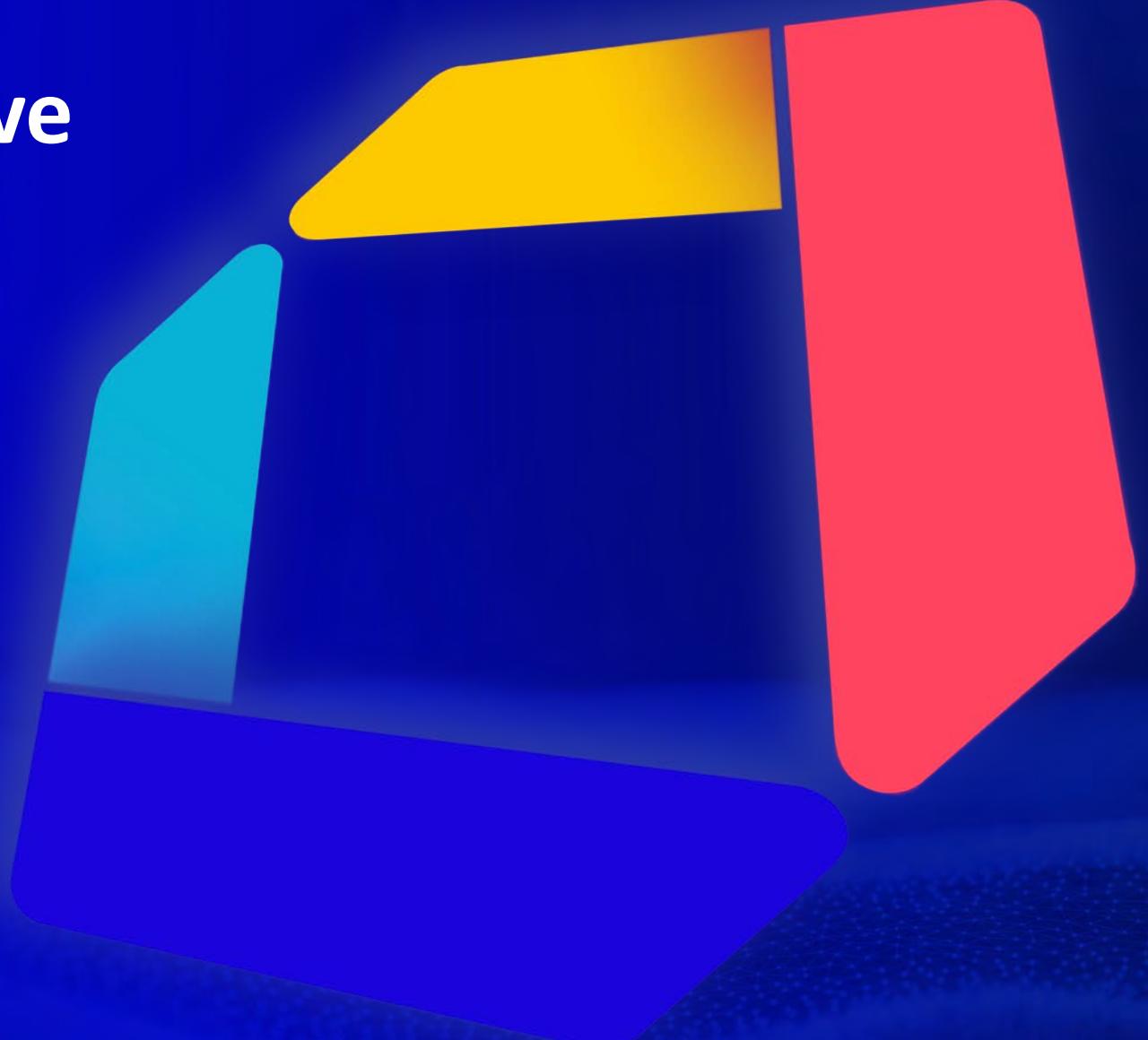


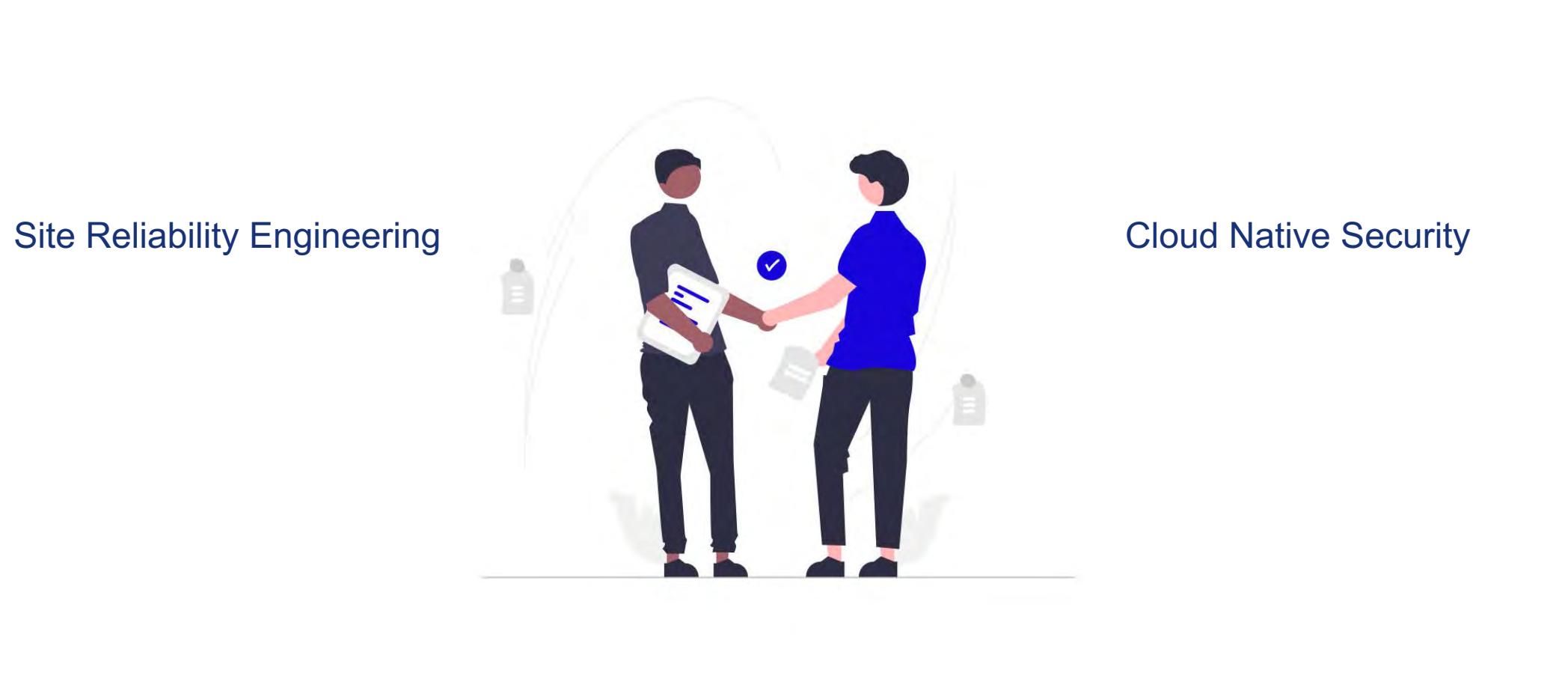


# Integrating Cloud Native Security into the SRE culture

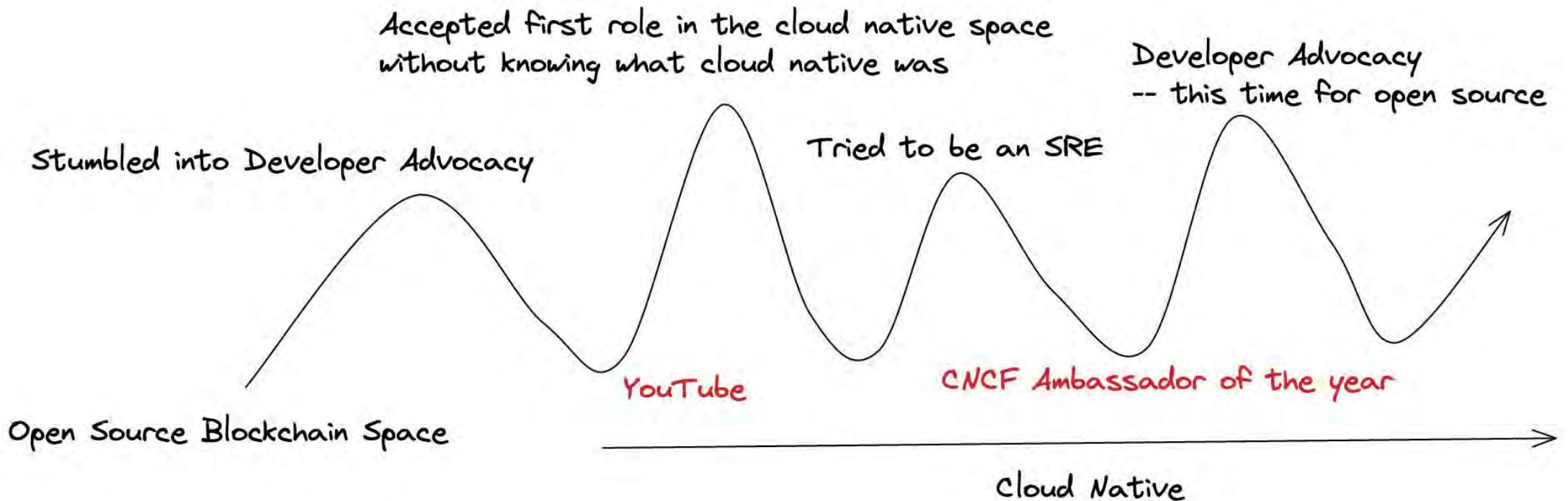
Anais Urlichs



# Topic



# Who am I?



# YouTube

The screenshot shows the YouTube channel page for "Anais Urlichs". The channel has 11.5K subscribers. The banner features a blue hexagonal icon with a white Kubernetes logo and the text "New Videos Weekly" and "#100DaysOfKubernetes". A video thumbnail for "Kyverno Overview – Defining Kubernetes Cluster Policies" is displayed, along with its description and resources. Below the banner, there are navigation links for HOME, VIDEOS, PLAYLISTS, COMMUNITY, CHANNELS, ABOUT, and sections for TRENDING and STATS. The main content area shows several video thumbnails from the channel's uploads, including topics like "Creating & using addons", "The best learning platform with FREE courses", "Kyverno Overview – Defining Kubernetes Cluster Policies", "How to become a Developer Advocate & My career path", "A deep dive into Helm Dependencies", and "lazytriv: Scan all your container images with one...". On the right side, social media links for Twitter (@urlichsanais) and YouTube (AnaisUrlichs) are shown, along with a link to anaisurl.com.

# Weekly DevOps Newsletter

## DevOps

DevOps Diary Weekly Newsletter and DevOps related blog posts

10 OCT 2022      **#69 Seven-Day DevOps — Weekly DevOps Newsletter**  
PUBLIC

Weekly Newsletter with updates, tutorials, videos, fun memes and more!



5 OCT 2022      **Writing a microk8s addon on Mac**  
PUBLIC

In this blog post, I first describe the steps I took to install microk8s and then to develop and test a microk8s addon for the Trivy...



22 SEP 2022      **#68 Seven-Day DevOps — Weekly DevOps Newsletter**  
PUBLIC

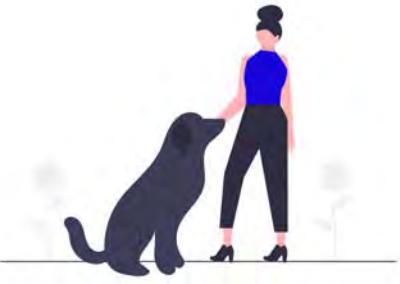
Some exciting updates, great content incl. tutorials, talks and more events. Enjoy!



8 SEP 2022      **#67 Seven-Day DevOps — Weekly DevOps Newsletter**  
PUBLIC

This week's newsletter is featuring lots of content on AWS, including tutorials, tweet-threads and podcasts – among other excitin...





# Supercluster design: compute



KubeCon



CloudNativeCon

North America 2021

Regional Hyper Converged Infrastructure



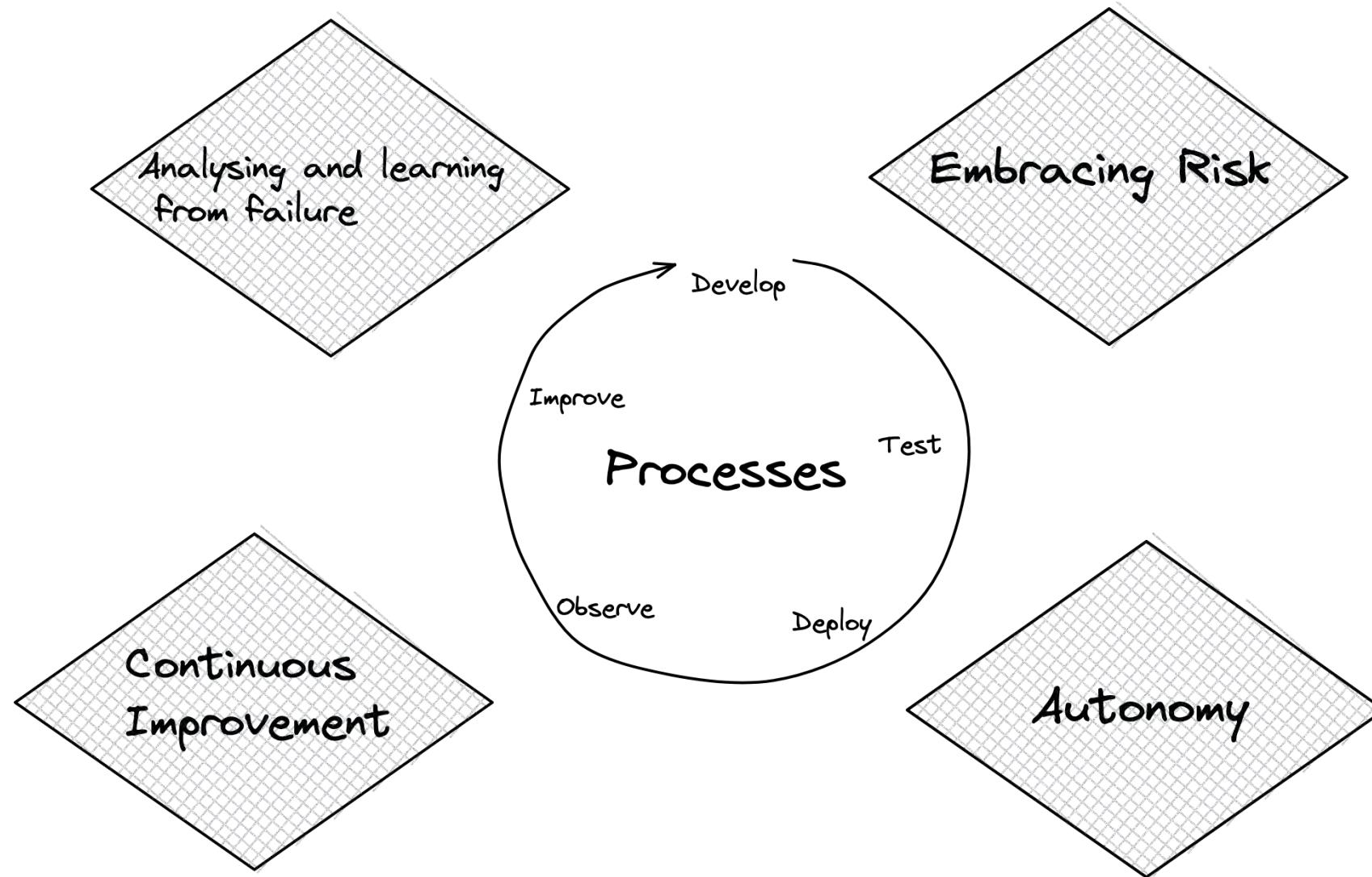
2xIntel Gold Xeon 20 core CPU  
384GB 2666Mhz DRAM  
100 Gb Networking



**OPEN**  
Compute Project®

@urlichsanais

# SRE Culture – What is it?



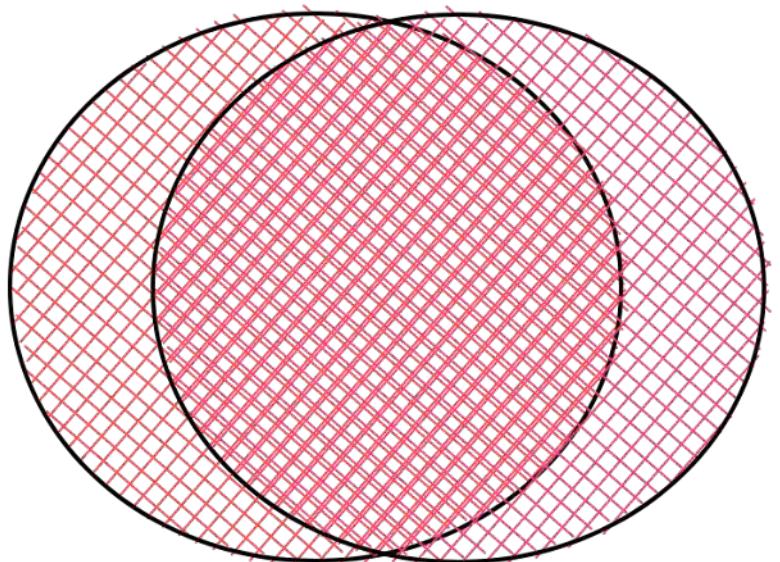
# DevSecOps/Cloud Native Security

## DevSecOps

Incorporating security into all other business functions  
by empowering people and creating accountability

# The premise

Security Practices  
&  
SRE Practices

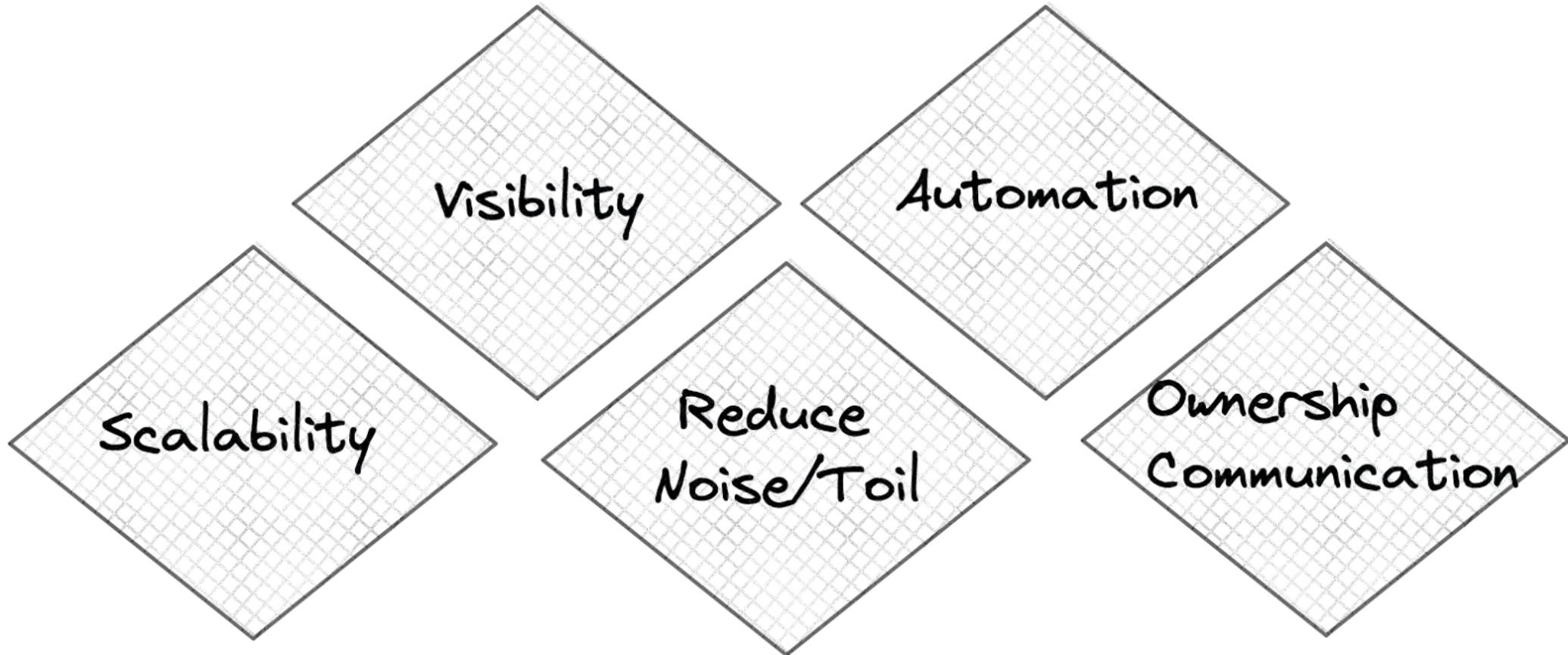


Healthy Services are Secure Services



Define what healthy services look like

# SRE Goals are Security Goals



## Practices

- Investing in runbooks and documentation
- Developing a robust delivery pipeline
- Define ownership
- Set explicit expectations on people
- Active communication with engineers
- Define procedures/processes
- Define use and expectations on tools

# Tools

## Observability



Grafana loki

## Management



# Where are the security tools?

# 10 Steps to integrating cloud native security





aqua  
**trivy**

## Scan Targets

Git Repository & Filesystem Scanning

Container Image Scanning

Dockerfile Scanning

IaC Scanning

Kubernetes Manifest Scanning

SBOM Generation & Scanning

AWS Account Scanning

In-cluster Scanning

# Step 1 Understanding your need

# Factors



Size of your team



Industry you are working in



Type of technologies you are working with



Company Goals and leadership



Budget and expertise



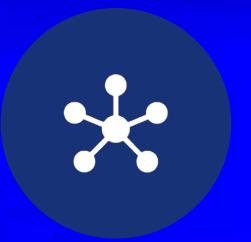
INSTALLATION



KUBERNETES  
RESOURCE TYPE



SCAN COVERAGE



INTEGRATION



FOCUS

# Needs – Based on Wise Engineering Blog post

- 1. Assign ownership of vulnerabilities**
- 2. Global view of the security state of services**
- 3. Develop Dashboards for different users and requirements**
- 4. Overcome difficult to use different UIs**

Note that these needs have been rewritten based on the following blog post <https://medium.com/wise-engineering/our-application-security-journey-part-1-fb7d449a7126>

# Step 2 Choosing a cloud native Security Scanner

## Open Source Security Scanning -- focus on cloud native

|  |  |  |   |
|--|--|--|---|
| <p><b>Vulnerability Scans</b></p> <ul style="list-style-type: none"> <li>Trivy</li> <li>clair</li> <li>Grype</li> <li>KubeClarity</li> </ul> | <p><b>IaC Misconfiguration Scans</b></p> <ul style="list-style-type: none"> <li>Trivy</li> <li>tfsec</li> <li>terrascan</li> <li>KICS</li> <li>checkov</li> <li>Conftest</li> <li>Kubescape</li> <li>regula</li> </ul> | <p><b>Policy Checks &amp; Schema validation*</b></p> <ul style="list-style-type: none"> <li>Trivy</li> <li>CloudQuery</li> <li>Cloud Custodian</li> </ul> <p>*a little vague since multiple tools that implement Rego etc. allow for policy checks</p> | <p><b>Compliance Scans</b></p> <ul style="list-style-type: none"> <li>kube-bench CIS</li> <li>kube-beacon CIS</li> <li>chain-bench CIS</li> <li>Starboard NSA</li> <li>Kubescape NSA</li> </ul> |
| <p><b>SBOM</b></p> <ul style="list-style-type: none"> <li>Trivy</li> <li>Syft</li> <li>CloudQuery</li> <li>KubeClarity</li> </ul>            | <p><b>In-Cluster Scans</b></p> <ul style="list-style-type: none"> <li>Trivy</li> <li>Kubescape</li> </ul>  | <p><b>Kubernetes Pentesting</b></p> <ul style="list-style-type: none"> <li>kube-hunter</li> <li>kdigger</li> <li>StackRox</li> </ul>   | <p><b>Secret Scanning</b></p> <ul style="list-style-type: none"> <li>tfsec (terraform)</li> <li>Trivy (K8s)</li> </ul> <p>many-more non cloud native</p>  |



## Scan Targets

Git Repository & Filesystem Scanning

Container Image Scanning

Dockerfile Scanning

IaC Scanning

Kubernetes Manifest Scanning

SBOM Generation & Scanning

AWS Account Scanning

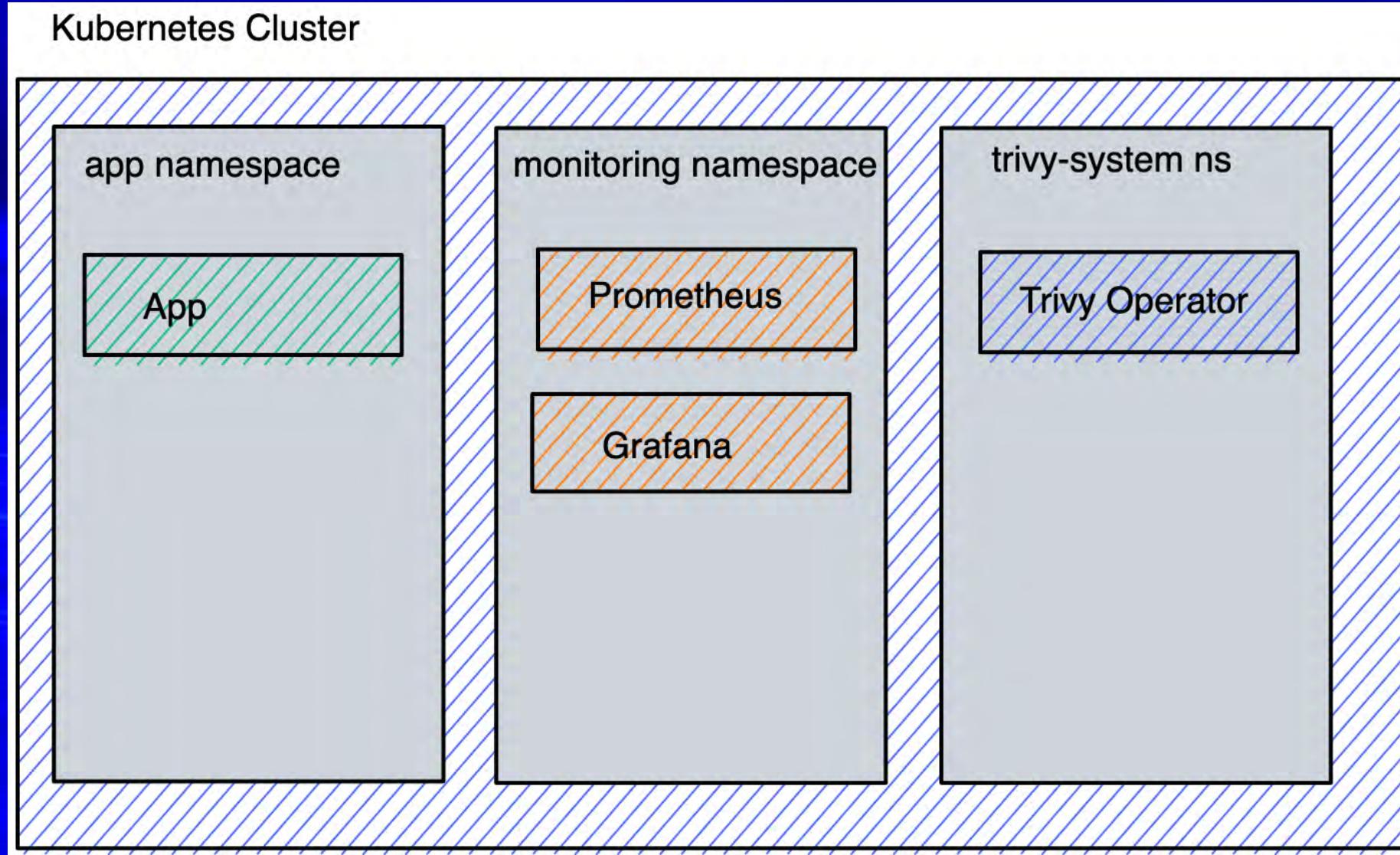
In-cluster Scanning

# Step 3 Setting it up & Making sure everything is running properly

- 1. Identifying the best installation option**
- 2. Deciding upon the configuration**
- 3. Testing custom configuration**
- 4. Ensuring everything is working together**



```
✓ MONITOR-SECURITY
  ✓ app-manifests
    ! deployment.yaml
    ! service.yaml
  > assets
  > dashboards
  ✓ observability-conf
    ! prom-values.yaml
    ! promtail-values.yaml
    ! tracee.yaml
    ! trivy-service-monit...
  README.md
```



If everything is a Kubernetes resource, you can use the same processes across your stack

```
● ● ●  
› kubectl get all -n trivy-system  
NAME                                     READY   STATUS    RESTARTS   AGE  
pod/trivy-operator-8556cdf857-snhdj     1/1     Running   0          4m56s  
  
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE  
service/trivy-operator   ClusterIP  None        <none>       80/TCP    4m56s  
  
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE  
deployment.apps/trivy-operator   1/1     1           1           4m57s  
  
NAME           DESIRED   CURRENT   READY   AGE  
replicaset.apps/trivy-operator-8556cdf857   1         1         1         4m57s  
    return go(f, seed, [])  
}
```







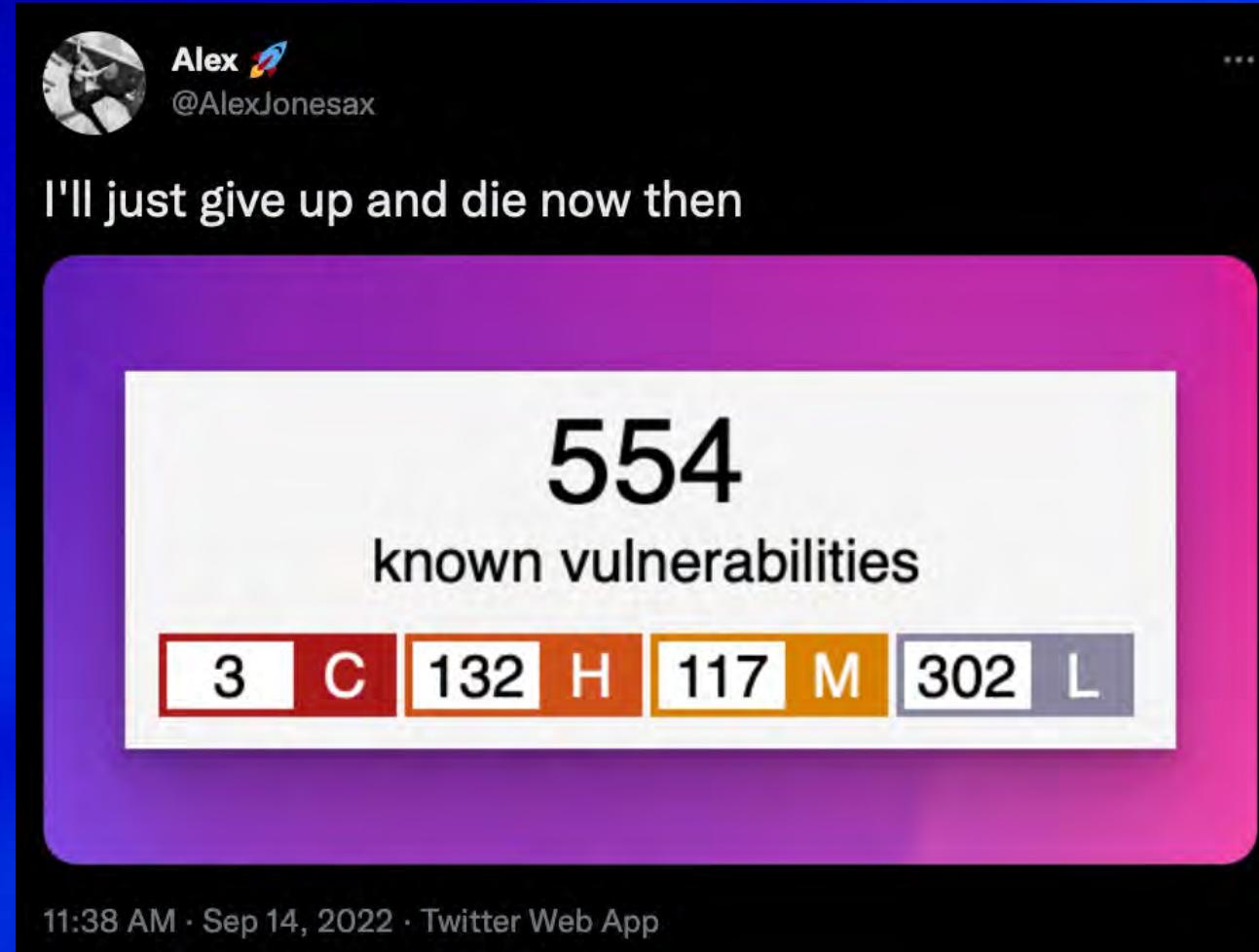


# Step 4 Setting up a dashboard



The dashboard can be found in the following repo: <https://grafana.com/grafana/dashboards/16652-trivy-operator-reports/>

# Step 5 Avoiding Vulnerability Hell



Tweet <https://twitter.com/AlexJonesax/status/1569998923955142657>

# Some possible strategies

- 1. Ignore all but Critical Vulnerabilities**
- 2. Don't scan everything at once**
- 3. Filter by Vulnerabilities with known-fixture**
- 4. Filter Vulnerabilities by team & by application**
- 5. Make the Vulnerabilities context-specific**

# Step 6 What are metrics without alerts

| State  | Name                   | Health | Summary |
|--------|------------------------|--------|---------|
| Normal | Critical Vulnerability | ok     |         |

[Silence](#)  [Show state history](#)  [View](#)  [Edit](#)  [Delete](#)

Evaluate Every 1m Data source  
For 5m Prometheus

Matching instances Search by label  State  
 Normal 1 Alerting Pending NoData Error

| State  | Labels   | Created |
|--------|--|---------|
| Normal | alertname=Critical Vulnerability grafana_folder=test | -       |

It's easy to ignore vulnerabilities –  
give them a "voice"

# Step 7 Correlating Metrics



## Replicas available by deployment



# Step 8 Some additional tips

## Just a few more things to think about...

- **Assign Ownership**
- **Don't introduce "too many" new tools at once**
- **Utilise existing workflows, platforms and processes**

# Step 9 Optimise based on what works for your team

The initial setup might be the same but everything else will be different

# Step 10 Don't stop at security scanning



<https://github.com/aquasecurity/tracee>



# Additional Resources

- [Our Application Security Journey \(Part 1\) by Wise Engineering](#)
- [The Aqua Open Source YouTube Channel](#)
- [The Trivy GitHub Repository and the Trivy Operator Repository](#)
- [The demo project on GitHub](#)

& You can [find us on Slack](#): slack.aquasec.com

# Questions



@urlichsanais

# Thanks



@urlichsanais