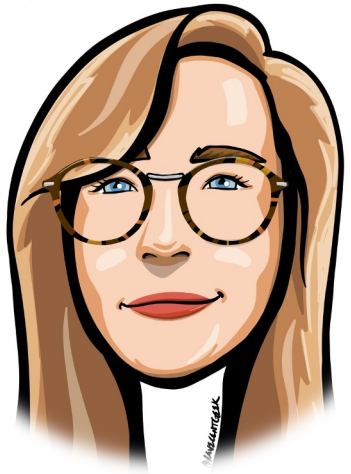
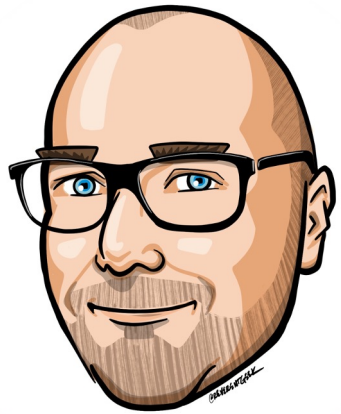
# State of Cloud Native Security in 2022

Annie Talvasto
Karl Ots



**Annie Talvasto**

Sr PMM @ Camunda
CNCF Ambassador, MVP



**Karl Ots**

Head of Cloud Sec @ EPAM
CISSP. MVP, RD

# Agenda

- Introduction
- Overview of Cloud Native Security
- Cloud Native Security (in the Enterprise)
- Takeaways & Resources

# Changes with Cloud Native Security

| | |
|---|---|
| Ephemeral workloads | • Event-driven and agentless monitoring |

# Changes with Cloud Native Security

| | |
|---|---|
| Ephemeral workloads | • Event-driven and agentless monitoring |
| Perimeter changes | • Identity-based perimeter<br>• Microsegmentation |

# Changes with Cloud Native Security

| | |
|---|---|
| Ephemeral workloads | • Event-driven and agentless monitoring |
| Perimeter changes | • Identity-based perimeter<br>• Microsegmentation |
| Growing share of OSS | • Supply chain lifecycle security |

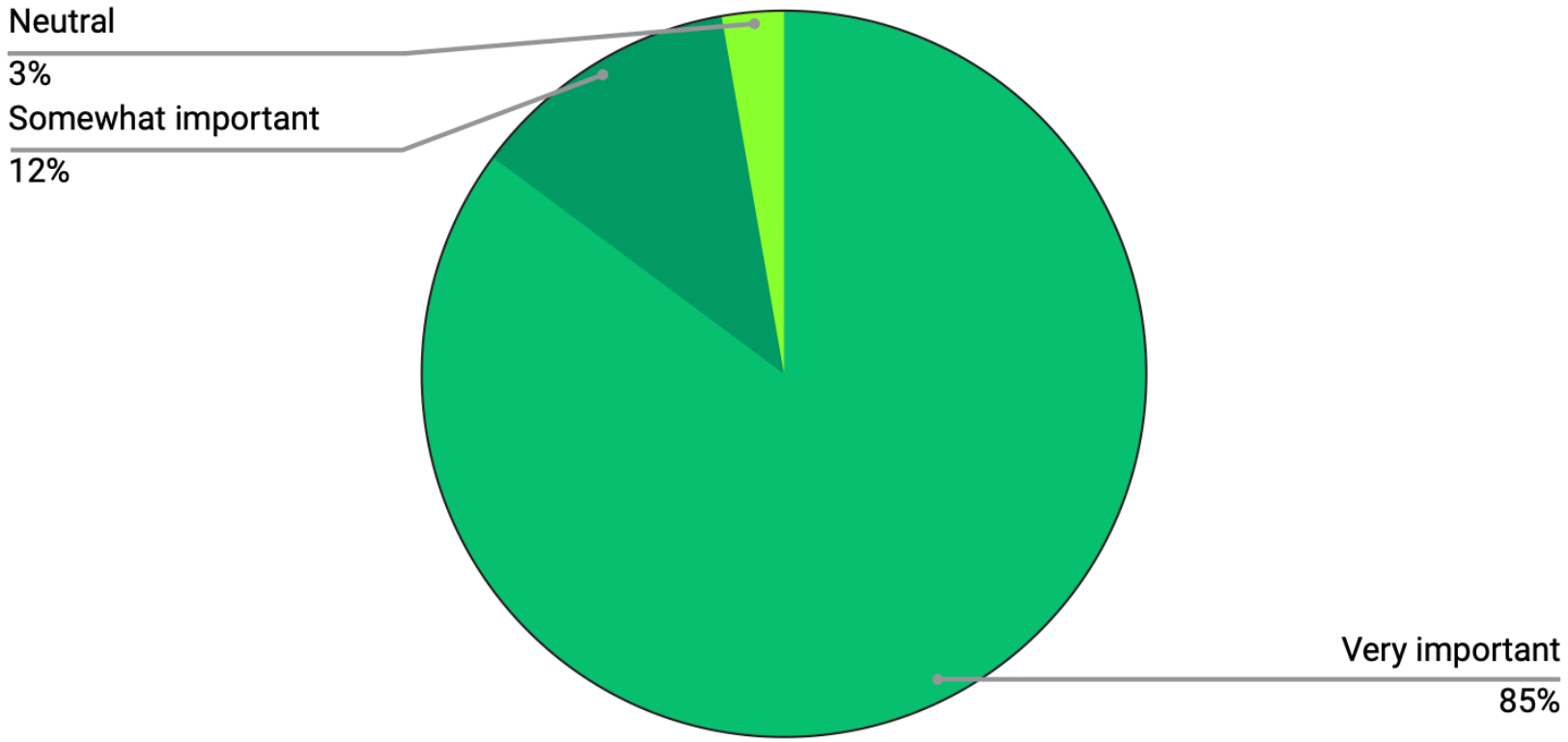
# CNCF Microsurvey on Security

## Cloud Native Security Microsurvey:

More than 80% of organizations want to build modern security systems with open source software

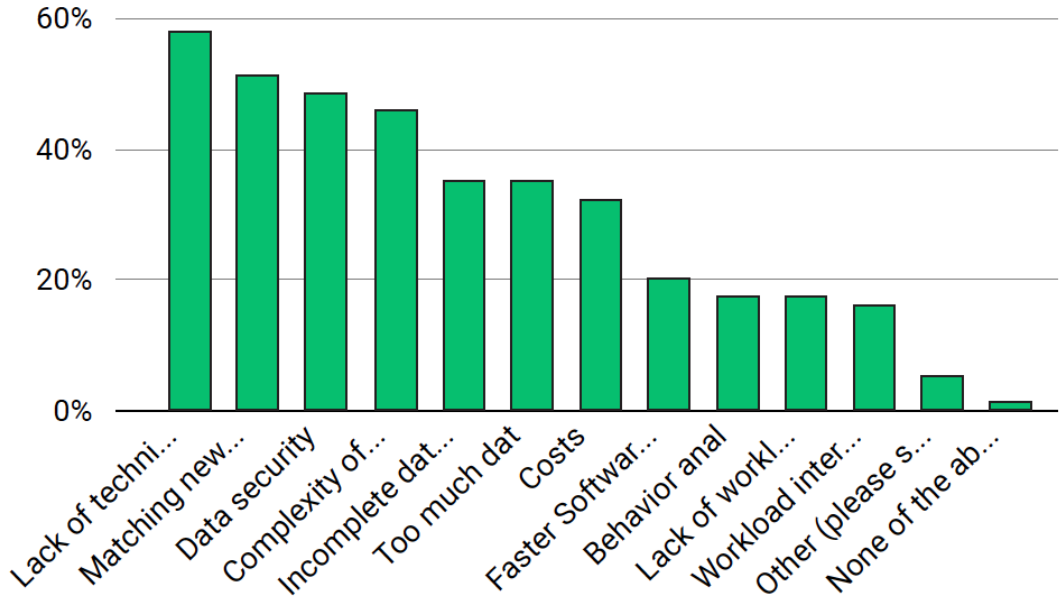
# Importance of Cloud Native Security

How important is modernizing security to your cloud native environments?



# Challenges in Cloud Native Security

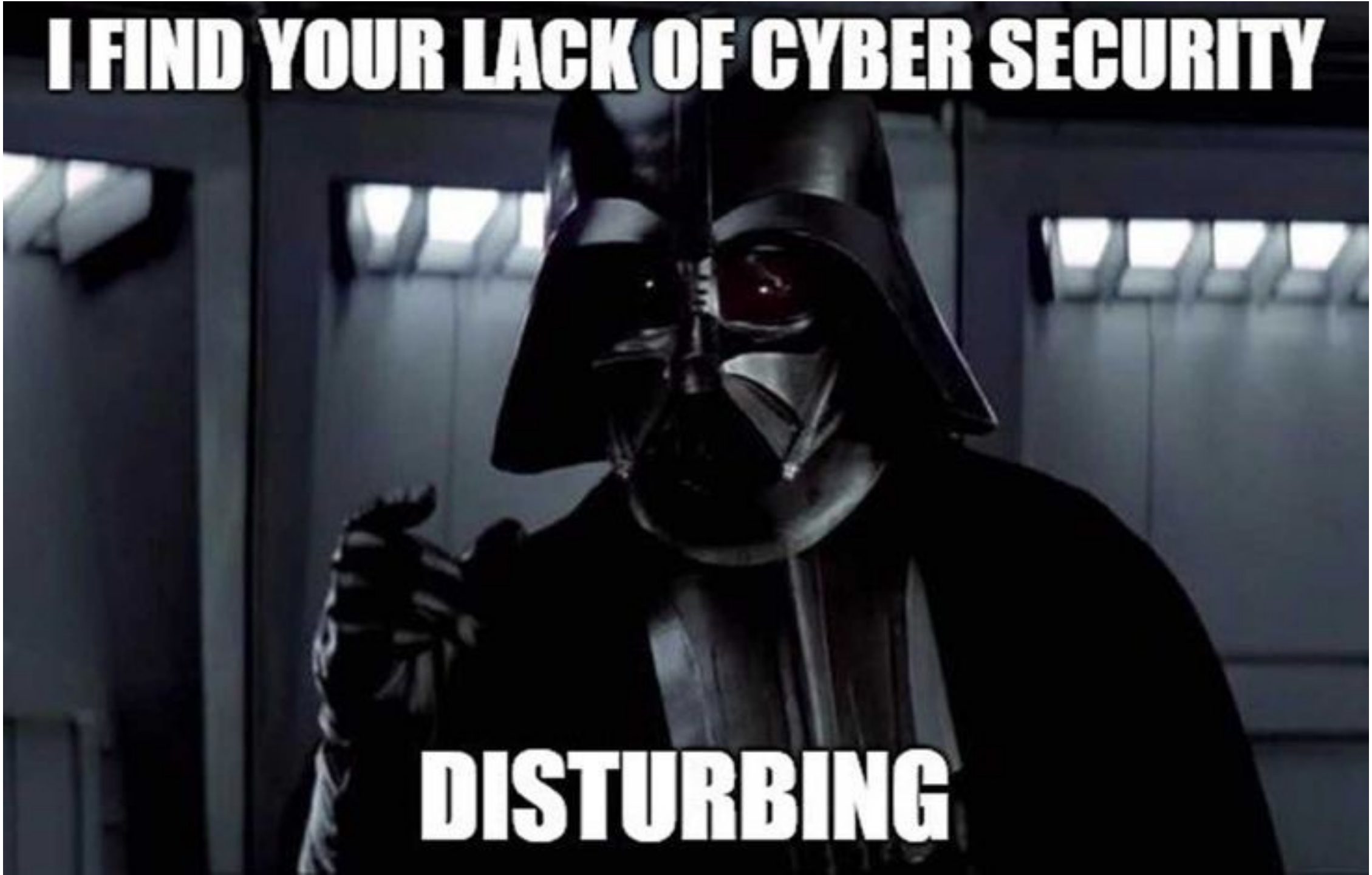
Which of the following challenges have you experienced in running cloud native environments? Please select all that apply.




- Lack of technical expertise 58%
- Matching new methods and processes like DevOps and CI/CD with existing requirements, tools, or processes 51%
- Data Security 49%
- Complexity of build, deploy, and management 46%
- Incomplete data for observing and managing applications and infrastructure 35%
- Too much data and/or inadequate tools for analysis and understanding 35%
- Costs 32%
- Faster software-delivery cycles 20%
- Behavior analysis (cryptomining, ransomware, botnet present) 18%
- Lack of workload authorization 18%
- Workload interoperability 16%
- Other (please specify) 5%
- None of the above 1%

# Challenges in Cloud Native Security

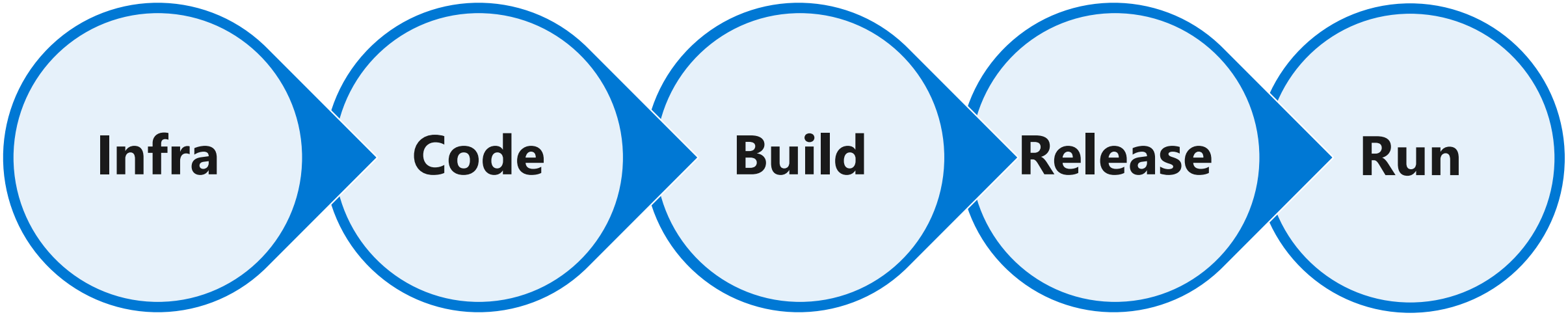
1. A lack of technical expertise – 58%
2. Trouble matching new methods and processes like DevOps and CI/CD with existing requirements, tools, or processes – 51%
3. Data security – 49 %
4. The complexity of building, deploying, and management – 46%


I FIND YOUR LACK OF CYBER SECURITY
DISTURBING

# Agenda

- Introduction
- Overview of Cloud Native security
- Cloud Native Security (in the Enterprise)
- Takeaways & Resources

# Security Across the Cloud Native Lifecycle




- Deploy the cluster using Infrastructure as Code
- **IAC SAST**

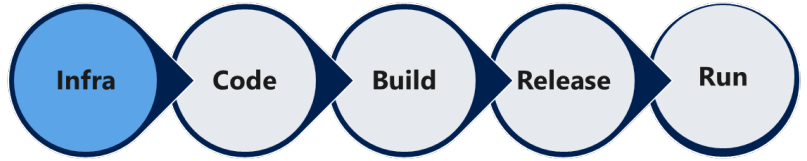
- Developer makes changes to code
- Developer checks in code to source control
- **Code scanning**

- Build application
- Build Dockerfile
- Push to Container Registry
- **Image scanning**

- Deploy the application into the cluster
- **Auditable and declarative CD**

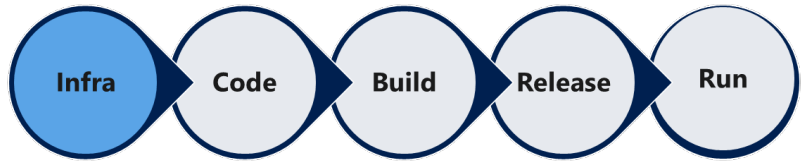
- Vulnerability management
- Operational security
- **Microsegmentation**
- **Policies and guardrails**
- **Runtime monitoring and detection**

# Security Testing IaC

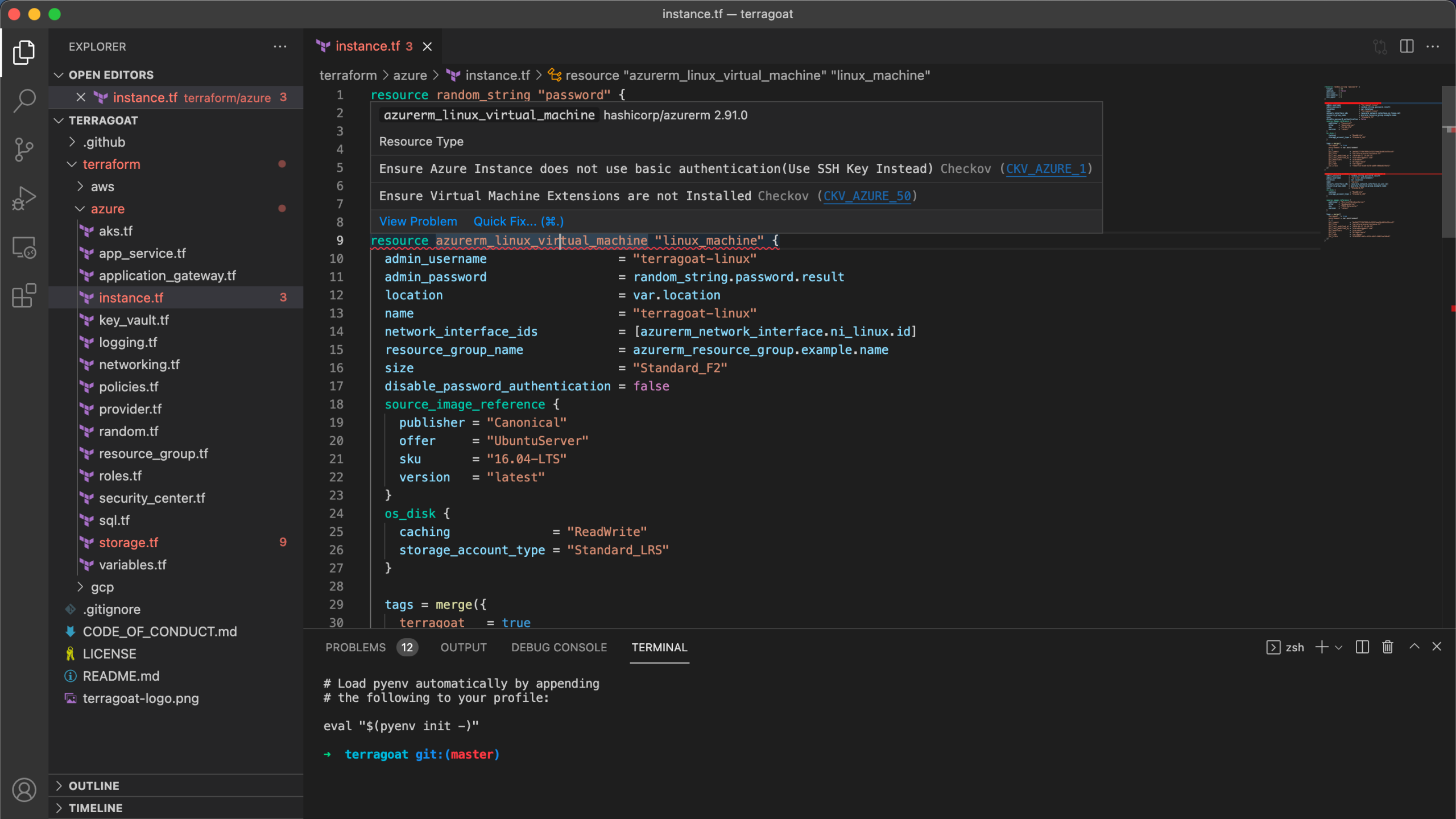



- Scan your IaC templates for vulnerabilities and security misconfigurations
- Run the scan continuously, as pre-deployment test or bring the rule set as recommendations in the IDE

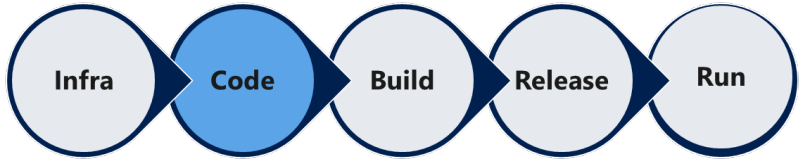
# Checkov



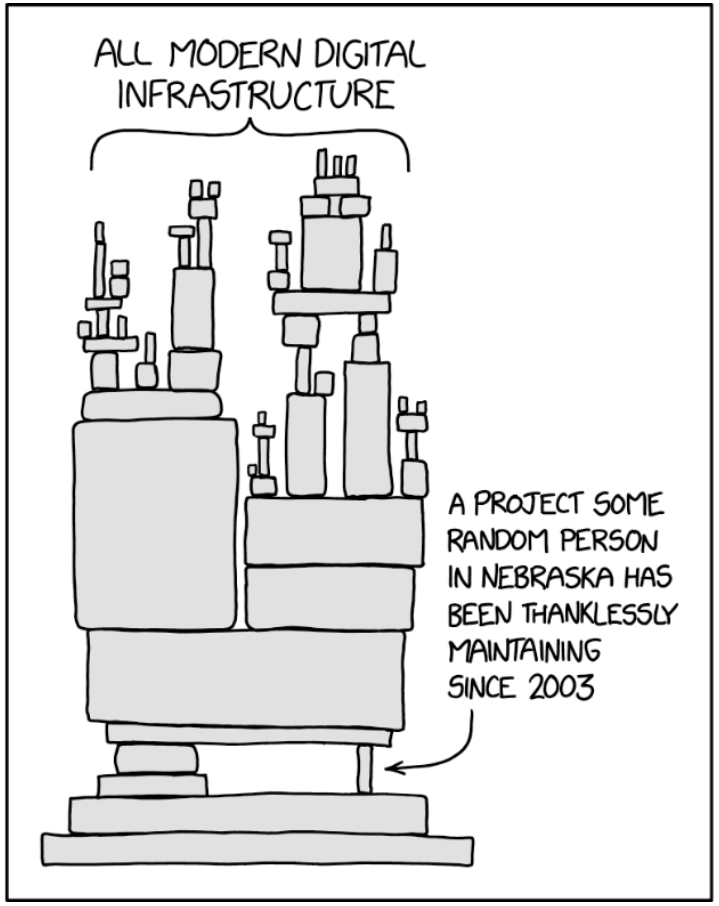

- Checkov scans cloud infrastructure configurations to find misconfigurations before they're deployed.
- Open source, commercial service available from BridgeCrew (part of Palo Alto)
- Provides common command line interface to manage IaC, and can scan across
  - Terraform
  - CloudFormation
  - Kubernetes
  - Helm
  - ARM templates
  - Serverless framework


instance.tf — terragoat
EXPLORER
OPEN EDITORS
instance.tf terraform/azure 3
TERRAGOAT
.github
terraform
aws
azure
aks.tf
app_service.tf
application_gateway.tf
instance.tf 3
key_vault.tf
logging.tf
networking.tf
policies.tf
provider.tf
random.tf
resource_group.tf
roles.tf
security_center.tf
sql.tf
storage.tf 9
variables.tf
gcp
.gitignore
CODE_OF_CONDUCT.md
LICENSE
README.md
terragoat-logo.png
OUTLINE
TIMELINE
instance.tf 3
terraform > azure > instance.tf > resource "azurerm_linux_virtual_machine" "linux_machine"
resource random_string "password" {
azurerm_linux_virtual_machine hashicorp/azurerm 2.91.0
Resource Type
Ensure Azure Instance does not use basic authentication(Use SSH Key Instead) Checkov (CKV_AZURE_1)
Ensure Virtual Machine Extensions are not Installed Checkov (CKV_AZURE_50)
View Problem Quick Fix... (⌘.)
resource azurerm_linux_virtual_machine "linux_machine" {
admin_username = "terragoat-linux"
admin_password = random_string.password.result
location = var.location
name = "terragoat-linux"
network_interface_ids = [azurerm_network_interface.ni_linux.id]
resource_group_name = azurerm_resource_group.example.name
size = "Standard_F2"
disable_password_authentication = false
source_image_reference {
publisher = "Canonical"
offer = "UbuntuServer"
sku = "16.04-LTS"
version = "latest"
}
os_disk {
caching = "ReadWrite"
storage_account_type = "Standard_LRS"
}
tags = merge({
terragoat = true
PROBLEMS 12
OUTPUT
DEBUG CONSOLE
TERMINAL
zsh
# Load pyenv automatically by appending
# the following to your profile:
eval "$(pyenv init -)"
→ terragoat git:(master)

# Code Scanning

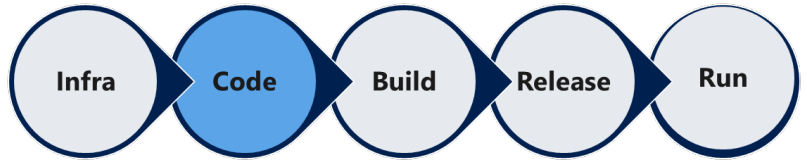



- Complexity of polyglot applications requires going beyond static security testing, to also include composition analysis
- Scanners produce Software Bill of Materials, and map dependencies with OSS risk



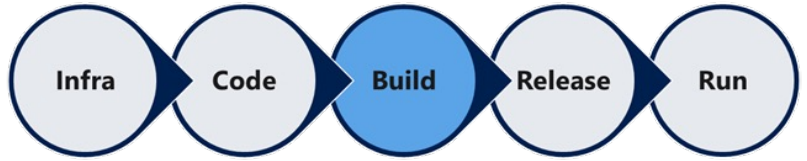

XKCD 2347

# SonarQube



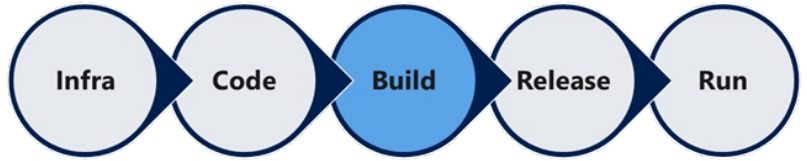

- Code scanning **enables vulnerabilities to be detected and remediated prior to release into production**, eliminating the cybersecurity risks that they pose.
- Enterprisey features include to systematic tracking of Code Quality and Code Security

# Container Image Scanning




- Do you **really** know all the dependencies of all the base images of the containers that you are running?
- Untrusted container images violate the integrity of your Kubernetes cluster
- Scan the images against security vulnerabilities before pushing to registry

# Docker Scan




- Native vulnerability scanner for Docker images

- Based on Snyk engine
- Quickly detect and learn how to remediate CVEs in your images by running docker scan IMAGE_NAME.
- Scan results:
  - Common Vulnerabilities and Exposures (CVEs)
  - Sources, such as OS packages and libraries
  - Versions in which they were introduced
  - Recommended fixed version

# Auditable and Declarative CD


- Sometimes referred to as GitOps
- Define your entire cloud native stack as code
  - Application
  - Configuration
  - Kubernetes environment
- Apply secure software development practices across the whole stack
  - Auditable
  - Repeatable

# ArgoCD


- Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.
- Application definitions, configurations, and environments should be declarative and version controlled.
- Kubernetes manifests can be specified in several ways:
  - kustomize applications
  - helm charts
  - jsonnet files
  - Plain directory of YAML/json manifests

# Runtime Security Monitoring


- Monitoring and detection of attacks in immutable and ephemeral environment is difficult
- You need to tap into new sources of logs and events
- You need real-time alerting

# Falco


- Cloud-Native Runtime Security
- **Falco**, the cloud-native runtime security project, is the de facto **Kubernetes threat detection engine.**
- Detects threats at runtime by observing the behaviour of your applications and containers.
- Extends threat detection across cloud environments with Falco Plugins.

# What does Falco check?


- Privilege escalation using privileged containers
- Namespace changes
- Read/Writes to well-known directories
- Executing shell binaries
- Executing SSH binaries such as ssh, scp, sftp, etc

# Cluster Security


- Make sure container images deployed to Kubernetes are from a trustworthy source
- Do not use the default cluster namespace to deploy applications
- Do not run your container images as root
- Set cluster-wide guardrails using **policy as code**

# Kyverno


- Kyverno is a policy engine designed for Kubernetes.
- With Kyverno, policies are managed as Kubernetes resources and no new language is required to write policies.
- This allows using familiar tools:
  - kubectl
  - Git
  - kustomize

# OPA – Open Policy Agent


- Policy-based control for cloud native environments
- The Open Policy Agent (OPA, pronounced “oh-pa”) is an open source, general-purpose policy engine that unifies policy enforcement across the stack.
- OPA provides a high-level declarative language that lets you specify policy as code and simple APIs to offload policy decision-making from your software.
- You can use OPA to enforce policies in microservices, Kubernetes, CI/CD pipelines, API gateways, and more.

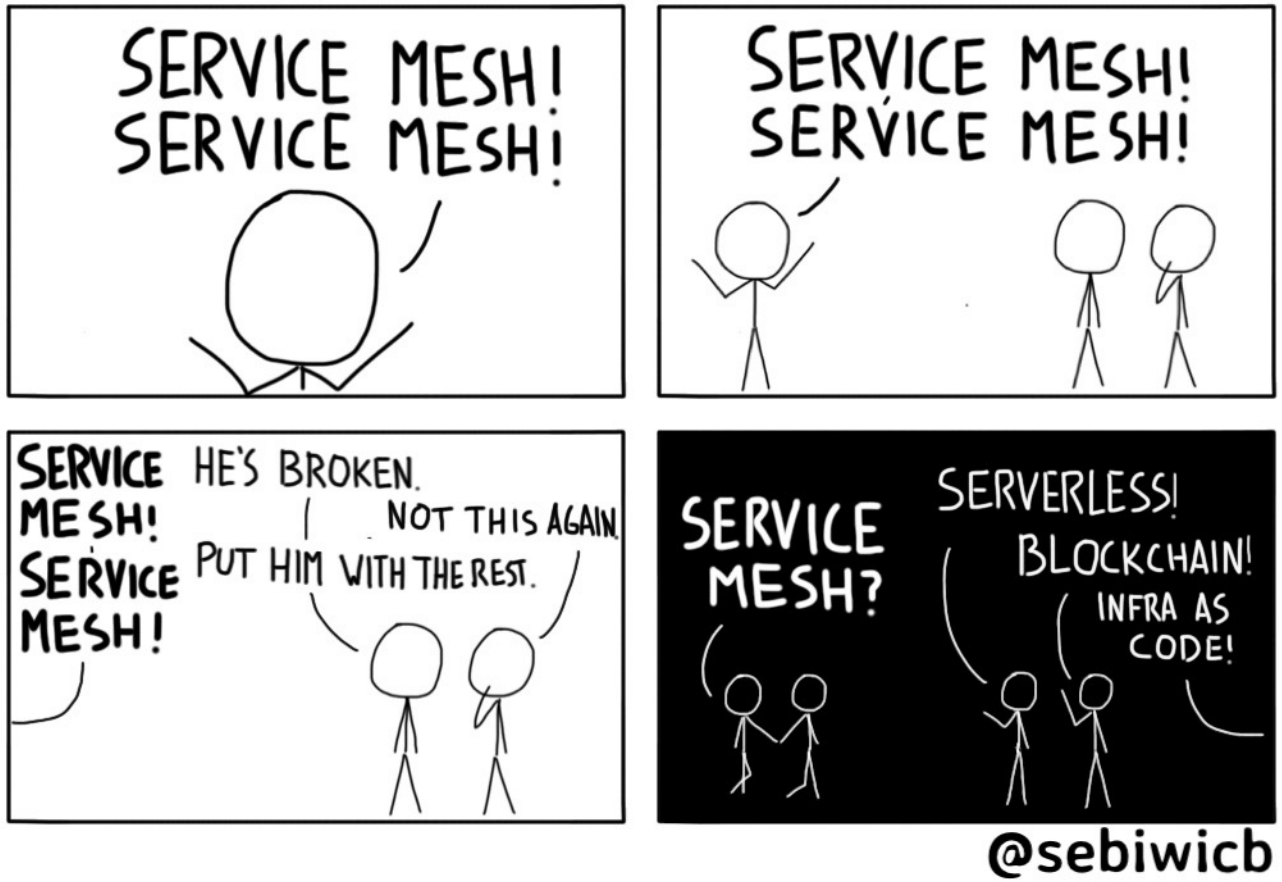
# Microsegmentation


- Control the flow of traffic **between pods** in the cluster
- **Network policies** are translated into sets of allowed and disallowed IP pairs
- Kubernetes implements these pairs as IPTable rules

```yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: backend-policy
  namespace: development
spec:
  podSelector:
    matchLabels:
      app: webapp
      role: backend
  ingress:
  - from:
    - namespaceSelector: {}
      podSelector:
        matchLabels:
          app: webapp
          role: frontend
```

# Microsegmentation: Service Meshes


- Manage ingress and egress
- Control cross-pod, cross-namespace, cross-node and cross-cluster networking
- Sidecar or agentless
- Projects: Linkerd, Istio, Open Service Mesh



# Linkerd


- Community, simple design
- Ultralight, ultrafast, security-first service mesh for Kubernetes.
- The overall goal is to reduce overhead of having a service mesh
- What does it do?
  - Observability
  - Realiability
  - Security

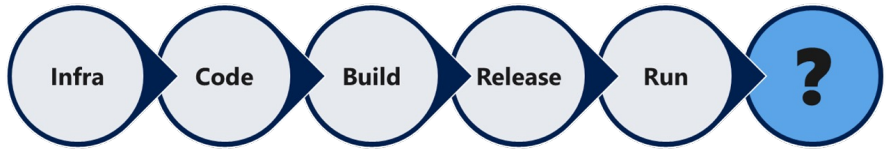
# KubeAudit

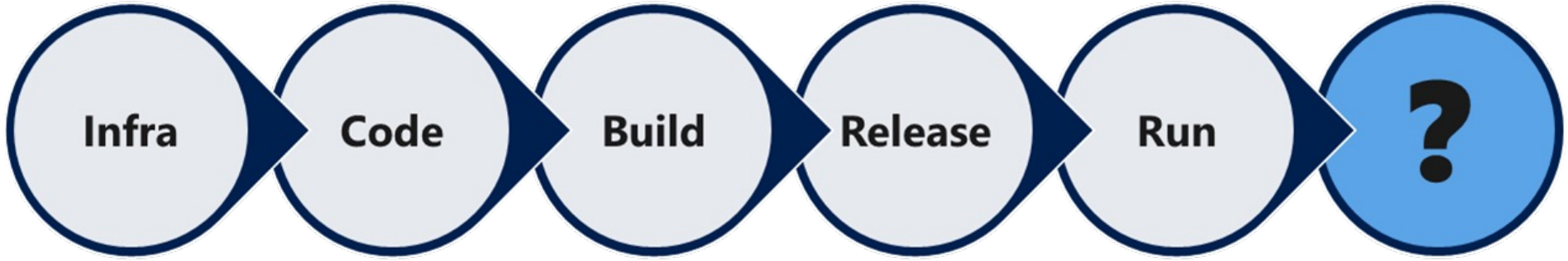

- kubeaudit is a command line tool and a Go package to audit Kubernetes clusters for various different security concerns, such as:
  - run as non-root
  - use a read-only root filesystem
  - drop scary capabilities, don't add new ones
  - don't run privileged

- tldr. kubeaudit makes sure you deploy secure containers

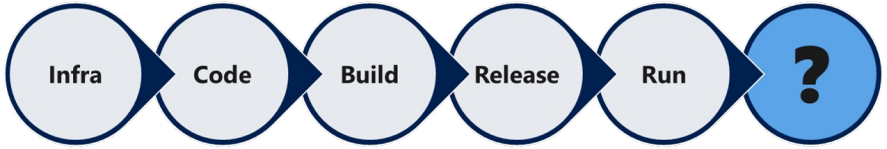
- Quick Demo: KubeAudit in action

# Future & Predictions



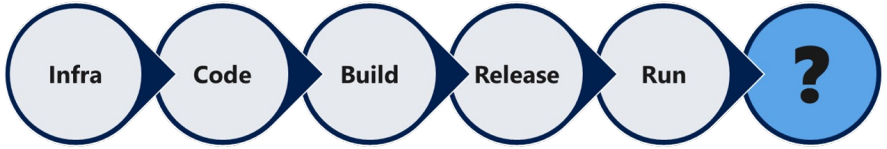
# Future & Predictions

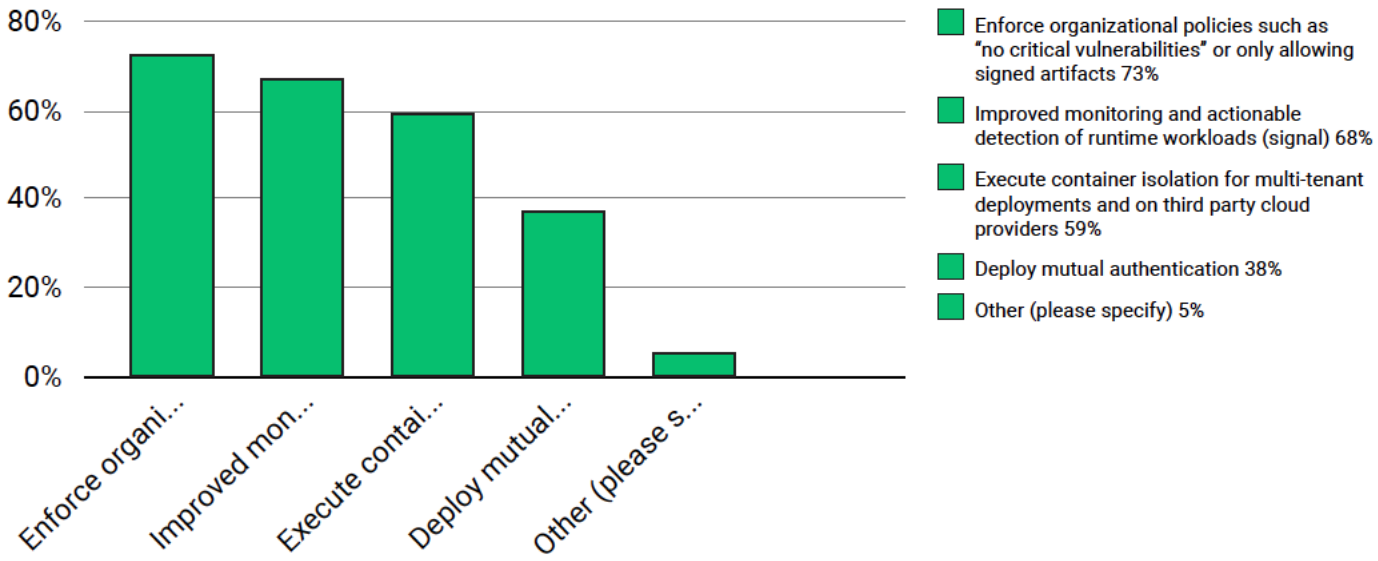



- Integration with the cloud

- Kubernetes will get closer to PaaS because cloud providers have better security defaults

# Security Investments



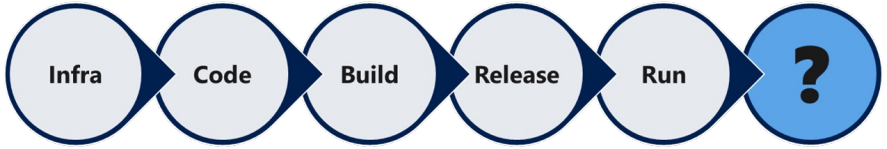




- Enforce organizational policies such as "no critical vulnerabilities" or only allowing signed artifacts 73%
- Improved monitoring and actionable detection of runtime workloads (signal) 68%
- Execute container isolation for multi-tenant deployments and on third party cloud providers 59%
- Deploy mutual authentication 38%
- Other (please specify) 5%

Where do you intend to focus your efforts for greater security in cloud nativeover the next 2-5 years?

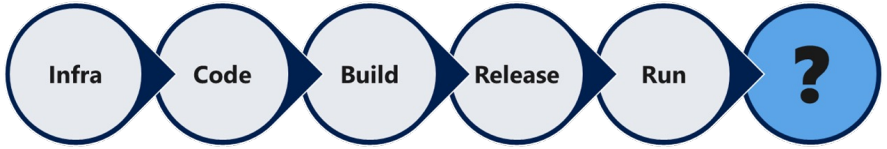
# Challenges in Cloud Native Security

1. A lack of technical expertise – 58%
2. Trouble matching new methods and processes like DevOps and CI/CD with existing requirements, tools, or processes – 51%
3. Data security – 49 %
4. The complexity of building, deploying, and management – 46%




# State of Cloud Native Security Report

# Cloud Native Security Report



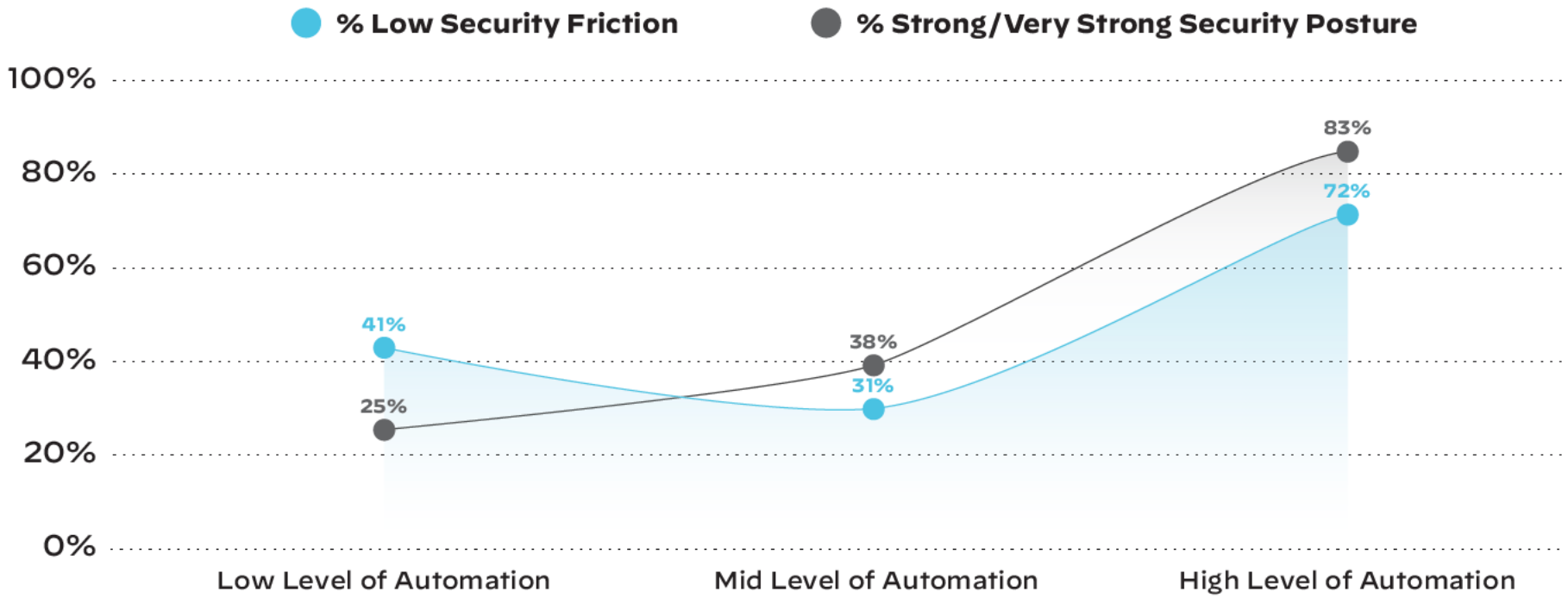
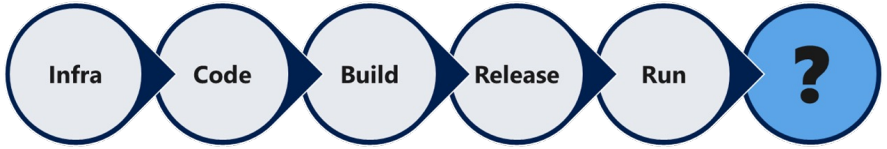




**Figure 16:** Automation as a driver of security outcomes

# Cloud Native Security Report



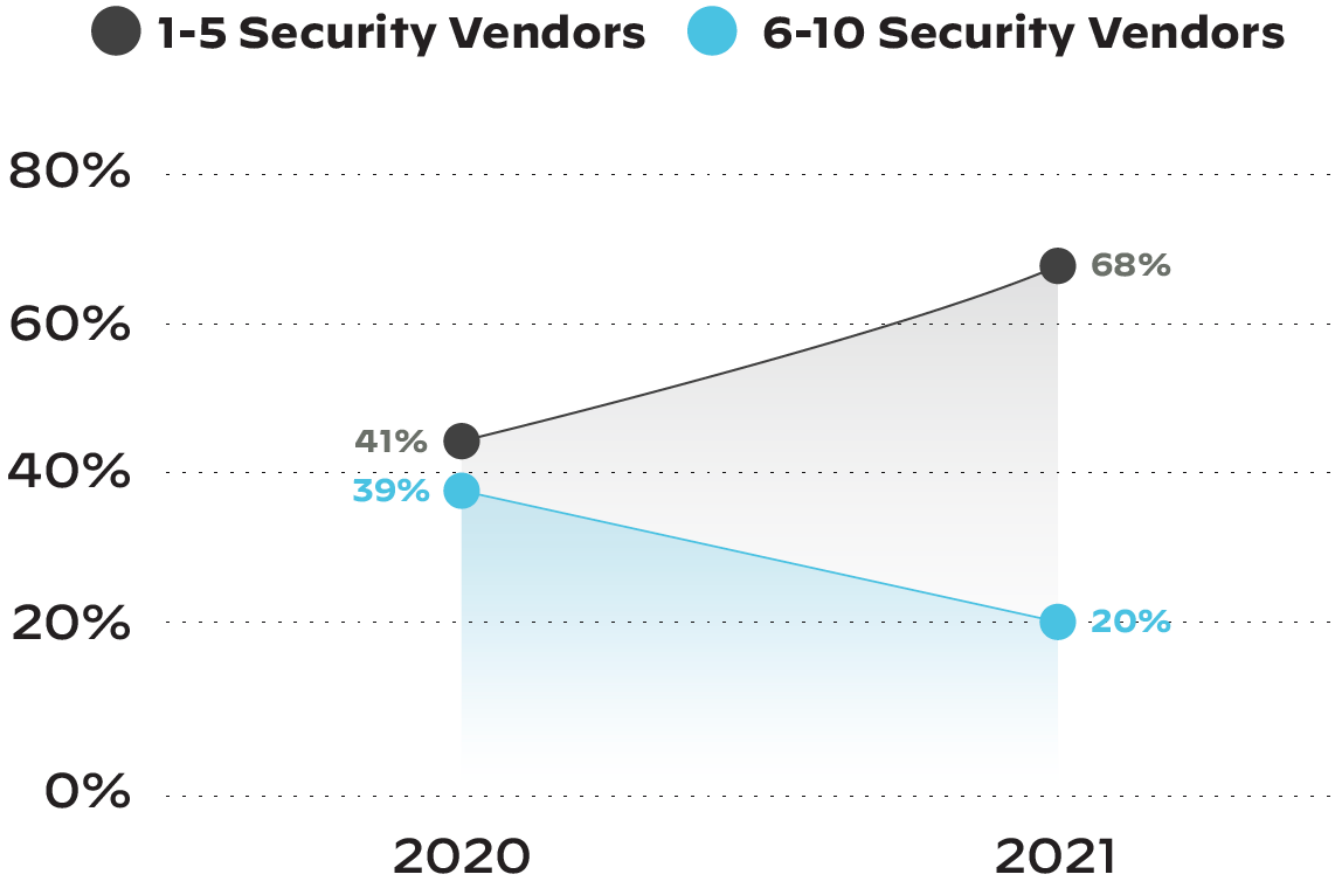
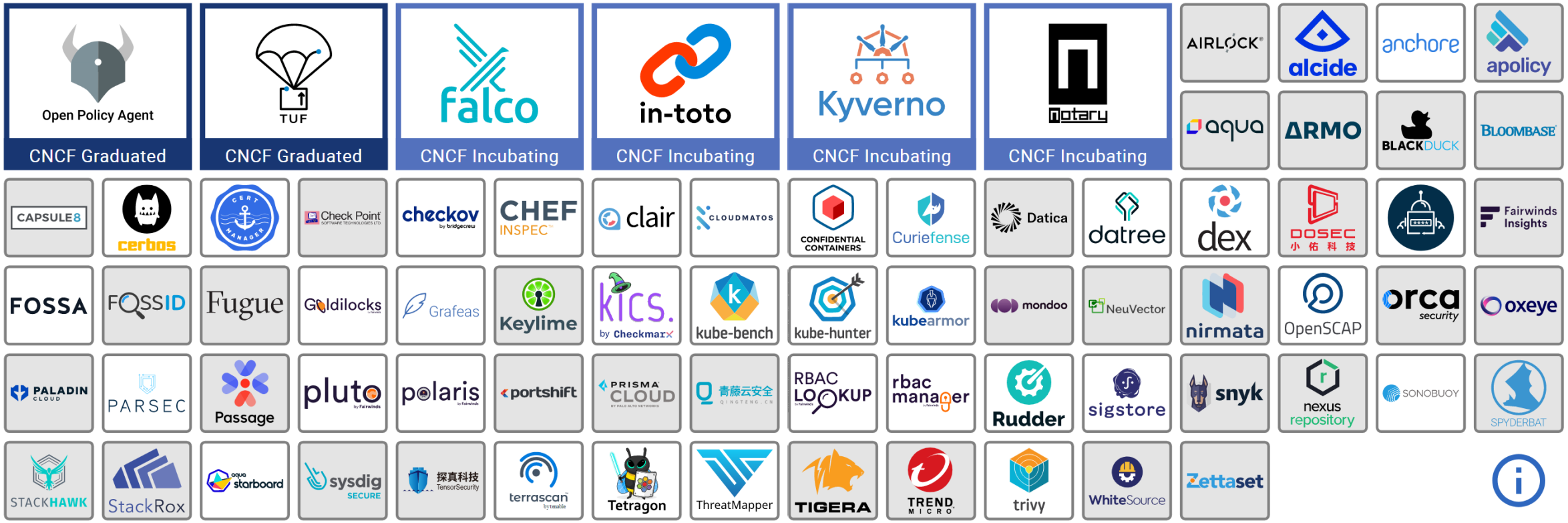




**Figure 8:** Changes to number of security vendors

# Agenda

- Introduction
- State of Cloud Native Security
- Cloud Native Security (in the Enterprise)
- Takeaways & Resources

# CNCF Landscape



# CNCF Resources

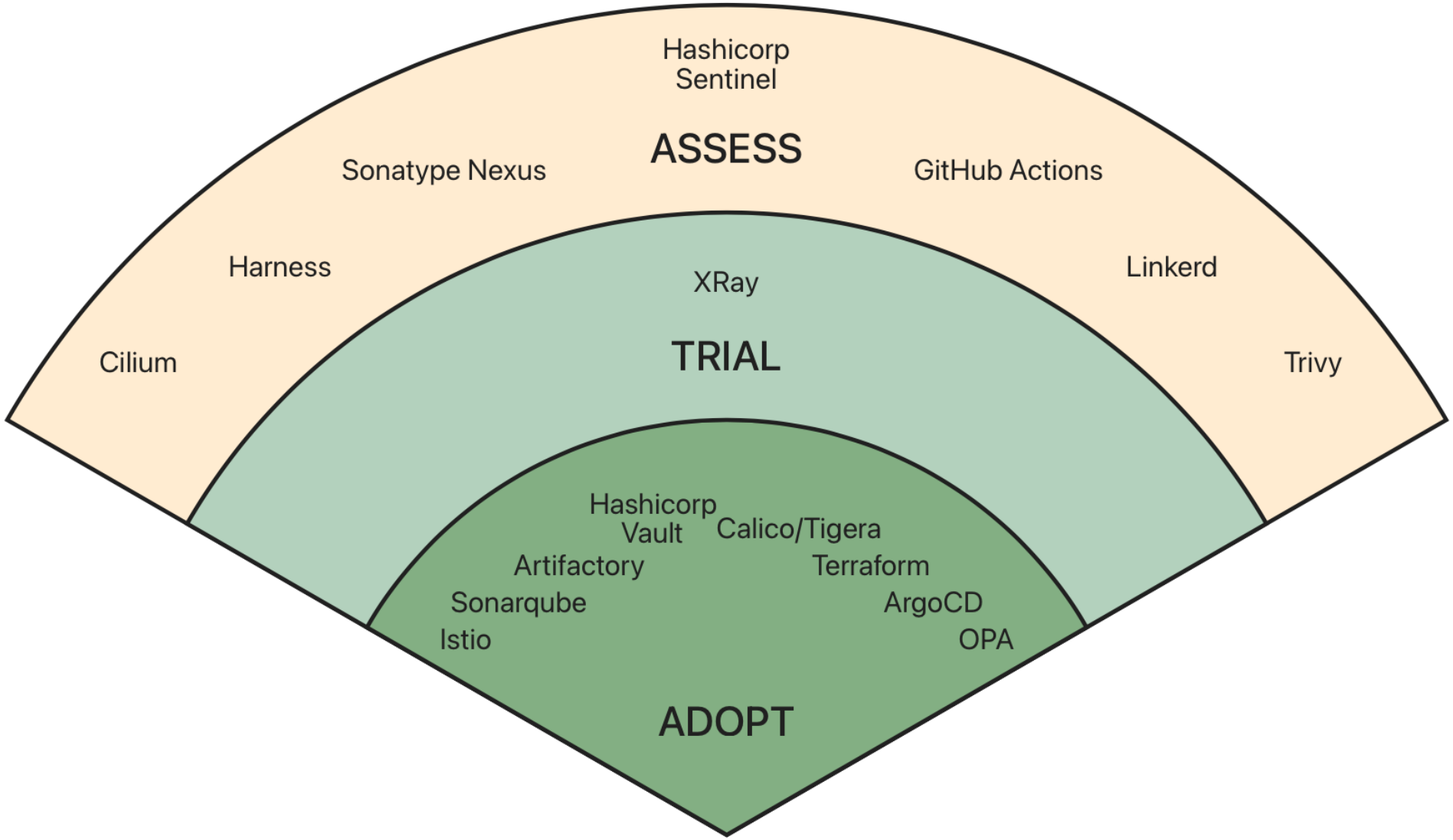
## Security TAG

- Advises other Kubernetes projects on security
- Coordinates the recurring pentest

## CNCF DevSecOps radar

## CNCF Microsurvey on security

# DevSecOps, September 2021




Download as svg or png

# CNCF Resources

## Security TAG

- Advises other Kubernetes projects on security
- Coordinates the recurring pentest

## CNCF DevSecOps radar

## CNCF Microsurvey on security

# CNCF Microsurvey on Security

## Cloud Native Security Microsurvey:

More than 80% of organizations want to build modern security systems with open source software

# Tools

- IAC SAST
  - Checkov
  - Tenable
  - KICS
- Scanning tools
  - docker scan
  - Kubeaudit
- Runtime security
  - Kubescape
  - Falco

- Materials
  - CIS Security Benchmark for Kubernetes
  - NSA Kubernetes hardening guide
  - Threat matrix for Kubernetes
  - Kubernetes Goat

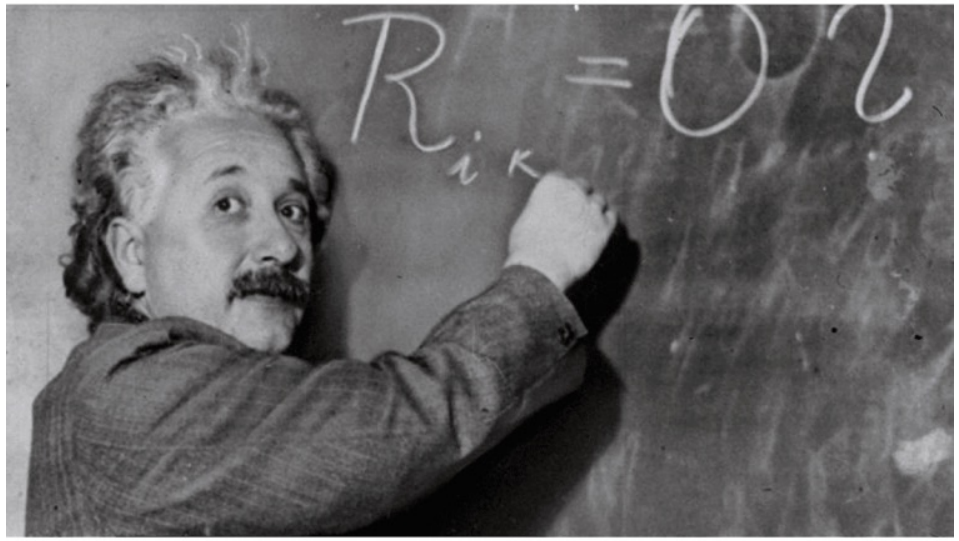
- Slides:
  - github.com/AnnieTalvasto/presentations

# Takeaways

- Compared to PaaS, Kubernetes allows for more security controls to be put in place
  - This comes with more responsibilities!
- Every application is different
  - You might not need all (or any) of the security controls listed in this session
- Cloud native security is continuously evolving
  - Check the backlog(s) and challenge your (perceived) security risks and controls

# Thank you!

How I think I look explaining cyber risk to the board



How I actually look