Maya Kaczorowski & Dan Lorenc
Google Cloud

BSidesSF
March 4, 2019

Maya Kaczorowski

Security PM, Google Cloud

@MayaKaczorowski

Dan Lorenc

Software Engineer, Container Tools, Google Cloud

Google Cloud

Agenda

1. Containers, Kubernetes, and a different model for security
2. Traditional software supply chain and patch management
3. Ideal software supply chain and best practices in image maintenance, patching, and validation
4. Demo: patching 0days
5. Using live migration in practice at Google

Google Cloud

Containers, Kubernetes, and a different model for security

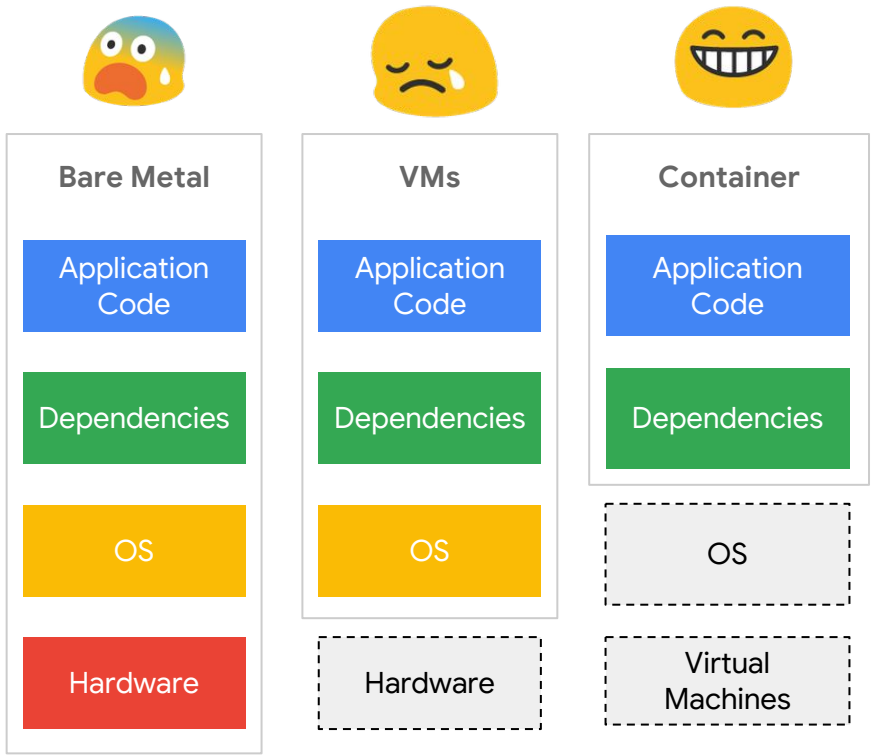
01

Google Cloud

First, what's a container?

Containers are all the rage these days, getting us close to the dream of write once, run anywhere.

They package and isolate applications and dependencies.

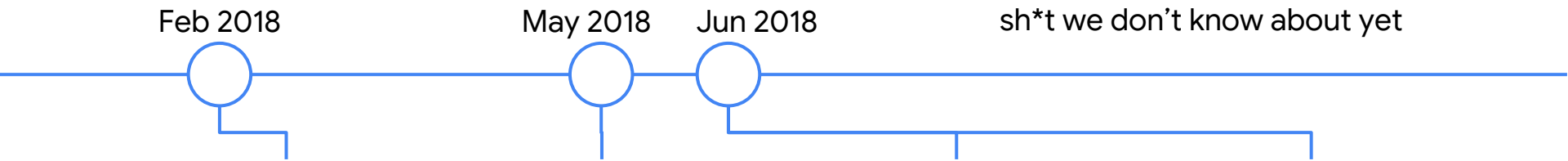


Google Cloud

Enter Kubernetes

- A portable, open-source, **container-centric** management platform
- Built-in primitives for **deployments, rolling upgrades, scaling, monitoring, and more**
- Inspired by **Google's internal systems**
- Get true **workload portability** and increased **infrastructure efficiency**

Containers are actually being attacked



Tesla

Unsecured Kubernetes dashboard with cloud account credentials

Used to mine cryptocurrency

Shopify

Researcher could access and replay kubelet credentials

Not exploited

Weight Watchers

Unsecured Kubernetes dashboard with sensitive data, including credentials

Not exploited

Docker Hub

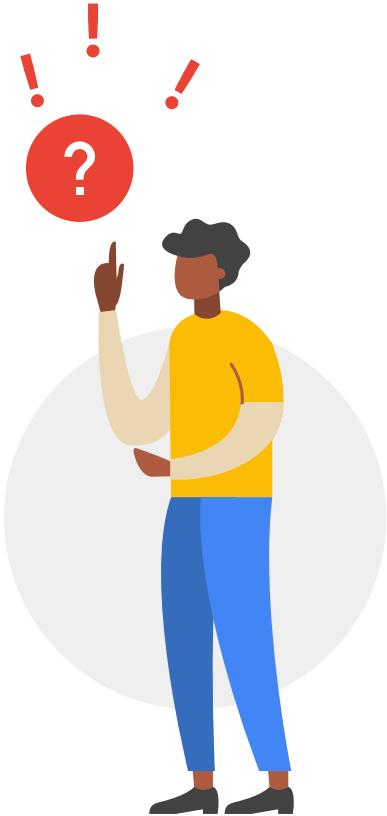
Public images with embedded cryptocurrency mining malware

Used to mine cryptocurrency

Google Cloud

Security people like to complain about containers and Kubernetes

- What's a koober net ease
- I can't use my IDS, firewall, ...
- Containers don't contain
- I am stuck with it, help me



How is securing a container different?

Surface of Attack

Minimalist host OS and limits the surface of an attack.

Resource Isolation

Host resources are **separated using namespaces and cgroups.**

Permissions

Access controls are for app privileges and shared resources.

Lifecycle

Containers have a **shorter, better defined lifecycle.**

Traditional software supply chain and patch management

Google Cloud

Traditional software supply chain

Traditional patch management

01

Get patch

From the distributor, some random mailing list, a vendor. Not always sent to the security team.

02

Take down server n=1 and apply patch

Test the patch in prod! Take some unimportant workload down to make sure nothing goes too bad.

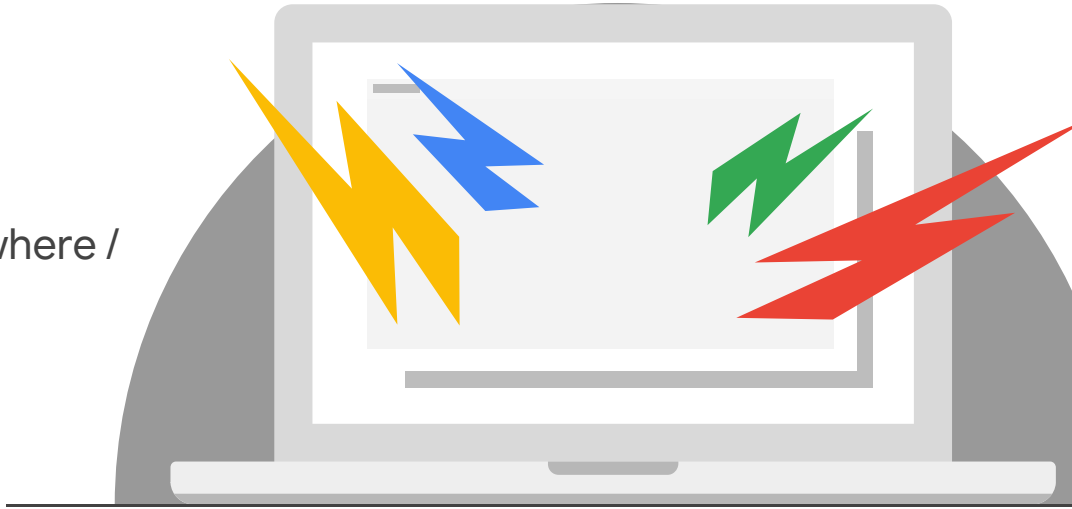
03

Repeat for n servers, where n is unknown

It worked! Now do it again, for everything you think is affected. Miss a bunch of it.

Problems with traditional patch management

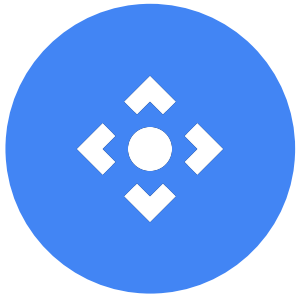
- Spreadsheet-driven management
- Down time
- 0days are scary
- Unclear what's running in your infrastructure / what's running where / if you even need a patch



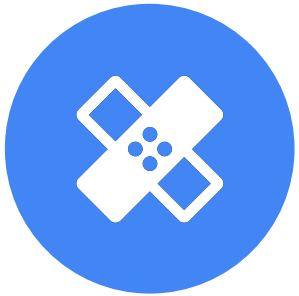
Ideal software supply chain and best practices

Google Cloud

Running containers allows you to adopt a fundamentally different security model



Containers give you a **software supply chain**



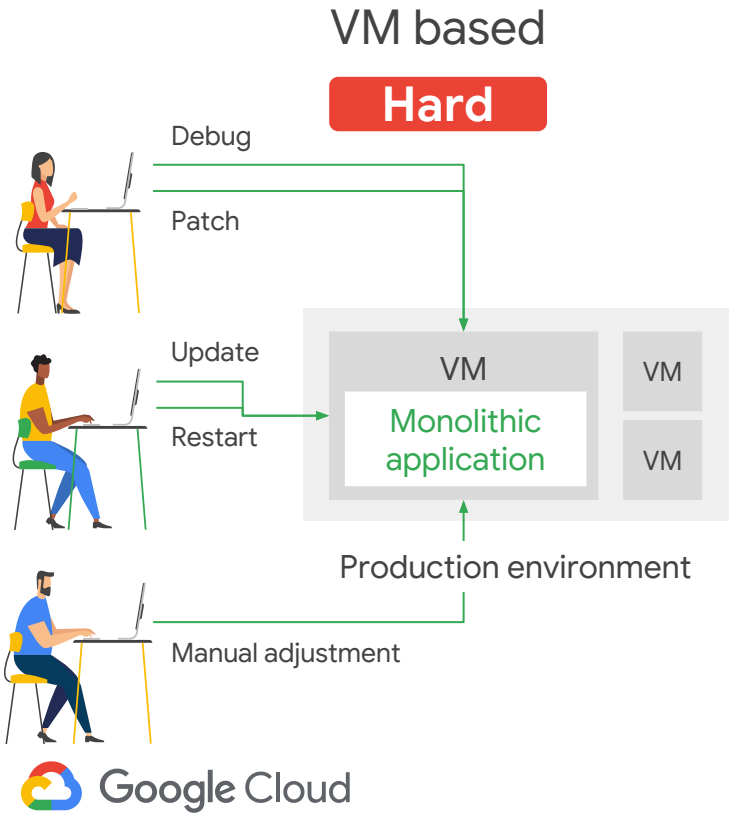
Containers let you **patch continuously**, automatically

Containers mean you can actually **tell if you're affected** by a new vulnerability

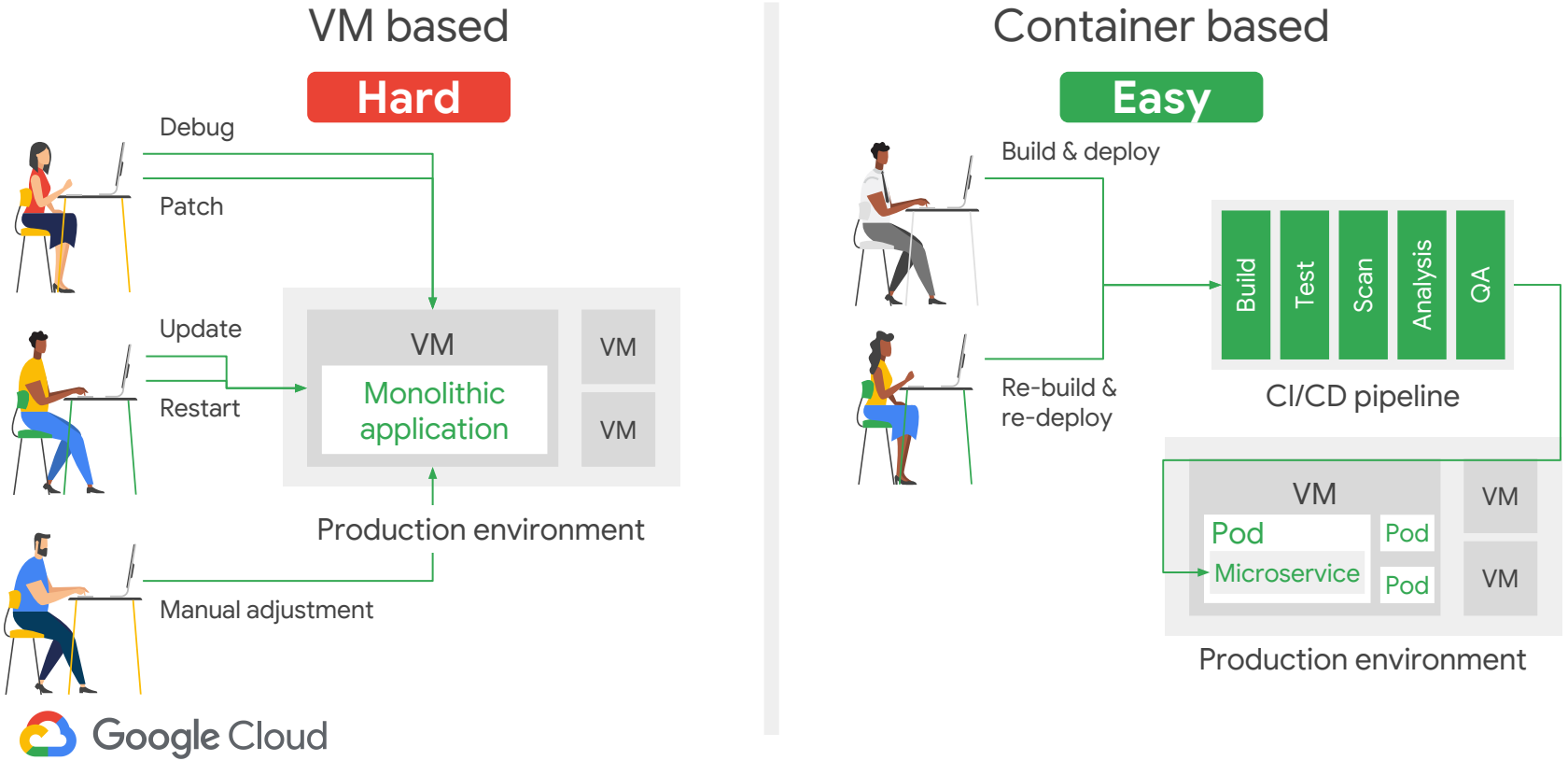
Google Cloud

Containers give you a software supply chain

What's different about supply chains with containers

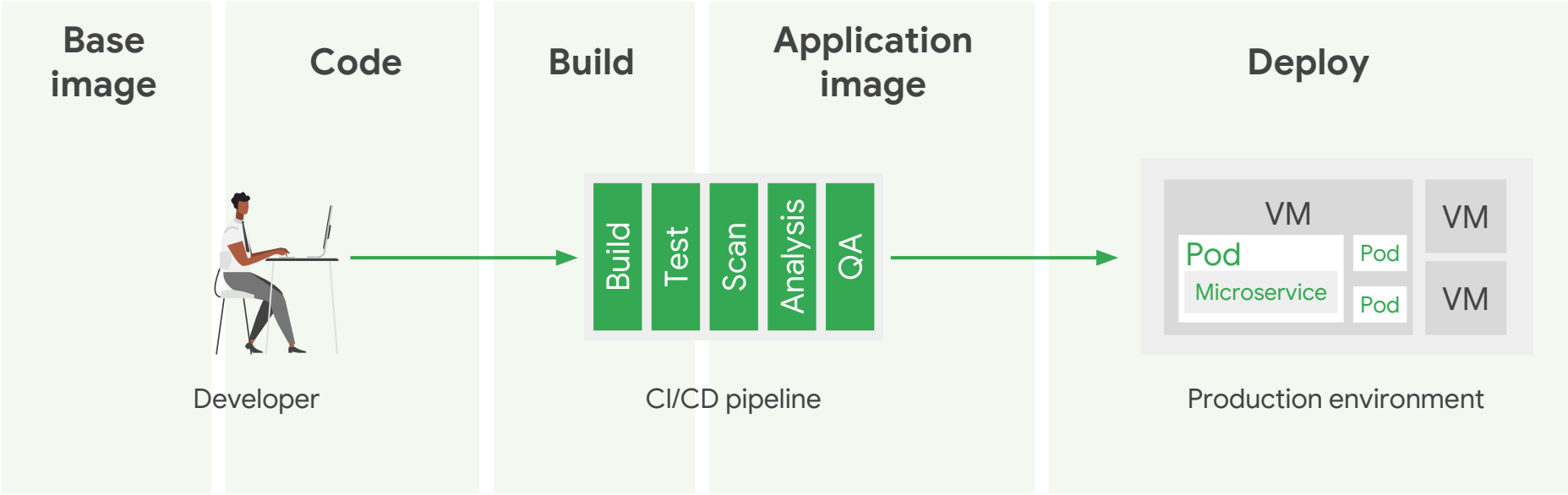


What's different about supply chains with containers



Google Cloud

Containers let you enforce a software supply chain



Google Cloud


Containers let you
patch continuously,
automatically
Google Cloud

Constantly patch your registry… and roll out as normal

01

Patch the image in your registry

Figure out what's affected, and apply the patch everywhere you need it.

02

Test, validate, and roll out

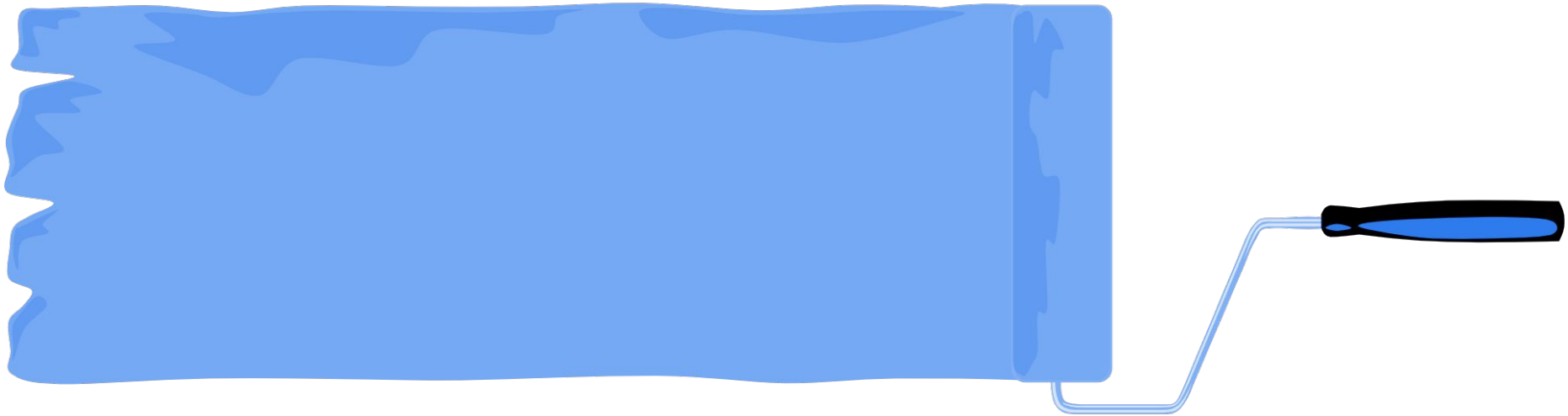
Roll out the patch like you would any other infrastructure change, going incrementally.

03

Load balance traffic over

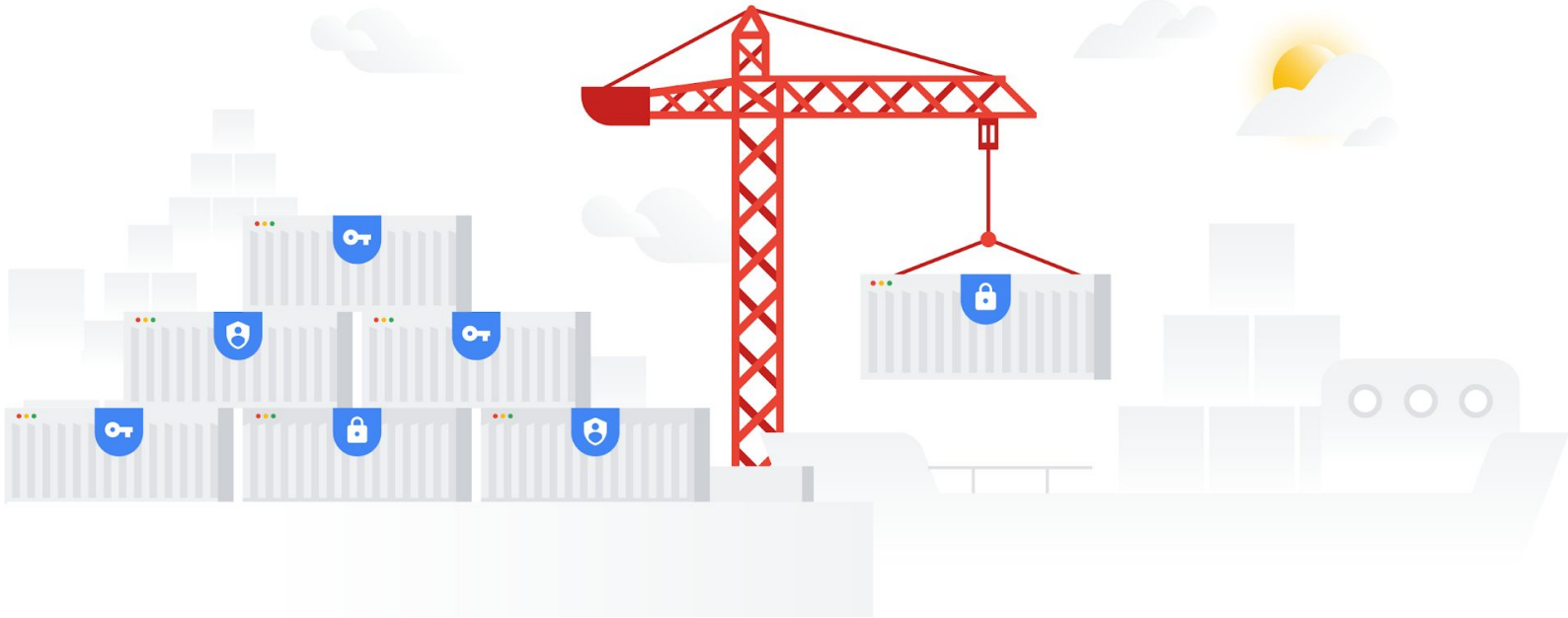
When testing is successful, move traffic over to the new, patched workload, with no downtime.

Rollout patches the same way other changes rollout



Google Cloud

Containers enable passive patching



Google Cloud

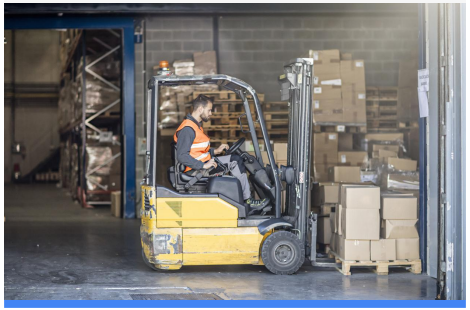
Vulnerability mitigation strategies

Update packages

apt-get update & upgrade gets you pretty far. Do this daily.

Remove packages

Do you really need 6.022×10^{23} debian packages installed on your production image?



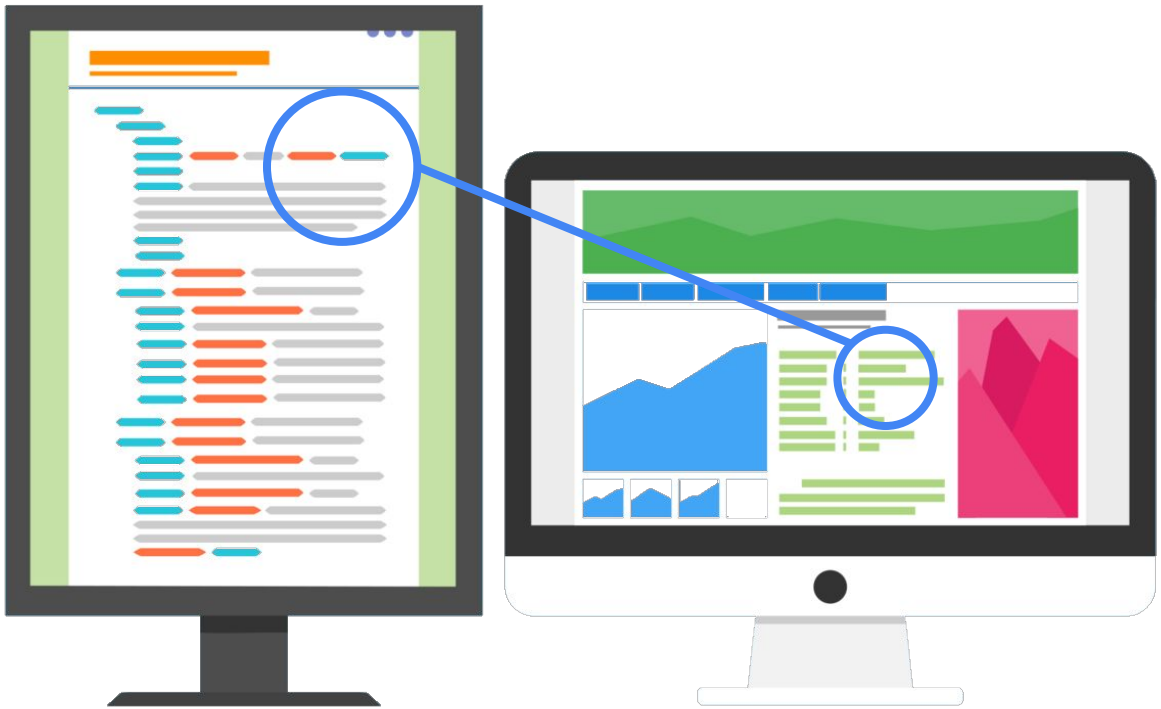
New distro

In many cases, you can get away with a smaller distro like Alpine or Debian Slim.

Containers mean you can actually tell if you're affected by a new vulnerability

Google Cloud

Check your registry and compare to what you deployed



Google Cloud

Figure out what's in prod

Find all the containers in prod

kubectl get pods
resolve everything to a digest

Find out what is in those containers

Package manifests, application dependencies

Find out what vulnz are in those packages

Cross reference BOM with CVE databases

Google Cloud

Demo:
Patching 0days

04

Google Cloud

Time to roll up your sleeves

Using live migration at Google

05

Google Cloud

Blue/green deployments

Blue
Existing workload

Green
New (patched) workload

Google Cloud

Borg and live migration

Large-scale cluster management at Google with Borg

Abhishek Verma† Luis Pedrosa‡ Madhukar Korupolu
David Oppenheimer Eric Tune John Wilkes
Google Inc.

Abstract

Google's Borg system is a cluster manager that runs hundreds of thousands of jobs, from many thousands of different applications, across a number of clusters each with up to tens of thousands of machines.

It achieves high utilization by combining admission control, efficient task-packing, over-commitment, and machine sharing with process-level performance isolation. It supports high-availability applications with runtime features that minimize fault-recovery time, and scheduling policies that reduce the probability of correlated failures. Borg simplifies life for its users by offering a declarative job specification language, name service integration, real-time job monitoring, and tools to analyze and simulate system behavior.

We present a summary of the Borg system architecture and features, important design decisions, a quantitative analysis of some of its policy decisions, and a qualitative examination of lessons learned from a decade of operational

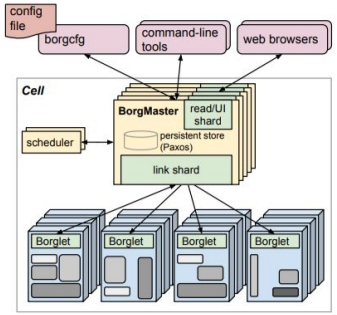


Figure 1: The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*

https://cloud.google.com/compute/docs/instances/live-migration

Google Cloud Why Google Solutions Products Pricing Getting started Docs Support

Compute Engine › Documentation

Live Migration

SEND FEEDBACK

Internal: Count: 20, Average: 4.7

Contents
How does the live migration proces
Live migration an
Live migration an SSDs
Live migration fo preemptible insta
What's next

Compute Engine offers live migration to keep your virtual machine instances running even when a host system event occurs, such as a software or hardware update. Compute Engine live migrates your running instances to another host in the same zone rather than requiring your VMs to be rebooted. This allows Google to perform maintenance that is integral to keeping infrastructure protected and reliable without interrupting any of your VMs.

Live migration keeps your instances running during:

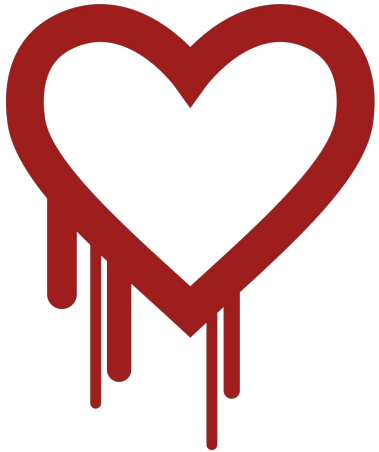
- Regular infrastructure maintenance and upgrades.
- Network and power grid maintenance in the data centers.
- Failed hardware such as memory, CPU, network interface cards, disks, power, and so on. This is done on a best-effort basis; if a hardware fails completely or otherwise prevents live migration, the VM crashes and restarts automatically and a `hostError` is logged.
- Host OS and BIOS upgrades.
- Security-related updates, with the need to respond quickly.
- System configuration changes, including changing the size of the host root partition, for storage of the host image and packages.

Live migration does not change any attributes or properties of the VM itself. The live migration process just transfers a running VM from one host machine to another host machine within the same zone. All VM properties and attributes remain unchanged, including internal and external IP addresses, instance metadata, block storage data and volumes, OS and application state, network settings, network connections, and so on.

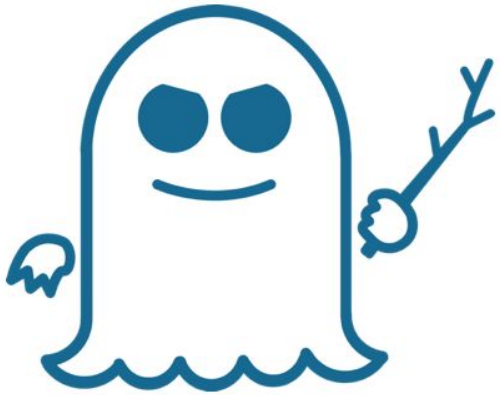
How does the live migration process work?

Patched under embargo

Heartbleed
2014

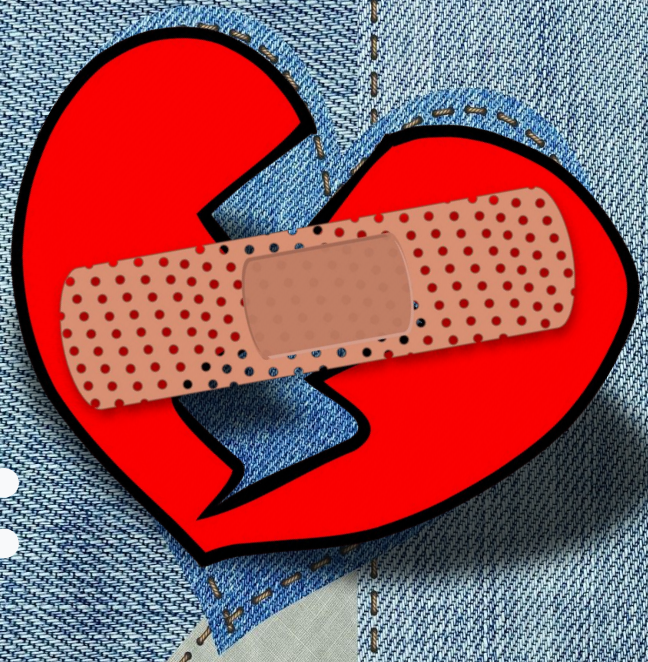


Spectre
2018



Meltdown
2018

Containers: patched



Learn more

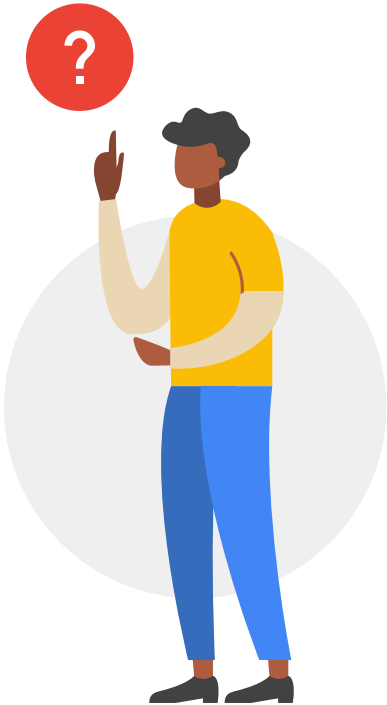
Blog post: goo.gl/Ew6hYa

cloud.google.com/containers/security

Google Cloud

Questions & answers

sli.do
#BSidesSF2019



Google Cloud

Join Google Cloud at

2019 Google Cloud Security Talks

Wednesday, March 6, 2019 - Thursday, March 7, 2019
8:30 AM - 5:15 PM

Bespoke @ Westfield San Francisco Centre
845 Market Street, LEVEL 4, San Francisco, CA

Topics include...

- Containers & Kubernetes
- Implementing BeyondCorp
- Trends on the accounts, passwords & malware threat landscapes

More information at **g.co/cloud/securitytalks2019**

Register **onsite** at Bespoke