Lessons from the Trenches of Container Scanning in CI/CD

Zach Hill

Chief Architect and Director of Engineering @ Anchore

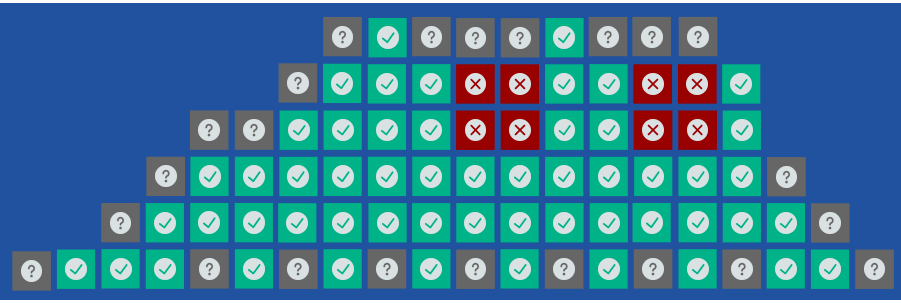
Overview

- Container Scanning in CI/CD & Anchore
- Lesson 1: You can automate away reliance on golden images
- Lesson 2: Align goals early
- Lesson 3: Understand the scanner & results

© 2020 All Rights Reserved.

A Challenging Security Environment

Complexity



Developers are building upon a large ecosystem of available containers with questionable quality & variable provenance

Velocity

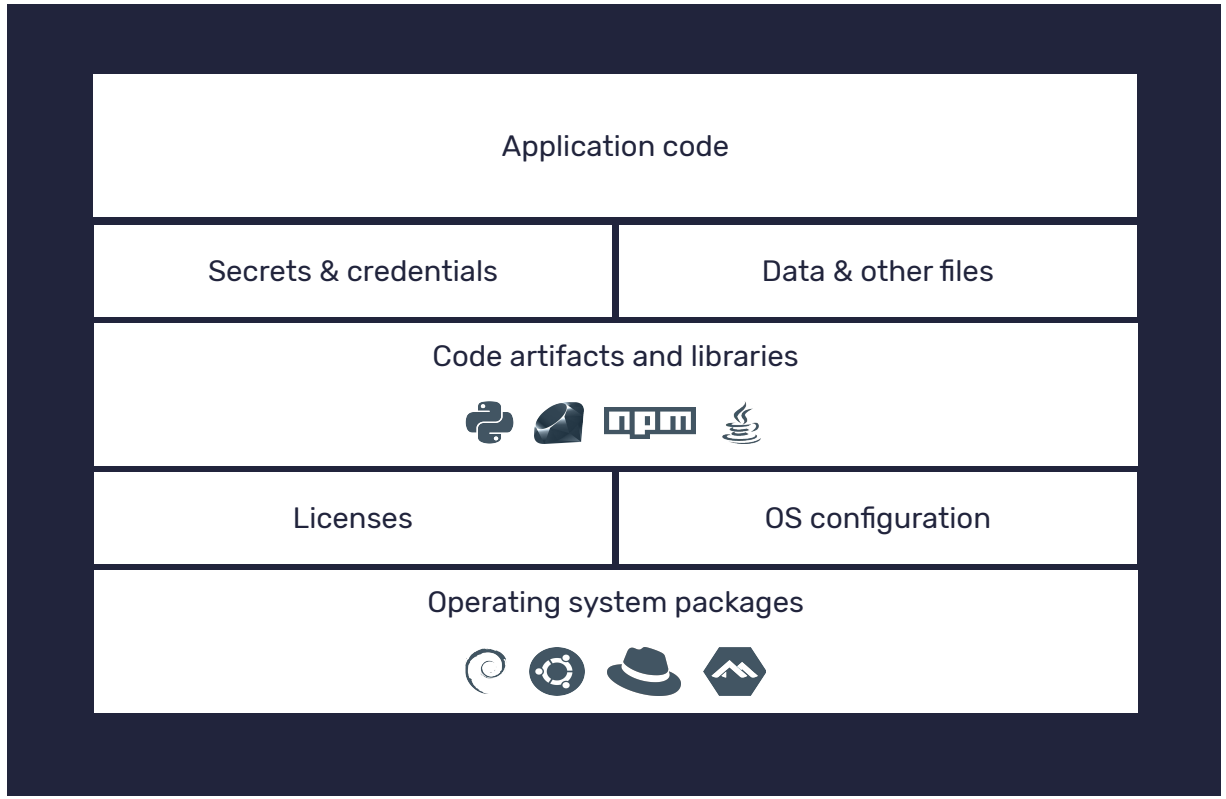
Organizations are adopting containers to improve devops efficiency, but security is still perceived as an encumbrance

Compliance

Compliance requirements are becoming increasingly complex in response to public security breaches

© 2020 All Rights Reserved.

Container Images



A **container image** is a single artifact that contains everything required to run an application or service.

Usually on a server, almost always on one running Linux.

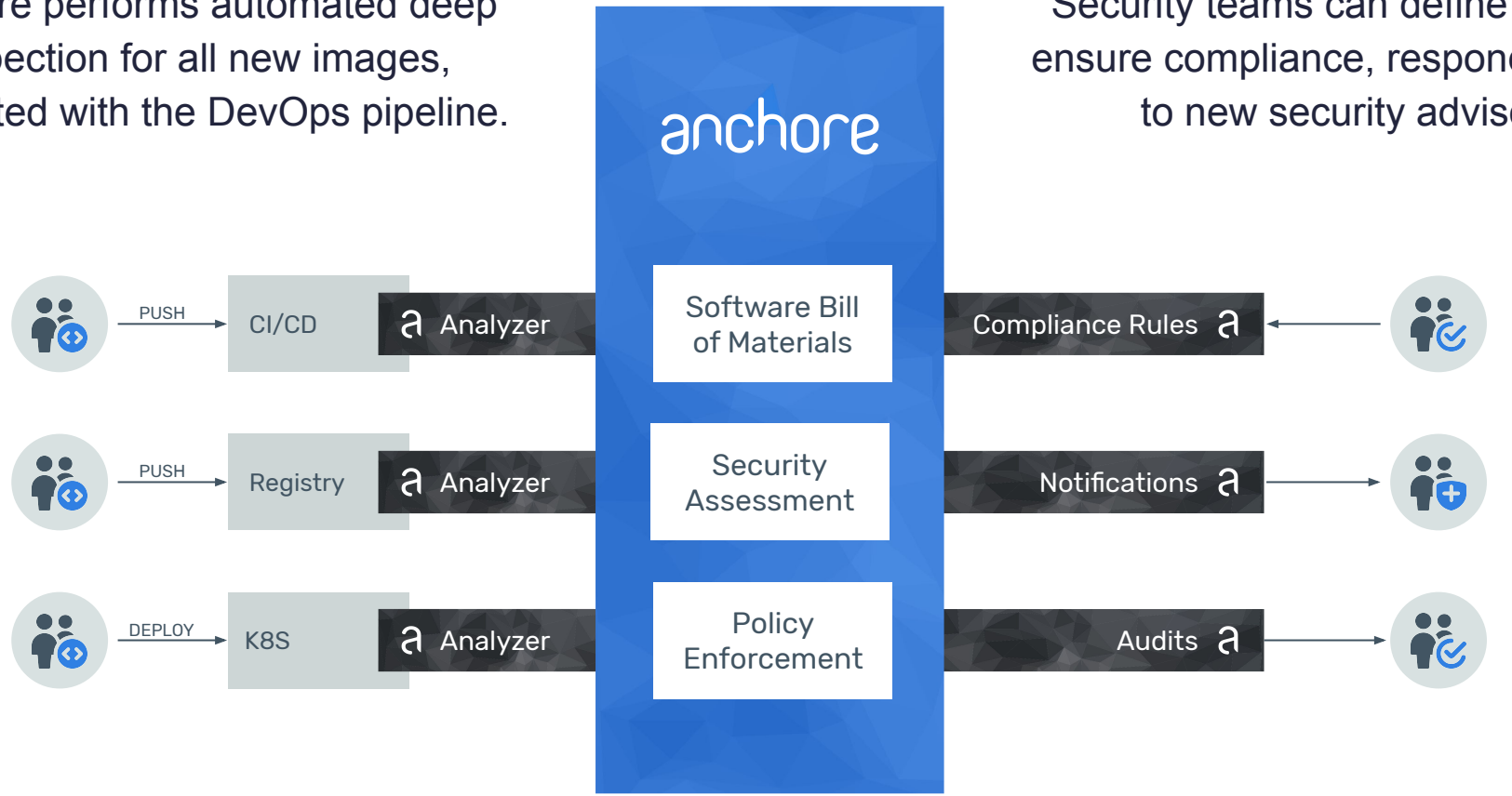
Containers provide greater consistency, which leads to more advanced tooling and heavy automation.

© 2020 All Rights Reserved.

Anchore Container Scanning and Compliance

Anchore performs automated deep inspection for all new images, integrated with the DevOps pipeline.

Security teams can define policies to ensure compliance, responding quickly to new security advisories.



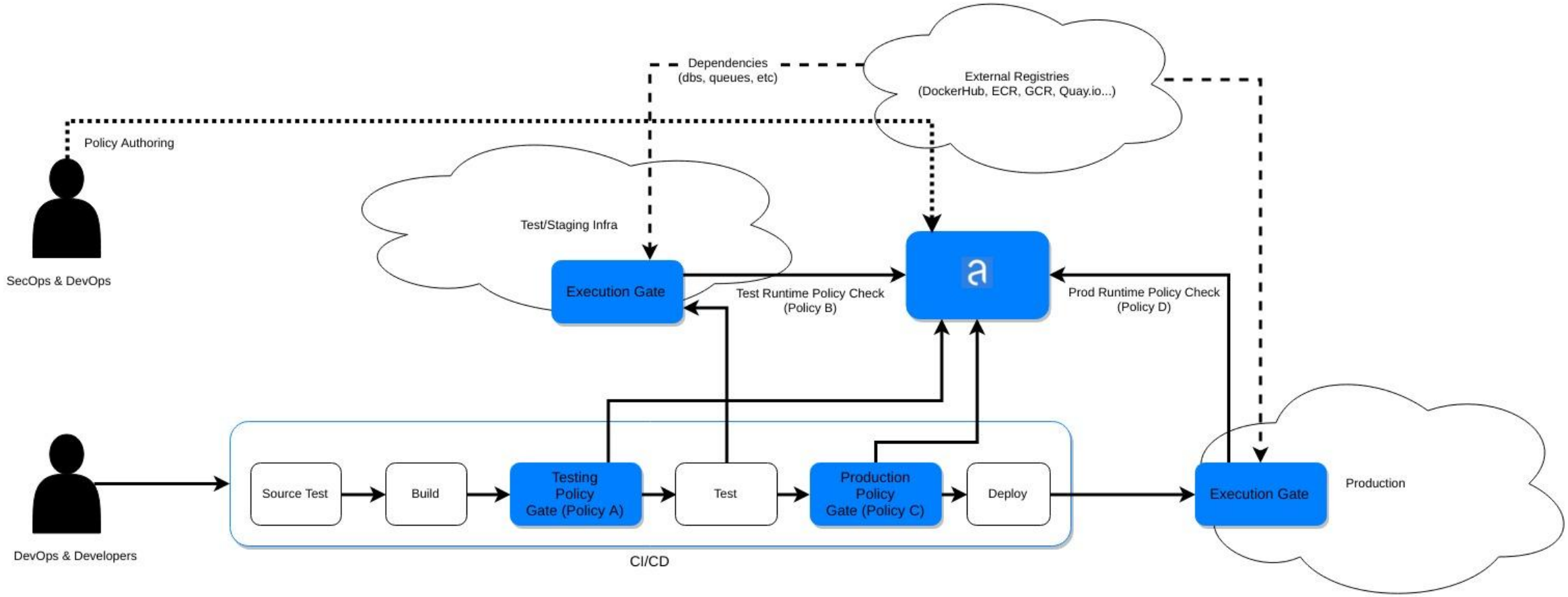
© 2020 All Rights Reserved.

Sources for These Lessons

- Work with Anchore open-source community and customers
  - SMBs, Enterprises, & Govs.
  - 1000+ open-source deployments
- Use-Cases
  - CI/CD
  - Registry audit
  - Execution gating
- Installations from 10s to 1000s of images scanned per day

© 2020 All Rights Reserved.

Container Scanning and Security in CI/CD



© 2020 All Rights Reserved.

Lesson 1

You can automate away reliance on golden images

DEVOPS WORLD
by CloudBees

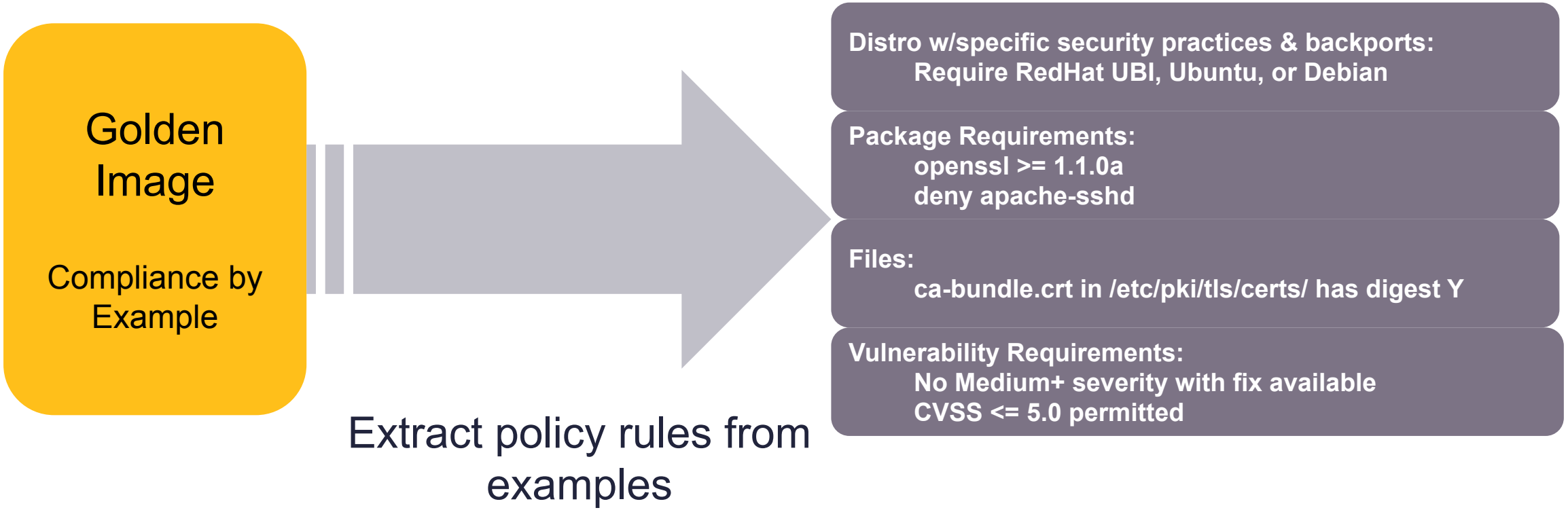
Golden Images

- Compliance by example
  - Known good baseline
  - Limit drift
- Limitations
  - Centralized maintenance
  - Limits flexibility and velocity
  - Creates dependencies in workflows

- Why move to policy-based? Flexibility, scalability, and time-to-solution

© 2020 All Rights Reserved.

Convert Compliance by Example to Policy-Based



© 2020 All Rights Reserved.

Incremental Approach

Overlap during validation

© 2020 All Rights Reserved.

Building Policy & Choosing Tools

- More than package db and vulnerability match
- Describe the properties you need to check
  - Files, packages, certs, permissions, users
- Anchore specifically designed for deep static inspection
  - Packages (rpm, npm, java...), file content & checksums, metadata
  - Policy language
    - Vulnerabilities
    - Package allow/deny
    - File content verification
    - Secrets

© 2020 All Rights Reserved.

Lesson 2

Align Goals Early

DEVOPS WORLD
by CloudBees

Approaches & Goals

- Golden Images
  - The image is the definition of compliance
  - Detect & limit deviation
  - Assumes golden is “good”
- Policy-Based
  - Tools need expressive policy
  - May need multiple tools
  - Block vs Audit w/async updates

© 2020 All Rights Reserved.

Gate & Audit in CI

- What are you evaluating?
  - Is it under the control of the upstream provider? Identify the owner
- Gating
  - Handling false-positives more urgent
  - Does developer have ability to fix the issue? Who owns all the images in the stack?
- Audit
  - Detect and record findings
  - Feedback loop to resolve with upstream
  - Higher throughput

© 2020 All Rights Reserved.

Gate & Audit in CD

- What artifact passed CI, what is scanned now?
- New deploy vs scale/config-update. GitOps?
- Have ‘break-glass’ option
- Plan for triaging findings
- Accessible scan results and scanned artifacts
- Ensure you have a clear remediation plan
- **Gate/Audit Hybrid is effective in CI & CD, but requires more sophisticated tools like Anchore**

© 2020 All Rights Reserved.

Lesson 3

Understand the Scanner & Results

DEVOPS WORLD
by CloudBees

Scanners and Noise

- All scanners will generate some noise
  - Sources of noise
    - Artifact identification (missing, or incorrect metadata)
    - Vulnerability data
- Verbosity
  - Filter or show everything?
  - For audit: more is ok, but don't overwhelm triage ability
  - For gating: keep focused on actionable items that are critical

© 2020 All Rights Reserved.

Understand the Scanner

- How does it identify artifacts?
  - Metadata vs binary scan/digests
  - These impact false-positive rate and detection sensitivity
- What are the vulnerability data sources and can you access it?
- How is matching done?
- Understand & document the process for triaging findings
- Does it support removing false positives or do you have to build that?

© 2020 All Rights Reserved.

Know Your Distro & Base Images

- OS images vs application stack containers
  - Backports
  - How is base managed? Tracks distro or locked?
- Vulnerability sources update frequently
  - Records added, updated, and **removed**
  - National Vulnerability Database NVD
  - Distro-specific sources & proprietary sources
  - Expect differences in severity, impact, etc for same CVE in different sources
  - CVSS vector helps, but not always applicable to containers. Good starting place

© 2020 All Rights Reserved.

Small Focused Images Are Easier to Understand

- Containers != VMs
- Not just lower surface area, also easier to understand and reason about
- Reduce time to answer: *“why is package X in here?”*
- Use side-car model for tools, logs, etc

© 2020 All Rights Reserved.

Conclusions

DEVOPS WORLD
by CloudBees

In Summary

- Scanning should be valuable and help accelerate velocity. It doesn't have to be a burden
- Can move from golden images to policy-based
- Be clear on your objectives for scanning
- Design the pipeline carefully
- Be realistic on results & understanding the variables
- Have a clear process for reviewing results

© 2020 All Rights Reserved.


anchore.com
info@anchore.com
github.com/anchore
@anchore
DEVOPS
WORLD
by CloudBees