CLOUD NATIVE
COMPUTING
FOUNDATION

Kubernetes Security Zooming In, Zooming Out

A comprehensive Container Security Strategy

Kavya Pearlman
Global Cybersecurity Strategist - Wallarm
@KavyaPearlman | @Wallarm

Rob Richardson
Technical Evangelist - MemSQL
@Rob_Rich | @MemSQL

Introducing Kavya...

Kavya Pearlman

- Well known as the “Cyber Guardian”
- Cybersecurity Strategist at Wallarm
- An Award-winning Cybersecurity Professional
- Founder and CEO of XR Safety Initiative
- Former Information Security Director Linden Lab
- Former Facebook Third Party Security Risk Advisor

Personal interests

Travel, Gaming, Virtual Worlds

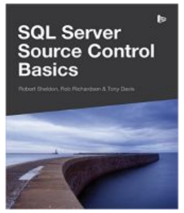
Introducing Rob...

Rob Richardson

- Tech Evangelist for MemSQL
- Microsoft MVP
- Leads the Southeast Valley .NET User Group
- AZGiveCamp Organizer

Personal interests

Travel, Coding, and Teaching


SQL Server Source Control Basics

Robert Sheldon, Rob Richardson & Tony Davis

Agenda

Let's Talk About Kubernetes!

- Overview of Containers
- Monolithic vs Microservices
- What is Kubernetes and its Benefits
- Securing K8 - Zooming in

    Essentials to build a secure Kubernetes environment
- Securing K8 - Zooming Out

    Do’s and Don’ts for Containerized Environments
- Conclusion

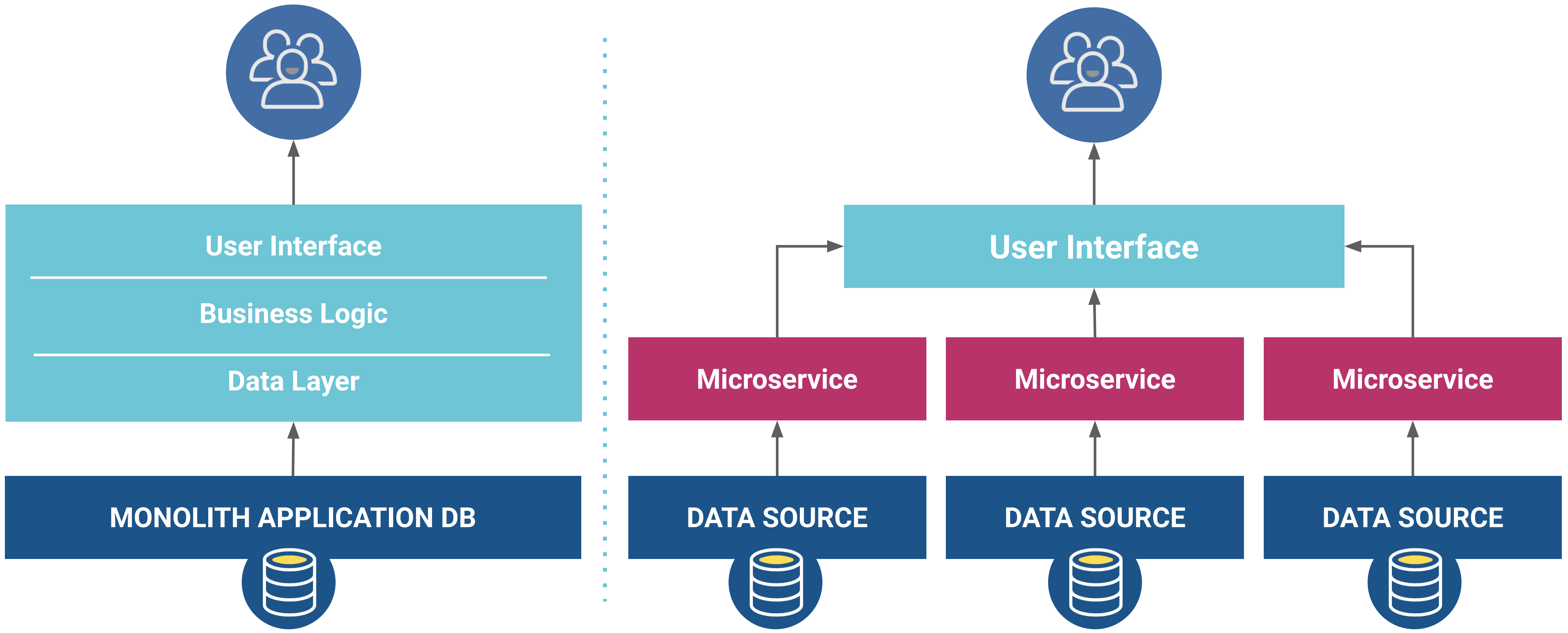
Kubernetes - Getting started

KUBERNETES NEEDS NEW SECURITY MINDSET

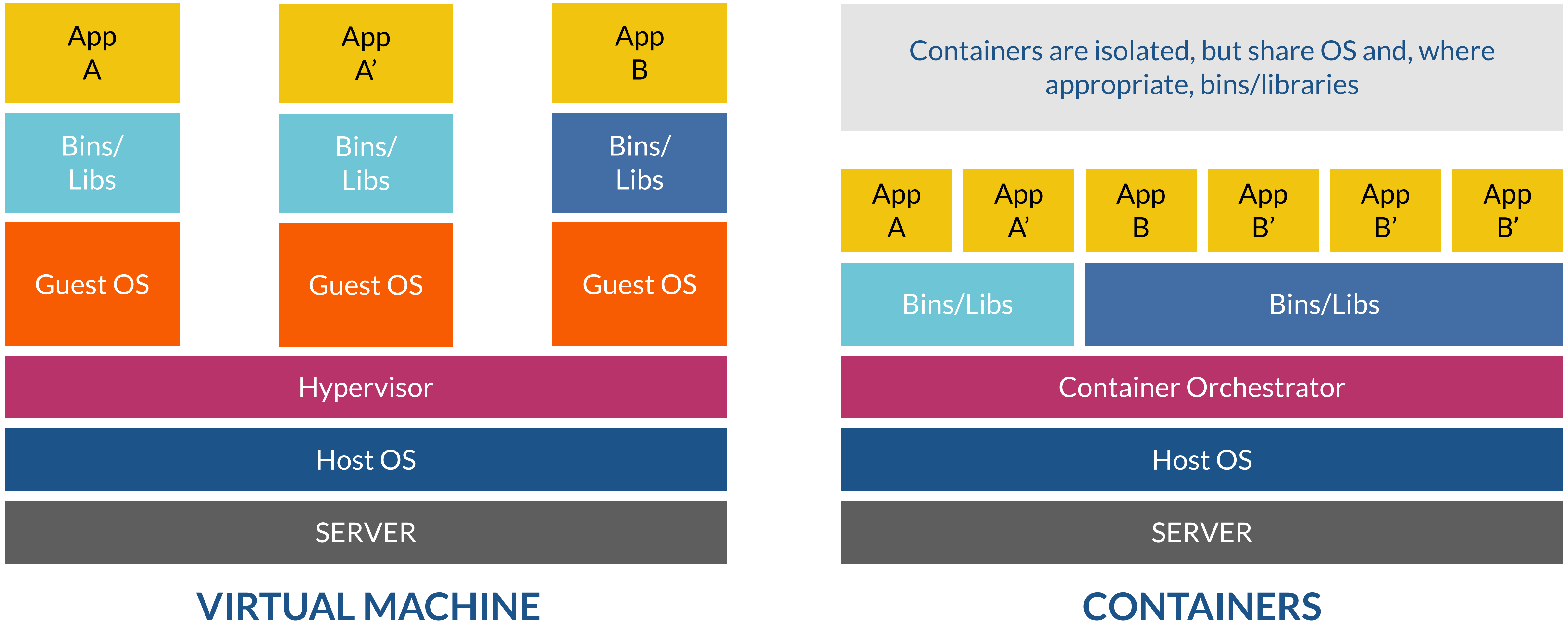
Cloud-native applications and infrastructure create several new challenges for all of us security professionals. We need to establish new security programs, have a new mindset and adopt advanced new tools that are focused primarily on securing cloud-native technologies.”

- Kavya Pearlman

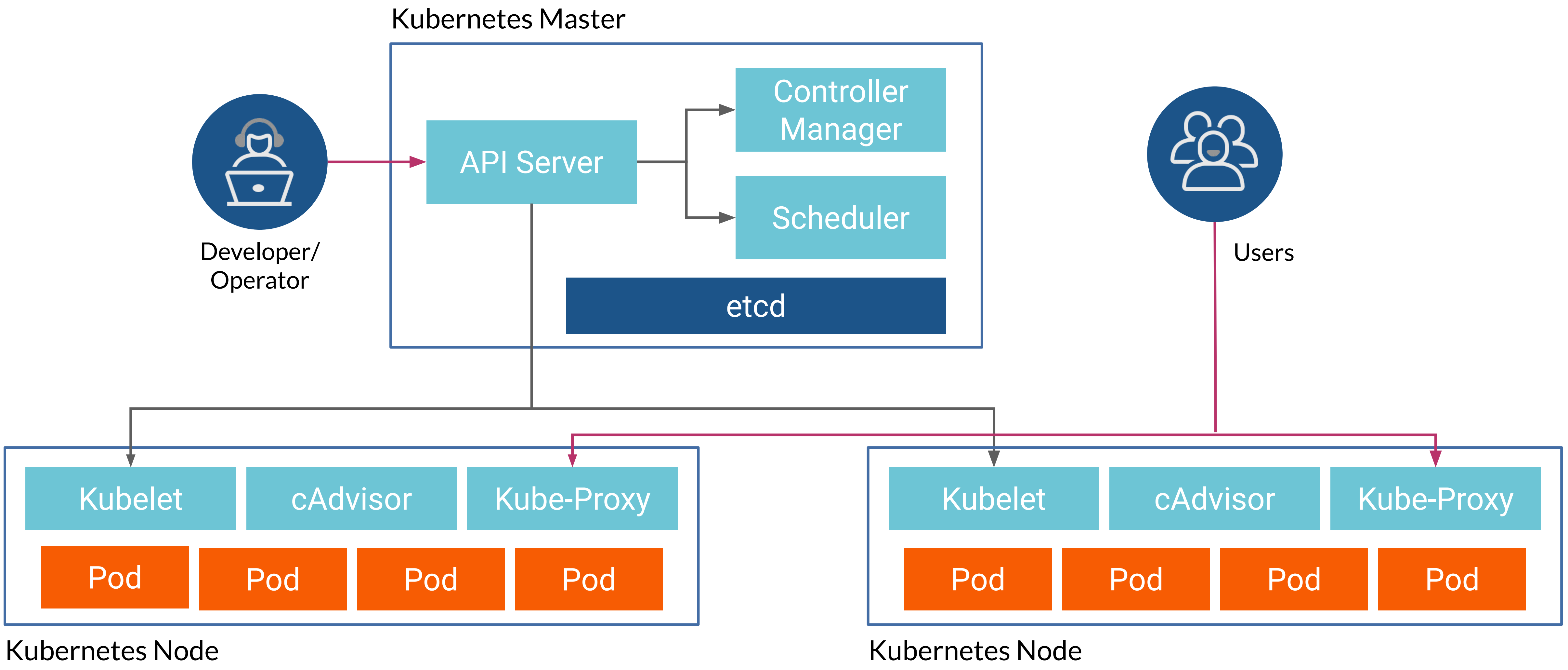
Monolith vs. Microservices



Containers vs. VMs



What is Kubernetes?



Benefits of using Kubernetes

Bring new products to market faster

Avoid vendor lock-in

Enjoy peace of mind that your applications are always on

Kubernetes self-heals

Kubernetes auto-scales

Benefits of using Kubernetes

It’s the de facto standard for running cloud-native applications at scale

Free community support or paid professional services

CLOUD NATIVE COMPUTING FOUNDATION

Kubernetes - Zooming In

The Essentials for Building a Secure Kubernetes Environment

Shopify Breach

Caused by lack of
K8 security Essentials

Exploited Weakness

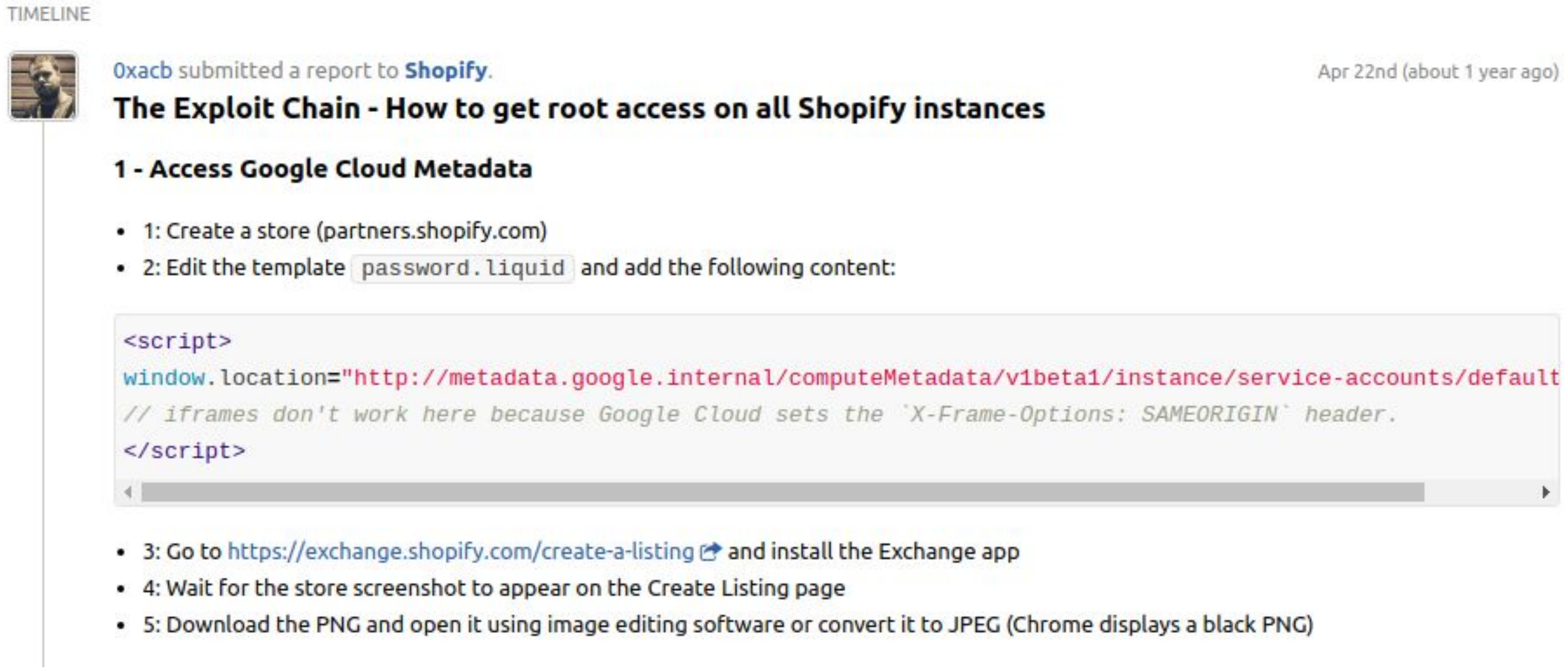
API configuration flaw

Type of attack

SSRF Attack whereby metadata used to steal API keys and credential packets

Effect

Thousands of stores and store-clients information was exposed



TIMELINE

0xacb submitted a report to **Shopify**. Apr 22nd (about 1 year ago)

The Exploit Chain - How to get root access on all Shopify instances

1 - Access Google Cloud Metadata

- 1: Create a store (partners.shopify.com)
- 2: Edit the template `password.liquid` and add the following content:

```
<script>
window.location="http://metadata.google.internal/computeMetadata/v1beta1/instance/service-accounts/default
// iframes don't work here because Google Cloud sets the `X-Frame-Options: SAMEORIGIN` header.
</script>
```

- 3: Go to https://exchange.shopify.com/create-a-listing and install the Exchange app
- 4: Wait for the store screenshot to appear on the Create Listing page
- 5: Download the PNG and open it using image editing software or convert it to JPEG (Chrome displays a black PNG)

Caused by lack of
K8 security Essentials

Exploited Weakness:

Kubernetes instance and an insecure administrative console

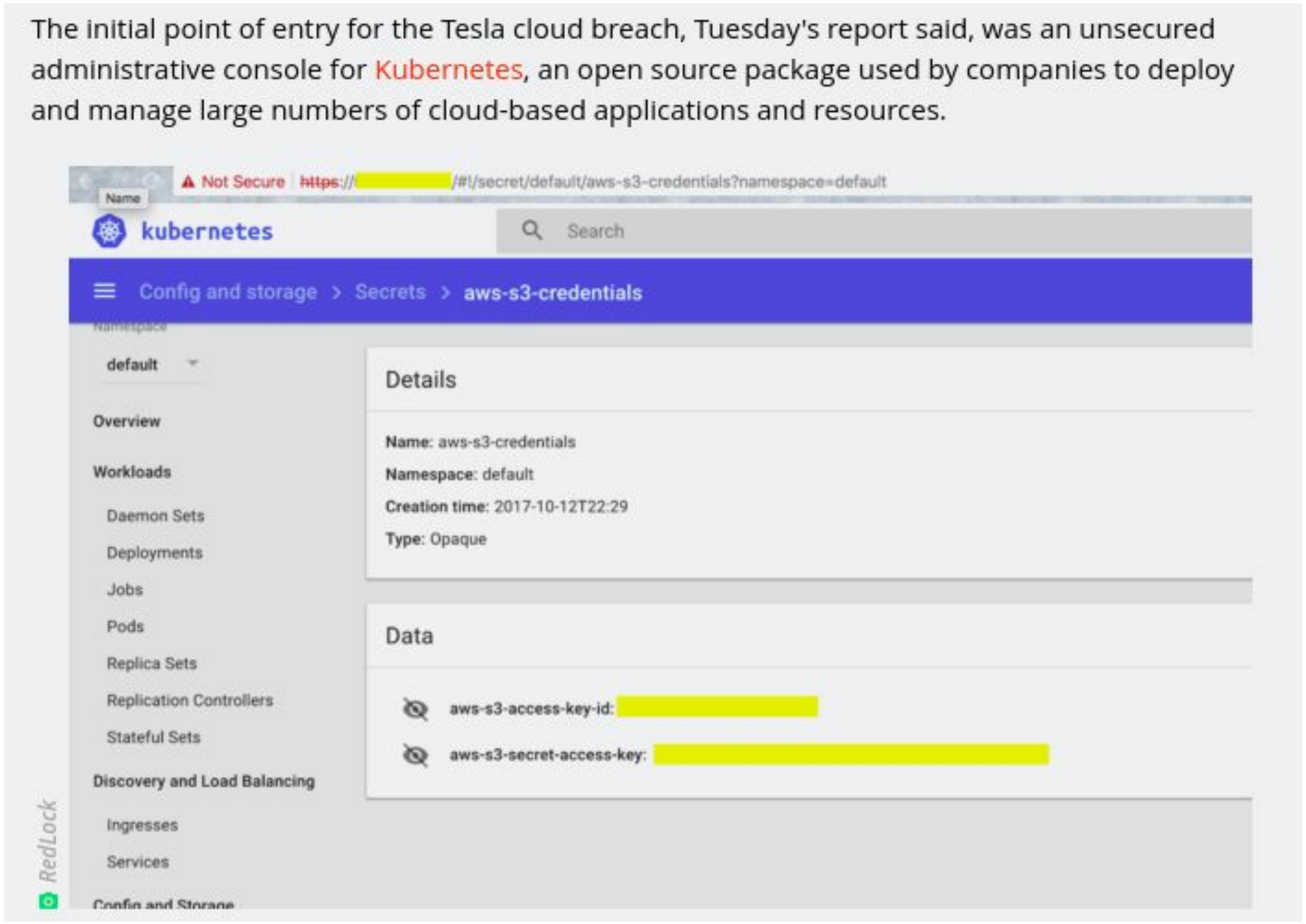
Type of attack

False credentials

Effect

The total scope of the breach is yet unknown

Tesla Breach



The initial point of entry for the Tesla cloud breach, Tuesday's report said, was an unsecured administrative console for Kubernetes, an open source package used by companies to deploy and manage large numbers of cloud-based applications and resources.

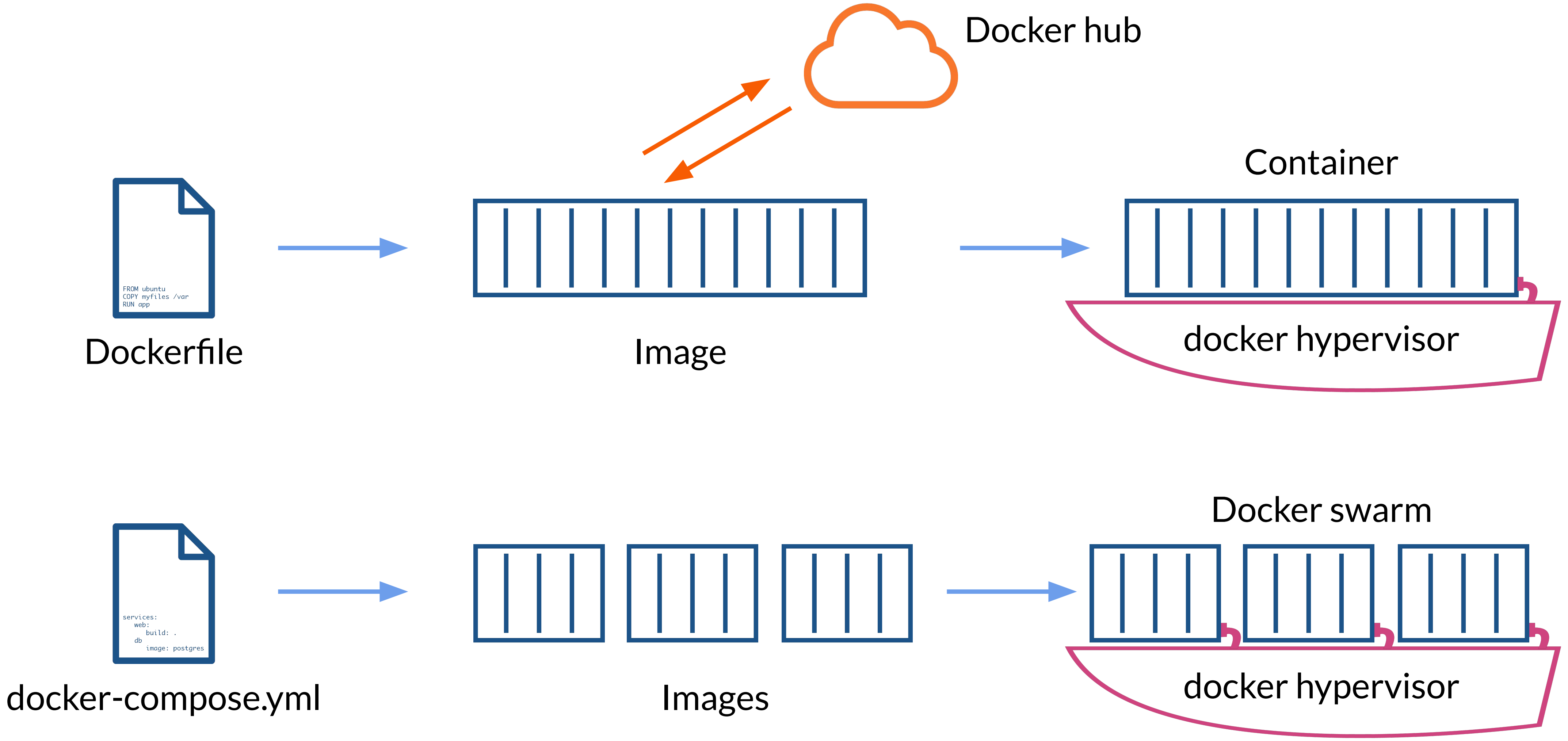
Details

Name: aws-s3-credentials
Namespace: default
Creation time: 2017-10-12T22:29
Type: Opaque

Data

aws-s3-access-key-id: [redacted]
aws-s3-secret-access-key: [redacted]

What is Docker?

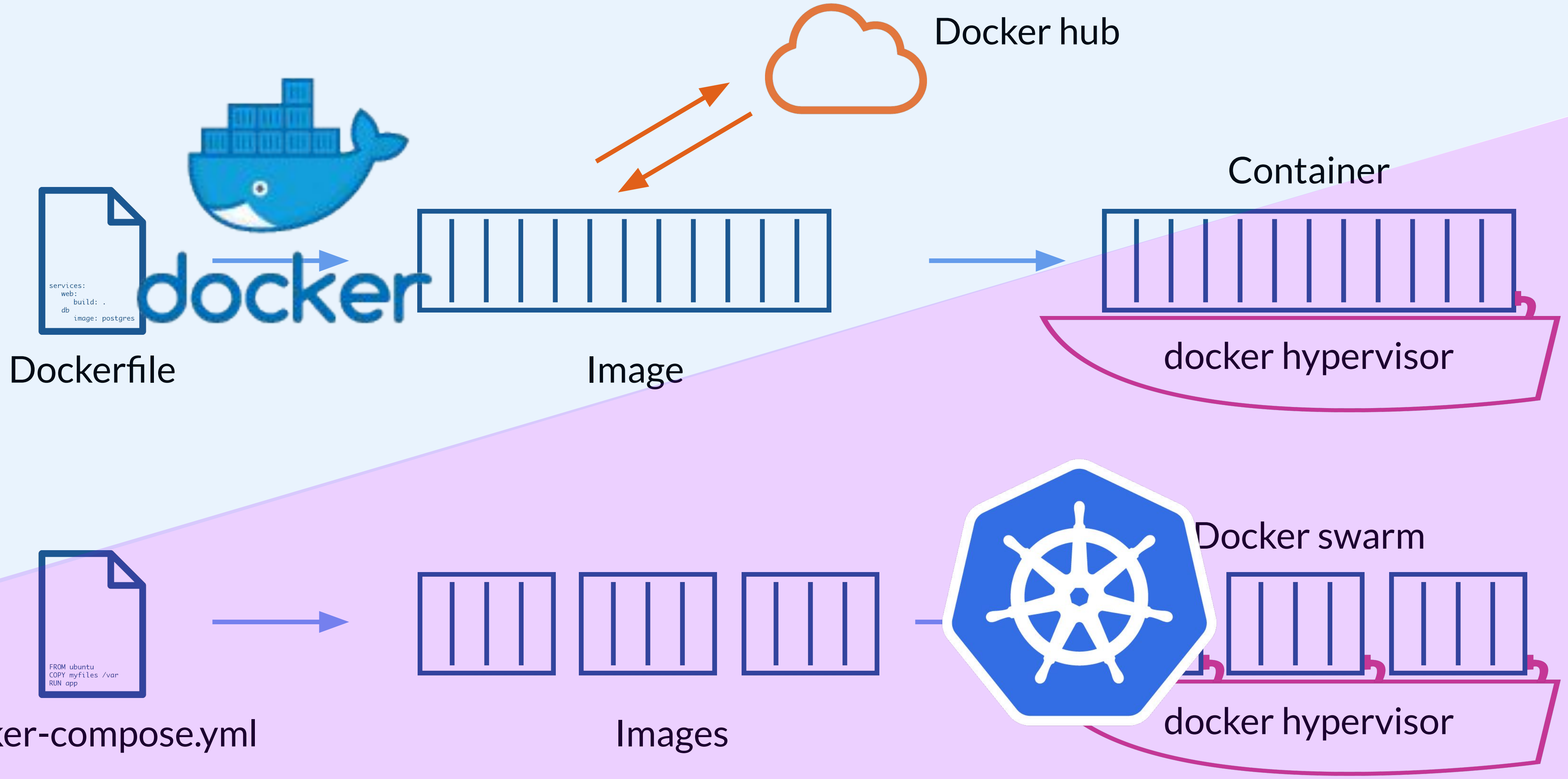


```
FROM ubuntu
COPY myfiles /var
RUN app
```

```
services:
   web:
      build: .
   db
      image: postgres
```

Docker ecosystem, infographic by Rob Richardson robrich.org

What is Kubernetes?



Namespaces

“K8s does not provide a mechanism to enforce security across Namespaces. You should only use it within trusted domains and not use when you need to be able to provide guarantees that a user of the cluster or pods be unable to access any of the other Namespaces resources”

--GCP Team

tl;dr: A namespace is not a security boundary for inter-pod communication.

Role based access control (RBAC)

Roles and ClusterRoles are a whitelist; essentially a list of the allowed permissions.

RoleBindings and ClusterRoleBindings marry users to roles:

- **Subject** includes the person, place, or thing that has been whitelisted.

  Ex) a developer, DevOps, a team member, user, or process.

- **Resource** is the kind of object

  Ex) pod, service, the cluster itself, or another logic instance related to Kubernetes.

- **Operations** that are whitelisted are action we permit the system to do. It's an action related to REST method.
- **Namespace** is the kubernetes section that is allowed.

Network Policies

“By default, pods are not isolated; they accept traffic from any source.” - GCP

https://kubernetes.io/docs/concepts/services-networking/network-policies/

Secure traffic between containers

using service mesh tools like Istio

Disable legacy APIs

etcd access from worker nodes

(Shopify)

Restrict API/ Dashboard access

(Tesla)

Kubernetes: Pod security policies

Run as non-root user

Smallest base container

Don’t install unnecessary software

Note: Don’t run as Root

Configuration Management

Config File in Container

must trust developers, registry, git repo

Environment Variables

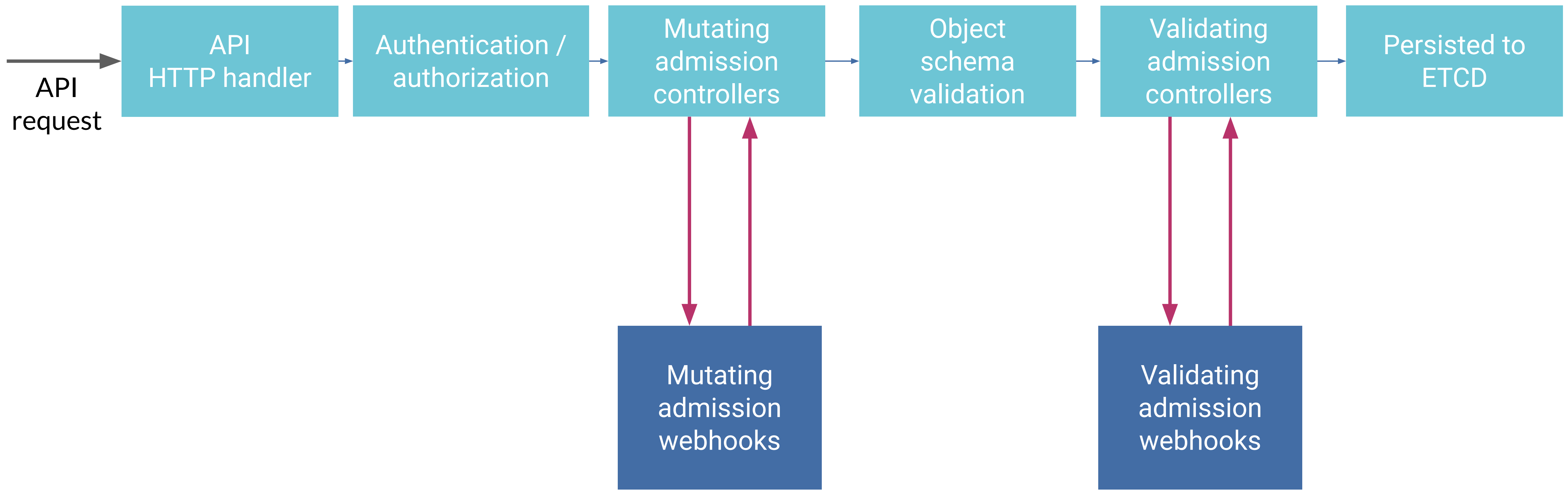
Must trust operations

External Key Vault

Must change application

Note: RBAC is usually best

Kubernetes API request lifecycle


API request
API HTTP handler
Authentication / authorization
Mutating admission controllers
Object schema validation
Validating admission controllers
Persisted to ETCD
Mutating admission webhooks
Validating admission webhooks

What Next?

APPLICATION SECURITY

AppSec follows from the above security methods.

Attacks can come from multiple directions. Separate application-specific vulnerabilities

- Orchestrator vulnerabilities
- Container content vulnerabilities
- Client-side elements

You cannot secure Kubernetes without securing applications.

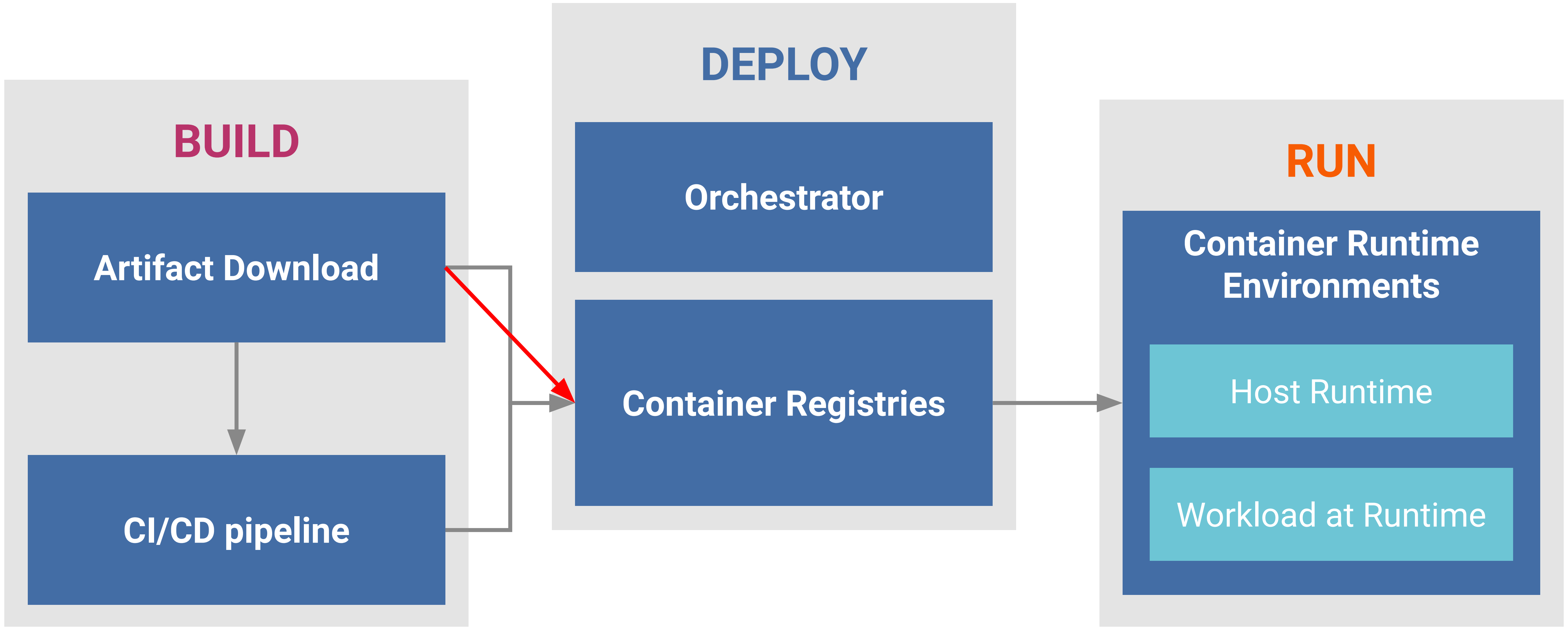
Note: Microservice environments are very useful, but they are not safe without special measures.

CLOUD NATIVE COMPUTING FOUNDATION

Kubernetes - Zooming Out

Do’s and Don’ts for Containerized Environments

Build. Deploy. Run.



DOs for Containerized Environments

CREATE IMMUTABLE CONTAINERS

RUN IMAGES ONLY FROM TRUSTED SOURCES

USE CONTAINER-NATIVE MONITORING TOOLS

NOT To Dos for Containerized Environments

- Don’t install an operating system in a container
- Don’t run unnecessary services
- Don’t store critical data in a container
- Don’t put hard-coded credentials for accessing Registry
- DON’T run a container as root

Kavya Pearlman

@KavyaPearlman

www.wallarm.com

Rob Richardson

@rob_rich

robrich.org