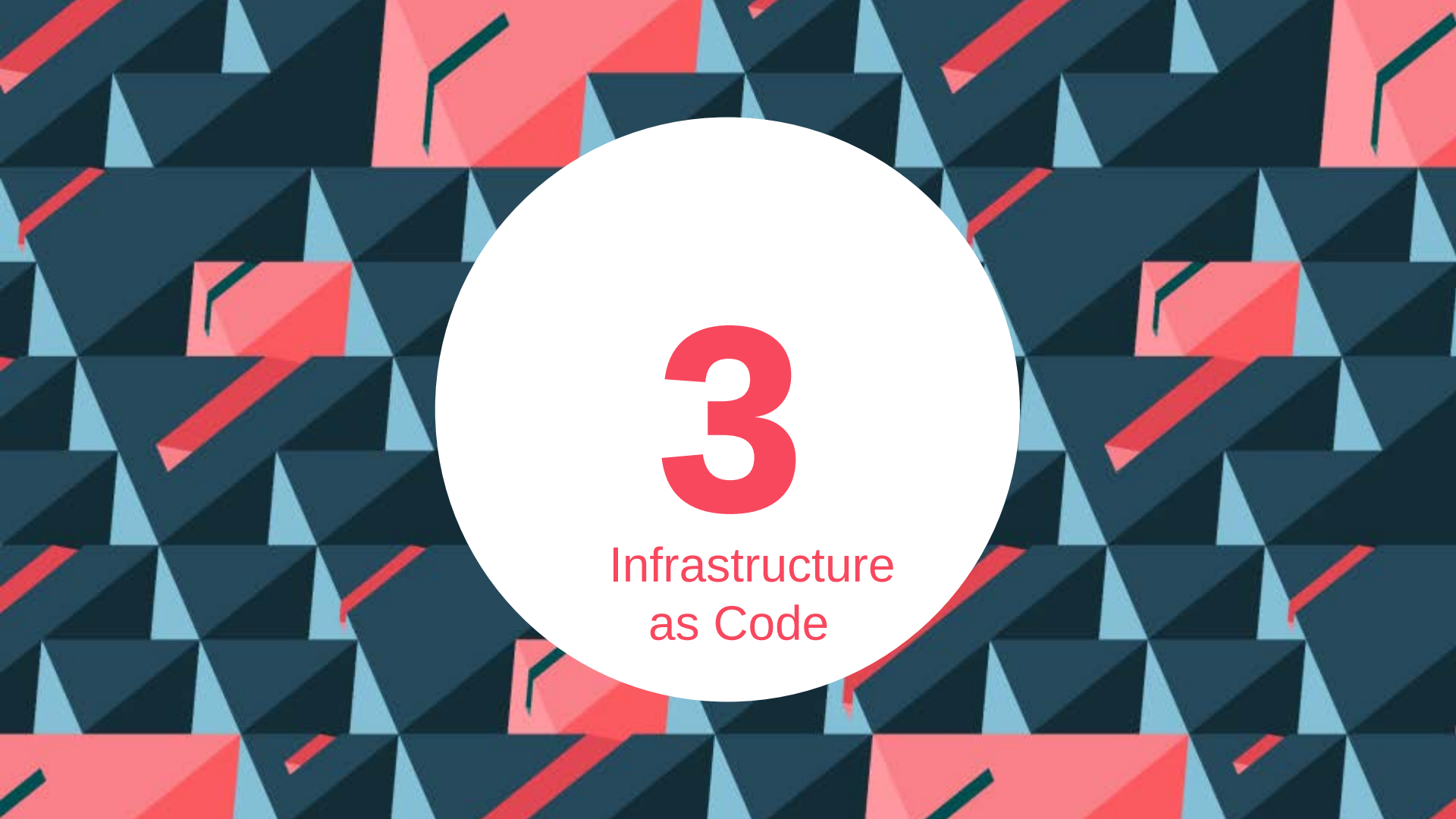
[Demo]: Container restriction 1



Containers vs Virtual machines

[Demo]: Container restriction 2

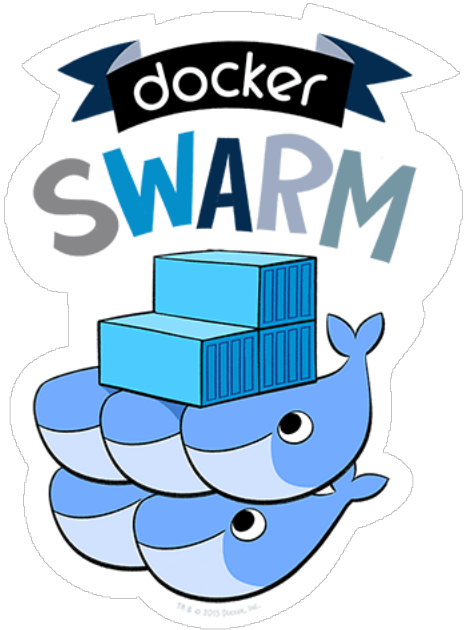



3
Infrastructure
as Code

Infrastructure as Code

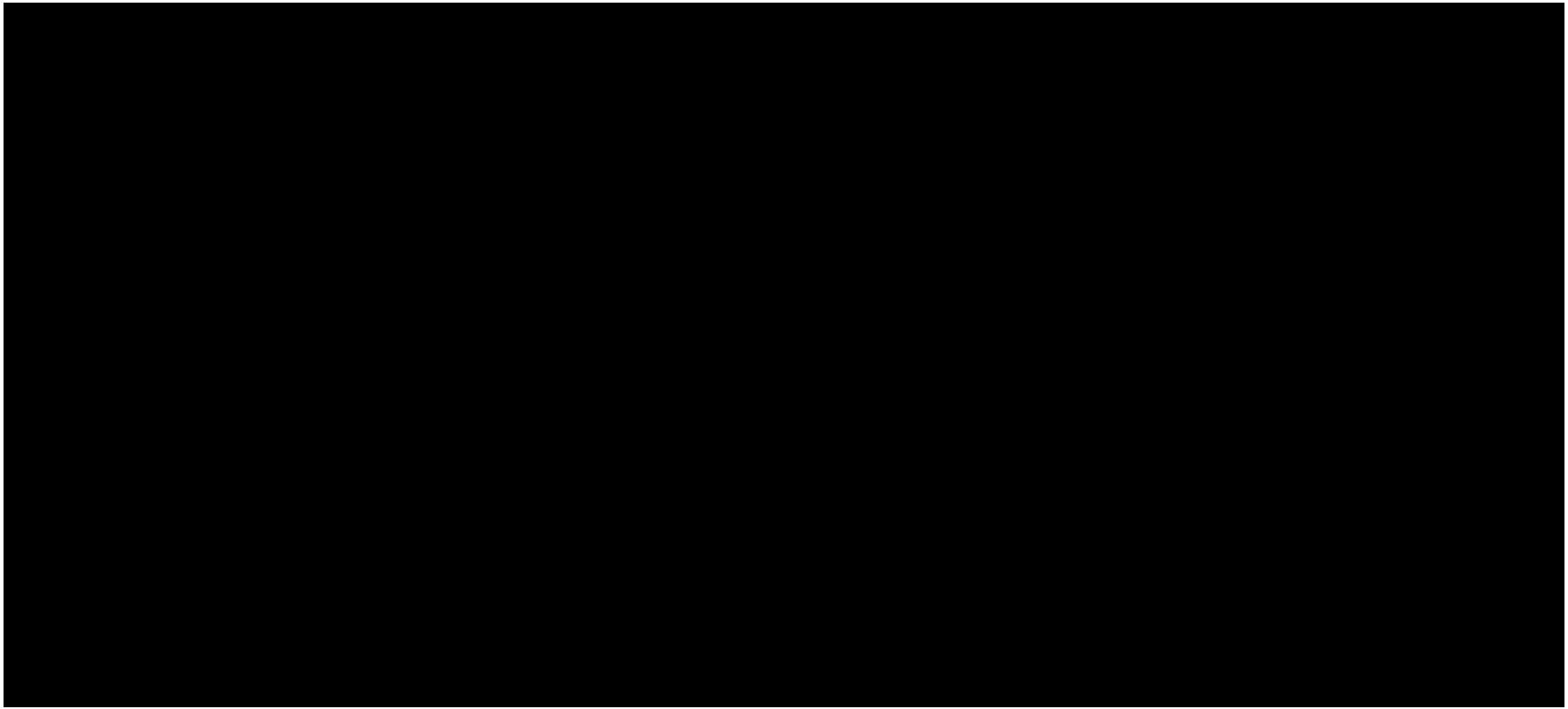
Orchestrators

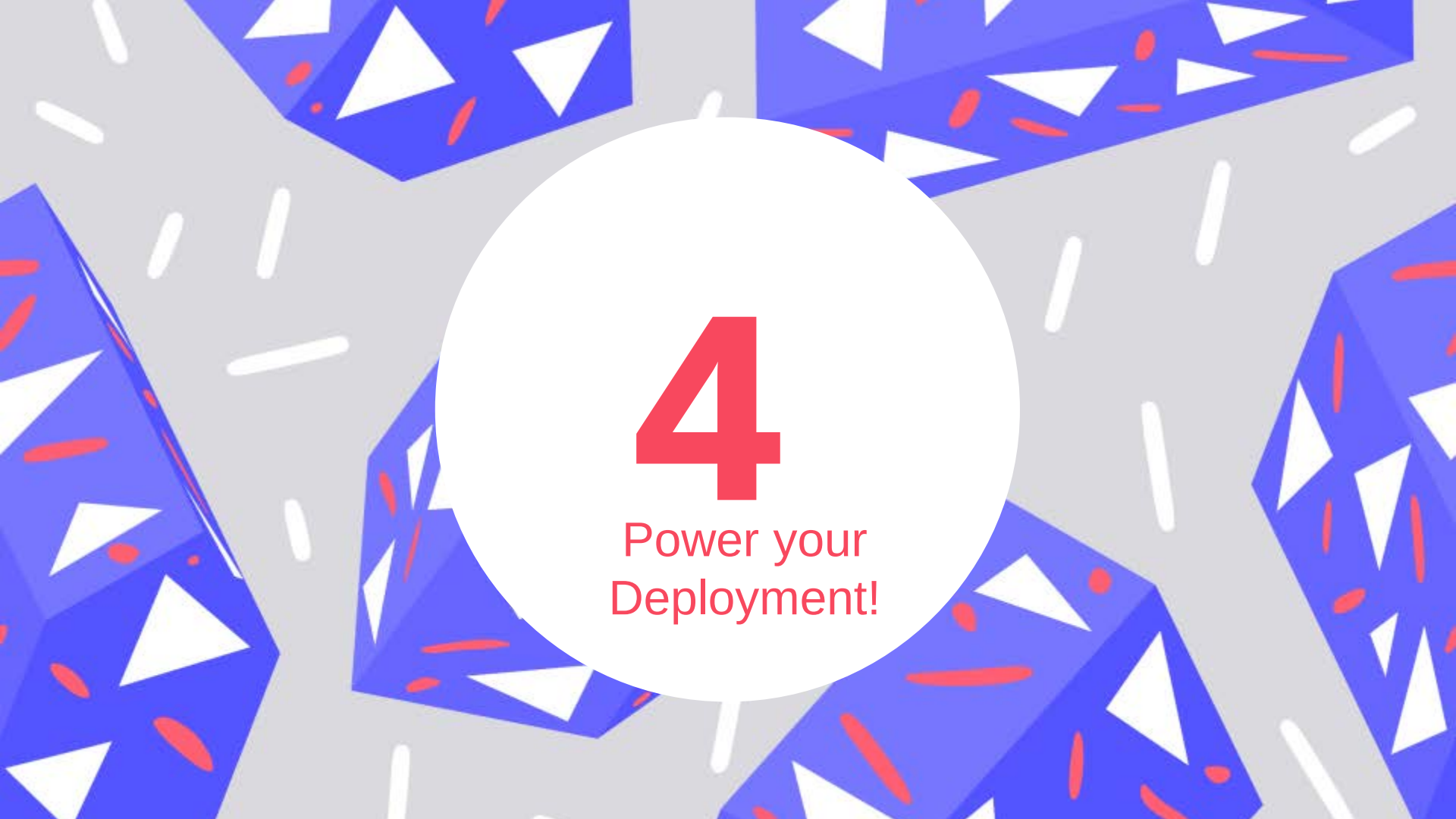
kubernetes



Infrastructure as Code

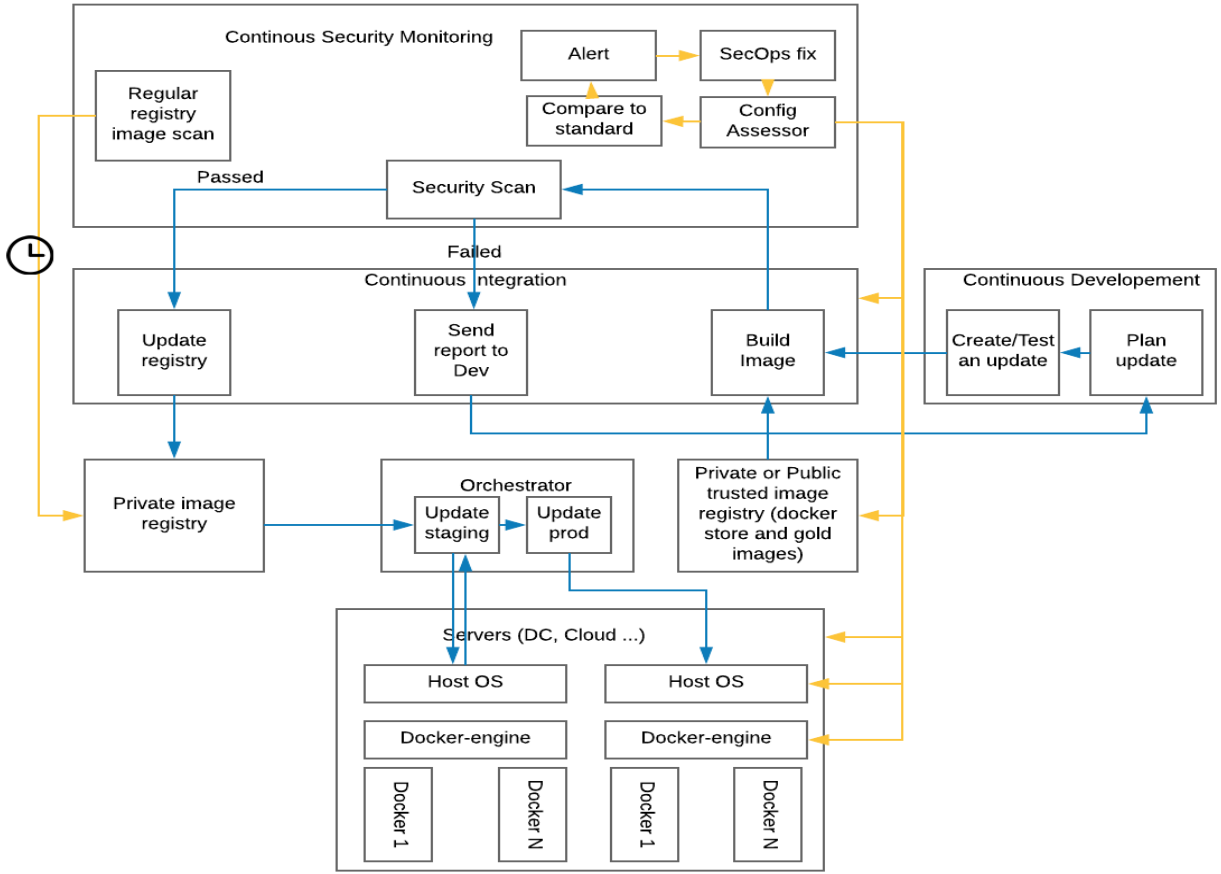
[Demo]: Patch and update a service




4
Power your
Deployment!

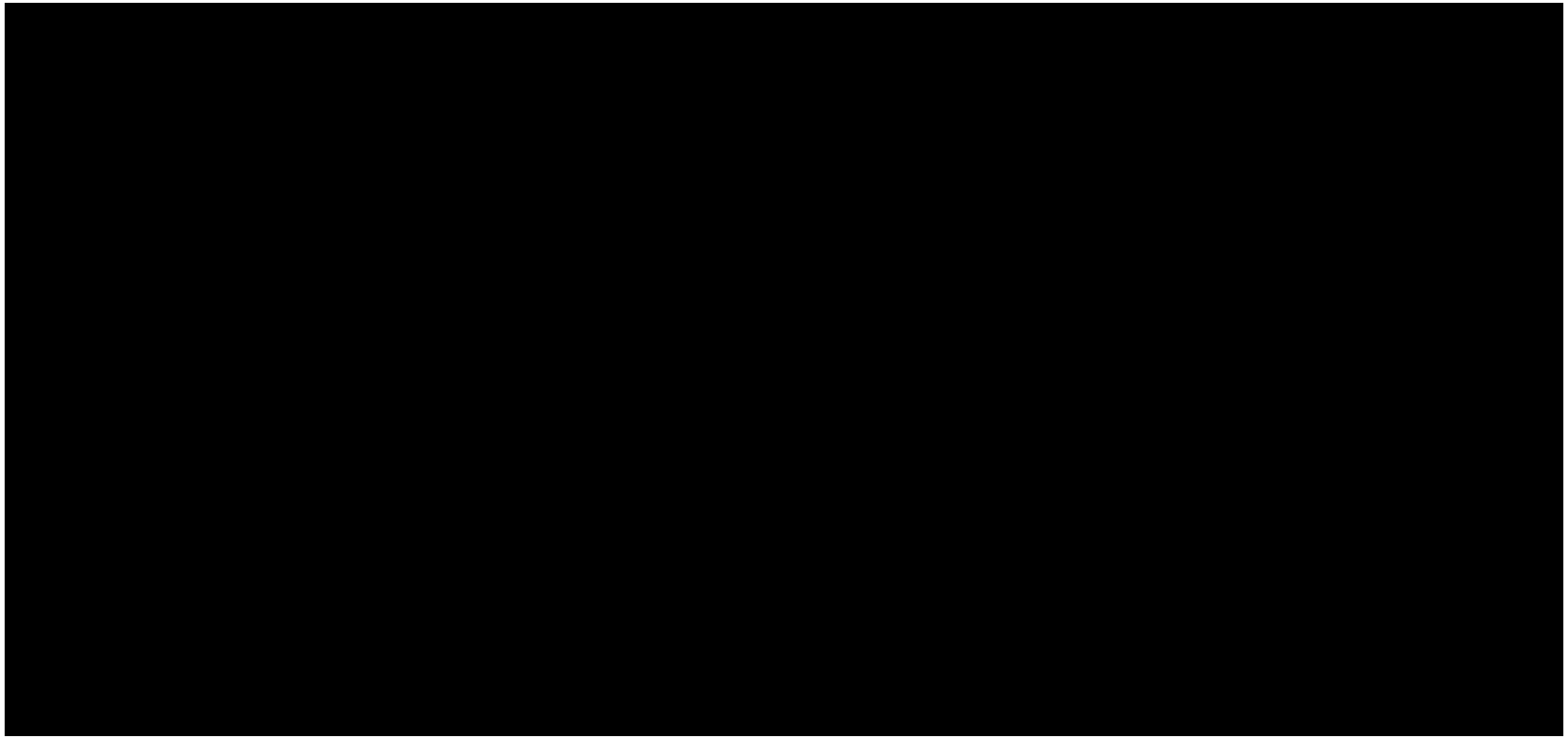
Power your Deployment!

The Container supply chain: build, distribute, run, monitor



Power your Deployment!

[Demo]: setup secure registry with TLS and upload an image



References:

Blogs:

- https://blog.aquasec.com/a -brief-history-of-containers-from-1970s-chroot-to-docker-2016
- https://cloudplatform.googleblog.com/2018/03/exploring-container-security-an-overview.html
- https://www.tripwire.com/state-of-security/featured/could-containers-save-the-day-ways-to-secure-docker/
- https://blog.aquasec.com/dirty-cow-vulnerability-impact-on-containers
- http://success.docker.com/article/secure-supply-chain

Whitepapers:

- Securing the Entire Container Stack, Lifecycle and Pipeline -- Tripwire

Aristide Bouix

Information Systems Security Consultant
Devoteam Risk & Security
+33 6 60 49 23 08

aristide.bouix@devoteam.com

devoteam

Questions?