
NOVATEC
Container & Kubernetes Security for Developers
Andreas Falk
DEVOXX™ UKRAINE

About Me

Andreas Falk

Novatec Consulting
(Stuttgart, Germany)

andreas.falk@novatec-gmbh.de
@andifalk

OWASP Open Web Application Security Project

OpenID

NOVATEC

Security @ Novatec

Our Offerings:

- OAuth 2.0 & OpenID Connect
  - Trainings, Audits and Coaching
  - Support in Architecture and Implementation
- Threat Modeling
  - Trainings & Workshops

https://www.novatec-gmbh.de/en
https://www.novatec-gmbh.de/en/consulting/oauth-2-0-and-openid-connect/

Agenda

1. Use Cases for Containers
2. What Can Go Wrong
3. The Path for Secure Development
   - Application Security
   - Container Security
   - Kubernetes Security
   - Kubernetes Secrets

Slides and the Code?

Demos

Iteration 1: Application Security

- Hello Spring Boot

Iteration 2: Container Security

- Root Container
- Rootless Container
- Rootless Container with JIB

Iteration 3: Kubernetes Security

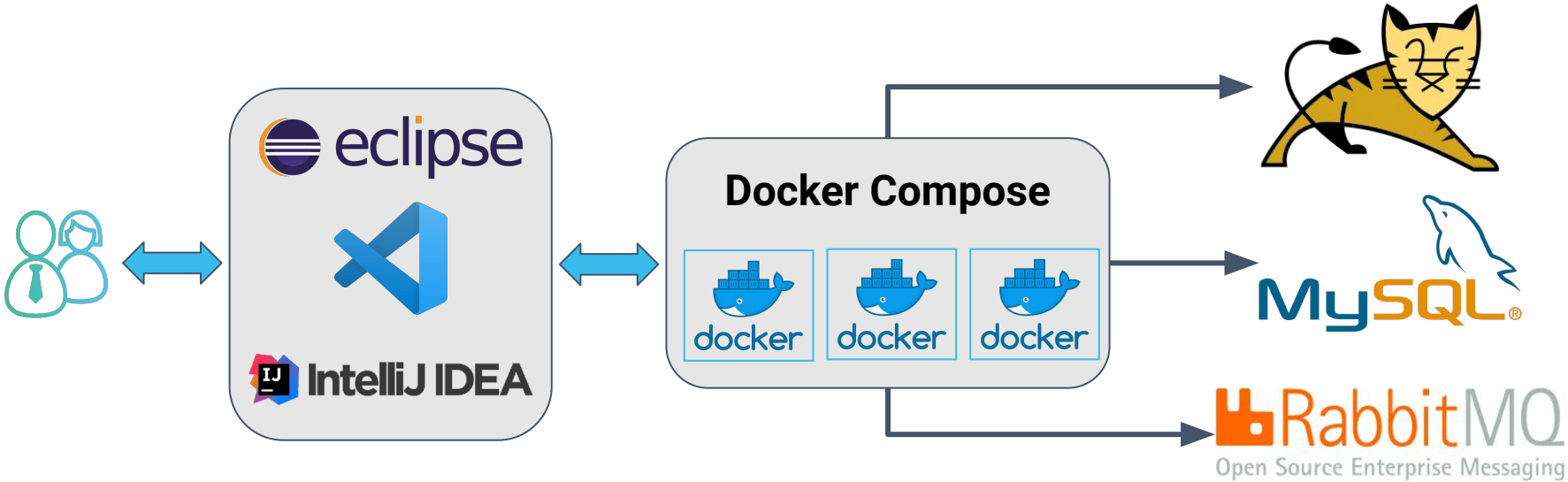
- Initial Unsafe Kubernetes Deployment
- Safe Kubernetes Deployment (Pod Security Context)
- Safe Kubernetes Deployment (Pod Security Policy)
- Safe Kubernetes Deployment (Open Policy Agent)

Look here:

https://github.com/andifalk/secure-development-on-kubernetes

Container Use Cases

Development Environments

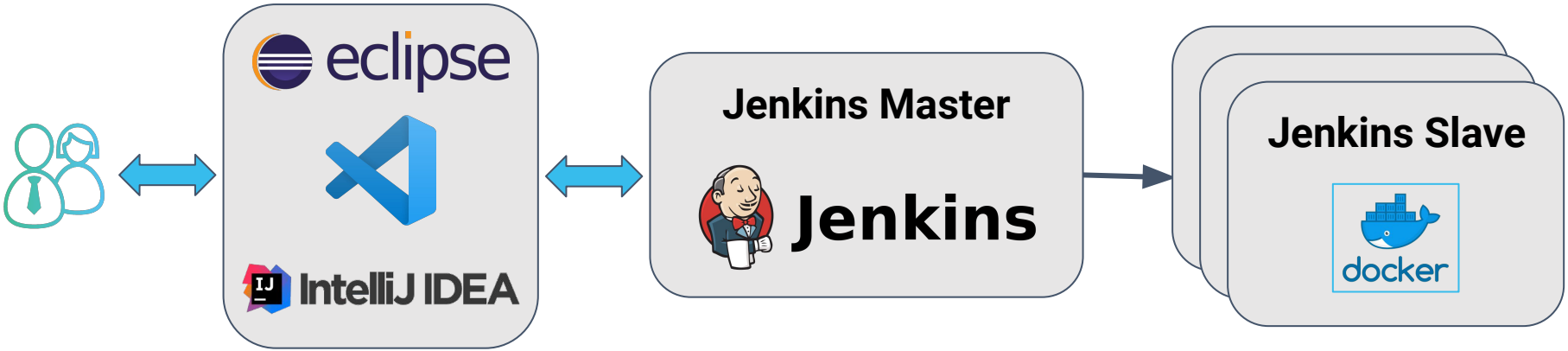


Integration Testing (TestContainers)

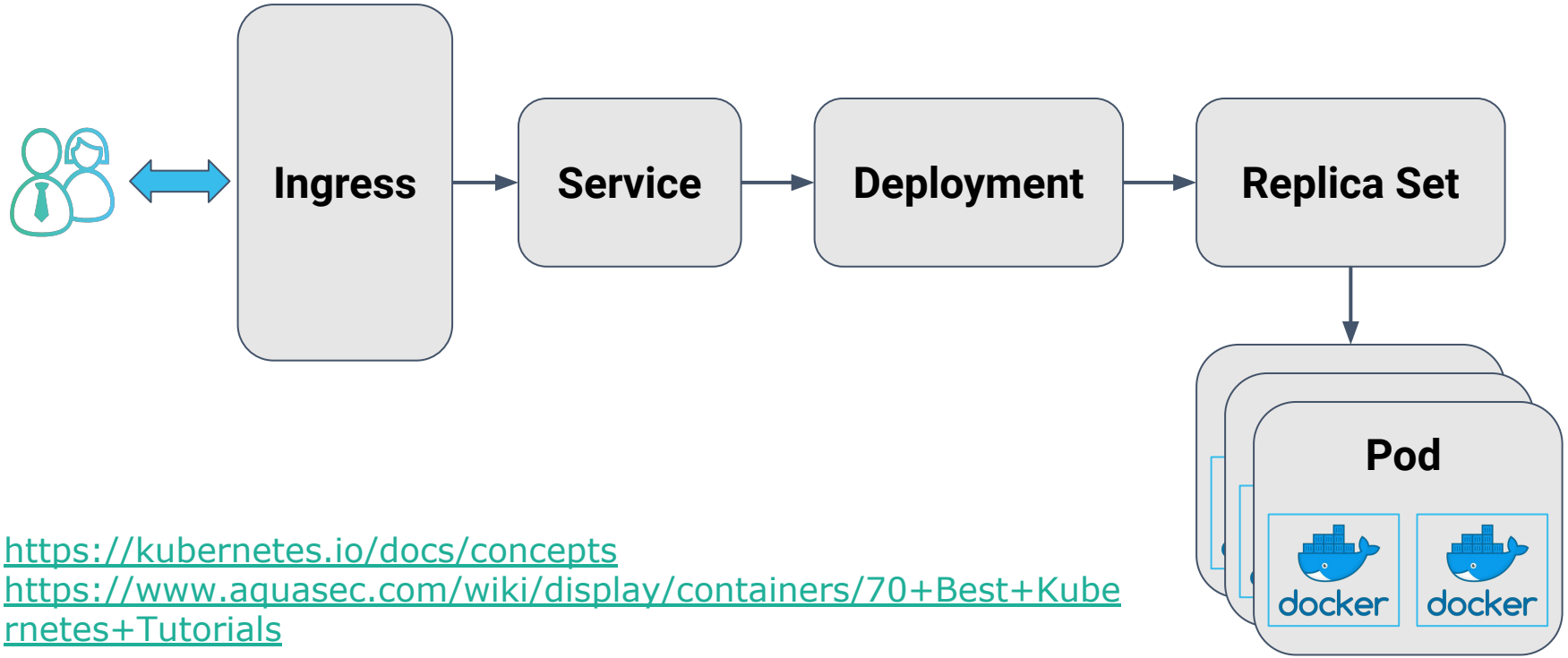
- Java library that supports JUnit tests
- Provides lightweight, throwaway instances of common 3rd party components that can run in a Docker container.
- Databases (DB2, Oracle XE, MS SQL, MySQL, …)
- Kafka, RabbitMQ
- Hashicorp Vault, Keycloak
- ...

https://www.testcontainers.org

CI-/CD Pipelines



Container & Kubernetes (Production)



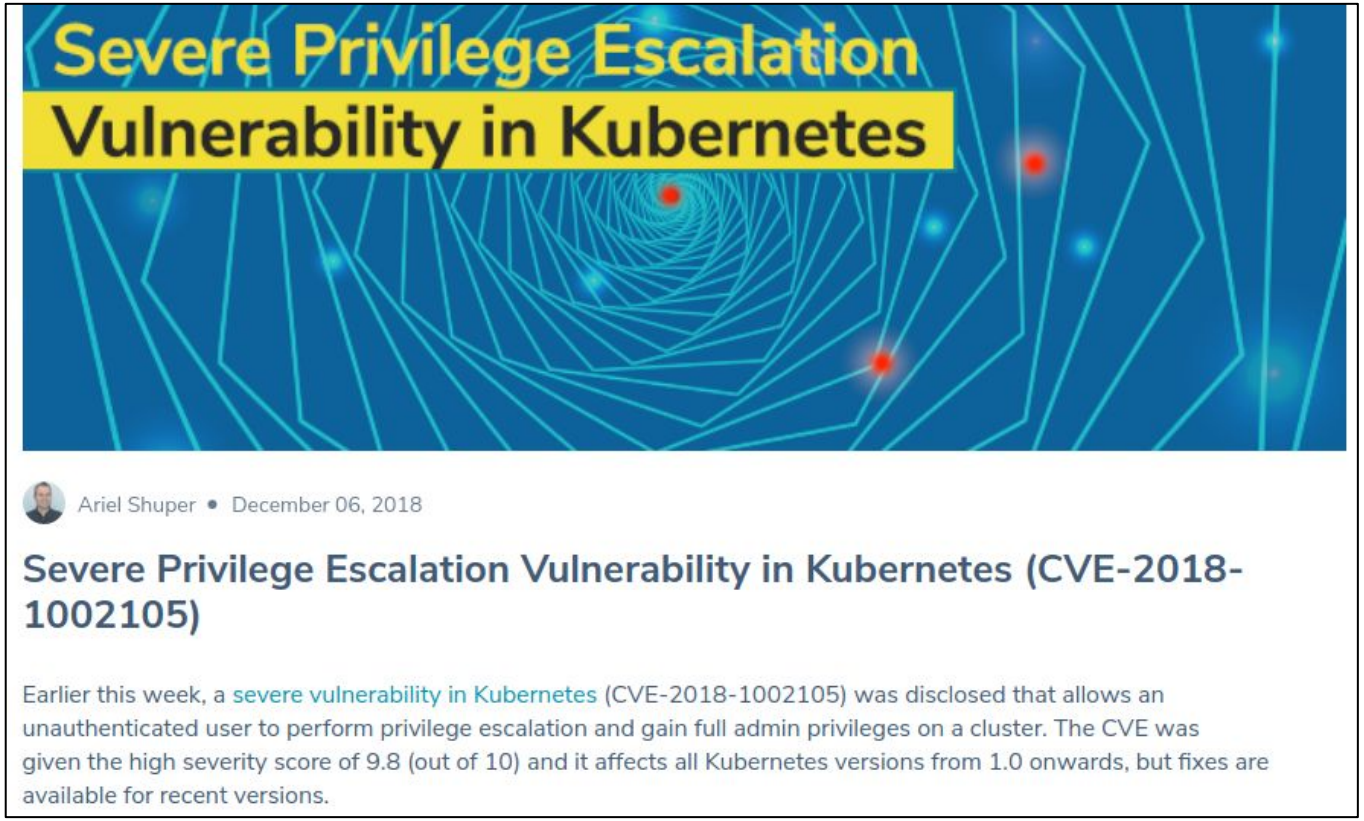
https://kubernetes.io/docs/concepts
https://www.aquasec.com/wiki/display/containers/70+Best+Kubernetes+Tutorials

What can go wrong?

Introduction

Severe Vulnerability in Kubernetes

Source: https://blog.aquasec.com



NOVATEC

Crypto Mining Via K8s Dashboard

Source: https://blog.heptio.com

On Securing the Kubernetes Dashboard

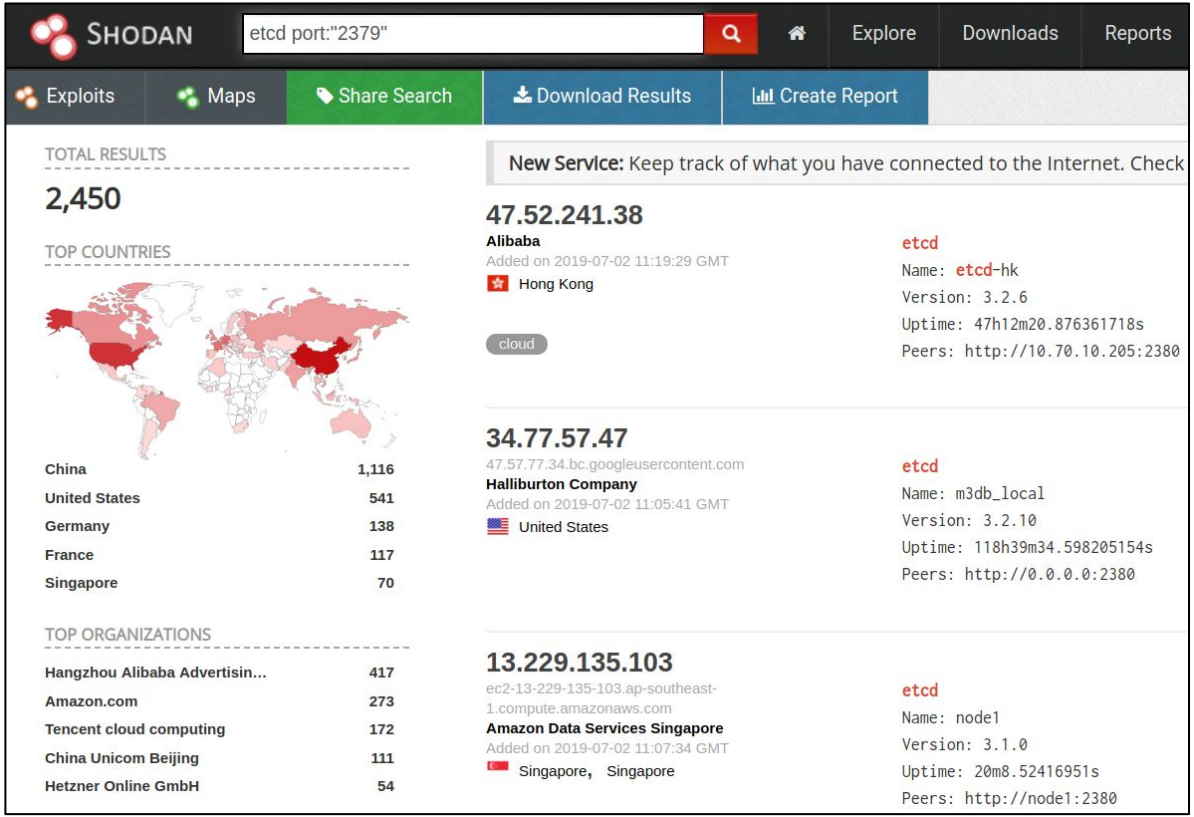
Joe Beda Follow

Feb 28, 2018 · 13 min read

Recently Tesla (the car company) was alerted, by security firm RedLock, that their Kubernetes infrastructure was compromised. The attackers were using Tesla's infrastructure resources to mine cryptocurrency. This type of attack has been called "cryptojacking".

The vector of attack in this case was a Kubernetes Dashboard that was exposed to the general internet with no authentication and elevated privileges. Not only this, but core AWS API keys and secrets were visible. How do you prevent this from happening to you?

Open ETCD Ports in Kubernetes (1)



https://shodan.io

NOVATEC

Open ETCD Ports in Kubernetes (2)


```
$ etcdctl --endpoints=http://xx.xx.xx.xx:2379
cluster-health

member b97ee4034db41d17 is healthy: got healthy
result
from http://xx.xx.xx.xx:2379
cluster is healthy
```

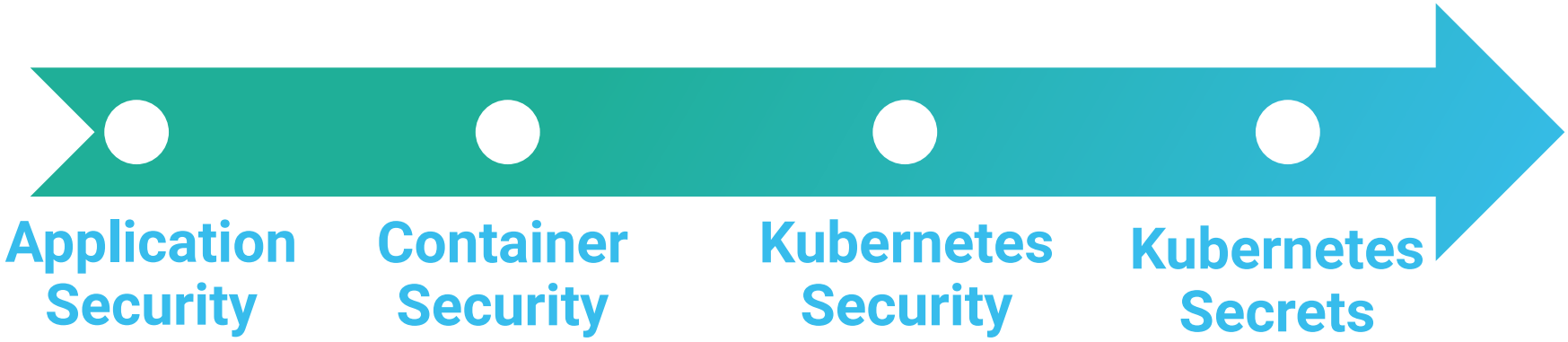
https://github.com/etcd-io/etcd/releases

NOVATEC

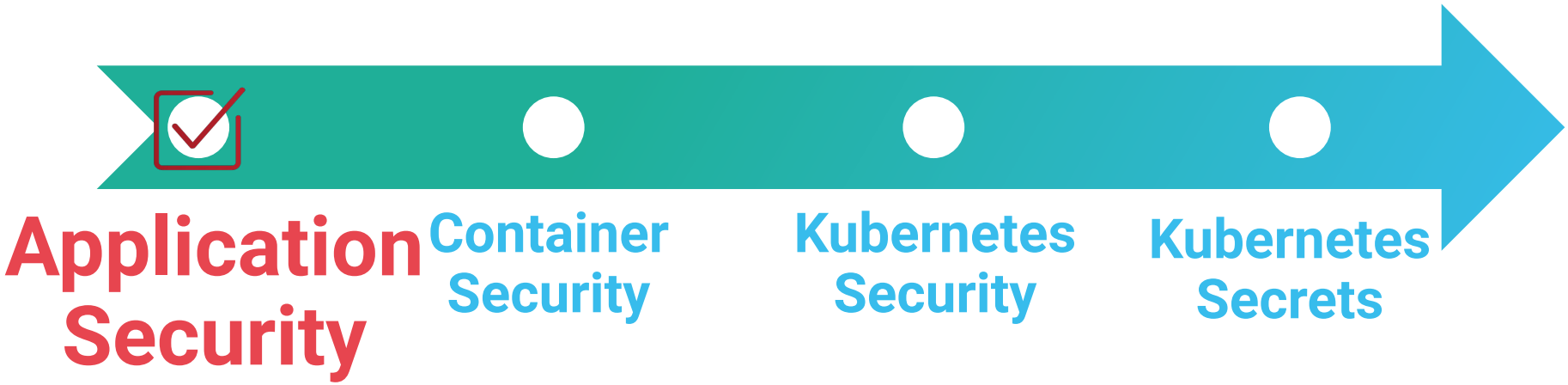
So what can WE do as Developers?

Application- / Docker- / K8s-Security

The Path for Secure Development on K8s



The Path for Secure Development on K8s



Application Security

Authentication

Authorization

SQL Injection

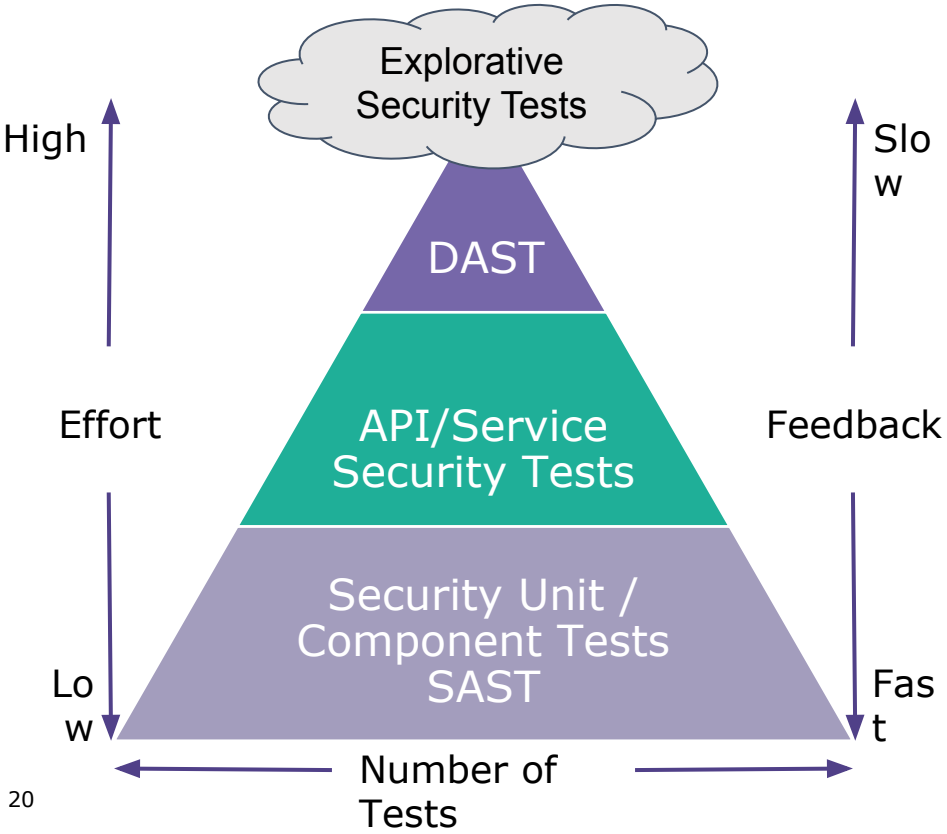
Cross Site Scripting (XSS)

Cross Site Request Forgery (CSRF)

Data Protection (Crypto)

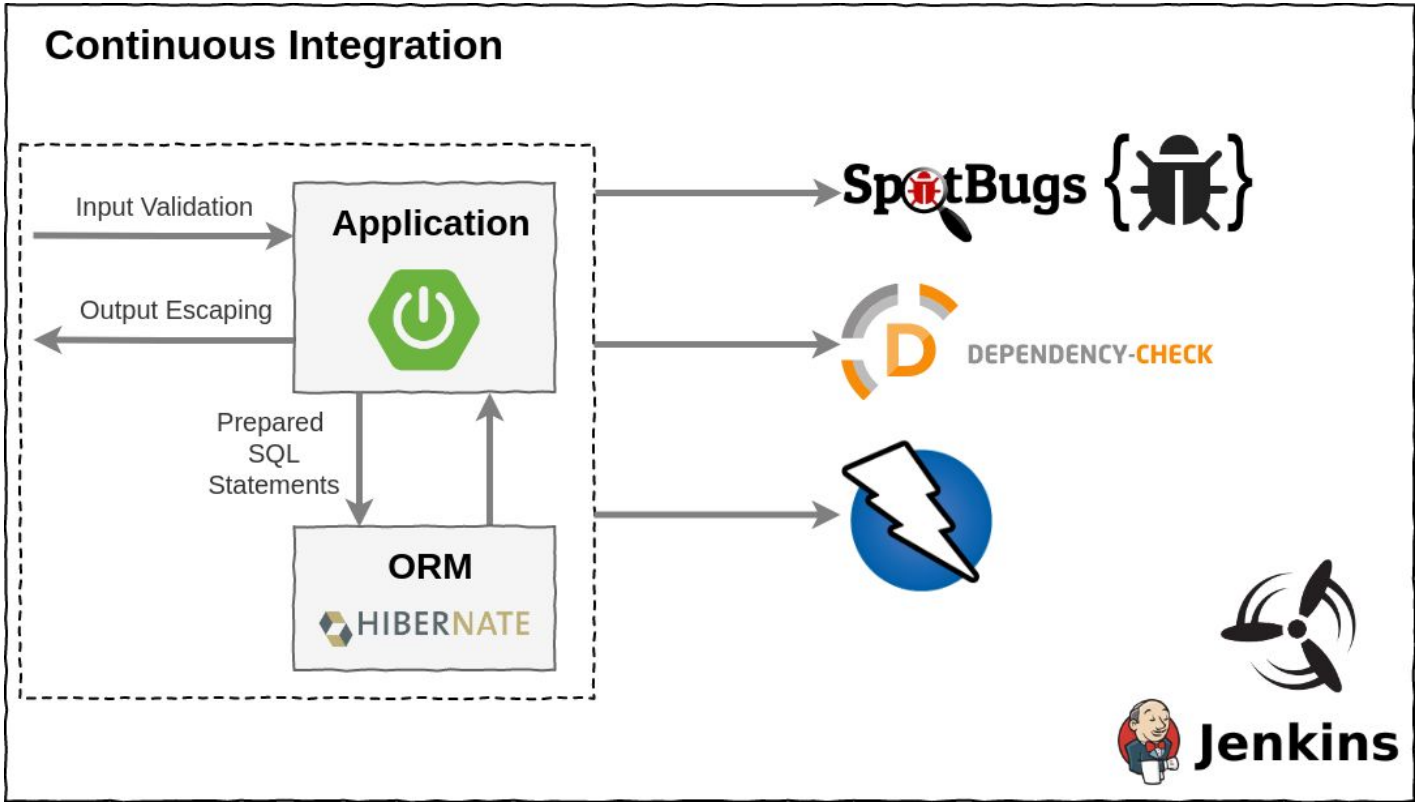
...

The Security-Testing Pyramid



- Security Charters in Explorative Testing
- Dynamic Application Security Testing (DAST), e.g OWASP Zap
- Input Validation, Authentication, Authorization, Session Management, Output Escaping (XSS), SQL Injection, Security Response Headers
- Input Validation, Authentication, Output Escaping (XSS), Bypass Business Logic, Error Handling & Logging
- Static Application Security Testing (SAST)

Automate All The Things!



Live Demo: Show me the code

Iteration 1: Application Security

https://github.com/andifalk/secure-development-on-kubernetes

Application Security 101

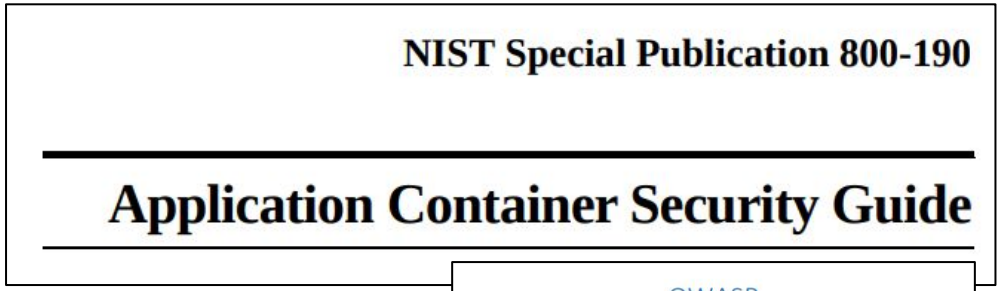
- Input Validation for ALL types of input
- Output Encoding to prevent XSS
- Use only Parameterized Queries/Prepared Statements
- Enforce Authentication & Authorization
- Never implement your own Crypto or Session-Management
- Check 3rd Party Dependencies for Vulnerabilities
- Use Static & Dynamic Application Security Testing

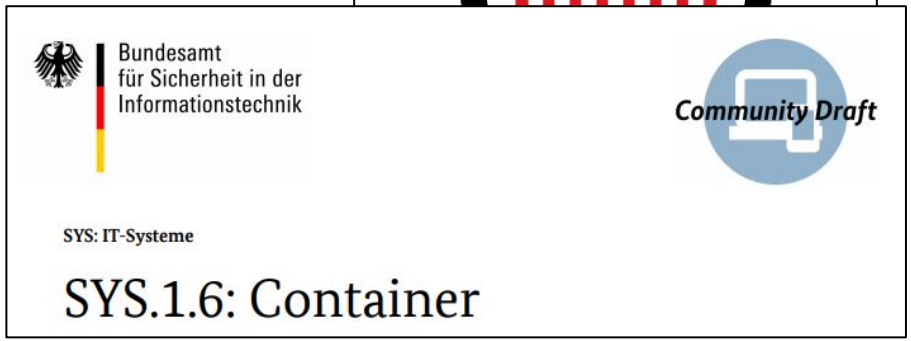
https://cheatsheetseries.owasp.org
https://owasp.org/www-project-top-ten
https://owasp.org/www-project-proactive-controls

The Path for Secure Development on K8s

OWASP Docker Top 10

1. Secure User Mapping
2. Patch Management Strategy
3. Network Segmentation and Firewalling
4. Secure Defaults and Hardening
5. Maintain Security Contexts
6. Protect Secrets
7. Resource Protection
8. Container Image Integrity and Origin
9. Follow Immutable Paradigm
10. Logging

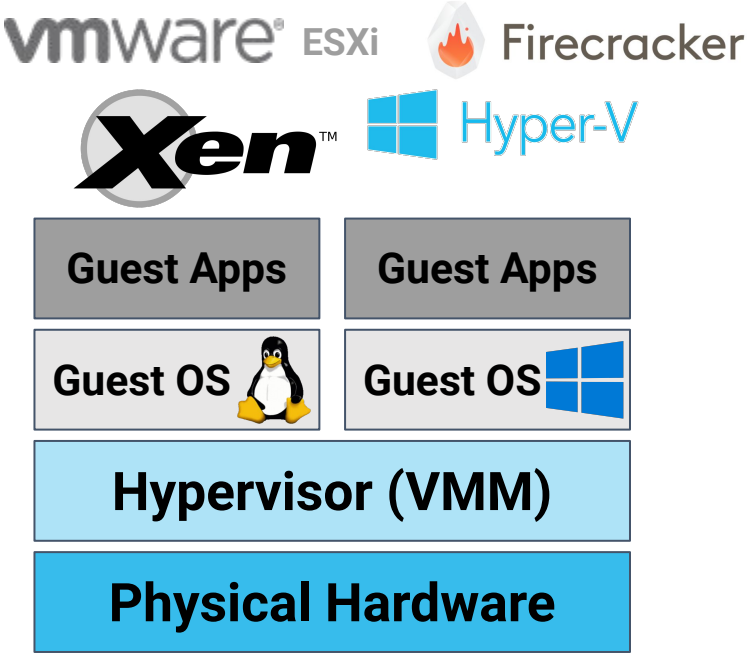




https://github.com/OWASP/Docker-Security
https://doi.org/10.6028/NIST.SP.800-190
https://github.com/OWASP/Container-Security-Verification-Standard
https://www.bsi.bund.de

NOVATEC

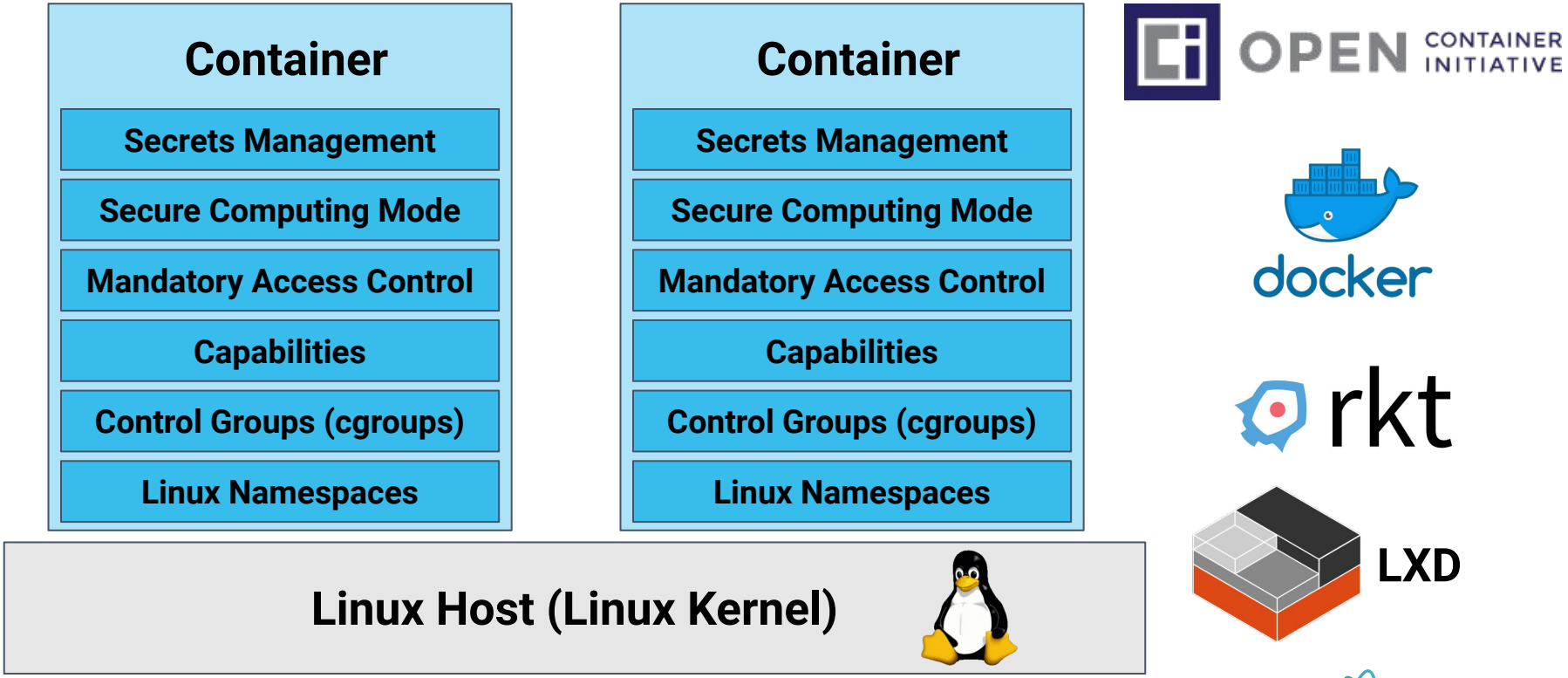
Virtual Machine (VM) Basics



Type 1 Virtual Machine Monitor

Type 2 Virtual Machine Monitor

Container (Security) Basics



NOVATEC

Linux Kernel Namespaces

- Process IDs
- Network
- Mount Points
- Inter-Process Communications (IPC)
- User & Group IDs
- Unix Timesharing System (UTS): hostname & domain names
- Control groups (cgroups)

```
$ man namespaces
$ sudo lsns
```

Linux Control Groups (cgroups)

- Resource Limits
  - CPU
  - Memory
  - Devices
  - Processes
  - Network

For Java this only works with container aware JDK versions as of **OpenJDK 8u192** or above
Recommendation: Use Java 11

Linux Capabilities

- Break up privileges into smaller units
  - CAP_SYS_ADMIN
  - CAP_NET_ADMIN
  - CAP_NET_BIND_SERVICE
  - CAP_CHOWN
  - ...

```
$ man capabilities
$ docker run --cap-drop=ALL --cap-add=NET_BIND_SERVICE
```

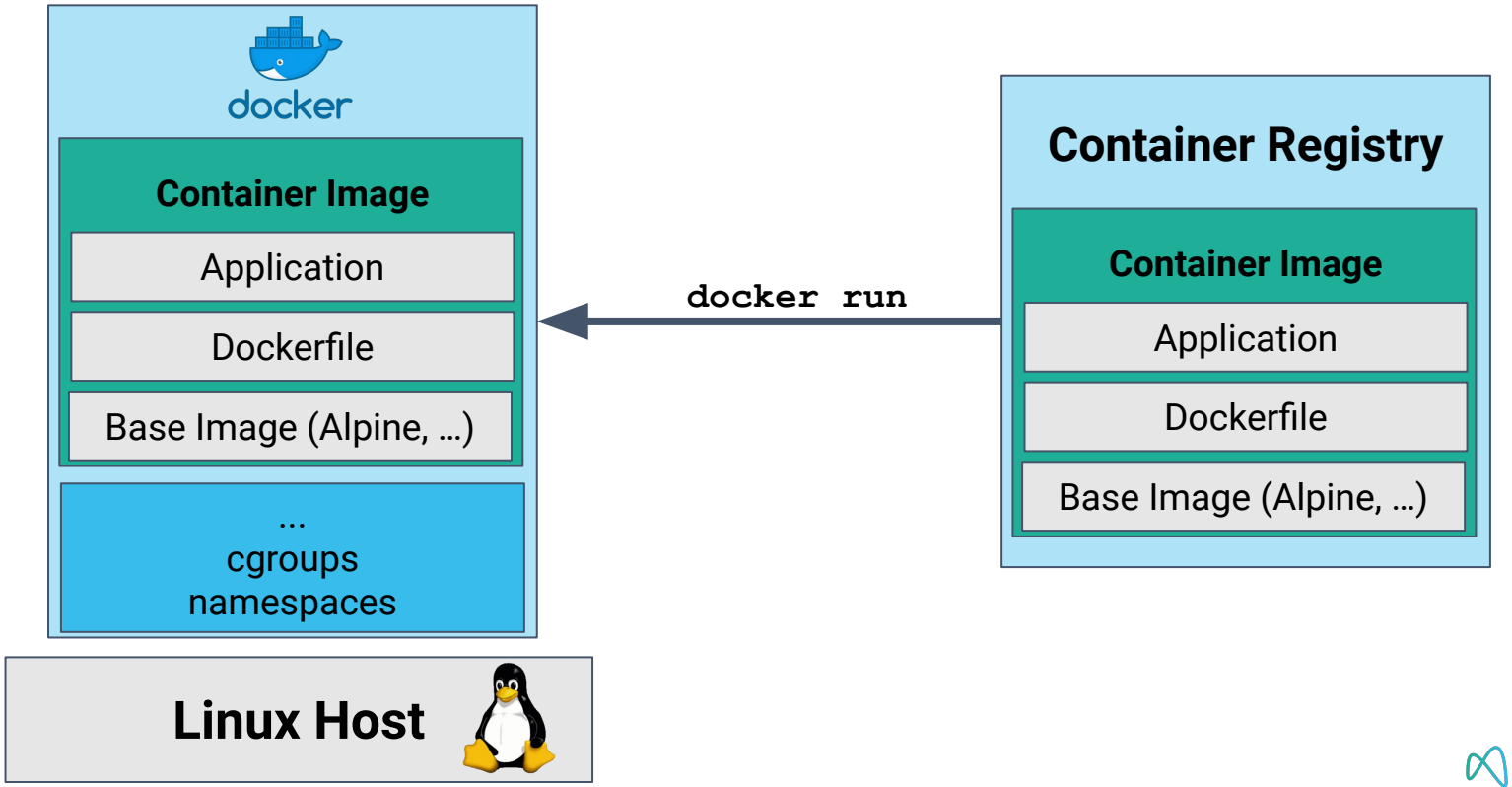
http://man7.org/linux/man-pages/man7/capabilities.7.html

Linux Mandatory Access Control & System Calls

- Restrict System Calls
  - Secure Computation Mode (seccomp)
  - Google gVisor
- Linux Kernel Security Modules (MAC)
  - AppArmor
  - Security-Enhanced Linux (SELinux)

https://docs.docker.com/engine/security/seccomp
https://apparmor.net
https://en.wikipedia.org/wiki/Security-Enhanced_Linux
https://gvisor.dev/docs

Docker Images



Breaking Container Isolation

- Containers Run as Root by Default
- The --privileged Flag and Capabilities
- Mounting Sensitive Directories
- Mounting the Docker Socket
- Shared Namespaces

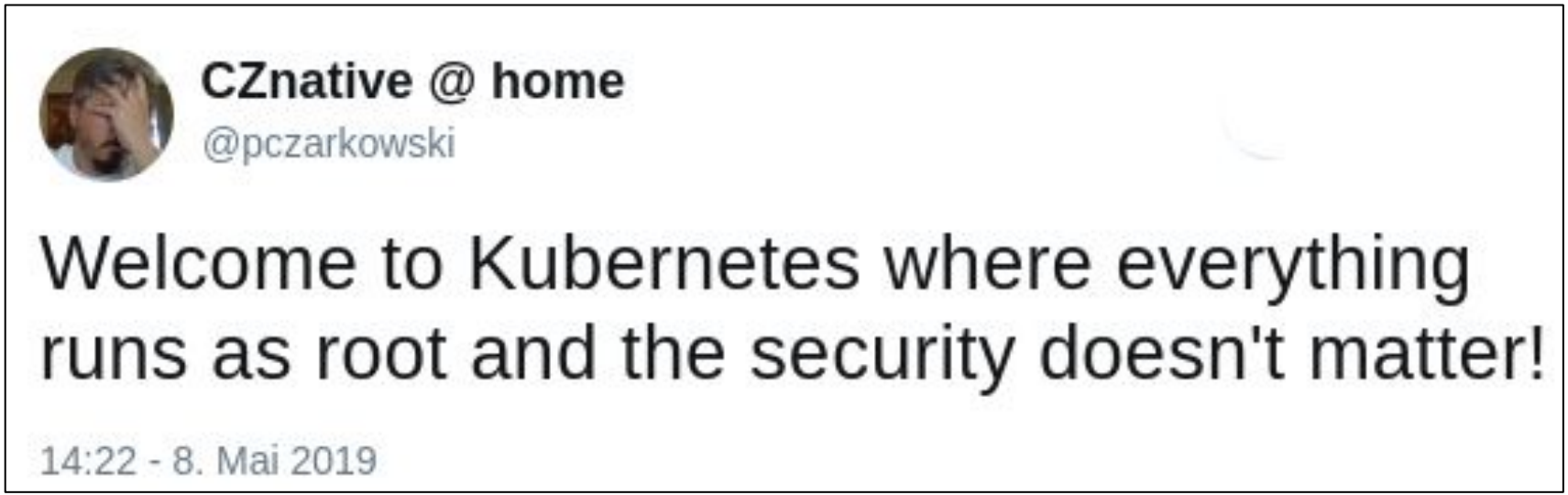
https://docs.docker.com/engine/security/rootless
https://docs.docker.com/engine/security/https

Mount sensitive Directories

```
$ touch /ROOT_FOR_HOST
$ docker run -it -v /:/hostroot ubuntu bash
root@91083a4eca7d:/$ ls /hostroot

ROOT_FOR_HOST bin etc home lib
lib64 media mnt root run srv sys var
..
.
```

All is Root


CZnative @ home
@pczarkowski
Welcome to Kubernetes where everything runs as root and the security doesn't matter!
14:22 - 8. Mai 2019

Say No To Root (1)

USER directive in Dockerfile

```
FROM openjdk:11-jre-slim
COPY hello-spring-kubernetes-1.0.0-SNAPSHOT.jar app.jar
EXPOSE 8080
RUN addgroup --system --gid 1002 app && adduser
    --system --uid 1002 --gid 1002 appuser
USER 1002
ENTRYPOINT java -jar /app.jar
```

https://opensource.com/article/18/3/just-say-no-root-containers

Say No To Root (2)

Use JIB and Distroless Images

```
plugins {
  id 'com.google.cloud.tools.jib' version '...'
}

jib {
 container {
   user = 1002
 }
}
```

https://github.com/GoogleContainerTools/jib

Secure Docker Run

docker run -t -i **--privileged** ubuntu bash

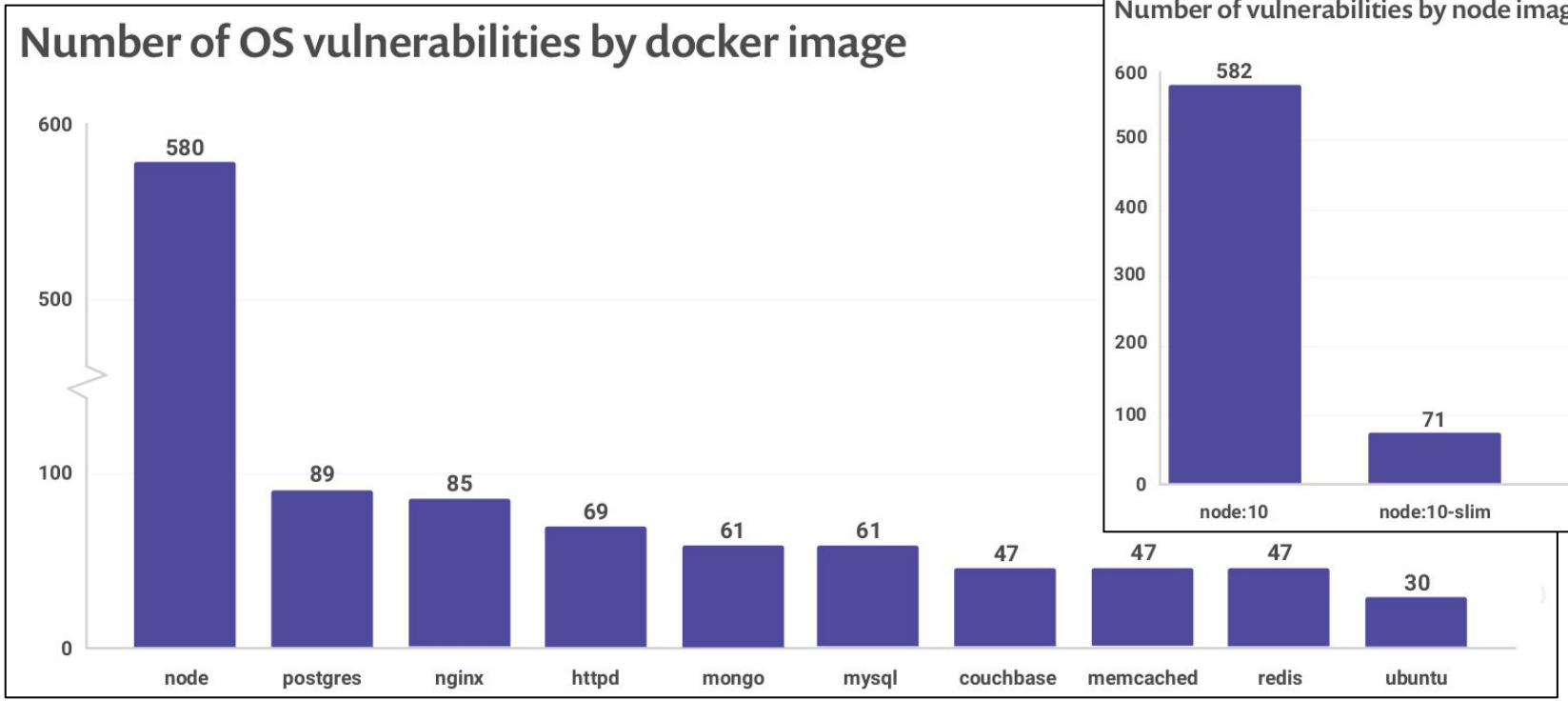
docker run -t -i ubuntu bash

docker run -t -i **--security-opt=no-new-privileges** ubuntu bash

docker run -t -i **--security-opt=no-new-privileges --cap-drop=ALL --cap-add=net_bind_service** ubuntu bash

docker run -t -i **--security-opt=no-new-privileges --cap-drop=ALL --cap-add=net_bind_service --cpu-shares=0.5 --memory=200m** ubuntu bash

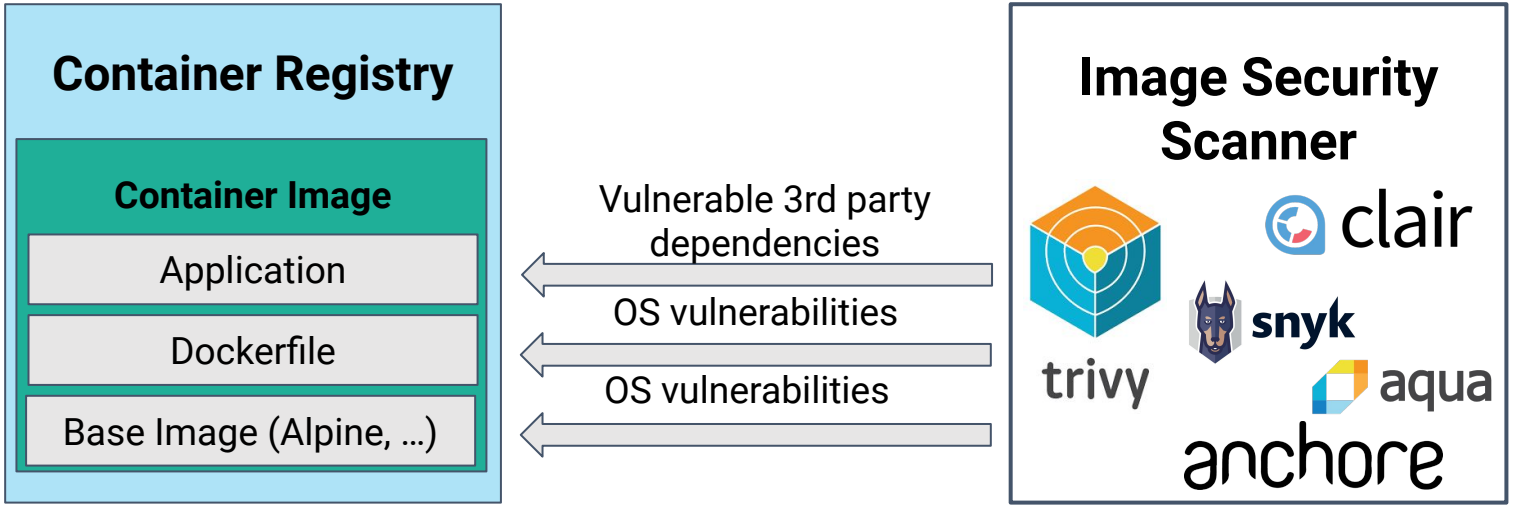
Vulnerable Docker Images



Source: The state of open source security report (snyk.io)

NOVATEC

Container Image Security



https://anchore.com/opensource/
https://github.com/coreos/clair
https://github.com/aquasecurity/trivy
https://www.docker.com/blog/announcing-scanning-from-snyk-for-docker

HARBOR

- Open source registry
- Secures artifacts with policies and RBAC
- Ensures images are scanned (Clair or Trivy)
- Signs images as trusted

https://goharbor.io/

https://www.docker.com/press-release/Docker-Snyk-Announce-Partnership-Container-Vulnerability-Scanning

Live Demo: Show me the code

Iteration 2: Container Security

https://github.com/andifalk/secure-development-on-kubernetes

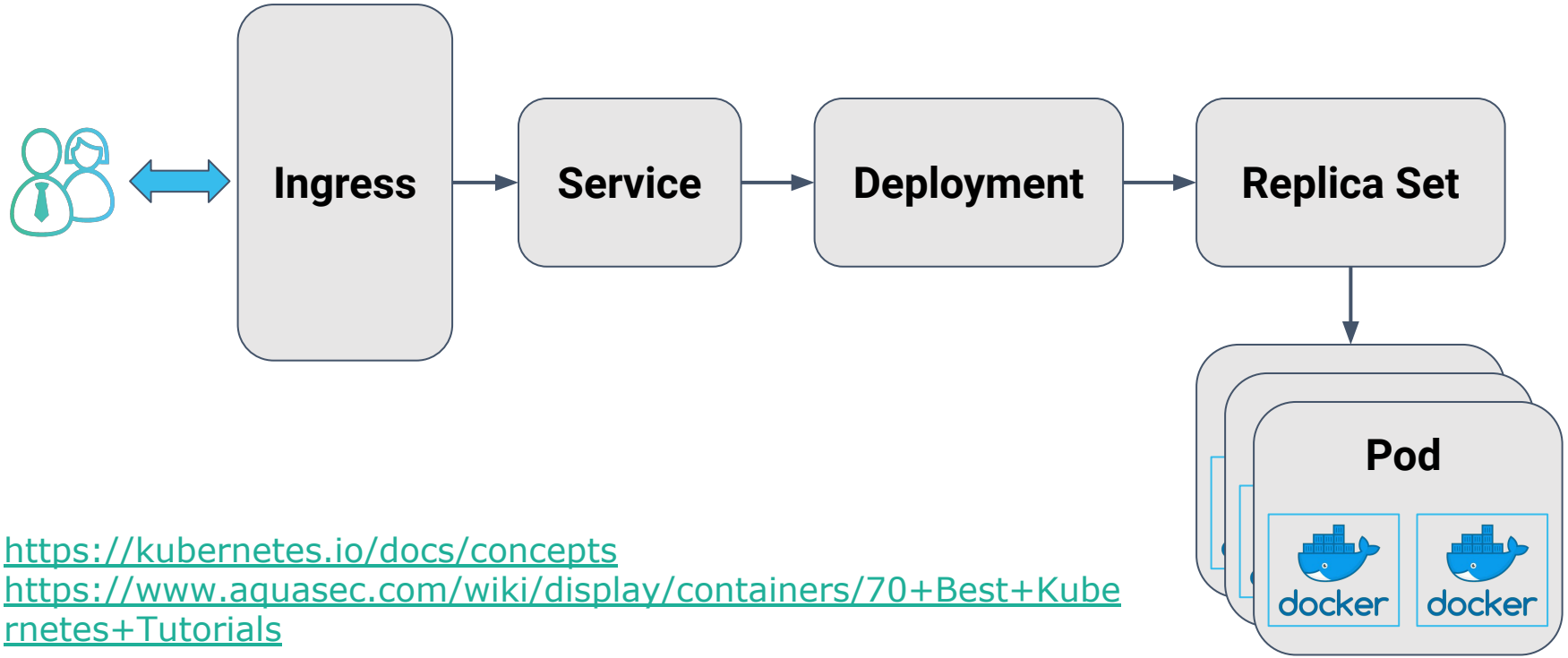
Container Security 101

- Learn Linux (Security) Basics
- Load Images from Trusted Registries Only
- Scan Images for Vulnerabilities (in CI/CD Pipeline)
- Say No To Root & Run with *--security-opt=no-new-privileges*
- Do NOT hardcode Secrets into a Container Image
- Limit resources (memory, CPU, processes, ...)
- Use Linux Security Module (seccomp, AppArmor, SELinux)

https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html
https://docs.docker.com/engine/security
https://blog.aquasec.com/docker-security-best-practices
https://blog.aquasec.com/devsecops-with-trivy-github-actions

The Path for Secure Development on K8s

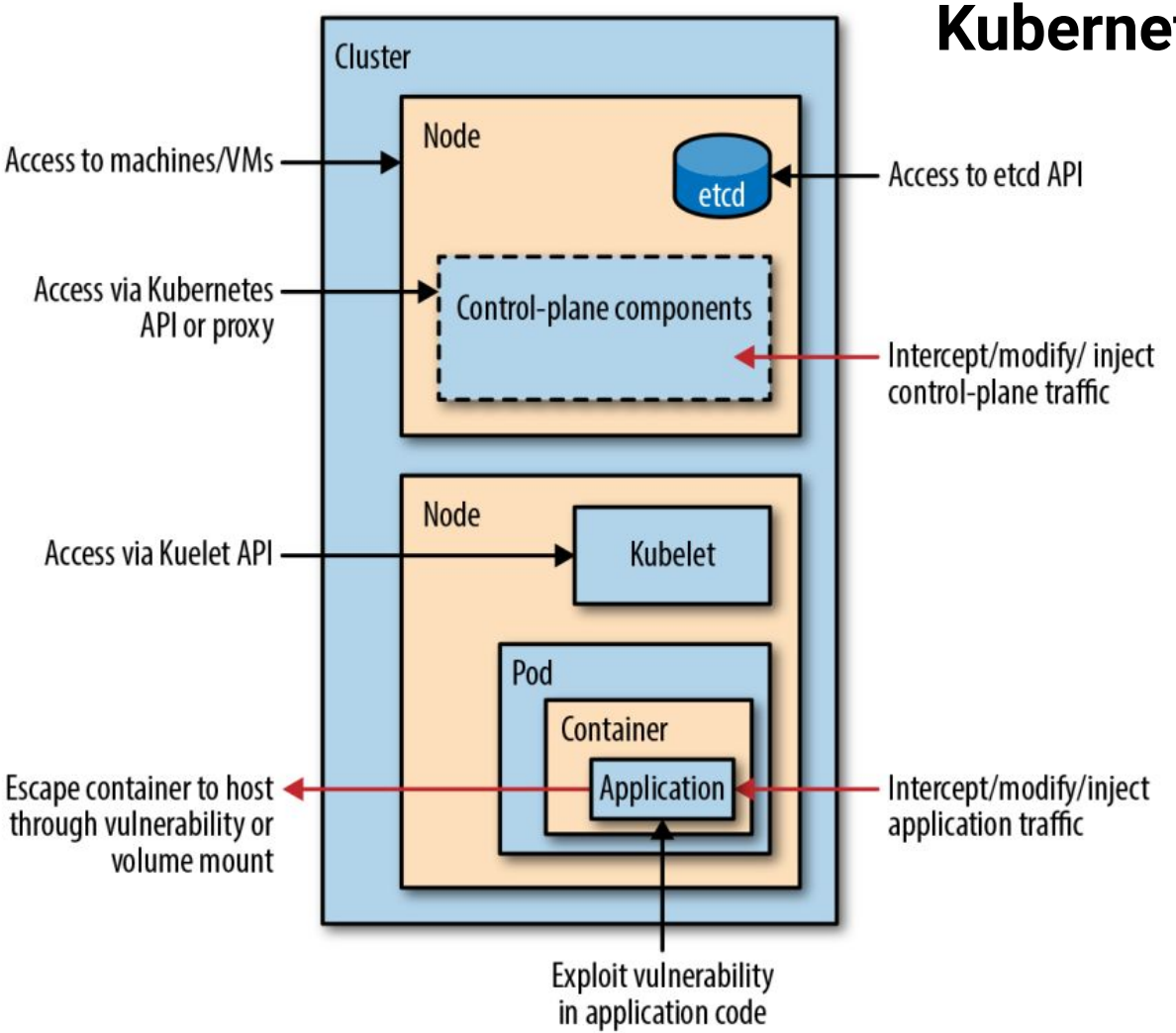
Kubernetes Basics



https://kubernetes.io/docs/concepts
https://www.aquasec.com/wiki/display/containers/70+Best+Kubernetes+Tutorials

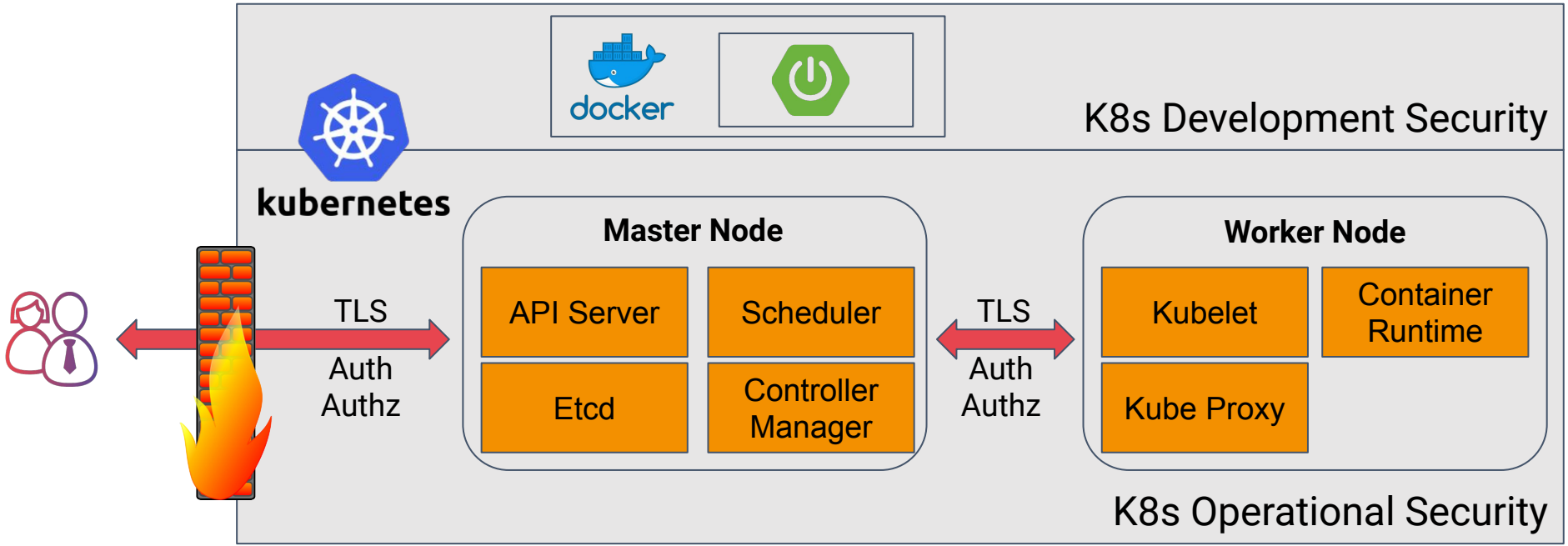
Kubernetes attack vectors

Source:
Kubernetes Security, O'Reilly, 2018



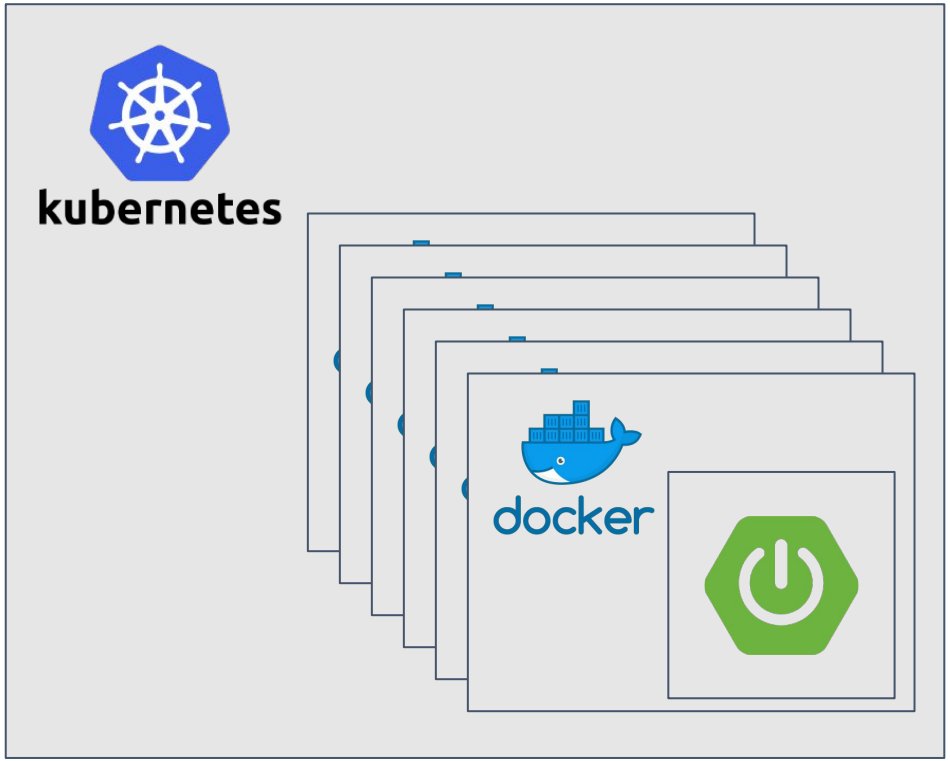
NOVATEC

Operational / Development K8s Security



https://kubernetes.io/docs/concepts/security/overview/#the-4c-s-of-cloud-native-security
https://learnk8s.io/production-best-practices/

Kubernetes Security



- Kubernetes Auditing
- Authentication
- Authorization (RBAC)
- Resource Limits
- Pod Security Context
- Pod Security Policy
- Open Policy Agent
- Network Policies

Kubernetes Authentication (1)

- Categories of users:
  - Service accounts managed by Kubernetes
  - Normal users

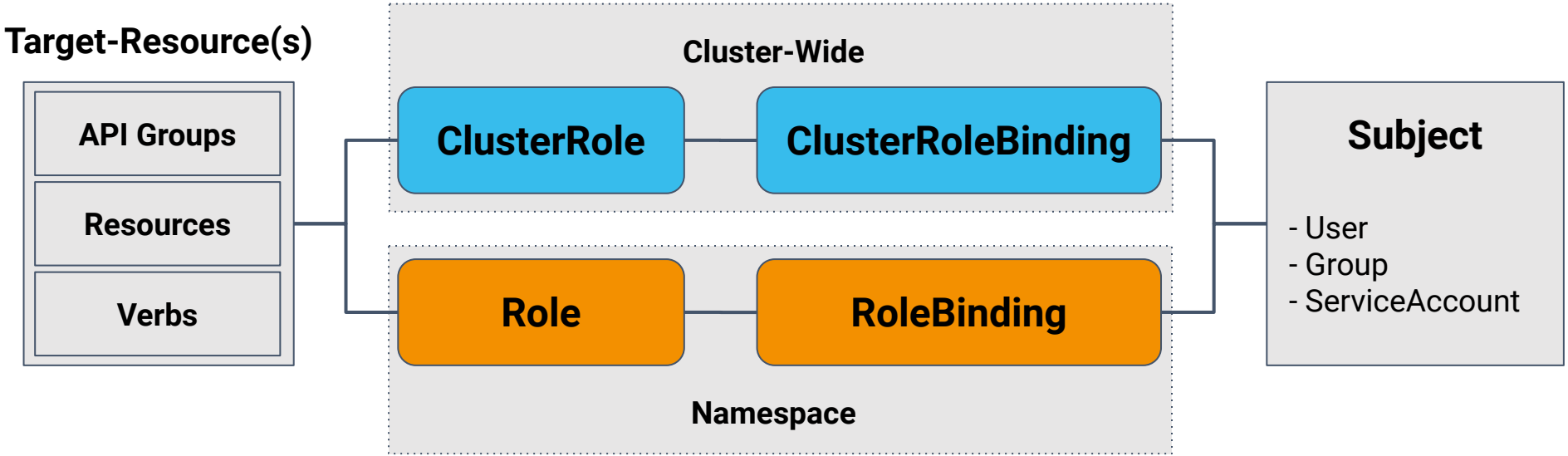
https://kubernetes.io/docs/reference/access-authn-authz/authentication/

Kubernetes Authentication (2)

- Authentication strategies
  - X509 Client Certs
  - Static Token File
  - Bootstrap Tokens
  - Service Account Tokens
  - OpenID Connect Tokens
  - Webhook Token Authentication
  - Authenticating Proxy

https://kubernetes.io/docs/reference/access-authn-authz/authentication/

Kubernetes Authorization (Role Based Access Control)



https://kubernetes.io/docs/reference/access-authn-authz/rbac/

Resource Limits

```
spec:
  ...
  containers:
    resources:
      limits:
        cpu: "1"
        memory: "512Mi"
      requests:
        cpu: 500m
        memory: "256Mi"
  ...
```

https://kubernetes.io/docs/tasks/configure-pod-container/assign-cpu-resource
https://kubernetes.io/docs/tasks/configure-pod-container/assign-memory-resource

Pod/Container Security Context

```
spec:
  securityContext:
    runAsNonRoot: true
  containers:
    securityContext:
      allowPrivilegeEscalation: false
      privileged: false
      runAsNonRoot: true
      readOnlyRootFilesystem: true
      capabilities:
        drop:
          - ALL
```

https://kubernetes.io/docs/tasks/configure-pod-container/security-context

Pod Security Policy

- Policies
  - Privileged
  - Baseline
  - Restricted

https://kubernetes.io/docs/concepts/security/pod-security-standards/#policies

Pod Security Policy (Still In Beta!)

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: no-root-policy
spec:
  privileged: false
  allowPrivilegeEscalation: false
  requiredDropCapabilities:
    - ALL
  runAsUser:
    rule: 'MustRunAsNonRoot'
  ...
```

https://kubernetes.io/docs/concepts/policy/pod-security-policy

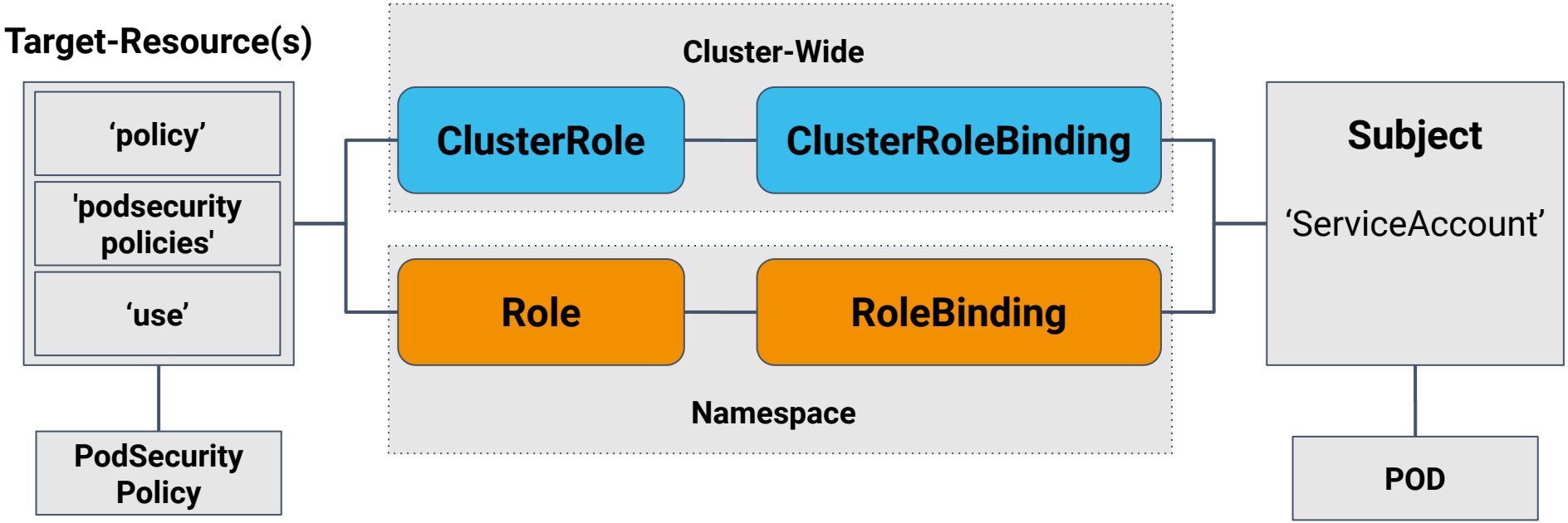
Pod Security Policy (Policy Order)

```
Policy order selection criteria:

1.  Policies which allow the pod as-is are preferred
2.  If pod must be defaulted or mutated, the first
    policy (ordered by name) to allow the pod is
    selected.
```

https://kubernetes.io/docs/concepts/policy/pod-security-policy/#policy-order
https://kubernetes.io/docs/reference/access-authn-authz/admission-controllers

Kubernetes RBAC + Pod Security Policies



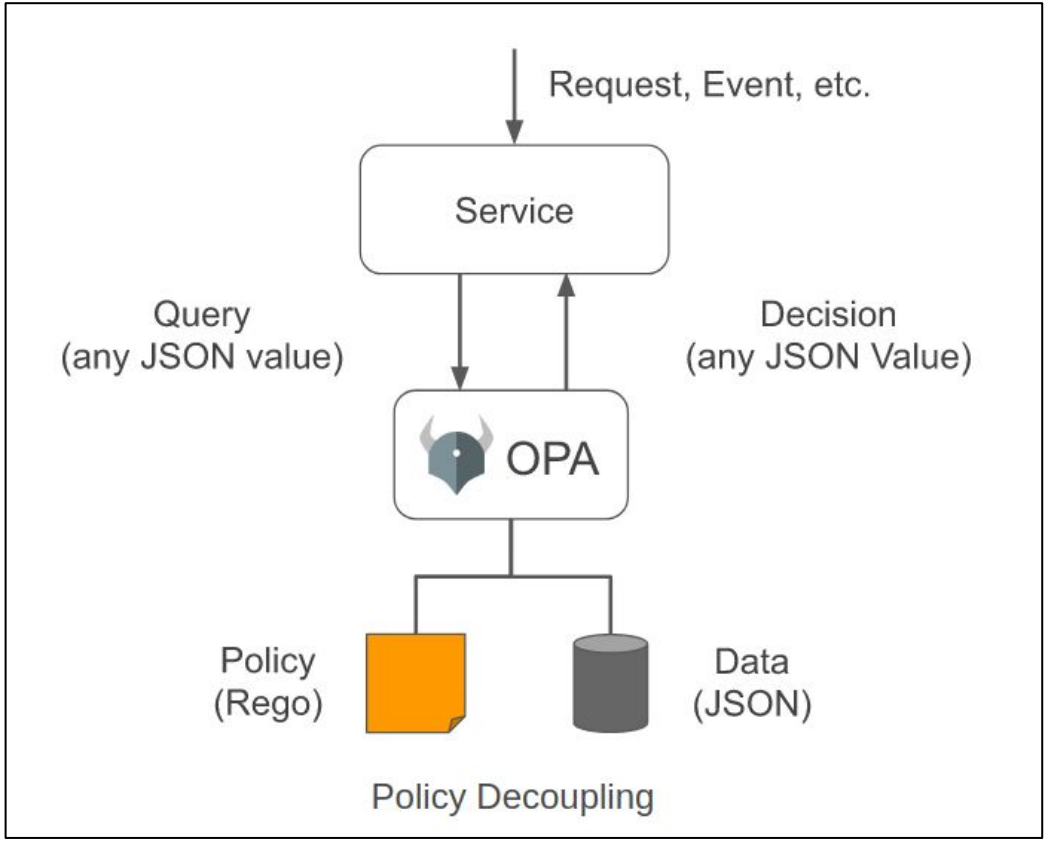
https://kubernetes.io/docs/reference/access-authn-authz/rbac/
https://kubernetes.io/docs/concepts/policy/pod-security-policy

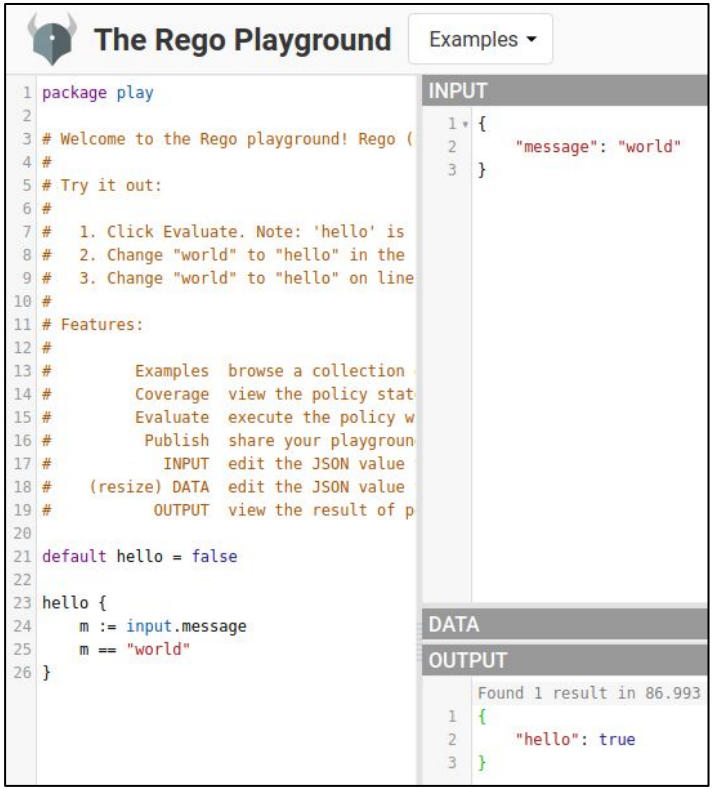
Kubernetes Role Based Access Control (RBAC)

apiGroups	extensions, apps, policy, ...
resources	pods, deployments, configmaps, secrets, nodes, services, endpoints, podsecuritypolicies, ...
verbs	get, list, watch, create, update, patch, delete, use, ...

https://kubernetes.io/docs/reference/access-authn-authz/rbac/

Open Policy Agent

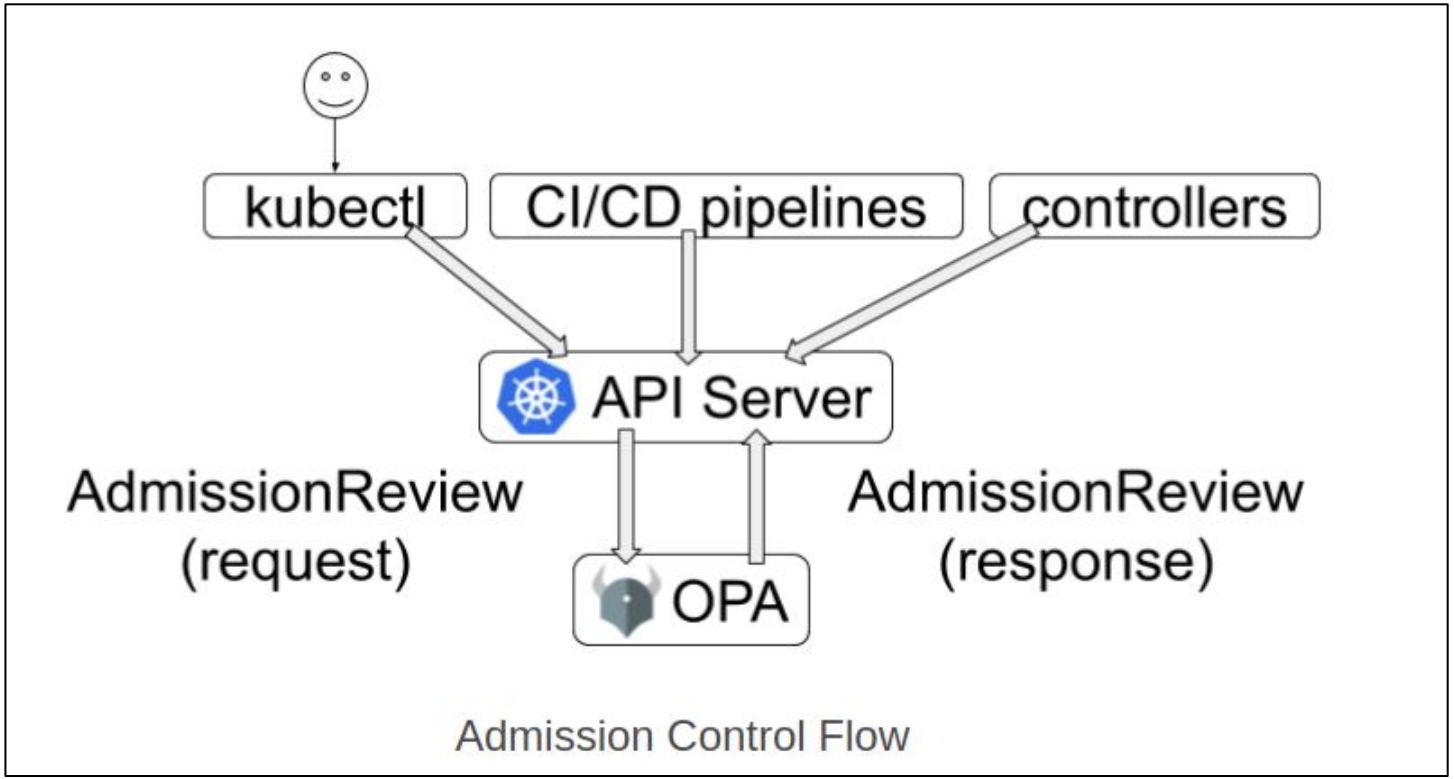




https://www.openpolicyagent.org
https://play.openpolicyagent.org

NOVATEC

Open Policy Agent - Kubernetes Gatekeeper



https://github.com/open-policy-agent/gatekeeper

Helm 3 Is Here! 😀

https://v3.helm.sh
https://helm.sh/docs/faq/#removal-of-tiller

Live Demo: Show me the code

Iteration 3: Kubernetes Security

https://github.com/andifalk/secure-development-on-kubernetes

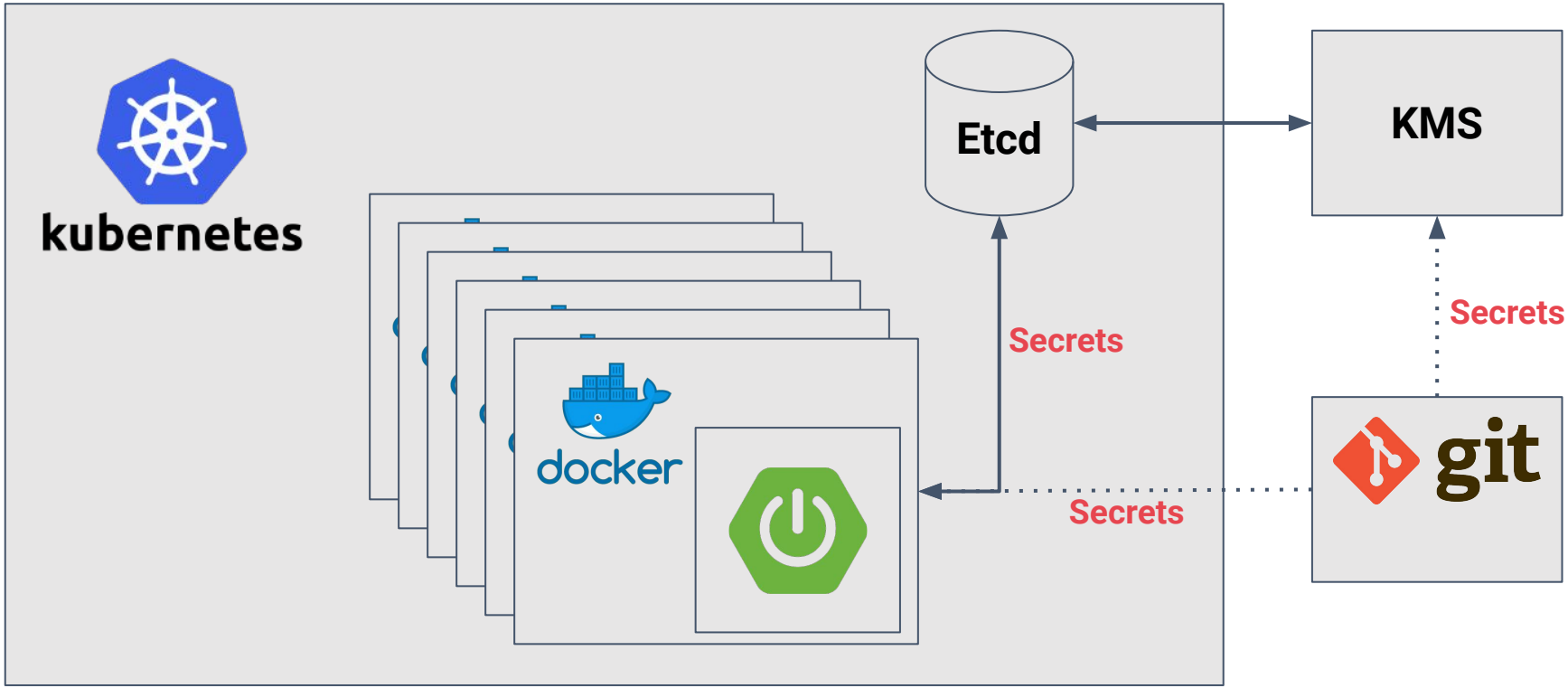
Kubernetes Security 101

- Follow Container Security 101
- Use a Managed Kubernetes Cluster
- Enable Audit Logs
- Enforce Authentication & Role Based Access Control
- Use Pod Security Policies / Open Policy Agent
- Upgrade to Helm Version 3.x (Remove Tiller)
- Monitor your Kubernetes Cluster

https://cheatsheetseries.owasp.org
https://owasp.org/www-project-top-ten
https://owasp.org/www-project-proactive-controls

The Path for Secure Development on K8s

Kubernetes Secrets



Kubernetes Secrets

```
apiVersion: v1
kind: Secret
metadata:
  name: hello-spring-cloud-kubernetes
  namespace: default
type: Opaque
data:
  user.username: dXNlcg==
  user.password: azhzX3VzZXI=
  admin.username: YWRtaW4=
  admin.password: azhzX2FkbWlu
```

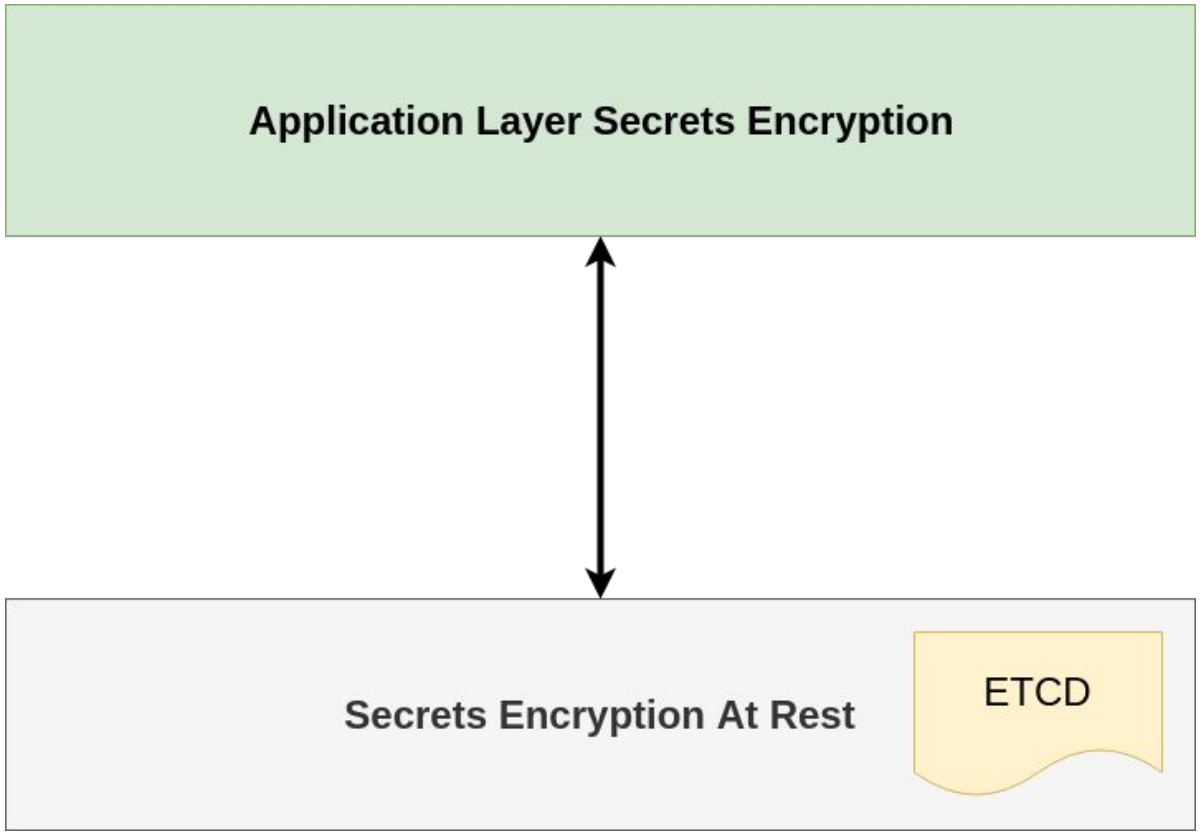
https://kubernetes.io/docs/concepts/configuration/secret

Pay Attention to Spring Boot Actuator

```
{
      "name": "applicationConfig: ...",
      "properties": {
        "greet.my-sec": {
          "value": "geheim",
          "origin": "class path resource ..."
        },
        "greet.password": {
          "value": "******",
          "origin": "class path resource ..."
        }
}
```

http://localhost:8080/actuator/env

Encryption Layers



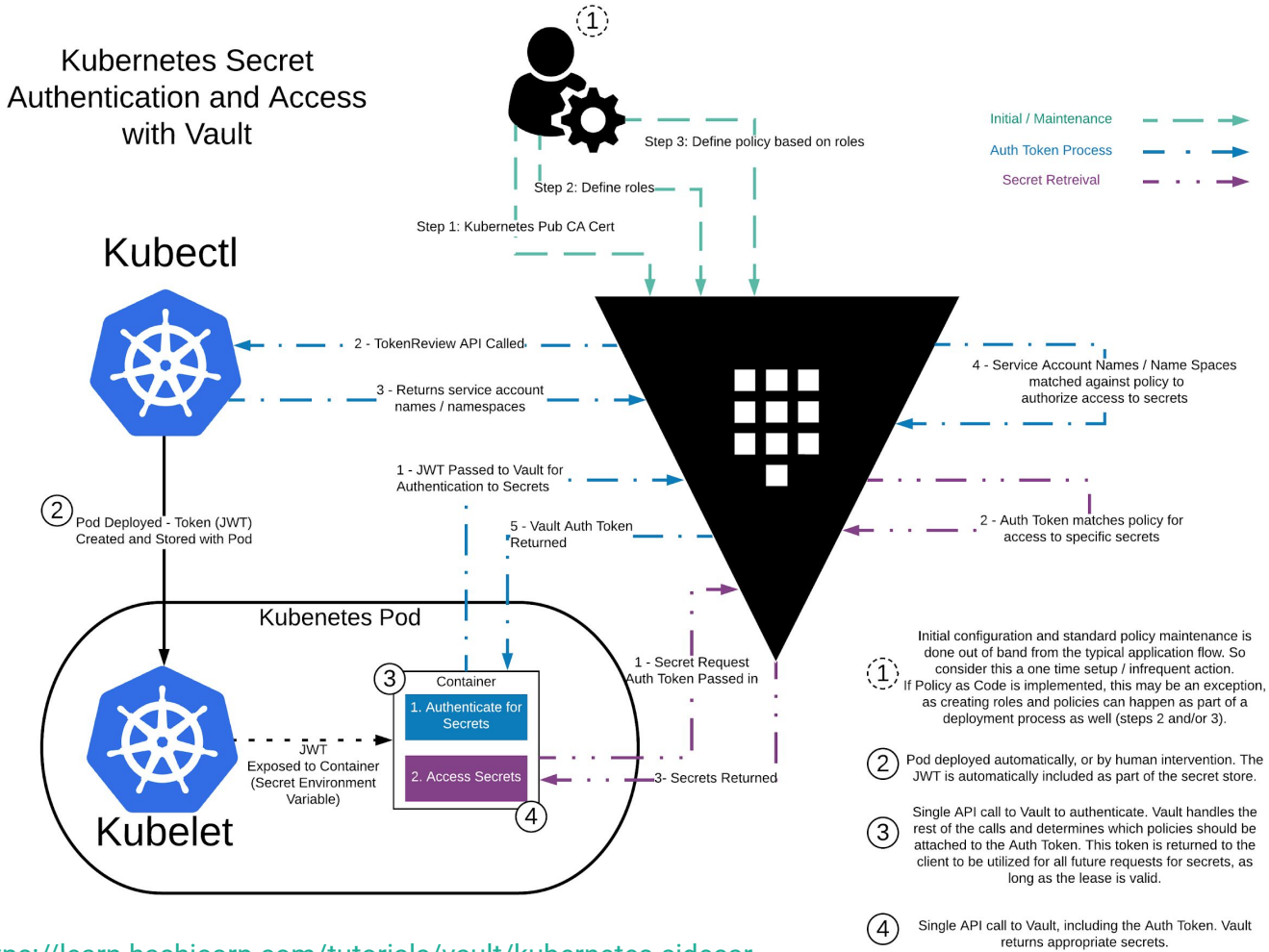
Envelope Encryption On Kubernetes

https://cloud.google.com/kms/docs/envelope-encryption
https://kubernetes.io/docs/tasks/administer-cluster/kms-provider

Key Management System (KMS) Providers

- Azure Key Vault
- Google Cloud KMS
- AWS KMS
- Hashicorp Vault
- ...

https://github.com/Azure/kubernetes-kms
https://github.com/Azure/kubernetes-keyvault-flexvol
https://cloud.google.com/kms
https://aws.amazon.com/de/kms
https://learn.hashicorp.com/vault/kubernetes/external-vault



https://learn.hashicorp.com/tutorials/vault/kubernetes-sidecar

NOVATEC

What about Secrets in 

- Sealed Secrets
- Helm Secrets
- Kamus
- Sops
- Hashicorp Vault

https://learnk8s.io/kubernetes-secrets-in-git
https://github.com/bitnami-labs/sealed-secrets
https://github.com/futuresimple/helm-secrets
https://github.com/Soluto/kamus
https://github.com/mozilla/sops
https://www.vaultproject.io

Kubernetes Secrets - Best Practices

- Encrypt Secret Data at Rest & in Transit
  - Only Base64 encoded by default in Etcd!
- Restrict interactions with secrets API (RBAC)
- Mount secrets instead of ENV Mapping

https://kubernetes.io/docs/concepts/configuration/secret/#best-practices
https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data

Summary

Summary / Key Insights

- Kubernetes is Complex - Check Alternatives!
- Follow Application Security 101
- Follow Container & Kubernetes Security 101
- Ensure your secrets are encrypted in K8s
- Never store secrets in Source Control (Git, …)
- Check out the Demos: https://github.com/andifalk/secure-development-on-kubernetes

Books and Online References

Books and Online References (1)

- Kubernetes Security, O'Reilly, 2018, ISBN: 978-1-492-04600-4
- Container Security, O'Reilly, 2020, ISBN: 978-1492056706
- https://github.com/andifalk/secure-development-on-kubernetes
- Crafty Requests: Deep Dive Into Kubernetes CVE-2018-1002105 - Ian Coldwater (Video)
- Ship of Fools: Shoring Up Kubernetes Security - Ian Coldwater (Video)
- https://kubernetes.io/docs/concepts/security/overview/#the-4c-s-of-cloud-native-security
- https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster
- https://opensource.com/article/18/3/just-say-no-root-containers
- https://github.com/GoogleContainerTools/jib
- https://anchore.com/opensource/
- https://github.com/coreos/clair
- https://github.com/aquasecurity/trivy
- https://www.owasp.org/index.php/OWASP_Docker_Top_10

Books and Online References (2)

- https://kubernetes.io/docs/tasks/configure-pod-container/assign-cpu-resource
- https://kubernetes.io/docs/tasks/configure-pod-container/assign-memory-resource
- https://kubernetes.io/docs/tasks/configure-pod-container/security-context
- https://kubernetes.io/docs/concepts/policy/pod-security-policy
- https://kubernetes.io/docs/reference/access-authn-authz/rbac/
- https://kubernetes.io/docs/concepts/configuration/secret
- https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data
- https://cloud.google.com/kms/docs/envelope-encryption
- https://kubernetes.io/docs/tasks/administer-cluster/kms-provider
- https://github.com/Azure/kubernetes-kms
- https://cloud.google.com/kms
- https://aws.amazon.com/de/kms

Thank You very much!
Questions?