DevOpsCon

redhat

Kubernetes Security

From Image Hygiene to Network Policies

Michael Hausenblas **@mhausenblas**

Developer Advocate, Red Hat

2018-05-29, DevOpsCon, Berlin

$ whois mhausenblas

- Developer Advocate @ Red Hat (Go, Kubernetes, OpenShift)
- Developer Advocate @ Mesosphere (Mesos, DC/OS, Kubernetes)
- Chief Data Engineer @ MapR (HDFS, HBase, Drill, etc.)
- Applied research (4y in Ireland, 7y in Austria)
- Nowadays mainly developing tools in Go (Python, Node, Java, C++)
- Kinda developer turned ops (aka appops)

admin

developer

architect

SRE

infosec

PM

Overview and terminology

container images

running container

authn & authz

communication

apps

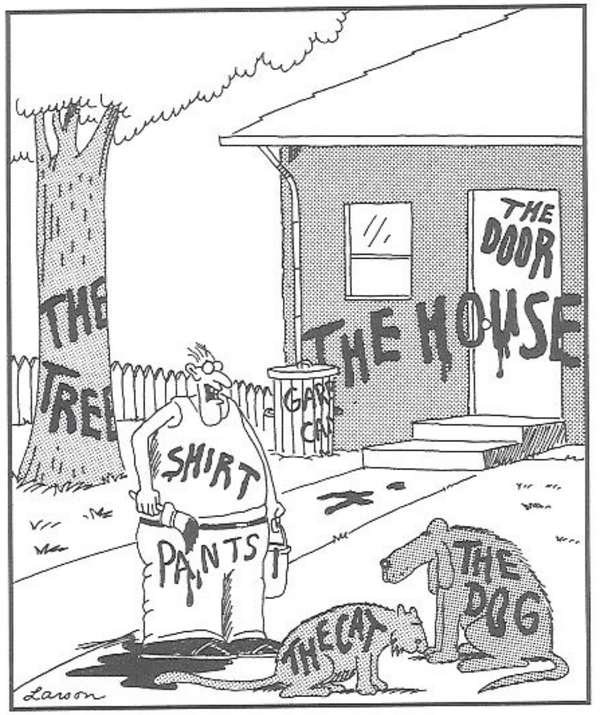
control plane

An expert is a person who has made all the mistakes which can be made, in a narrow field.

Updated from Niels Bohr, Danish physicist, Nobel Prize for Physics in 1922

Terminology

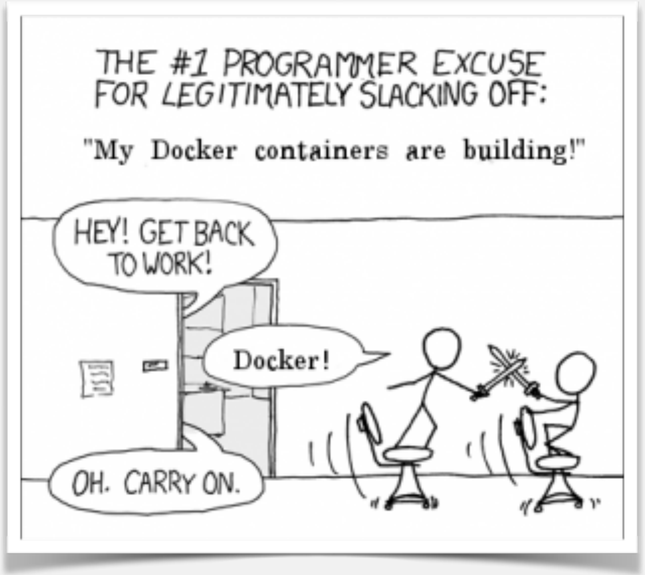
- build-time vs run-time
- immutability & automation
- responsibilities
- moving parts



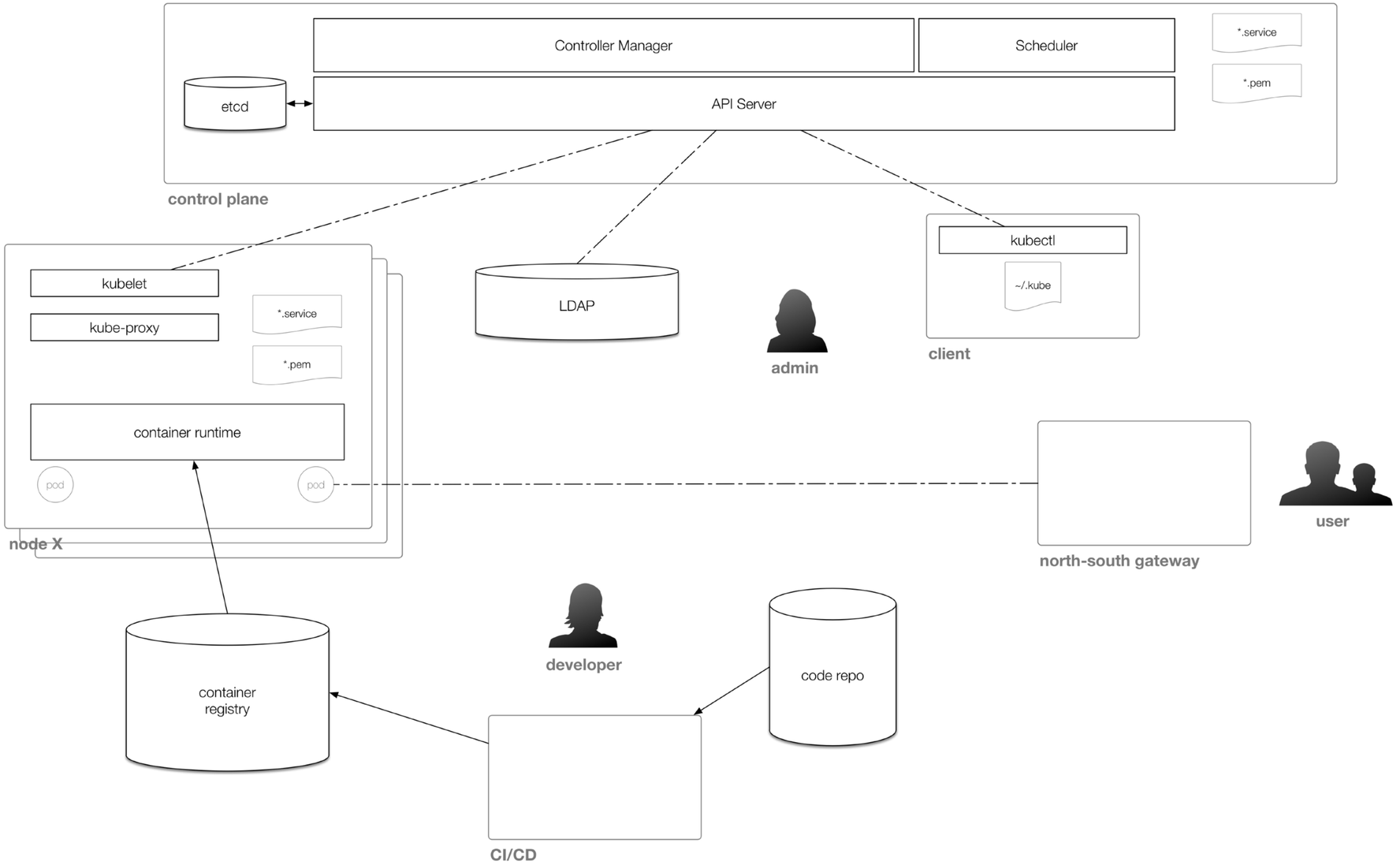
"Now! … *That* should clear up a few things around here!"

Responsibilities

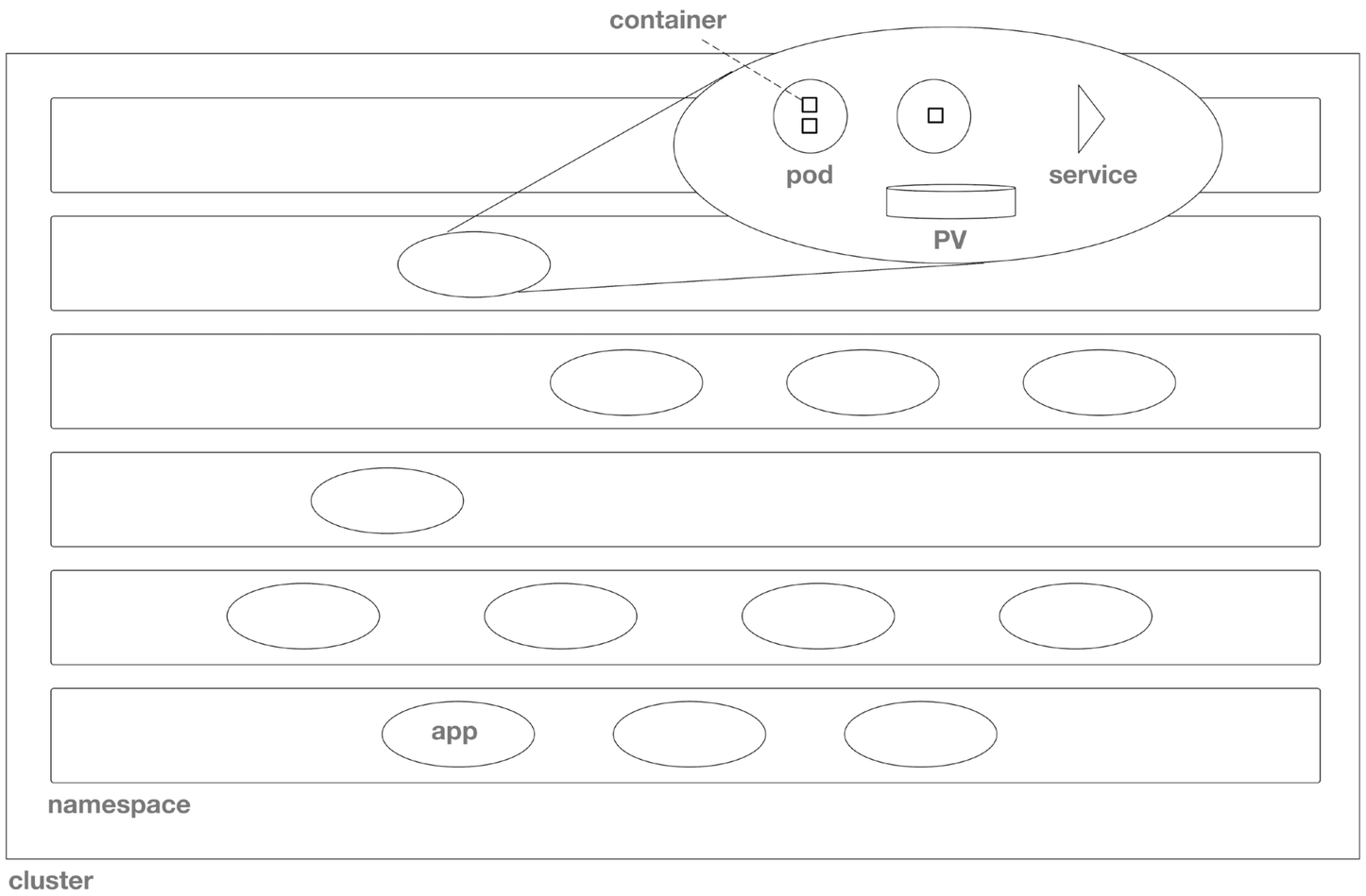
- infrastructure admin
- namespace admin
- developer



Moving parts—physical view



Moving parts—logical view



Building container images

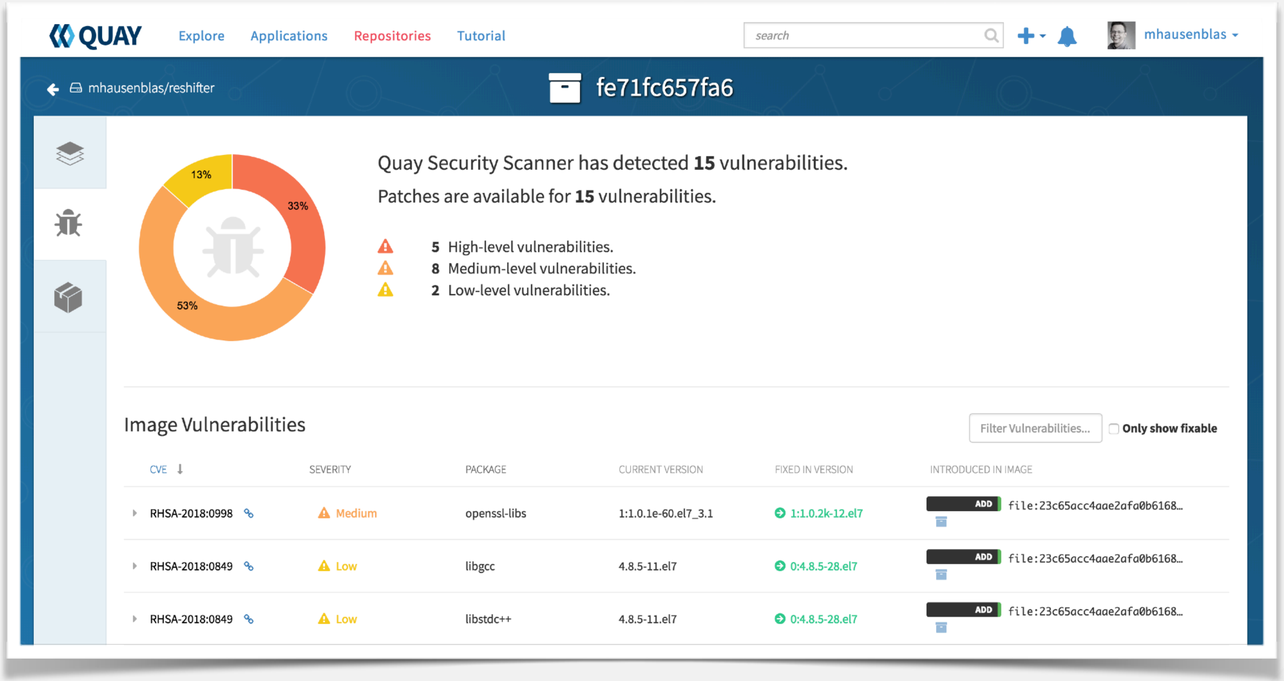
What?

Build container images that …

- have a small attack surface
- are checked for vulnerabilities
- are reproducible

Good practices

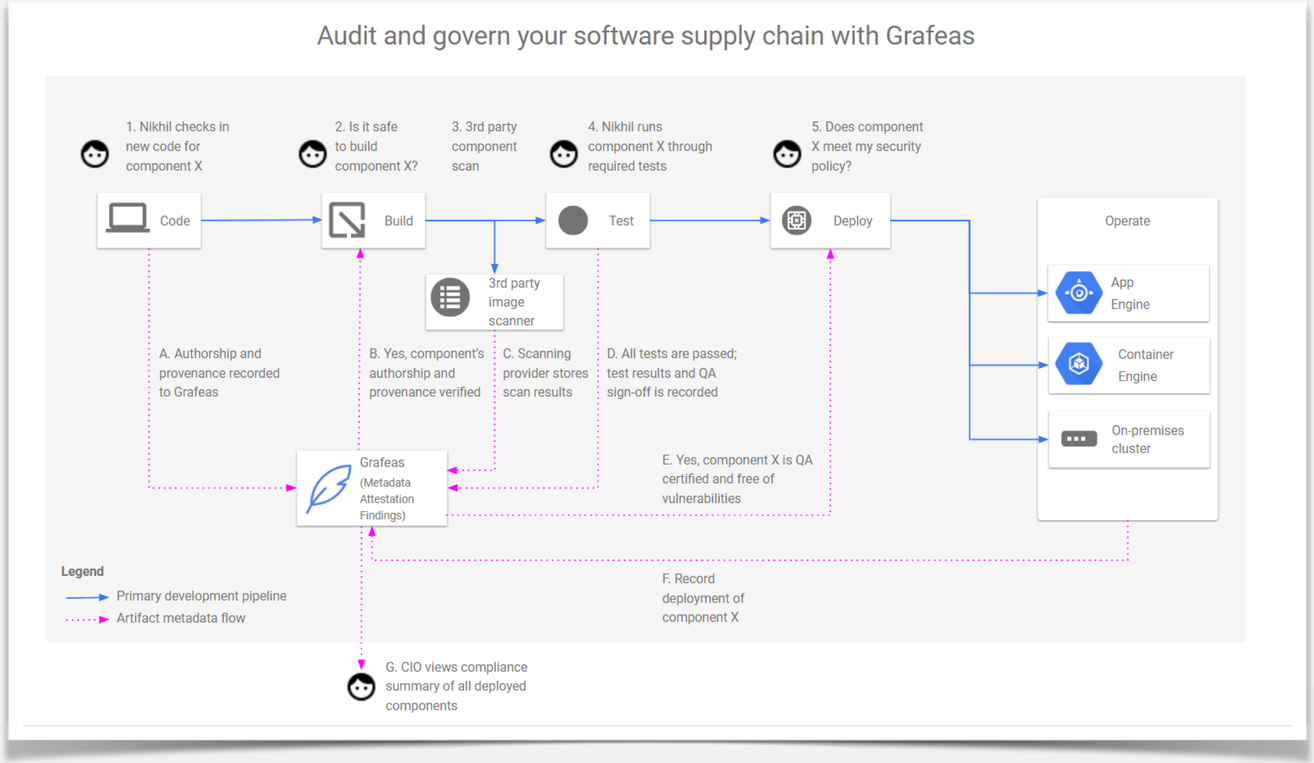
- use trusted base images
- define user in image
- perform automated CVE scans
- use private registries
- pin dependencies (reproducible builds)



redhat.

Grafeas

- supply chain management
- structured metadata API for annotating artefacts + enforcement (Kritis)
- backed by Google, JFrog, Red Hat, IBM, Black Duck, Twistlock, Aqua Security



Source: Introducing Grafeas: An open-source API to audit and govern your software supply chain

https://grafeas.io

Tooling

- https://docs.docker.com/docker-cloud/builds/image-scan/
- https://github.com/coreos/clair
- https://www.open-scap.org/tools/
- https://www.aquasec.com/use-cases/continuous-image-assurance/
- https://neuvector.com/container-compliance-auditing-solutions/
- https://github.com/theupdateframework/notary
- https://github.com/in-toto

Further reading

- Establishing Image Provenance and Security in Kubernetes
- Image Management & Mutability in Docker and Kubernetes
- Container security considerations in a Kubernetes deployment
- Building Container Images Securely on Kubernetes
- The OpenShift Build Process
- Introducing Grafeas: An open-source API to audit and govern your software supply chain

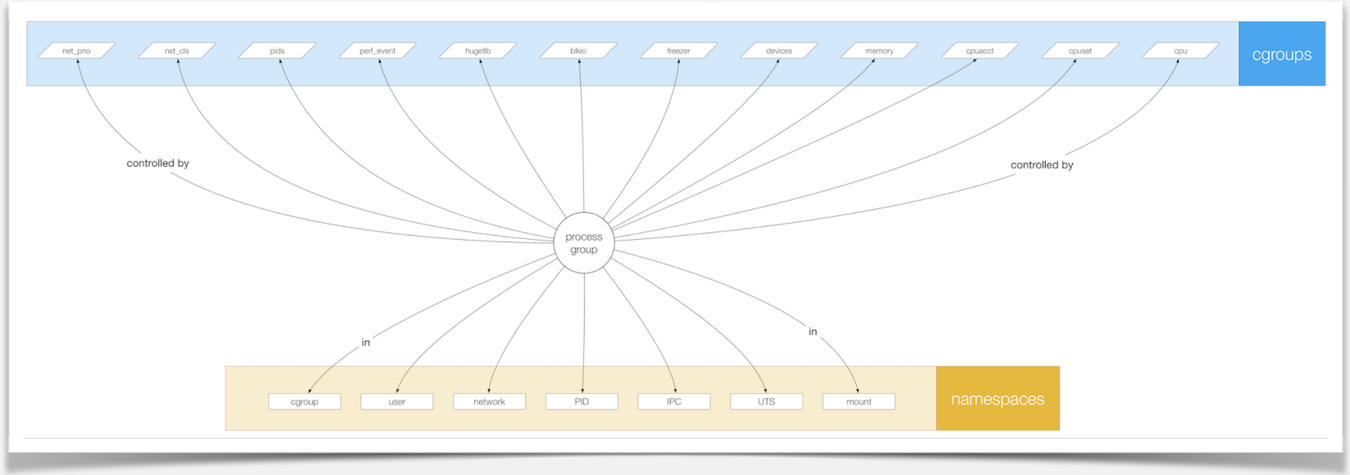
Running containers

What?

- Run containers …
  - with known origin (image & registry)
  - use least privilege to carry out the task
  - do only minimal host mounts necessary

Good practices

- verify the defaults
- don’t run as root
- use security context & policies
- use security benchmarks



http://containerz.info

Pod security context and policies

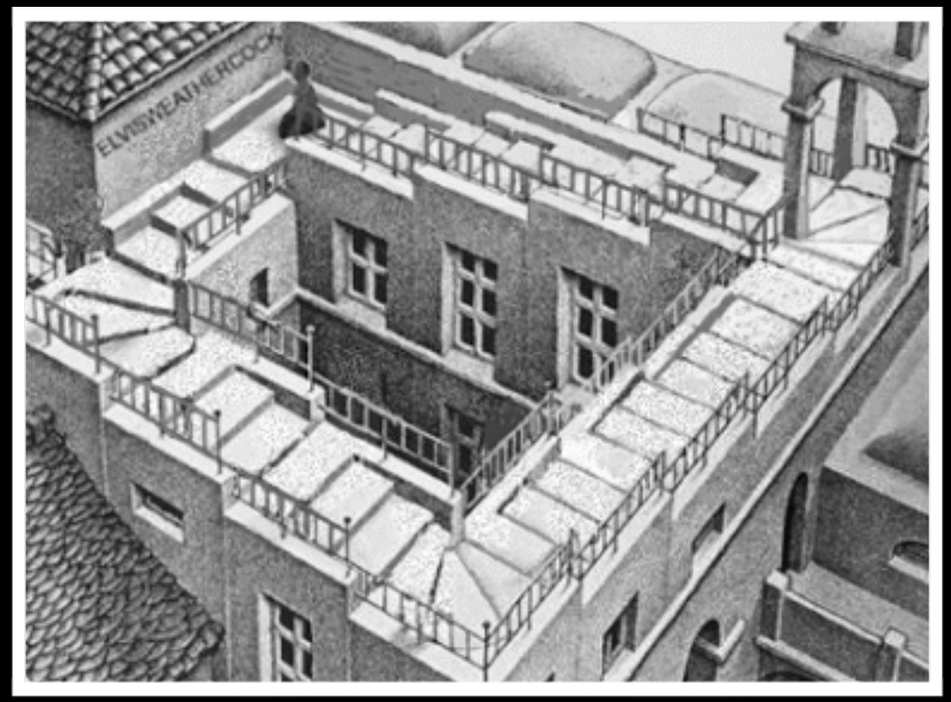
- security context
  - pod or container level
  - defines privilege and access control settings
  - seccomp, capabilities, SELinux, AppArmor
- security policies
  - cluster resource
  - via admission controller

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: security-context-demo
spec:
  sec
    r
    f
  vol
  - n
    e
  con
  - n
    i
    v
    -

    s
```

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: privileged
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
```

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



… demo time!

Tooling

- https://github.com/aquasecurity/kube-bench
- https://github.com/docker/docker-bench-security
- https://sysdig.com/opensource/falco/
- https://kubesec.io/
- https://www.twistlock.com/

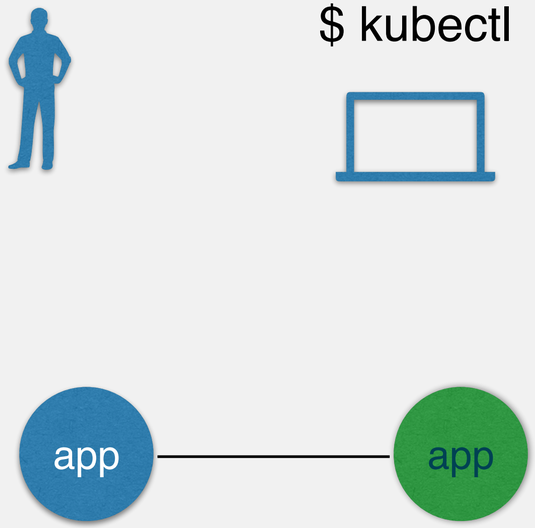
Further reading

- Just say no to root (in containers)
- Exploring Container Mechanisms Through the Story of a Syscall (slides | video)
- Improving your Kubernetes Workload Security
- Container Isolation at Scale (Introducing gVisor) (slides | video)

Authentication & Authorization

Identity

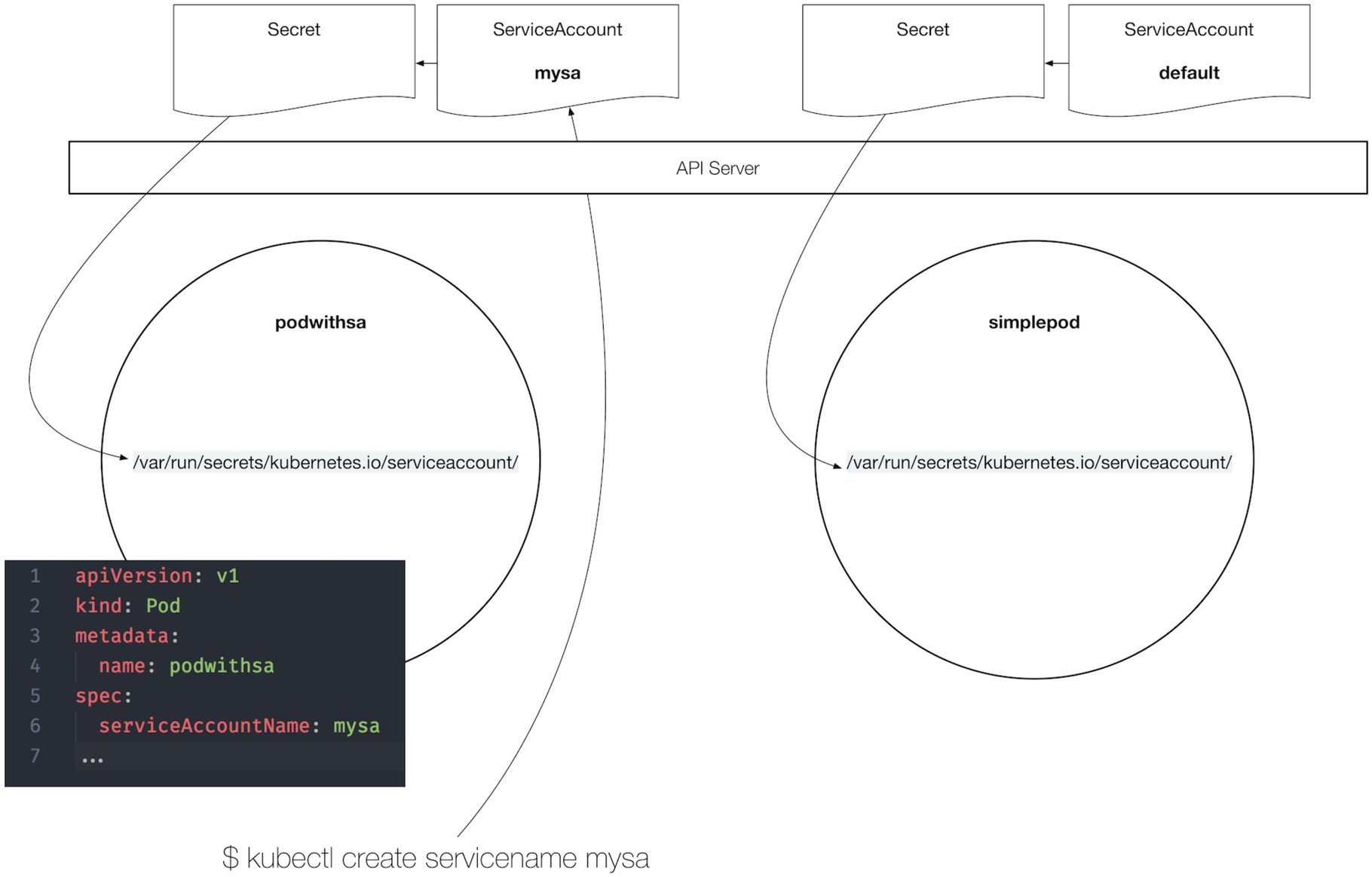
- human users
  - managed outside of Kubernetes
  - LDAP, SAML, Kerberos, etc.
- apps
  - running in containers in pods
  - first class resources via service accounts



Service accounts

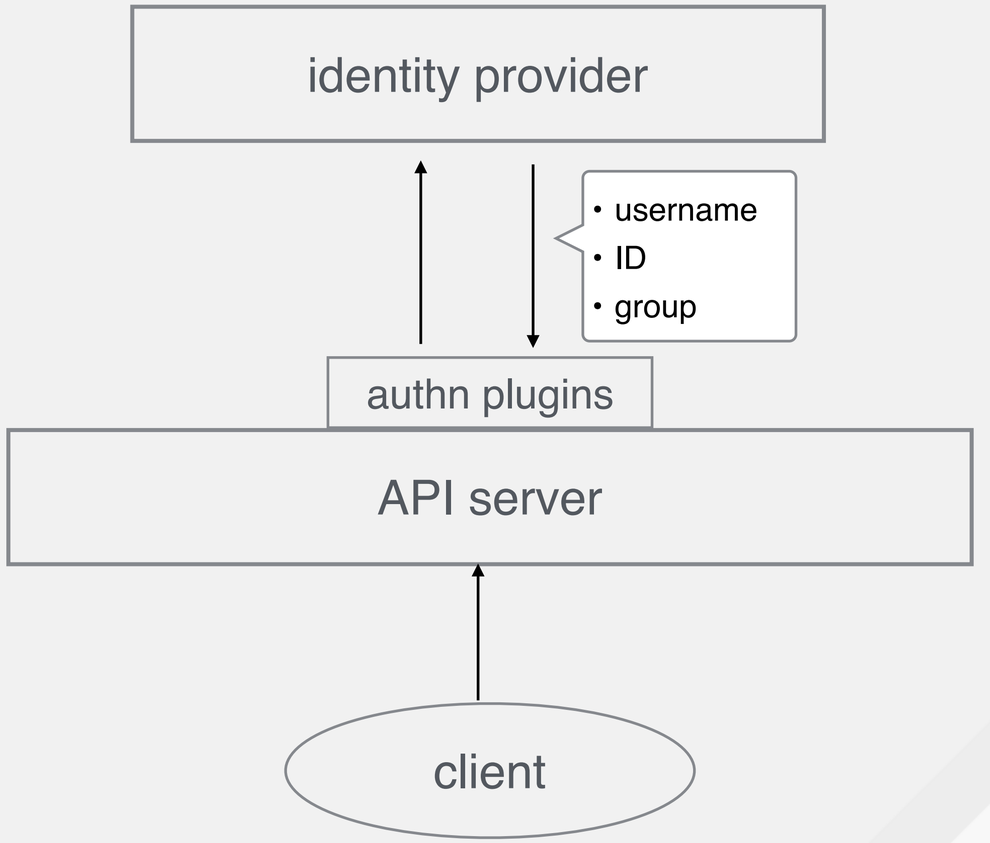
- provide identity for an app
- namespaced resources
- credentials via secret mounted into pod
- default service account per namespace

```
system:serviceaccount:$NAMESPACE:$NAME
```


Secret
ServiceAccount
mysa
Secret
ServiceAccount
default
API Server
podwithsa
/var/run/secrets/kubernetes.io/serviceaccount/
simplepod
/var/run/secrets/kubernetes.io/serviceaccount/
1 apiVersion: v1
2 kind: Pod
3 metadata:
4 name: podwithsa
5 spec:
6 serviceAccountName: mysa
7 ...
$ kubectl create servicename mysa

Authentication

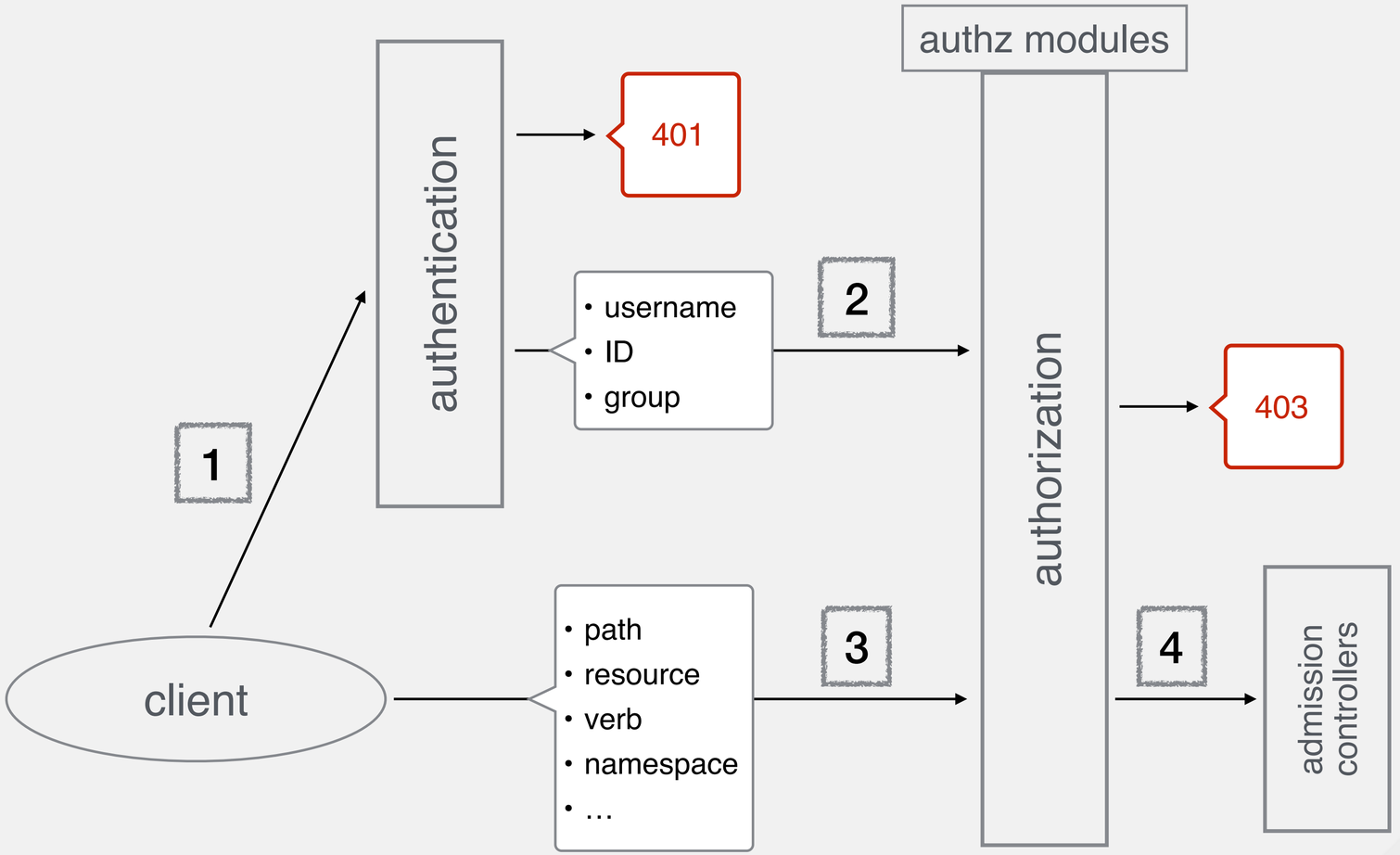
- static password/token file
- X509 client certs
- proxy+header
- OpenID Connect
- custom via Webhook



https://kubernetes.io/docs/admin/authentication/

Authorization

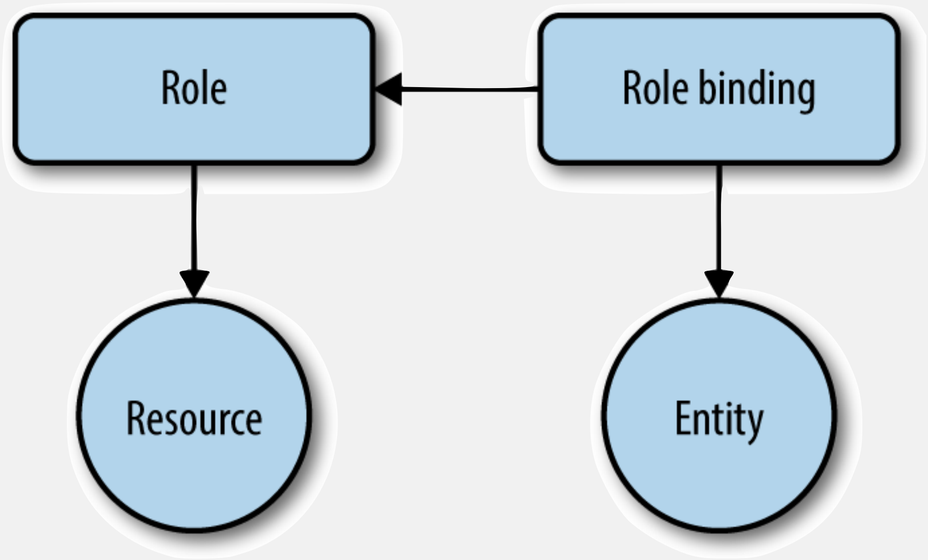
- Node (kubelet)
- ABAC (outdated)
- **RBAC**
- Webhook (external)



https://kubernetes.io/docs/admin/authorization/

RBAC

- stable as of 1.8
- entities: service accounts, user, group
- scope: namespace or cluster
- roles and bindings
- privilege escalation prevention



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

Defaults

user-facing

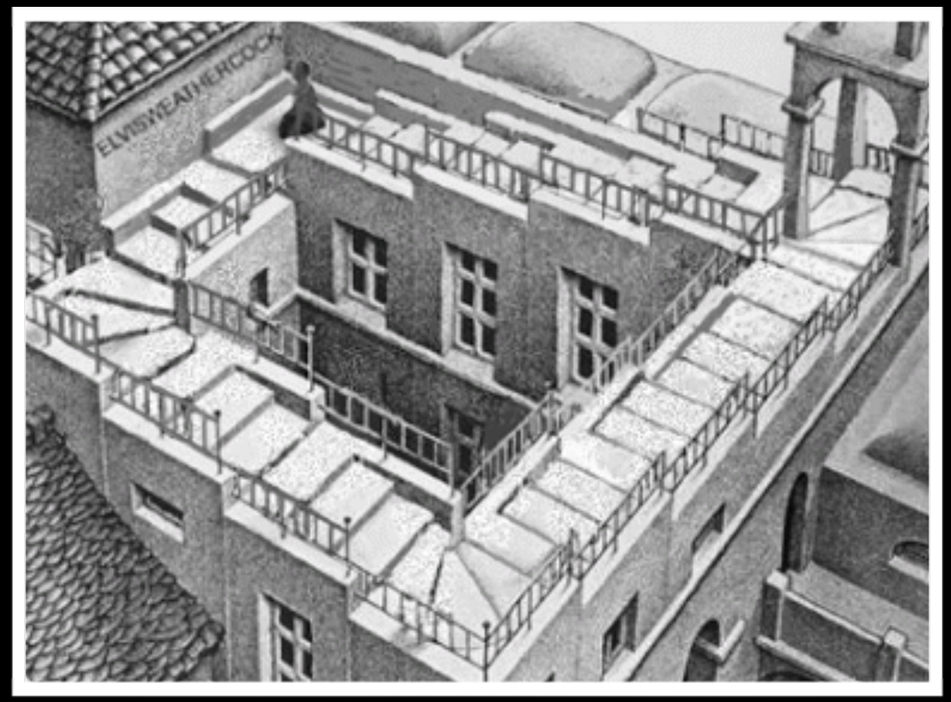
Default ClusterRole	Default ClusterRoleBinding
cluster-admin	**system:masters** group
admin	None
edit	None
view	None

core-components

Default ClusterRole	Default ClusterRoleBinding
system:kube-scheduler	**system:kube-scheduler** user
system:kube-controller-manager	**system:kube-controller-manager** user
system:node	None in 1.8+
system:node-proxier	**system:kube-proxy** user

other components

Default ClusterRole	Default ClusterRoleBinding
system:auth-delegator	None
system:heapster	None
system:kube-aggregator	None
system:kube-dns	**kube-dns** service account in the **kube-system** namespace
system:kubelet-api-admin	None
system:node-bootstrapper	None
system:node-problem-detector	None
system:persistent-volume-provisioner	None



… demo time!

Tooling

- kubectl create, kubectl auth
- https://github.com/coreos/dex
- https://github.com/heptio/authenticator
- https://github.com/liggitt/audit2rbac

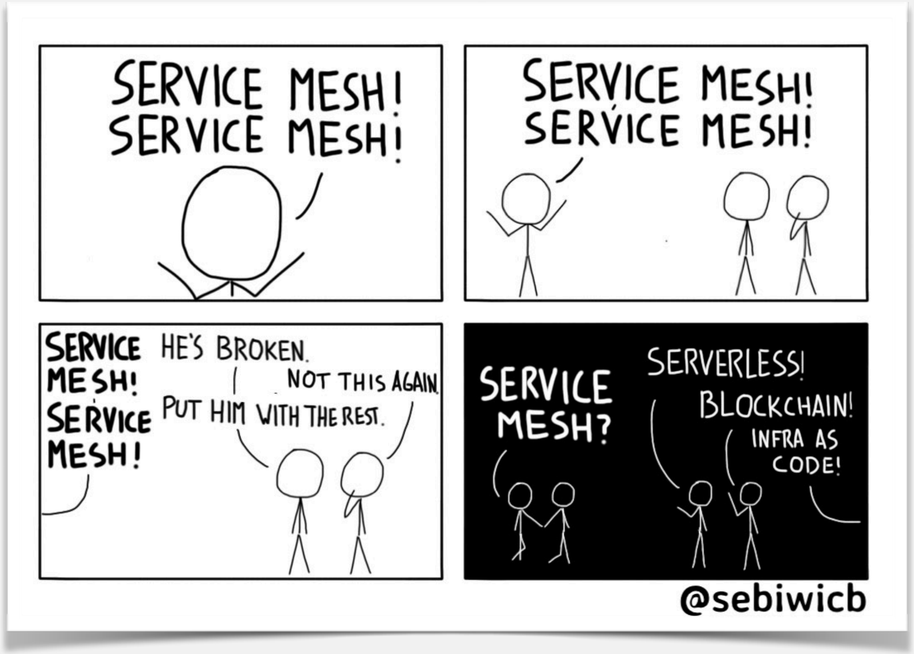
Further reading

- Controlling Access to the Kubernetes API
- Kubernetes deep dive: API Server – part 1
- Certifik8s: All You Need to Know About Certificates in Kubernetes
- Kubernetes Auth and Access Control
- Effective RBAC
- Single Sign-On for Kubernetes: An Introduction
- Let's Encrypt, OAuth 2, and Kubernetes Ingress

Communication

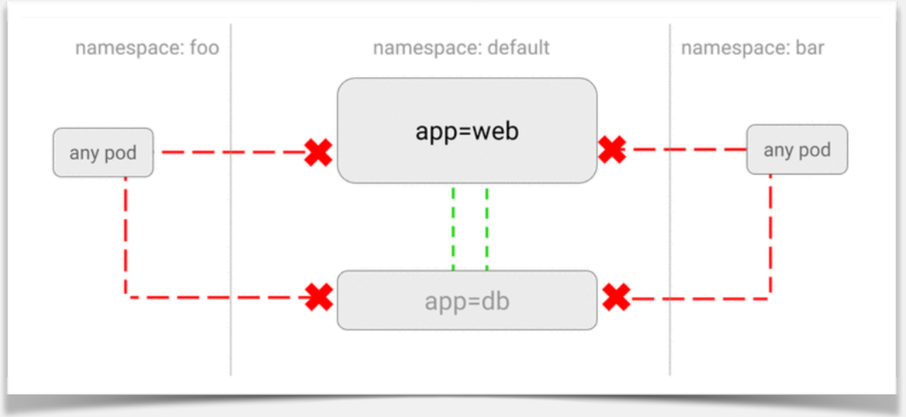
What?

- encryption on the wire (TLS everywhere)
- network policies
- service meshes



Network policies

- defines pod-to-pod communication
- enforced by network plugin

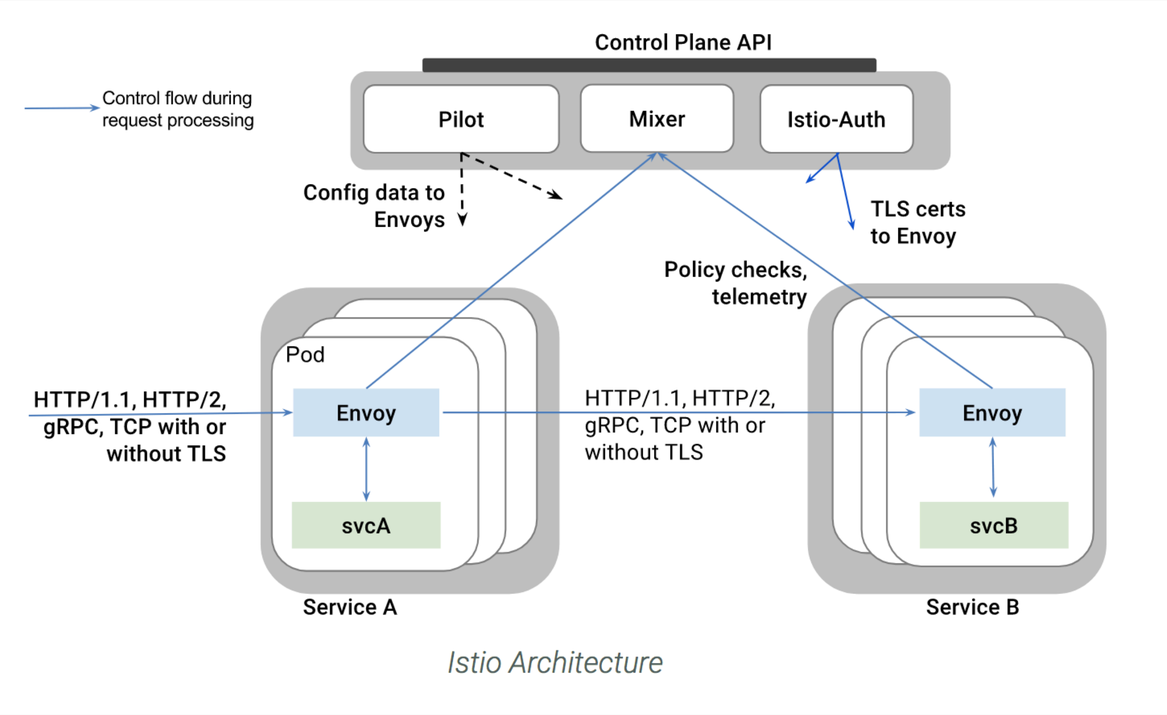


Source: Securing Kubernetes Cluster Networking by Ahmet Alp Balkan

https://kubernetes.io/docs/concepts/services-networking/network-policies/

Service meshes

- traffic management
- policy enforcement
- monitoring & tracing
- no app code changes



Istio Architecture

istio.io

Tooling

- https://github.com/aporeto-inc/trireme-kubernetes
- https://github.com/jetstack/cert-manager/
- https://spiffe.io/
- https://www.openpolicyagent.org/
- https://linkerd.io/
- https://conduit.io/

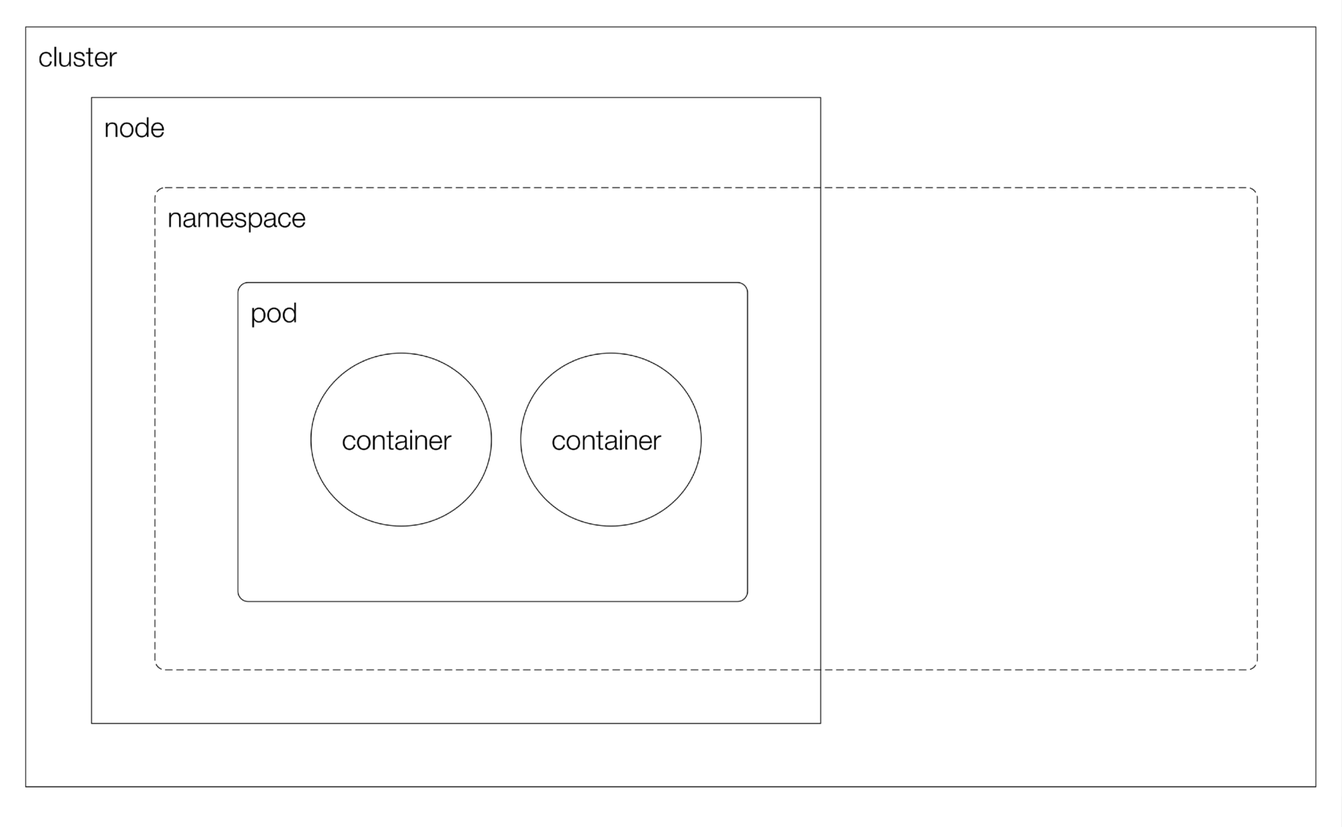
Further reading

- How Kubernetes certificate authorities work
- Securing Kubernetes Cluster Networking
- Tutorials and Recipes for Kubernetes Network Policies feature
- Kubernetes Security Context and Kubernetes Network Policy
- Kubernetes Application Operator Basics

Apps

What?

- security boundaries
- segregation
- secrets



Based on: Exploring container security: Isolation at different layers of the Kubernetes stack

Segregation

- namespace-level: visibility and access
- node-level:
  - separate sensitive workloads via affinity and taints
  - minimize blast radius (node authorizer)
- pod-level: limit communication via network policies or service mesh

Secrets

- Namespaced objects to store sensitive information
- Access via volume or environment variable
- Data is stored in tmpfs volumes
- Per-secret size limit of 1MB
- Only base64 encoded, need to enable encryption at rest

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  username: YWRtaW4=
  password: MWYyZDFlMmU2N2Rm
```

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

Tooling

- https://github.com/kelseyhightower/konfd
- https://github.com/hashicorp/vault-plugin-auth-kubernetes
- https://github.com/bitnami-labs/sealed-secrets
- https://github.com/shyiko/kubesec
- https://github.com/weaveworks/flux

Further reading

- Docs:
  - Configure a Security Context for a Pod or Container
  - Pod Security Policies
- Shipping in Pirate-Infested Waters: Practical Attack and Defense in Kubernetes
- Exploring container security: Isolation at different layers of the Kubernetes stack
- Security Best Practices for Kubernetes Deployment

Control plane

Good practices

- secure API server, etcd, dashboard
- secure kubelet
- limit access to cloud provider metadata
- limit access to metrics
- perform auditing

Tooling

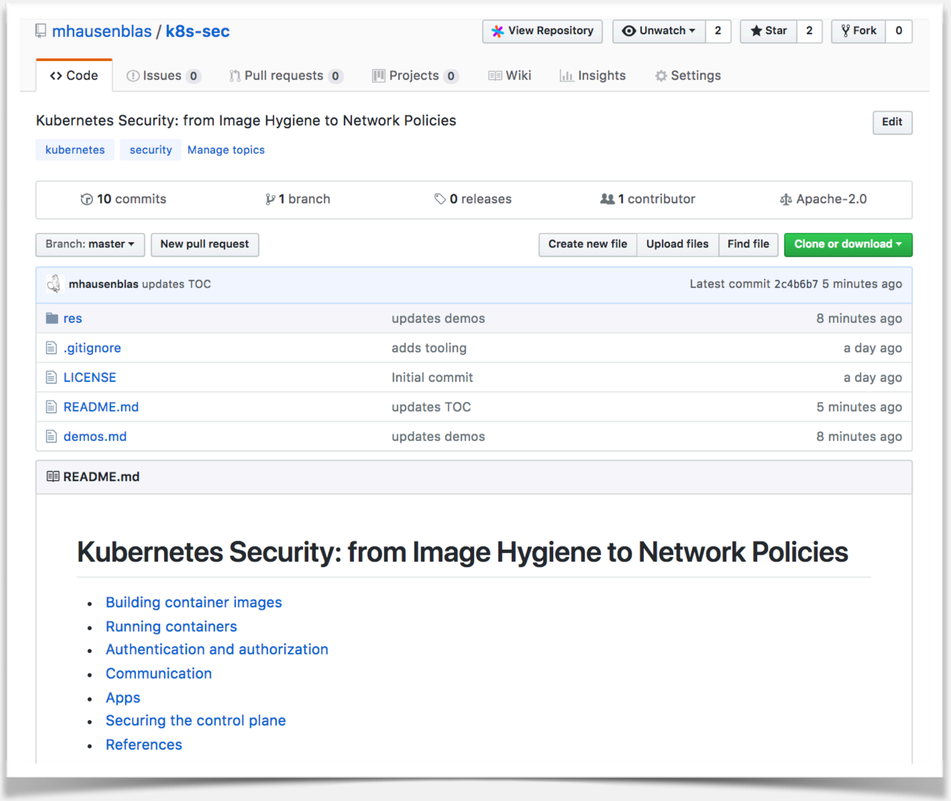
- https://github.com/bgeesaman/kubeatf
- https://github.com/Shopify/kubeaudit
- https://k8guard.github.io/
- https://www.vaultproject.io/

Further reading

- Docs:
  - Securing a Cluster
  - Encrypting Secret Data at Rest
  - Auditing
- Securing Kubernetes components: kubelet, etcd and Docker registry
- K8s security best practices
- Kubernetes Security - Best Practice Guide
- Lessons from the Cryptojacking Attack at Tesla

Resources

Demos and references



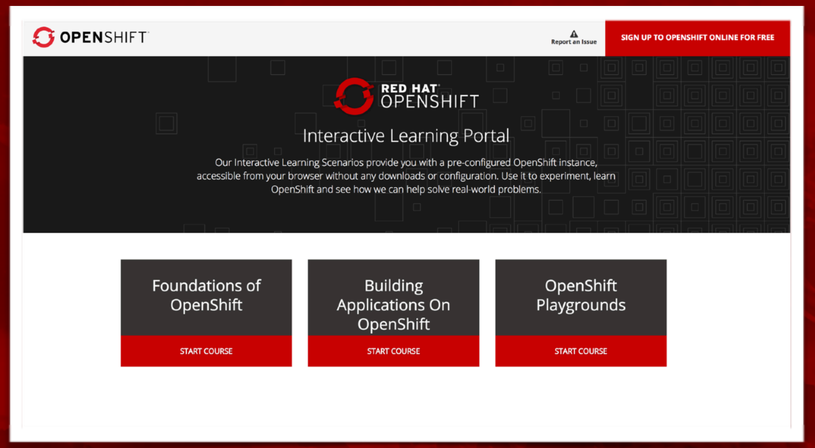
https://github.com/mhausenblas/k8s-sec

Articles

- NIST Special Publication 800-190: Application Container Security Guide
  https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf
- Hacking and Hardening Kubernetes Clusters by Example, Brad Geesaman, KubeCon 2017
  https://www.youtube.com/watch?v=vTgQLzeBfRU
- Kubernetes Security Best Practices, Ian Lewis, FOSDEM 2018
  https://www.youtube.com/watch?v=pzAwTC8KYV8
- Continuous Kubernetes Security, Andrew Martin, microXchg 2018
  https://www.youtube.com/watch?v=YtrA7eauSSg
- What Does “Production Ready” Really Mean for a Kubernetes Cluster?
  https://weave.works/blog/what-does-production-ready-really-mean-for-a-kubernetes-cluster

redhat

learn.openshift.com



plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos