
KubeSec
Enterprise vSummit 2021
aqua
The Container Supply Chain Model
A Strategy Discussion
Jimmy Ray
10/12/2021

# Disclaimer

The views and ideas expressed in this presentation are my own.

# Agenda

- Introduction to the problem
- Container challenges/DiD/Concerns
- Introduction to the Container Supply Chain Model (CSCM)
- Summary
- Q&A

# Reasons for Containers

- Accelerate software development
- Build modern applications
- Automate operations
- Reduce attack surfaces and threat vectors
- Enforce best-practices and policies

# The Problem

- Building container-ready applications
- Managing and securing all the processes, tools, and artifacts involved in moving from code to containers
- Implementing a layered-approach for complementary and redundant security
- Responding to events and making data-driven decisions

# Container-Ready Applications

"Container-ready applications have small footprints and take advantage of free services offered by orchestration layers such as logging, federated security, and metrics gathering. They are atomic and robust units of compute that can be partitioned for elasticity. Whether applications are designed to be container-native from the start, or refactored to be container-ready, embracing MSA and Twelve-Factor App methodology enables deployment into modern container orchestration layers." – Jimmy Ray, January 2019

Link: https://medium.com/capital-one-tech/container-ready-applications-with-twelve-factor-app-and-microservices-architecture-16af683a767f?source=friends_link&sk=eac45fea533fcde3601e7c788a40a1f2

# Some Container Challenges

- Processes usually run on a shared host (OS/kernel)
- Isolation implemented by Linux namespaces, cgroups, and chroot
- Immutable and Shorter Time-to-Live (ephemeral)
- Traditional/legacy security software is becoming container-aware
  - Firewalls
  - IDS/IPS
  - DLP
  - Forensics

# Some More Container Challenges

- Detecting and controlling container image provenance and contents
- Preventing unwanted or dangerous artifacts from getting into container images (bill-of-materials)
- Controlling from where container images are sourced
- Detecting issues with images and understanding the impact on downstream execution environments
- Detecting and preventing misbehaving containers
- Erecting guardrails for developers as well as for execution environments
- Compliance and Regulatory requirements

# Defense-in-Depth (DiD)

- Defensive mechanisms are layered
- Multi-layered approach increases the security of the system as a whole
- Intentional redundancies
- Designed to address different attack vectors
- Includes persistent forensics

# Some Important Management Concerns

- Risk
- Threats
- Impact
- Exposure and Blast-radius
- Resiliency and Disaster Recovery (including RTO/RPO)
- Service Level Agreements/Indicators/Objectives (SLA/I/O)
- Regulatory Requirements/Audits
- Cycle-time

# A Supply Chain

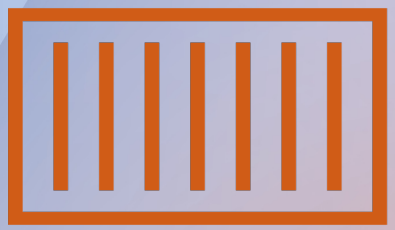
*"A **supply chain** is a network between a company and its suppliers to produce and distribute a specific product to the final buyer. This network includes different activities, people, entities, information, and resources. The supply chain also represents the steps it takes to get the product or service from its original state to the customer.*

*Companies develop supply chains so they can reduce their costs and remain competitive in the business landscape." - Ivestopedia*

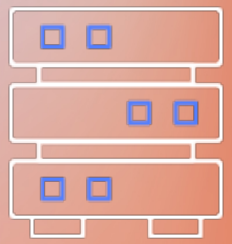
# Example Container Lifecycle

## Build

## Ship



## Run

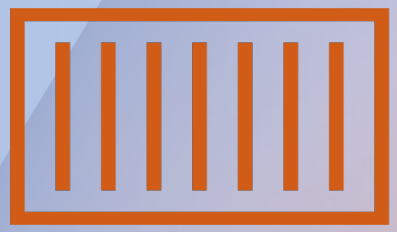


Source: https://www.docker.com/sites/default/files/Infographic_OneDocker_09.20.2016.pdf

# Container Supply Chain Model (CSCM)

## Development

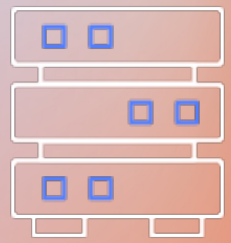
## Image Curation



Trust Boundaries

Security Boundaries

## Container Execution

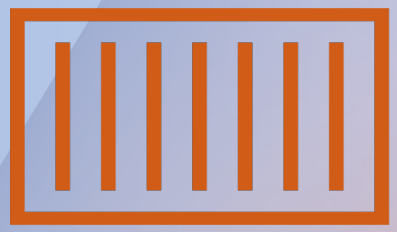


Build → Test → Store → Encrypt → Scan → Deploy → Monitor

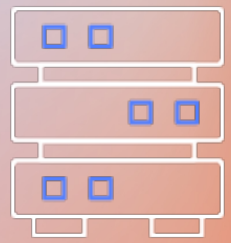
# Container Supply Chain Model (CSCM)

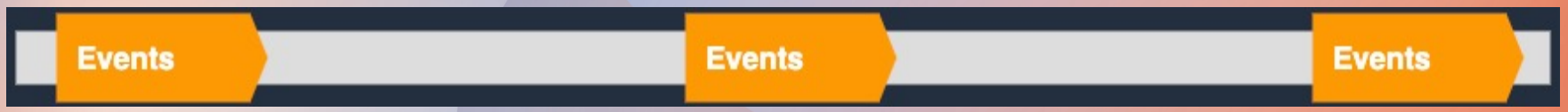
## Development

## Image Curation



## Container Execution

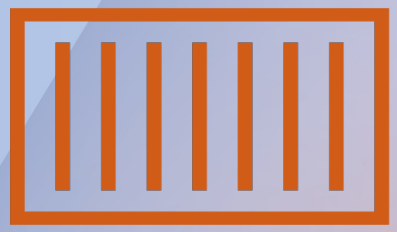




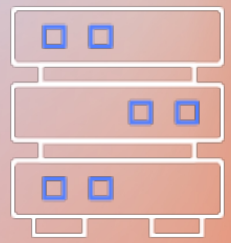
# Container Supply Chain Model (CSCM)

## Development

## Image Curation



## Container Execution



Metadata → Metadata → Metadata

KubeSec
Enterprise vSummit 2021

# Container Supply Chain Model

A container supply chain is the system that moves code/artifacts to execute as containers. It includes:

- Writing/testing/scanning software
- Generating images
- Curating Images (storing/scanning/encrypting/vending)
- Authorizing containers to execute
- Executing/scanning/remediating containers
- Event-driven decisions and actions, with feedback loops

# Container Supply Chain Security Principles

- Self-service automation with Least Privilege
- Defense-in-Depth
- Attack-surface Reduction
- Blast-radius Reduction
- Segregation-of-Duties (SoD), arranged into trusted (but verified) tiers
- Admission Control (tier transitions)
- Automated decisions and actions based on lifecycle events and collected metadata

# Trust Boundaries

- A.k.a. Security Boundaries or Data Boundaries
- “Gatekeepers” for application data
- Formed, within an application or architecture, anywhere the level of trust and reliability of data change.
- Applied to a container supply chain, afford more control and flexibility, while progressively reducing variability and increasing rigor and trust, as application lifecycles progress from code creation to container execution
- Artifacts transitioning from an upstream trust boundary should progress through automation that includes Admission Control

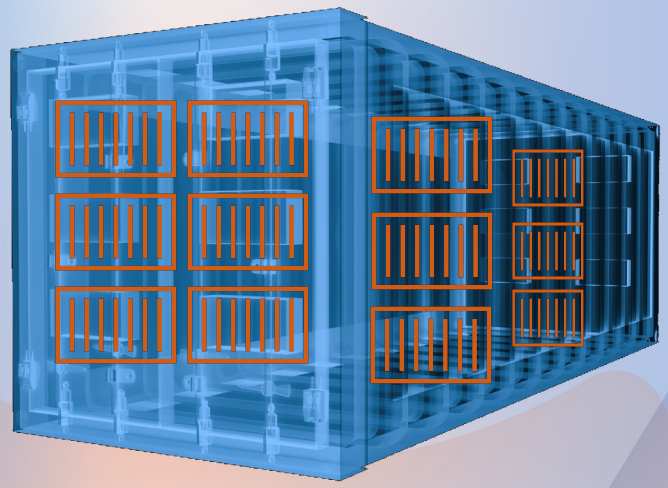
# Development Tier

- Least amount of rigor in the model
- Developers work here with IDE tools
- Code is written/integrated/tested (SCA/SAST/DAST)
- Container images created
- Automated (CICD) pipelines
- Guardrails are erected to guide developers
- Admission to next supply chain tier is controlled

# Development Tier Levers

- 12-Factor App Methodology
- Microservices Architecture
- Container design patterns/principles
- Signal handling
    - SIGTERM
- Automated testing
- Small base images
    - Minimal images
- Internal vs external base images
- GitOps

# Container Image Curation Tier

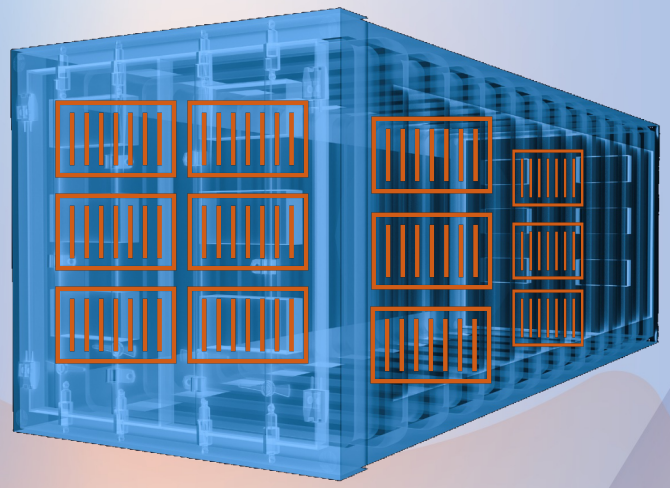


- About container images
- CICD pipelines store images into controlled registries/repositories
- Images are stored/scanned/encrypted
- Image tags are made immutable
- Image repositories are secured
- Event-driven controls applied to repositories, based on lifecycle configurations and events (scans, etc.)

# Image Curation Events and Actions

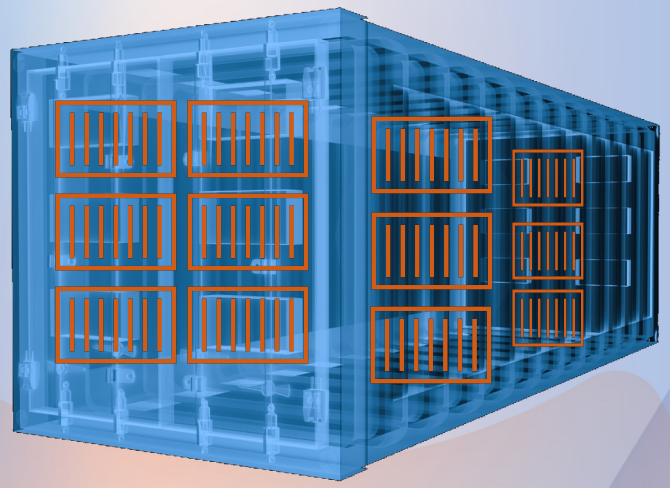
- Encryption
- Scans
- Signing
- Lifecycle events (based on age, tag, etc.)
- Repository restrictions
- Metadata Collection (advanced)
- Vending

# Container Execution Tier



- About containers, not container images
- Automation-driven configuration
- Separating application and configuration lifecycles
- Enforcing rules/profiles for container execution
    - user-space vs. kernel-space
- Compliance
    - Policy-as-code
- Scanning and monitoring containers
- Preventing mutability and drift

# Container Execution Tier (continued)



- Reacting to upstream events
- Repudiating unauthorized container execution
- Verifying container image signatures
- Mutating and validating inbound container configurations
- Remediating and isolating misbehaving or otherwise compromised containers
- Regularly collecting and persisting possible forensics data
- Conformance
- Chaos

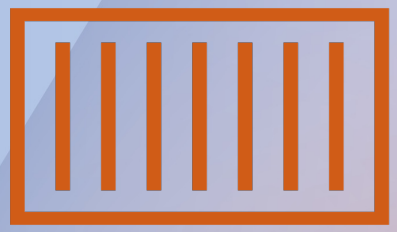
# Container/Host Placement Strategies

- Workload types
- Resource requirements
- Scaling
- Compute requirements
- Tenancy
- Isolation Factors (Data classification, etc.)

# Container Supply Chain Model (CSCM)

## Development

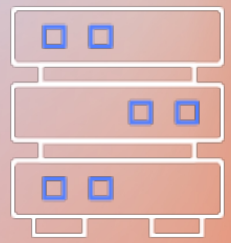
## Image Curation



Trust Boundaries

Security Boundaries

## Container Execution

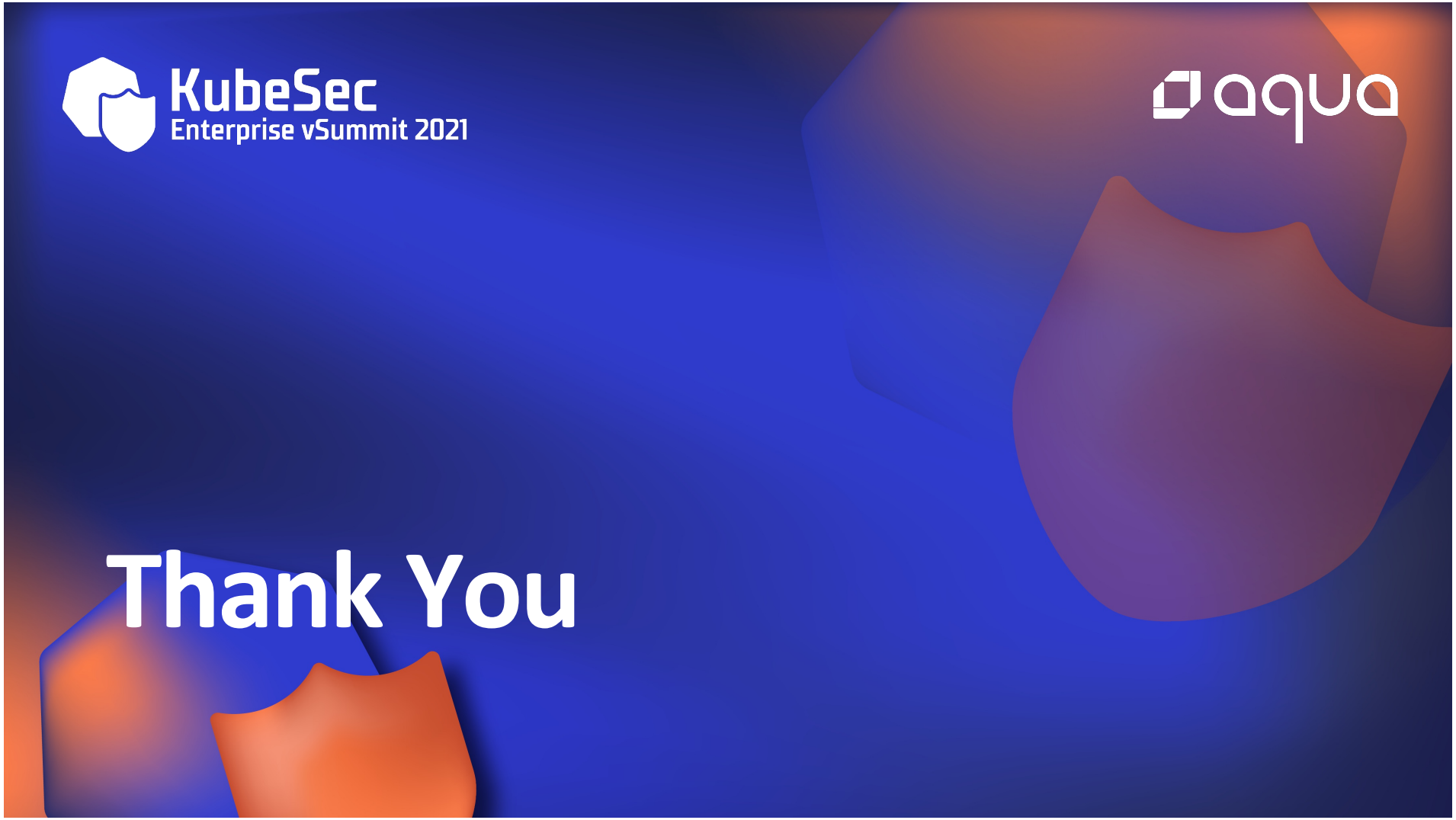


Build → Test → Store → Encrypt → Scan → Deploy → Monitor

# Summary

- The CSCM helps distill simplicity from complexity
  - Decomposes problem into manageable areas of focus
  - Helps to apply Defense-in-depth
- Tiers provide trust boundaries, separation-of-duties, focus
  - Trust buy verify – admission control
- Upstream events impact downstream tiers
  - Events
  - Metadata
    - Bill-of-materials (BOM) for artifacts and images
    - Evidence and attestations
    - Provenance


KubeSec
Enterprise vSummit 2021
aqua
Q & A


KubeSec
Enterprise vSummit 2021
aqua
Thank You