DigitalOcean Kubernetes

digitalocean.com

Kubernetes Supply Chain Security

Recommended Security Practices for Developers and SMBs

Bikram Gupta, Product - DigitalOcean
Marco Morales, Partner Solutions Architect - Snyk

What is your top-of-mind Security Pain Point for Kubernetes?

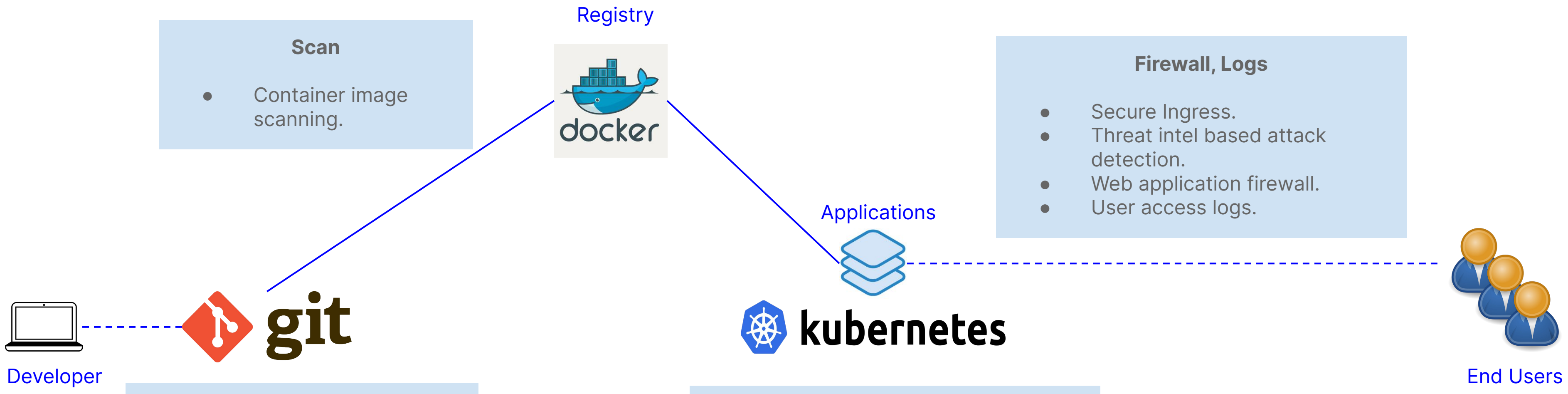
Can be an issue, or concern, or even what you do not fully understand.

What You’ll Learn

- Big-picture Security for Kubernetes and Applications
- Components of Supply-chain Security
- How to incorporate Snyk into your CI/CD

Big-picture view of Securing Kubernetes

Kubernetes Security - Big Picture



Scan

- Container image scanning.

Firewall, Logs

- Secure Ingress.
- Threat intel based attack detection.
- Web application firewall.
- User access logs.

Scan, Sign

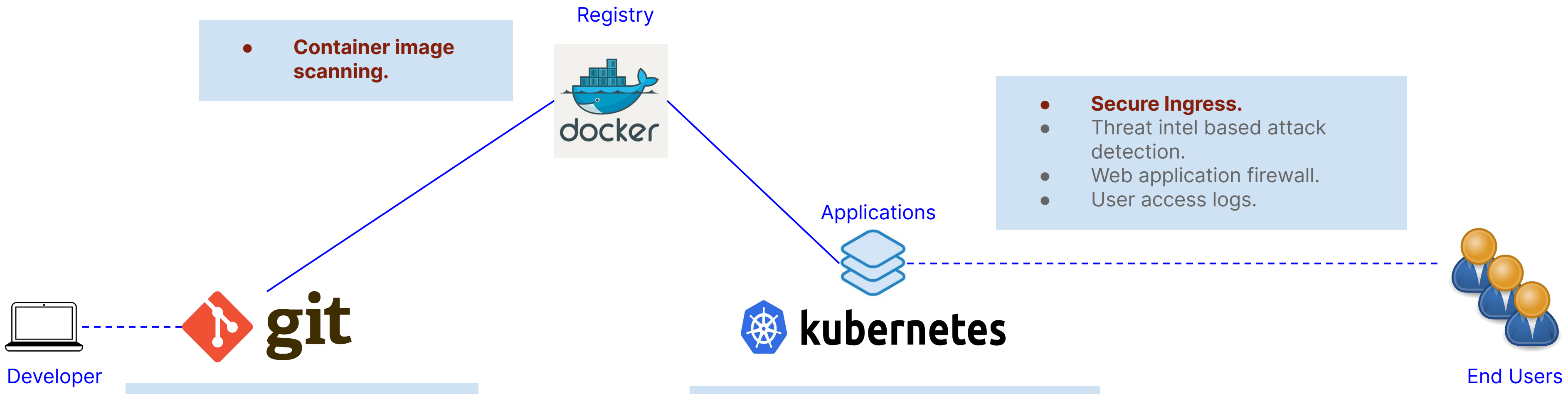
- Security Scanning on Dockerfile, manifests, and Source code.
- Image and Manifests Signing.

Control plane, OS, Network

- Ongoing Scans for nodes and containers.
- Image admission control.
- Security compliance checks.
- Network policy.
- Secure secrets.
- Audit logs.
- Outbound threat detection (eg. Crypto, Command and Control).
- Run-time threat detection.

How many of you have a dedicated security staff in your organization?

Kubernetes Security - Big Picture



- **Container image scanning.**

- **Secure Ingress.**
- Threat intel based attack detection.
- Web application firewall.
- User access logs.

- **Security Scanning on Dockerfile, manifests, and Source code.**
- Image and Manifests Signing.

- **Ongoing Scans for nodes and containers.**
- **Image admission control.**
- **Security compliance checks.**
- **Network policy.**
- **Secure secrets.**
- **Audit logs.**
- ACL for Kube API.
- Outbound threat detection (eg. Crypto, Command and Control).
- Run-time threat detection.

Optimize your Security Effort

- Automate.
    - Remember to upgrade your tools time-to-time.
- Open-source as a selection criteria.
    - Pick SaaS vendors with freemium (feature complete based on usage) model.
- Factor in the cost of maintaining security.
    - Logs and metrics can be expensive very fast.
    - Detection is simple. Remediation, upgrade testing etc. takes time.

Supply-chain Security

The Log4j Vulnerability

- Apache Log4j is an open source Java-based logging framework that collects and manages information about system activity.
- As part of log parsing, it had a feature to fetch a Java class from a remote server and execute it locally using Java Naming and Directory Interface (JNDI).
    - Hackers could enter malicious inputs to any public-facing system and exploit it.
- Millions of systems affected.
- Triaging was a difficult part. How do you know if one of your systems is using Log4j?
    - By analyzing your code. But when you are using many packages with downstream, undocumented dependencies, it is a hard problem.
    - Using the right tools and integrating those at the right places (supply chain) can help detect and mitigate the the issues faster.

Why Security should be Baked into the Supply Chain?

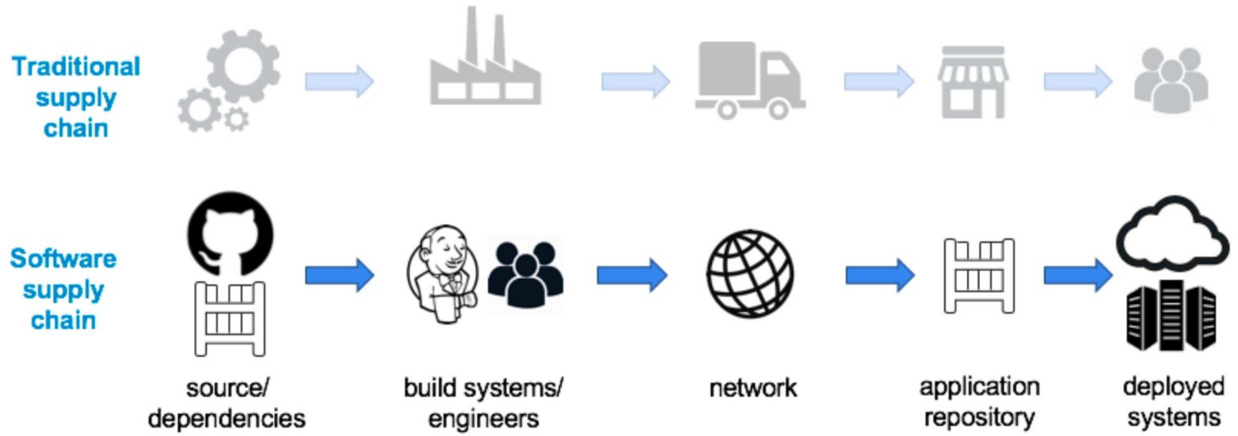


Fig. 1: *https://blog.convisoappsec.com/en/is-your-software-supply-chain-secure/*

- **Intangible** - the software supply chain is an intangible object made up of a series of virtual and digital components that make it extremely difficult to count, discover, and understand their operation end-to-end.
- **Mercurial** - as more teams reduce their time to market, it is reasonable to expect the shape and structure of a software supply chain to continue to change.
- **Iterative reuse** - one product's supply chain, is n number of products and n supply chains, an endless supply of turtles all the way down.

Source: https://github.com/cncf/tag-security/blob/main/supply-chain-security/supply-chain-security-paper/CNCF_SSCP_v1.pdf

Securing the Supply Chain

- Source Code
  - Secure authentication and authorization.
  - Prevent storing secrets in clear-text.
  - Automated scanning and testing.
  - Hardened infrastructure-as-code (IaC).
  - Slimmed Dockerfile.
- Materials
  - Dependencies scanning.
  - Software bill of materials.
  - Trusted repositories.
  - Licensing issues.
- Build Pipelines
  - Secure build components and steps.
  - Reproducible builds.
  - Pipeline as code.
- Artefacts
  - Signed artefacts.
- Deployments
  - Verify artefacts.
  - Access to latest artefacts.
  - Admission control.

What products/tools are you using today for kubernetes security?

Supply-chain Security Tooling Example

Signing: Sigstore, Docker

Scanning: Trivy, Snyk, Kubescape.

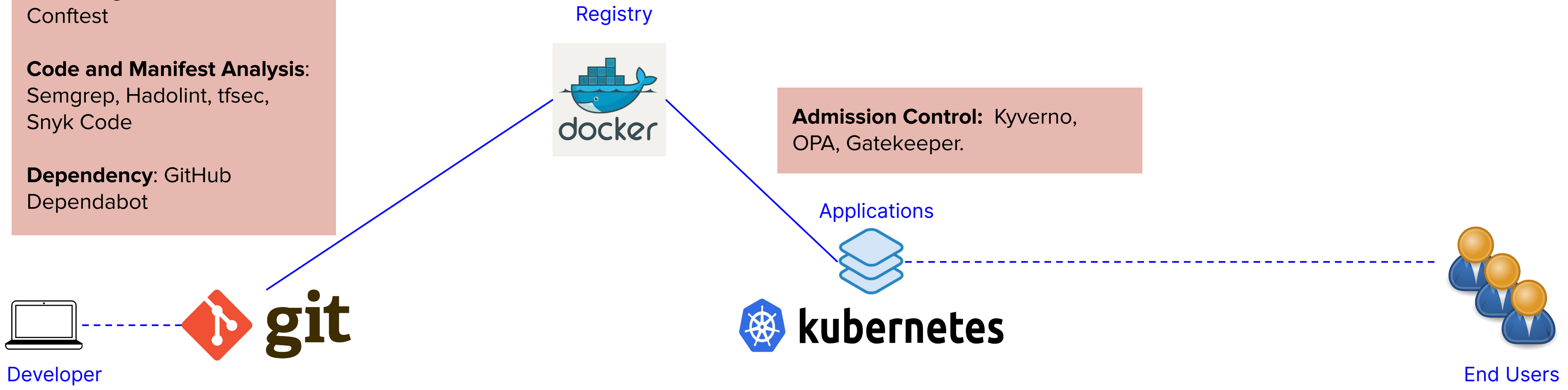
Misconfiguration: Datree, Conftest

Code and Manifest Analysis: Semgrep, Hadolint, tfsec, Snyk Code

Dependency: GitHub Dependabot

Image Scanning: Trivy, Snyk, Kubescape.

Admission Control: Kyverno, OPA, Gatekeeper.



Optimize Docker image: multistage dockerfile, docker-slim

Supply-chain Security for SMBs

- Reduce Docker image file size. This improves security, performance, and productivity, while reducing cost.
- Use the recommended practices of your source-code repositories, eg. GitHub, GitLab etc.
- Invest in one/more good tools to identify vulnerabilities and insecure configurations upfront and resolve those prior to deployment.
- Lock down the deployment to accept infrastructure-as-code (IaC). GitOps is recommended.
- Use a managed build pipeline, eg. GitHub, GitLab, CircleCI etc.
- Additional vulnerability information are learned every week. Hence periodic scanning is necessary.
- If you have a reasonable team size, and with a continuously used supply chain, lock down authorizations, sign images and artefacts, and verify those in deployment.

Develop Fast.
Stay Secure.
snyk

Snyk for securing the Supply-chain

What is Snyk?

Snyk is a developer security platform. Its main goal is to help you detect and fix vulnerabilities in your application source code, third party dependencies, container images, and infrastructure configuration files (e.g. Kubernetes, Terraform, etc).

- Snyk Code - helps you find and fix vulnerabilities in your application source code.
- Snyk Open Source - helps you find and fix vulnerabilities for any 3rd party libraries or dependencies your application relies on.
- Snyk Container - helps you find and fix vulnerabilities in container images or Kubernetes workloads used in your cluster.
- Snyk Infrastructure as Code - helps you find and fix misconfigurations in your Kubernetes manifests (Terraform, CloudFormation and Azure are supported as well).

Snyk Pricing

- Free Tier includes unlimited Developers. Tests limited to as follows.
  - 200 Open Source tests/month
  - 100 Container tests/month
  - 300 IaC tests/month
  - 100 Code tests/month
- Paid plans include team, business and enterprise tiers.

Reference: https://snyk.io/plans/

Demo - Incorporate Snyk in CI/CD Pipeline

Solution Guide

https://github.com/digitalocean/container-blueprints/tree/main/DOKS-supply-chain-security

Steps

- Fork an application repo.
- Sign-up for Snyk, and connect to Github.
- Trigger Snyk workflow via Github actions (CI/CD).
- Check the results, and remediation actions.

Summary

- Implementing proper supply-chain security for Kubernetes can eliminate a lot of security concerns up-front.
- Positive side-effects
  - Automation of the pipeline.
  - Performance (eg. slimmed-down Dockerfile, hardened OS).
- Follow the recommended security hygiene from your version control provider (eg. GitHub, GitLab).
- 3 areas of security scanning in priority order (image, manifests, source code).

Resources

- Securing the supply chain for DigitalOcean Kubernetes: **https://github.com/digitalocean/container-blueprints/tree/main/DOKS-supply-chain-security**
- https://docs.snyk.io/products/snyk-container/image-scanning-library/digitalocean-image-scanning/scan-container-images-from-digitalocean-in-snyk
- https://docs.snyk.io/products/snyk-container/image-scanning-library/digitalocean-image-scanning/container-security-with-digitalocean-integration
- https://sysdig.com/blog/dockerfile-best-practices/

Q & A

24