ELASTX

# Automated and sustainable IT

- Automated
- Swedish operations and support
- No lock-in
- Security

# Intro

Säkerhet i en cloud native miljö

- Fokus på open source verktyg
- Hands on på det mesta
- Hyfsat tempo
- Fristående labbar (förutom 1 & 2)
- Preppa kommande labb under genomgång
    - (starta minikube)

GIT: https://github.com/elastx/tech-fika

# Dagens övningar

- Admission Controllers (hands-on)
- Pod Security Policies (hands-on)
- Secure Containers
  - Kata & gVisor
- Network Trust
  - Network Policy (hands-on)
  - Istio
- FIKA (hands-on)
- Image trust
- Image Security
  - MicroScanner (hands-on)
- Hardening Kubernetes
  - Kube-Bench (hands-on)
- Runtime Security
  - Falco

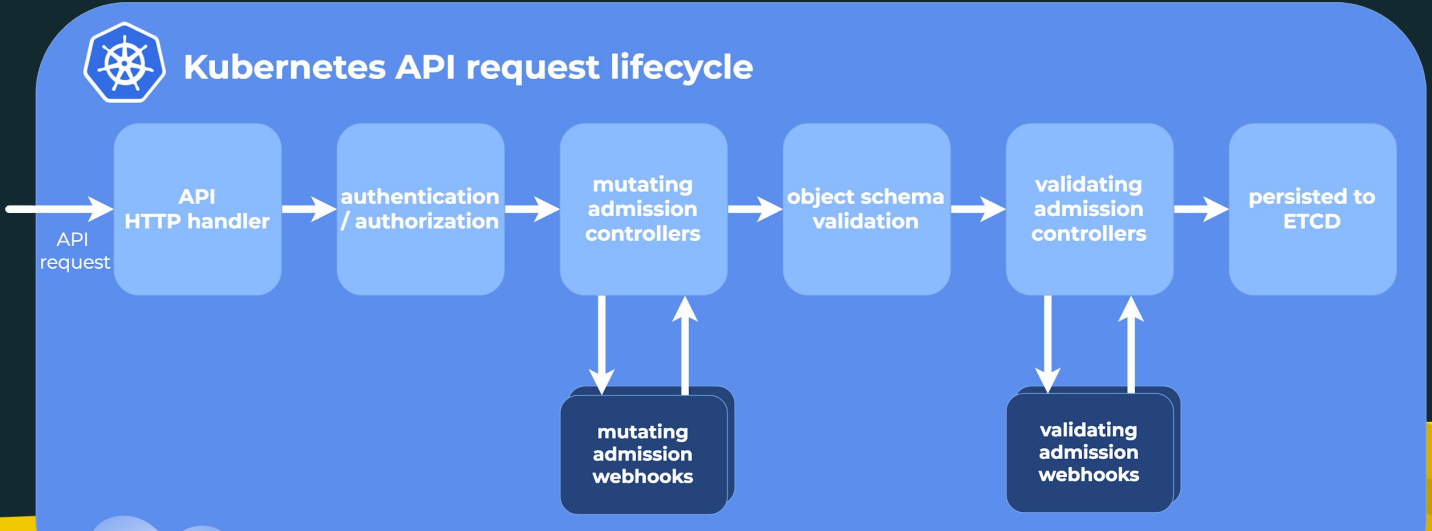
YOUR SECURITY CAN'T GET BREACHED
IF YOU DON'T GOT ANY SECURITY
memegenerator.net

# Admission Controllers

Plugin som aktiverar funktioner i Kubernetes.

Finns ett antal som laddas som standard och dom som kan laddas efter behov.

kube-apiserver --enable-admission-plugins=...


Kubernetes API request lifecycle
API request
API HTTP handler
authentication / authorization
mutating admission controllers
object schema validation
validating admission controllers
persisted to ETCD
mutating admission webhooks
validating admission webhooks

# Admission Controllers

- AlwaysPullImages (bra i multitenant miljö)
- NamespaceAutoProvision
- PodSecurityPolicy
- DenyEscalatingExec
- EventRateLimit

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

# Pod Security Policy

Begränsar vad en pod får göra

Är normalt inte aktiverad som standard

Den aktiveras i “deny-all” läge

En pod måste matcha en policy

- privileged
- allowPrivilegeEscalation
- runAsUser
  - MustRunAsNonRoot
- runAsGroup
- volumes

# Secure Containers

To get better pod isolation

- Kata Container
  - A small VM per Pod
- gVisor
  - a user-space kernel for containers

# Network Policies

- By default all open, everything can talk to everything!
- Check that your network stack support policies (calico, etc)
- Stateful rules
- Use podSelectors to match which pods should be affected
- Egress and Ingress policies
- Once a policy is applied, everything is blocked except what is allowed in the policy
- Target rules based on namespaces, pod selectors or ip-ranges (cidr) and tcp/udp ports
- Recommend to use namespace/pod selector as ip-ranges is difficult because of NAT and the dynamic nature of IP’s in kubernetes

# Istio service mesh

- Load balancing
- Service-to-service authentication
- Monitoring
- Traffic behavior control
- Rate limit
- Quota

GRATIS FIKA?
JAG KOMMER
quickmeme.com

# Image trust

- DCT, Docker Content Trust
  - When enabled you can only run signed images
- Notary
  - Based on TUF (The Update Framework)
- Portieris (IBM)
  - Admission Controller that only works with IBM Notary service
- Grafeas
  - An open-source API to audit and govern your software supply chain
  - https://github.com/kelseyhightower/grafeas-tutorial

# Image security

- Scan image for vulnerabilities
  - Include as late step in CI/CD pipeline
  - Fail build on defined severity level
  - Follow up on failed builds
  - Running containers may also be vulnerable

- Multiple ways of implementation
  - SaaS Registry (Quay.io, Docker Hub Private)
  - SaaS Policy Engine (MicroScanner)
  - Open-Source (Anchore Engine, Clair)

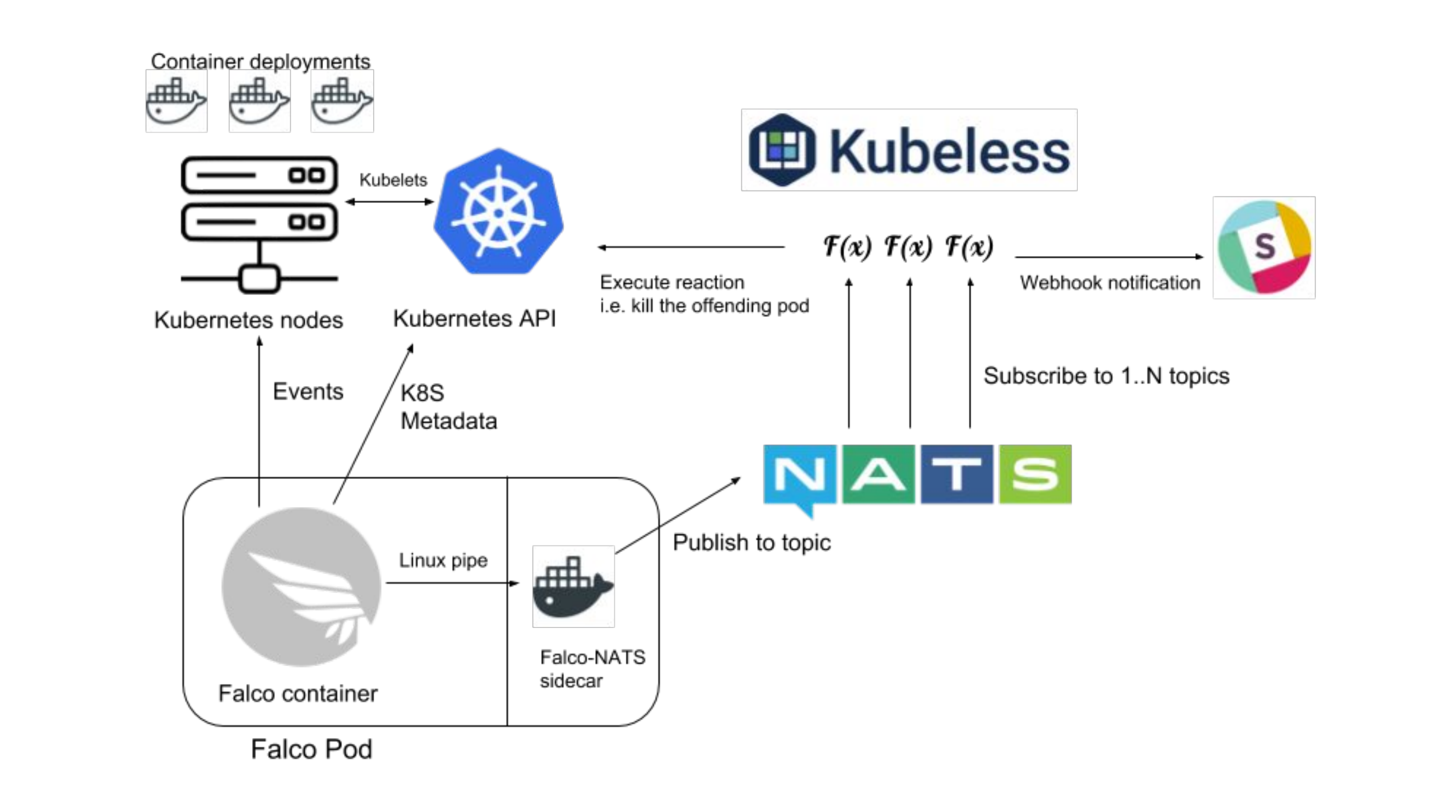
# Hardening basics

- Do not run containers as root
- Protect etcd from the rest of the cluster (etcd holds critical data which can be used to gain access to the system, either by reading or modifying), it should only be reached from master API services
- Use network policies
- Use a trusted registry
- Run a service mesh for a “zero trust” environment
- Use TLS everywhere! Identity trust is more important than encryption
- Use PodSecurityPolicy
- Disable ABAC and transition to RBAC
- Run kube-bench to get CIS benchmark score
- Check master and worker node IPTables rules so that containers can not access kubelet api for example

# Container runtime security

- monitoring the behavior at execution time
  - runtime visibility agent
  - response engine
  - security playbooks
- Falco
  - Monitors system calls
  - Transparent for the container
  - Container native, understands the env.
  - Runs as a DaemonSet
  - Installs a kernel module

# Runtime Security Stack



# Falco rules

https://github.com/draios/falco-extras/blob/master/rules/

```
kubectl exec -it redis-master-0 cat /etc/passwd
```

```
$ kubectl logs falco-gsppx
...
09:35:58.678355216: Notice Unexpected process spawned in redis container
(command=cat /etc/passwd pid=9811 user=<NA> k8s.pod=redis-master-0
container=ab6769a7c1d2
image=bitnami/redis@sha256:e50375d55ea5e5912f985ae1bf8f7c95a00ec2ff
7f4c18e3c9afe7b98dcdaf43) k8s.pod=redis-master-0
container=ab6769a7c1d2
```

# Falco match containers

```
- macro: app_nginx
  condition: container and container.image contains "nginx”

macro: node_app_frontend
condition: k8s.ns.name = node-app and k8s.pod.label.role = frontend

- list: container_image_whitelist
  items:
["sha256:8ac48589692a53a9b8c2d1ceaa6b402665aa7fe51ccc03002300856d8c7",
"sha256:f3fcd0775c4e15d4b73d2b62f60efb1872328be52a0fc9a2b0951bbcc1e"]
```

# Thanks for listening

# Bra källor

https://github.com/freach/kubernetes-security-best-practice

https://sysdig.com/blog/kubernetes-security-guide/

https://sysdig.com/blog/oss-container-security-runtime/

https://dev.to/petermbenjamin/kubernetes-security-best-practices-hlk

https://sysdig.com/blog/container-security-docker-image-scanning/