white duck

How to Prevent Your Kubernetes Cluster From Being Hacked

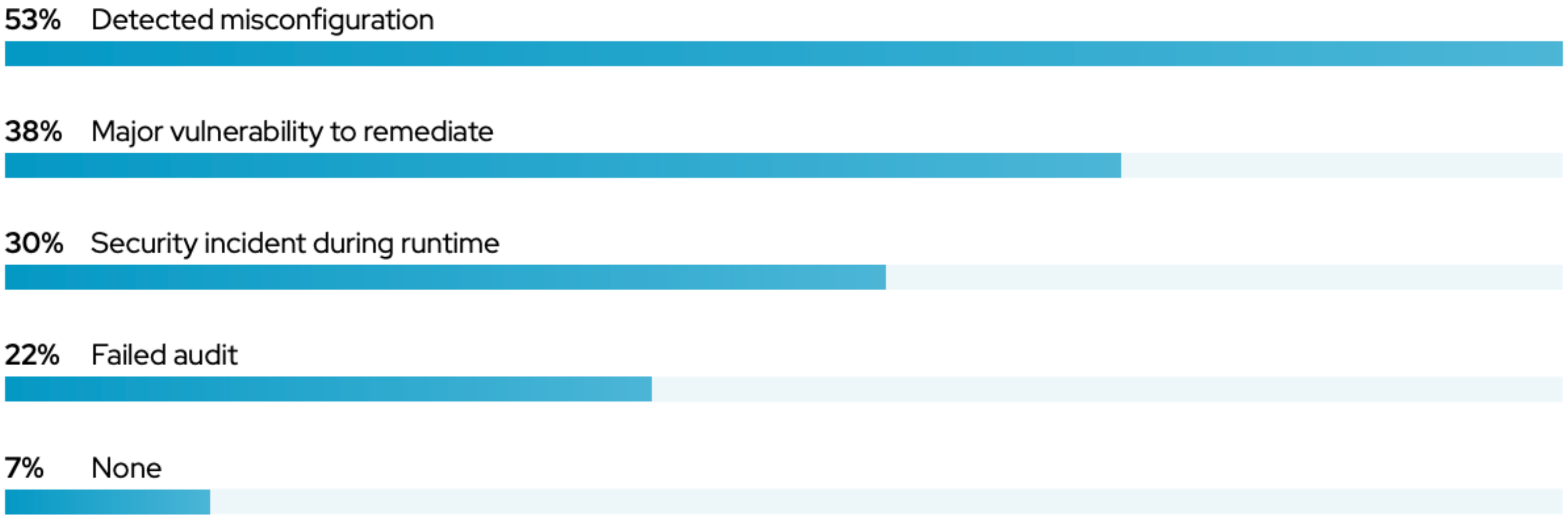
Nico Meisenzahl

- Head of DevOps Consulting & Operations at white duck
- Microsoft MVP, GitLab Hero
- Cloud Native, Kubernetes & Azure

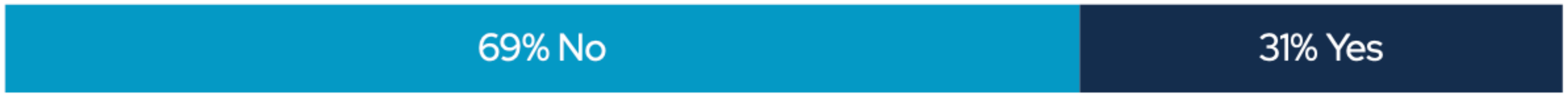
Email: nico.meisenzahl@whiteduck.de
Twitter: @nmeisenzahl
LinkedIn: https://www.linkedin.com/in/nicomeisenzahl
Blog: https://meisenzahl.org

Why do we need to care about security?

In the past 12 months, what security incidents or issues related to containers and/or Kubernetes have you experienced? (pick as many as apply)



In the last 12 months, have you experienced revenue/customer loss due to a container/Kubernetes security or compliance issue/incident?



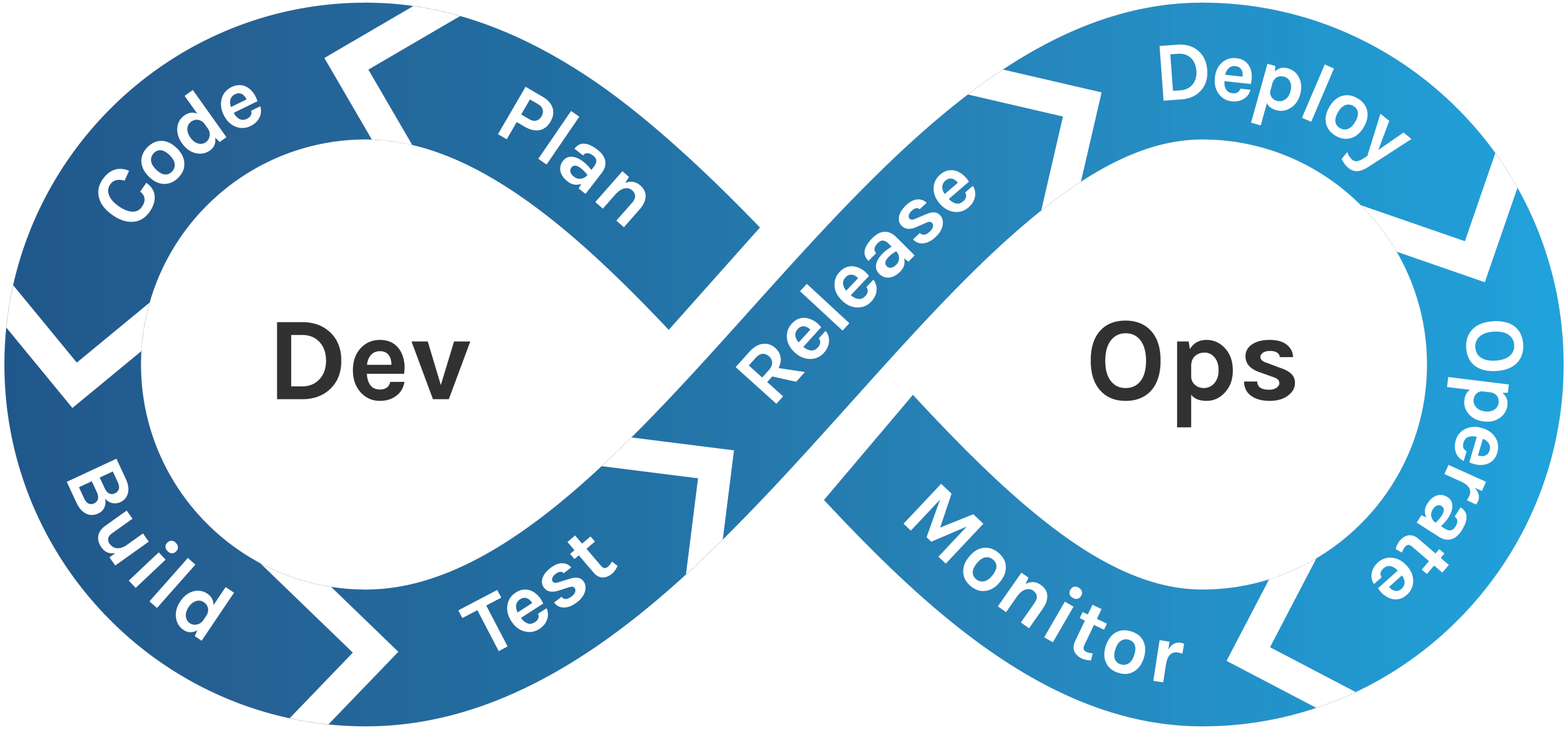
https://www.redhat.com/en/resources/state-kubernetes-security-report

It can be quite simple …

- you don't think so?
- check out my “Hijack Kubernetes” talk
  - https://github.com/nmeisenzahl/hijack-kubernetes
  - recordings on Youtube

white duck

Security quick wins through the DevOps cycle



You should think about

- rely on best practices and quick wins to get started
- ensure secure application & deployment code
- build secure container images
- implement Kubernetes policies
- introduce Kubernetes network policies
- many more …

Things we will focus on today

- build secure images with Wolfi
- image verification with Cosign
- container runtime security with Tetragon

Build secure images with Wolfi

- “the first Linux (Un)distro designed for securing the software supply chain”
  - Undistro what? → Distroless v2
- provides a high-quality, build-time SBOM as standard for all packages
- packages (based on apk) are designed to be independent
- fully declarative and reproducible build system (if you like)

Software Bill of Materials (SBOM)

- “list of ingredients” for all your software and dependencies
  - supports hierarchy and therefore multi-level dependencies
- without you don’t have the full visibility
- in an ideal world you would only need to care about your own stuff
- SBOMs can be the baseline for your vulnerability scanning

Demo: Wolfi in Action

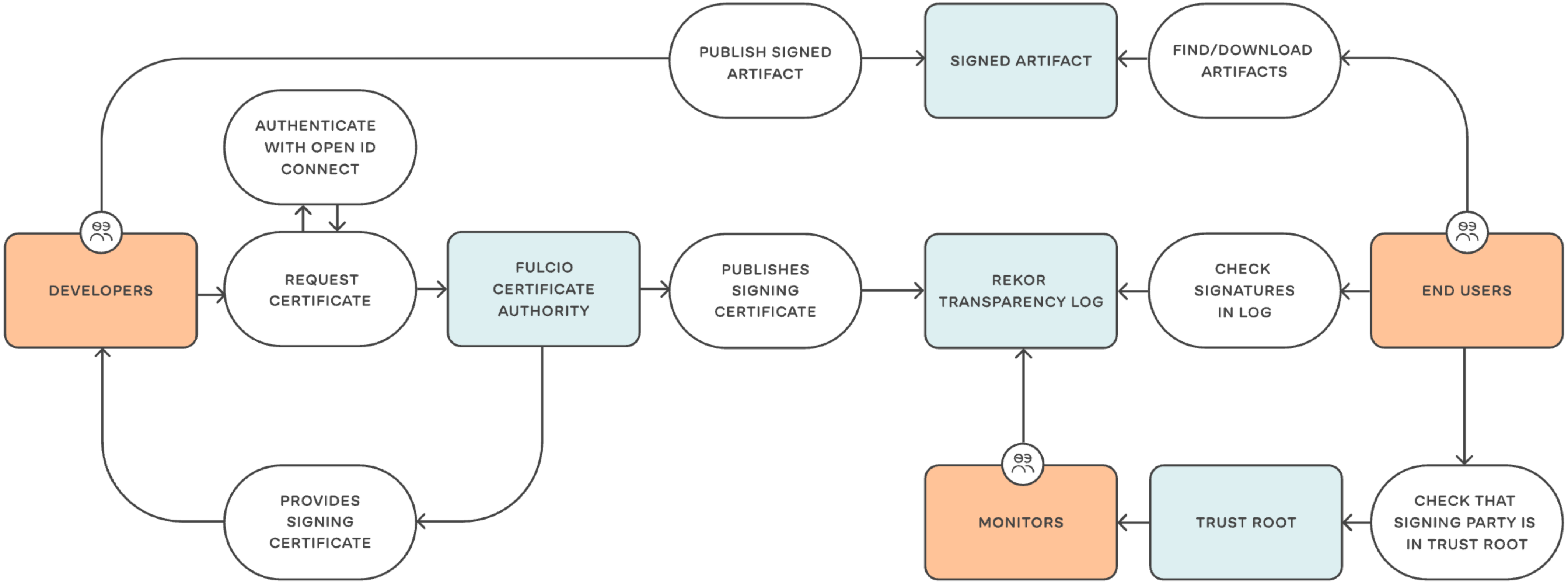
- we will build a Wolfi as base image
  - compare against others (size, vulnerabilities, …)
- then build an image declarative and reproducible with apok & melange
- more details
  - https://edu.chainguard.dev/open-source/wolfi
  - https://github.com/wolfi-dev
  - https://github.com/chainguard-dev/melange
  - https://github.com/chainguard-dev/apko
  - https://github.com/chainguard-images/images/tree/main/images/node

Image verification with Cosign

- “Cosign signs OCI containers using Sigstore”
- integrated with K8s policies Cosign allows validating the source of images
  - verifying third-party images
  - signing and validating your own images
- integrations are available with OPA Gatekeeper and Kyverno

white duck

Keyless Signing with Sigstore



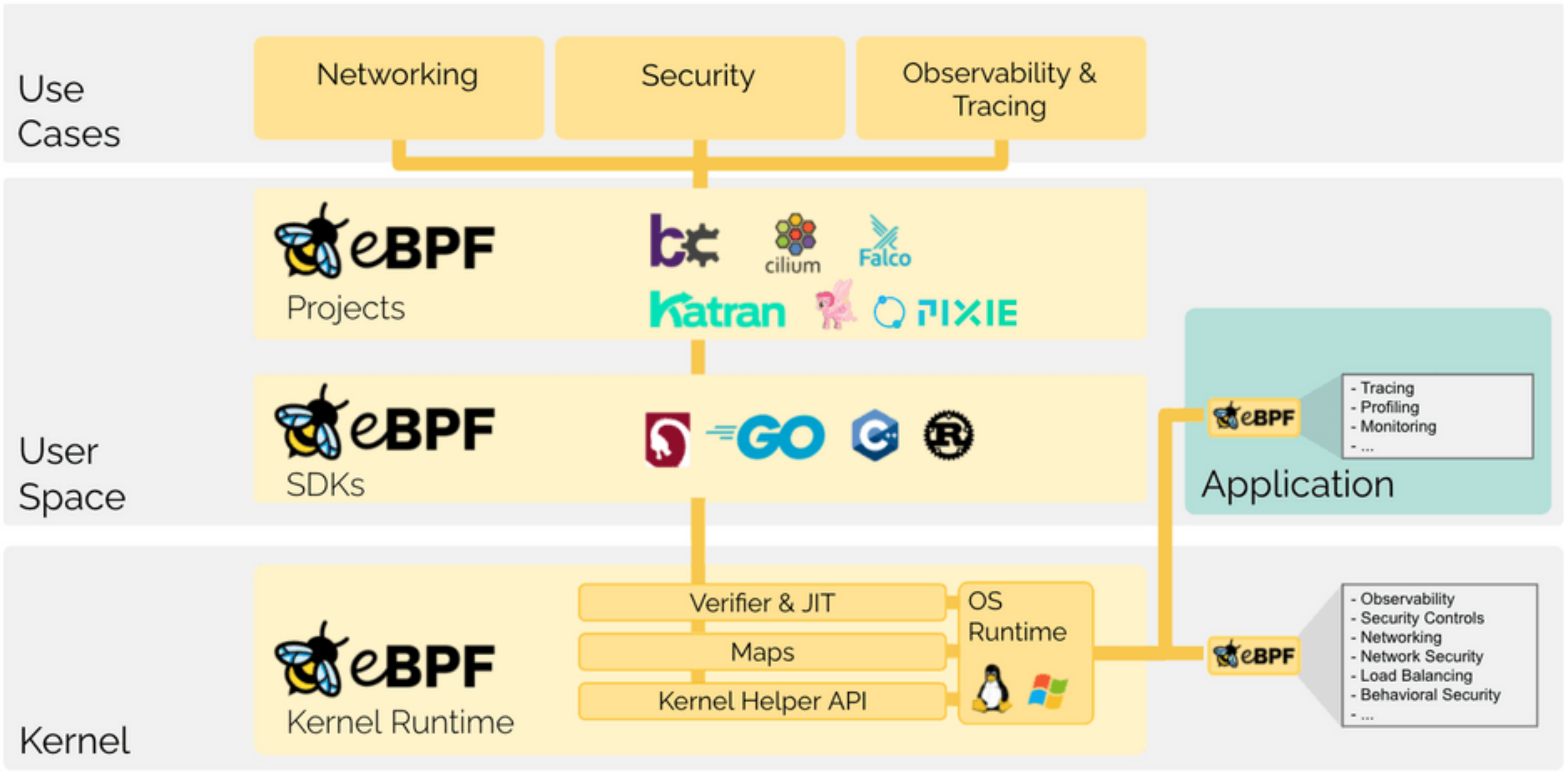
Demo: Cosign and Kyverno in Action

- we will deploy a policy
- then run signed images & sign our own
- more details
  - https://kyverno.io
  - https://github.com/sigstore/cosign
  - https://www.sigstore.dev

Container Runtime Security with Tetragon

- “eBPF-based Security Observability and Runtime Enforcement”
- gives you awareness into your cluster
  - without that you won't know what is going on
- alerts you on malicious events and workloads
- real-time enforcement

What is eBPF?


Use Cases
Networking
Security
Observability & Tracing
User Space
eBPF
Projects
cilium
Falco
Katran
PIXIE
eBPF
SDKs
GO
Application
eBPF
- Tracing
- Profiling
- Monitoring
- ...
Kernel
eBPF
Kernel Runtime
Verifier & JIT
Maps
Kernel Helper API
OS Runtime
eBPF
- Observability
- Security Controls
- Networking
- Network Security
- Load Balancing
- Behavioral Security
- ...

Demo: Tetragon in Action

- we will inject into a Pod via Log4Shell
- then observe the process execution and block it
- more details
  - https://github.com/cilium/tetragon

Questions?

- Slides
  - https://www.slideshare.net/nmeisenzahl
- Demo
  - https://github.com/nmeisenzahl/prevent-your-k8s-from-being-hacked

Email:	nico.meisenzahl@whiteduck.de
Twitter:	@nmeisenzahl
LinkedIn:	https://www.linkedin.com/in/nicomeisenzahl
Blog:	https://meisenzahl.org