SysEleven

// ContainerDay Workshops 2023 - Munich

Mastering Production Grade Deployments

K8s Deep Dive

November 23 | Christian Hörl

About

SysEleven

Estd. 2007

- based in Berlin
- 3 Datacenter regions in Germany
- Managed Hosting, Openstack
- Managed Kubernetes with “MetaKube”

METAKUBE

Topics

- **K8s on-board features**
  - Pod Design: Good Practices
  - Deployment Strategy & Affinity
  - Resources: Requests, Limits, Priority
  - Scaling
  - Disruption Budget, Security

- **3rd-party Tools**
  - external-dns, cert-manager
  - ingress-nginx & canary deployments
  - Helm & Helmfile

Production-grade Deployments?

Production readiness?

Redundancy

Redundancy

- having “more than 1”
- on different layers
- K8s *Nodes* and *Replicas*

High Load & Scaling

High Load & Scaling

- react
- scale fast
- automate
- K8s *Pod / Node auto-scaling*

Persistence

Persistence

- centralized, decentralized
- distribute persistent data
- move with the consumer
- K8s *persistentVolumes*

Self-healing Architecture

Security

Security

- enable, configure, use
- intercept workload behavior
- K8s *securityContext*

Backup & Recovery

Quality Assurance

The Pillars of Production-Readiness

K8s features

3rd-party tools

reactive & supportive

automation & continuous deployment

K8s features

Get ready

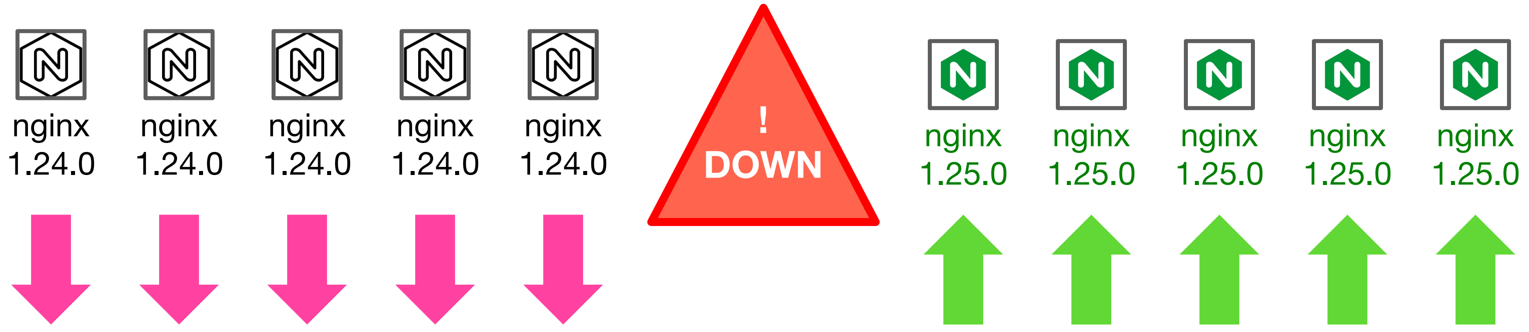
Hands-on repo
github.com/syseleven/containerday-workshops

Kubeconfig
sys11.it/production-grade-2023

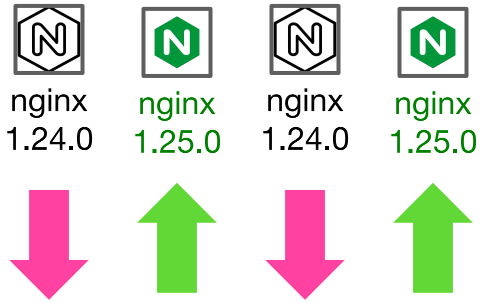
Pod Configuration Best Practices

Deployment Strategy

Deployment strategy: Recreate



Deployment strategy: Rolling Update

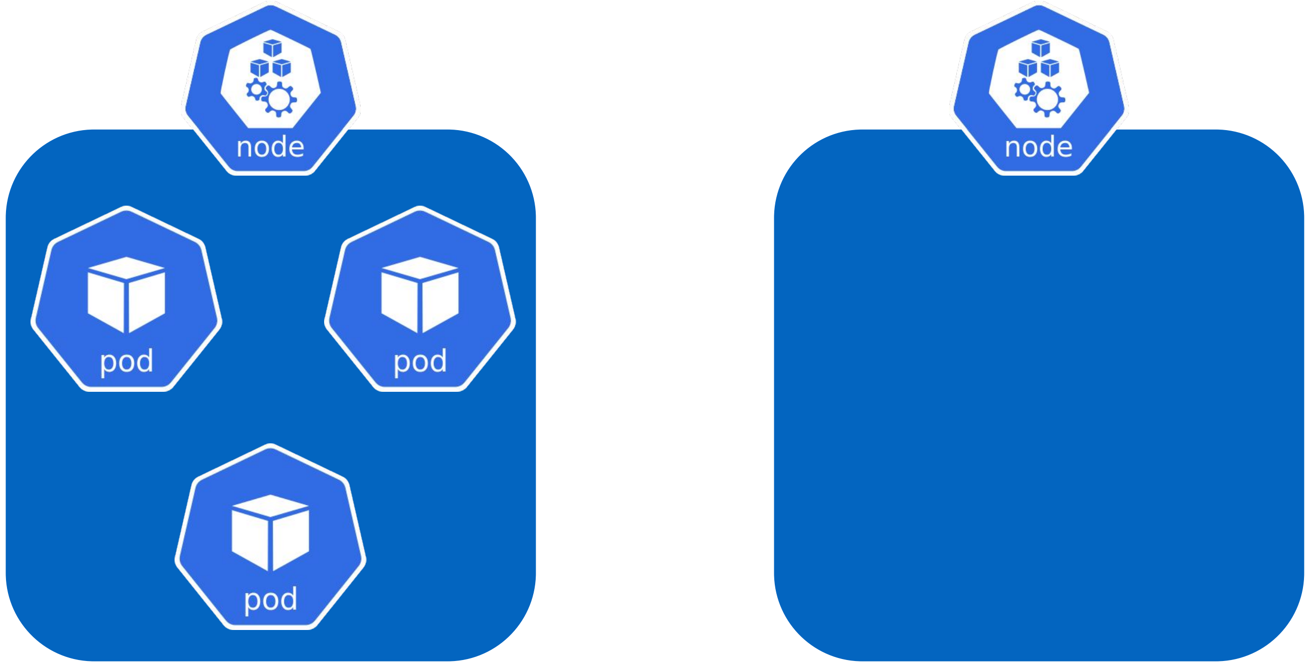


Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/100-deployment-strategy

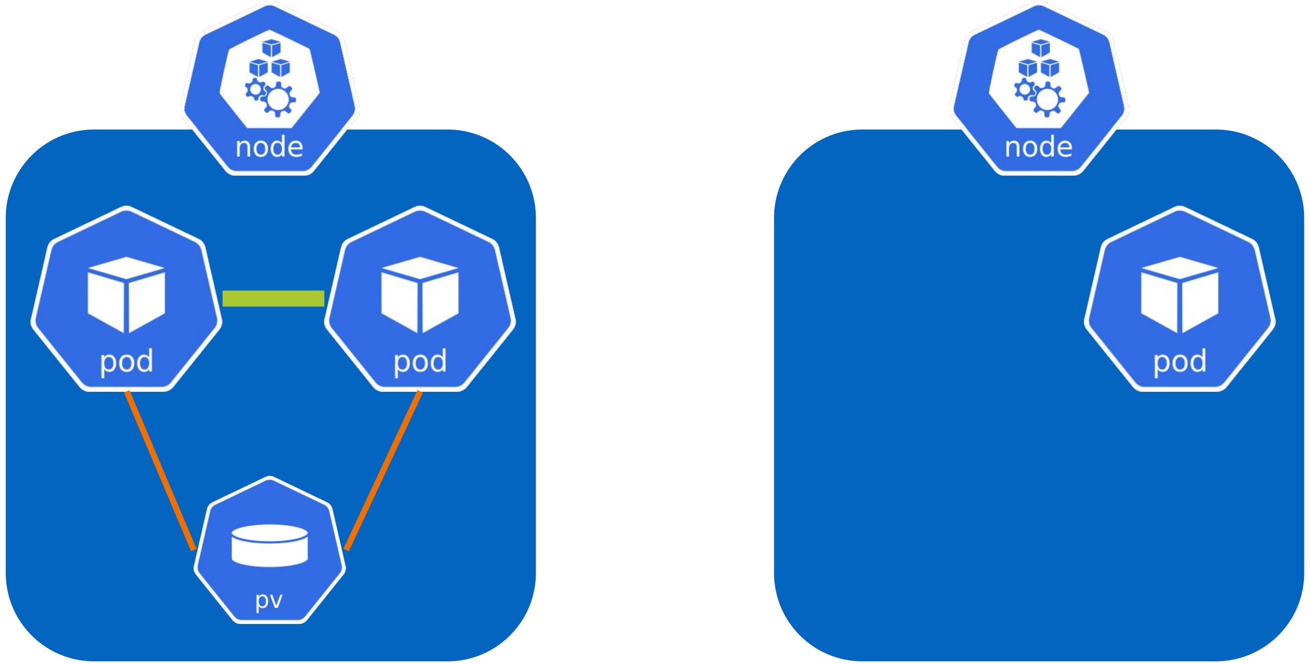
Affinity

Node Affinity



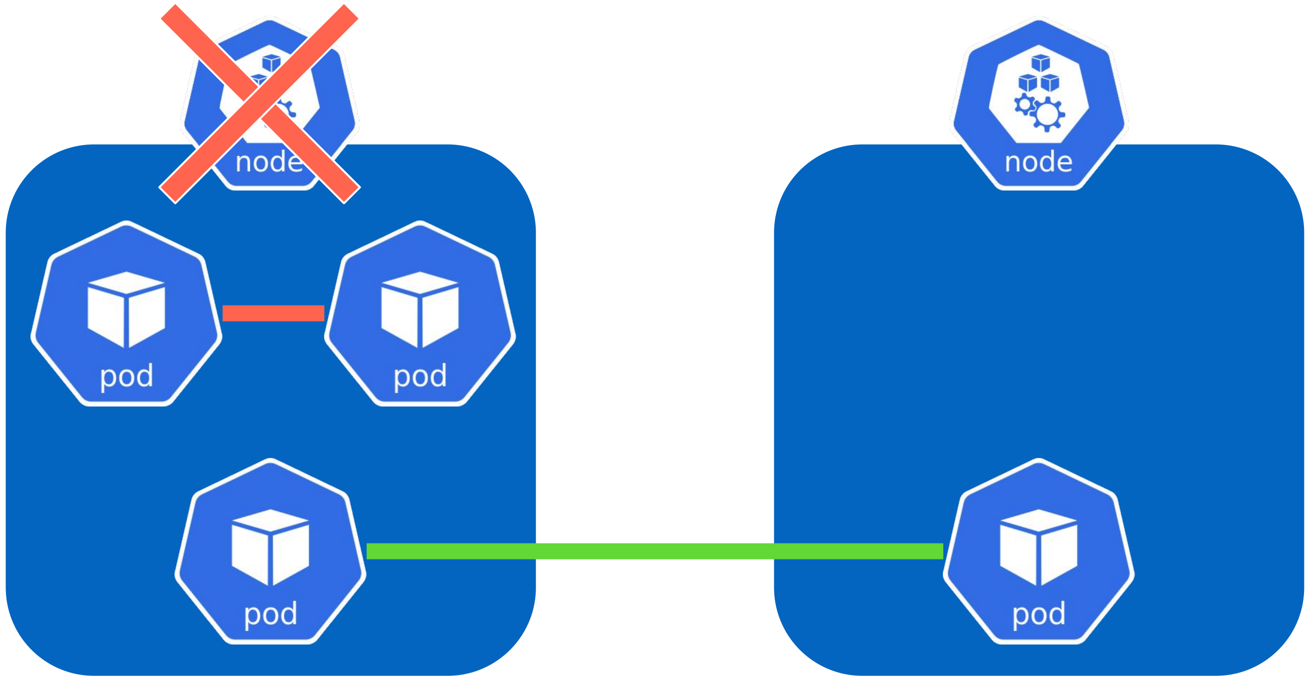
controlled by Node labels

Pod Affinity



controlled by Pod labels

Pod AntiAffinity



controlled by Pod labels and topology key

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/105-affinity

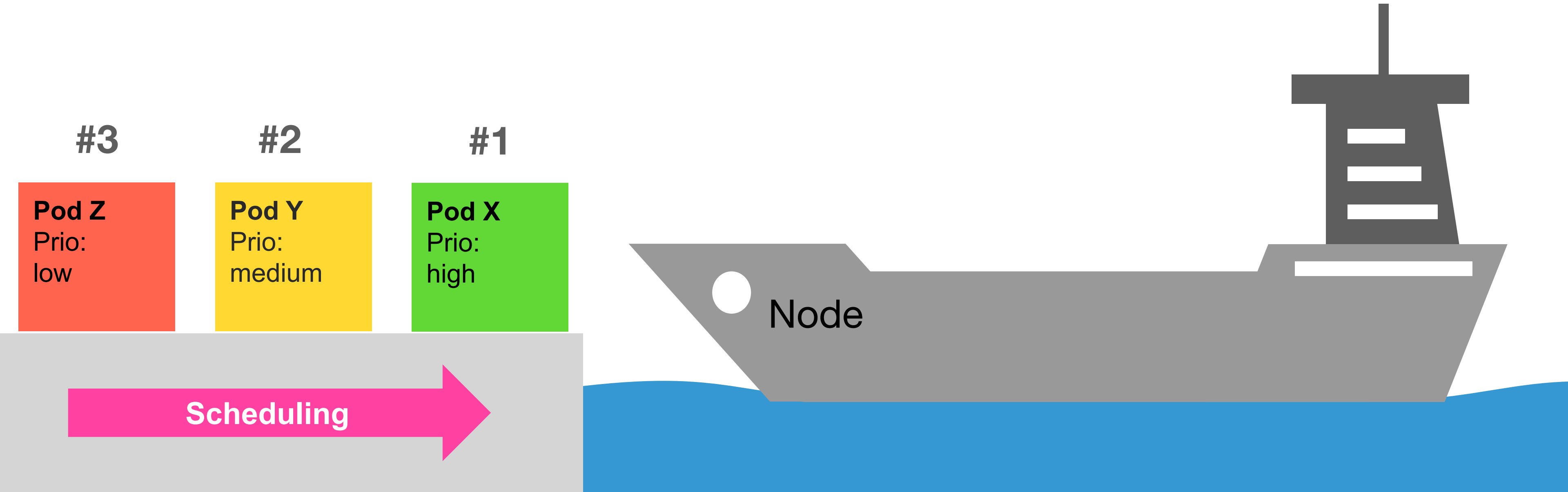
Resource Requests & Limits

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/110-resource-requests-limits

PriorityClasses

PriorityClass - Faster scheduling of pods with higher priority

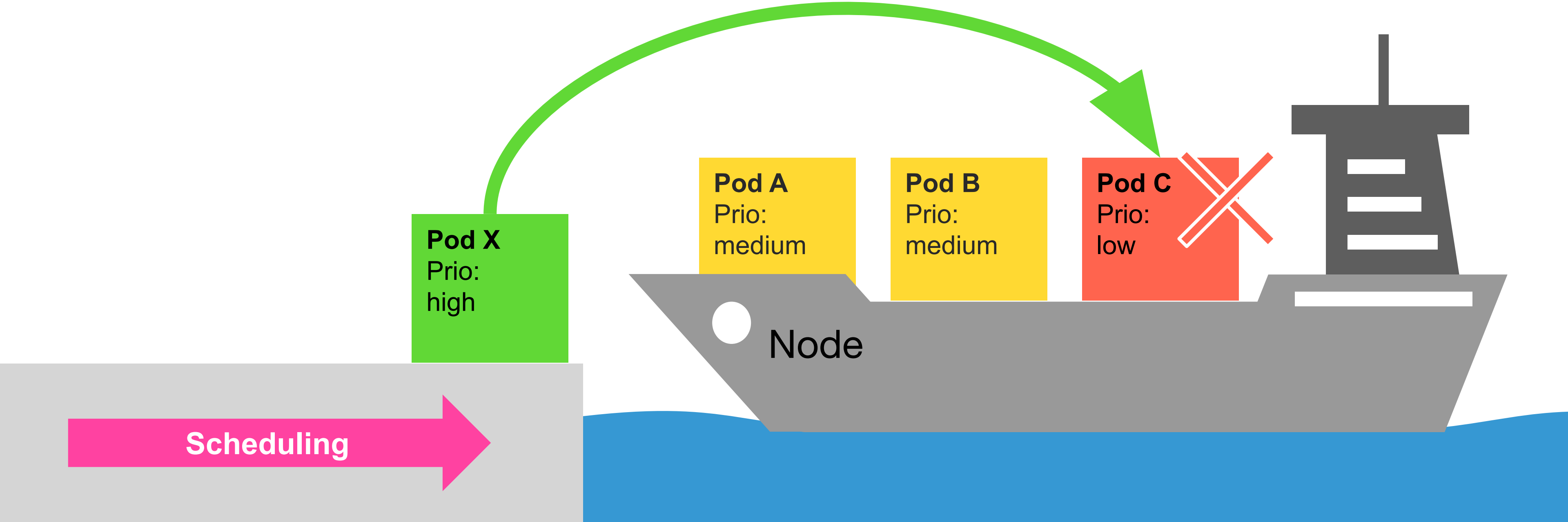


```yaml
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: high
value: 1000000
description: "For most important apps”
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx
  priorityClassName: high
```

preemptionPolicy - Eviction of pods with lower priority

- protects high prio Pods from being evicted



Probes

startupProbes

- at start time
- checks if application is started properly
- executed before any other probe
- deactivates other probes if unsuccessful

readinessProbes

- at start time
- Pod “ready” when probe successful
- receives Traffic when “ready”

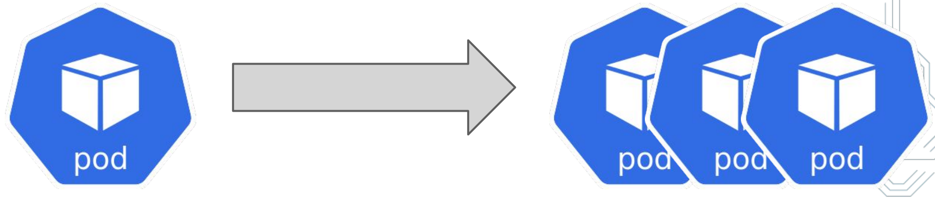
livenessProbes

- regular checks at runtime
- if probe not successful, Pod is restarted

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/115-probes

Horizontal Pod Autoscaler



Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/120a-hpa

Pod Disruption Budget

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/120b-pdb

Sidecar Containers

Authentication and TLS offloading

Backup

Metrics & Healthchecks

Logs

Repair & Maintenance

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/200-sidecar

Init Containers

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/300-init-container

Security for Pods & Containers

Security Context

Pod level

- fsGroup
- fsGroupChangePolicy
- runAsGroup
- runAsNonRoot
- runAsUser
- seLinuxOptions
- seccompProfile
- supplementalGroups
- sysctls

Container level

- allowPrivilegeEscalation
- capabilities
- privileged
- procMount
- readOnlyRootFilesystem
- runAsGroup
- runAsNonRoot
- runAsUser
- seLinuxOptions
- seccompProfile

more specific than Pod level

Security Context

control user permissions of the running process

- runAs…

use security frameworks & kernel features

- seccompProfile
- seLinux
- allowPrilvilegeEscalation
- capabilities

Pod Security Standards (PSS)

- **successor to Pod Security Policies** (PSP) (removed in v1.25)
- works on **cluster level** or in **single namespace**
- can be added and removed at any time

Pod Security Standards (PSS)

- 3 profiles:
  - **privileged** (unrestricted)
  - **baseline** (minimal security)
  - **restricted** (heavily restricted, best-practice security)

Pod Security Standards (PSS)

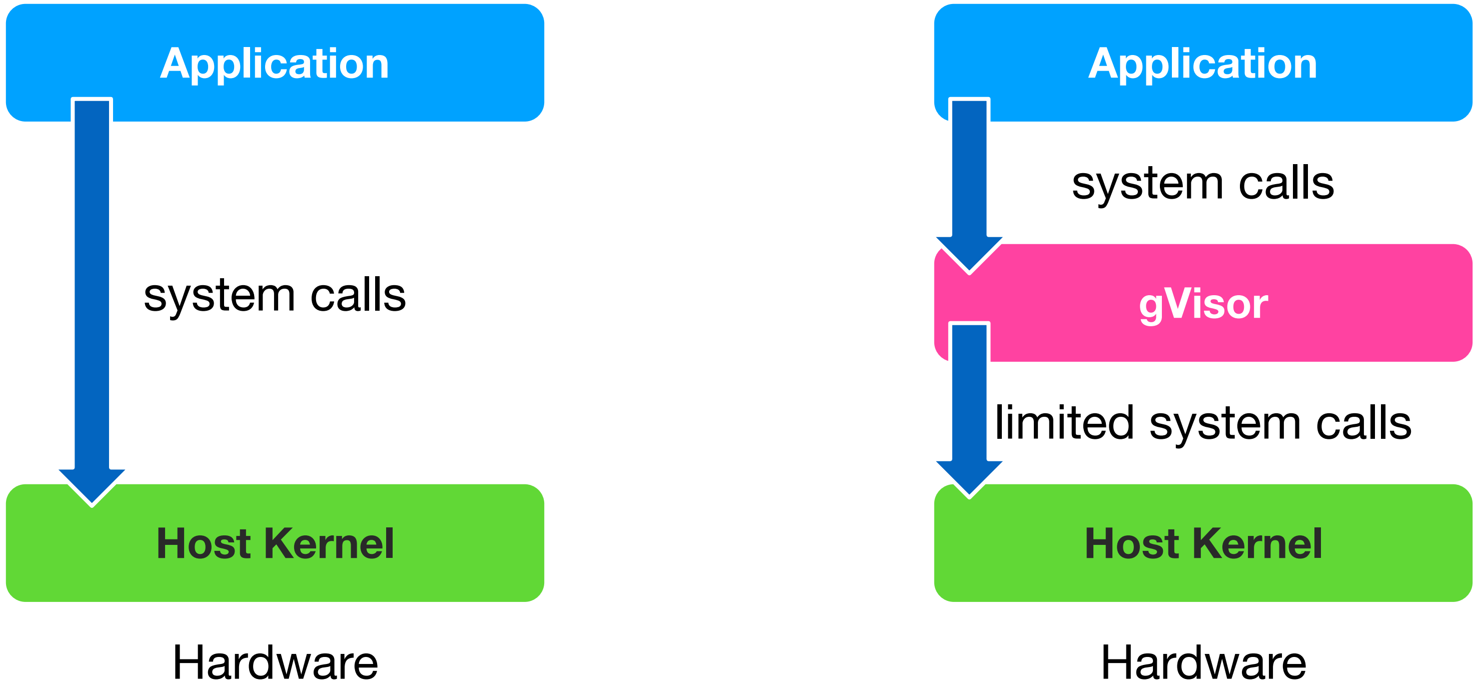
- 3 modes:
  - **warn** (Pod will be created, user visible warning)
  - **audit** (Pod creation permitted but will be logged)
  - **enforce** (Pod cannot be created)

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/400-security

Sandboxing

Sandboxing


Application
system calls
Host Kernel
Hardware
Application
system calls
gVisor
limited system calls
Host Kernel
Hardware

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/410-sandboxing

Persistence

PersistentVolumeClaim

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/350-persistent-storage

3rd-party tools

External-DNS

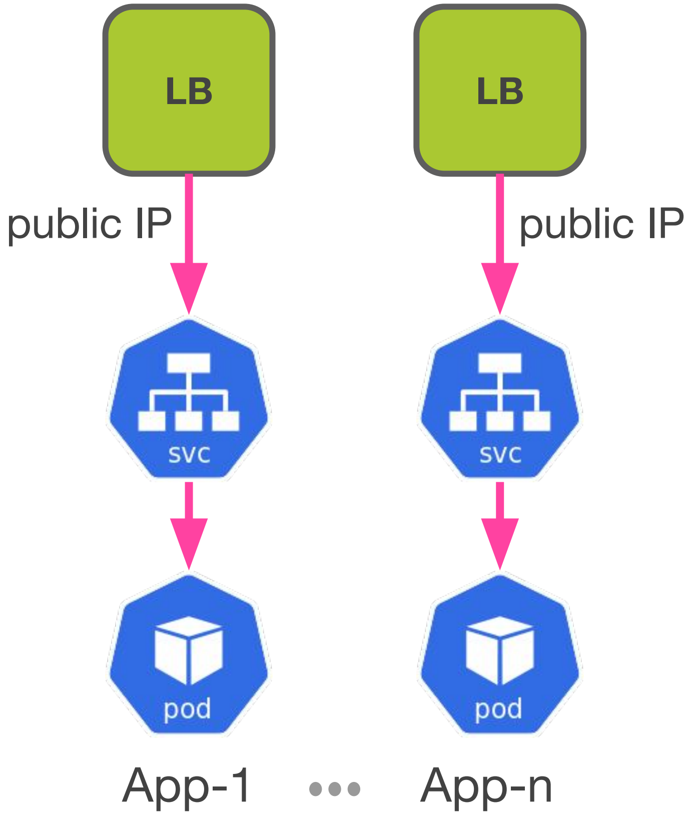
EXTERNAL DNS FOR kubernetes

- synchronize DNS records
- multi-provider capable
- controlled by K8s resources (svc, ing)
- annotations
- automation

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/500-external-dns

Recap: Without ingress controller



Ingress Controller

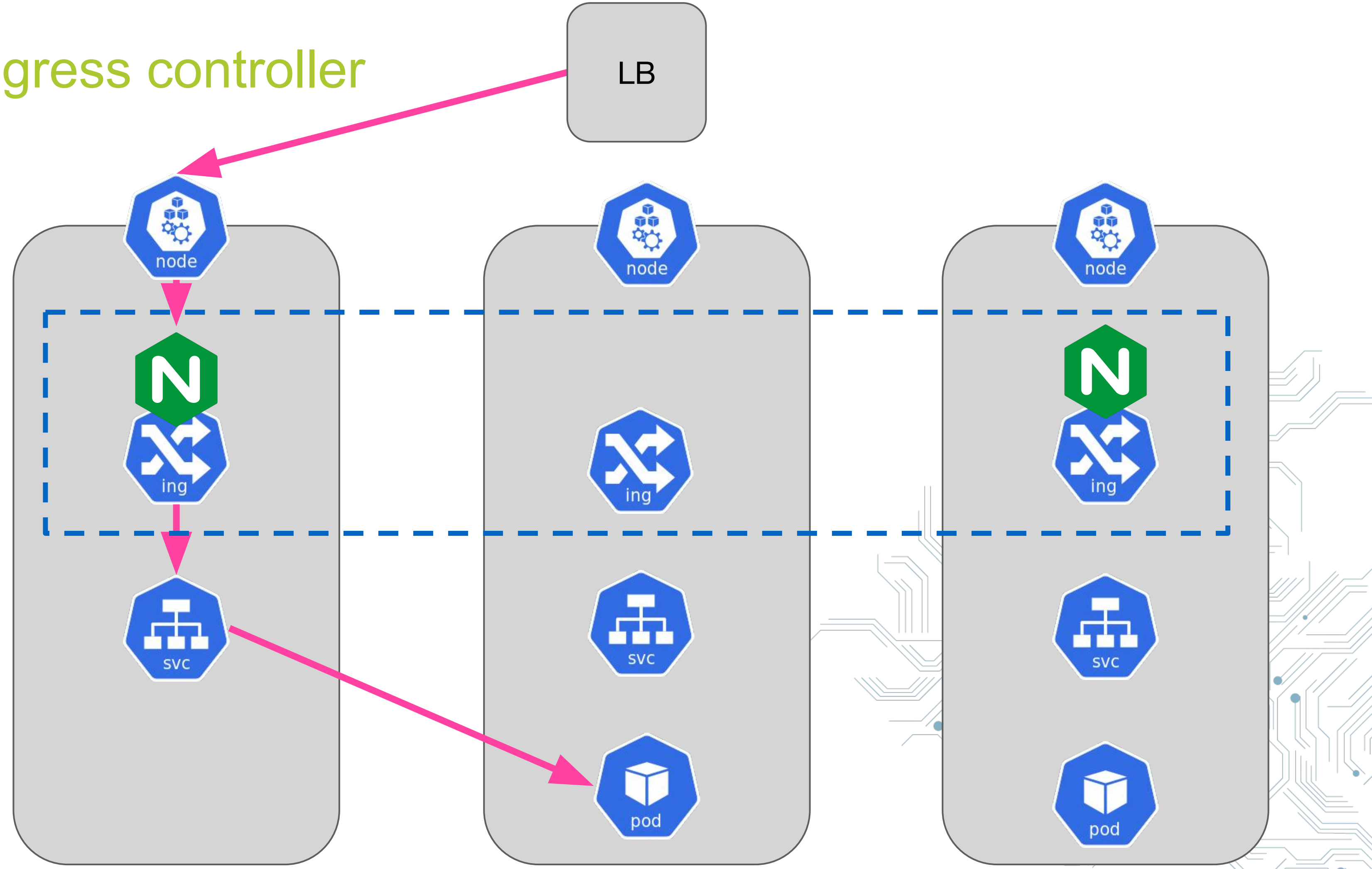
HAPROXY

træfik

ingress-nginx

- works on Layer7 HTTP/S
- routing based on
  - domain name
  - URL path
- SSL offloading
- basic authentication
- IP filtering

With ingress controller



Ingress Controller

Ingress Ressource
“Kind: Ingress”

Ingress Controller vs. Ingress Resource

Ingress Controller

Ingress Ressource
“Kind: Ingress”

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/600-ingress-nginx

Cert-Manager

- Let's Encrypt
- HTTP / DNS challenge
- wildcard certificates (DNS challenge)
- auto renew
- controlled by K8s resource (ing)

Hands-on

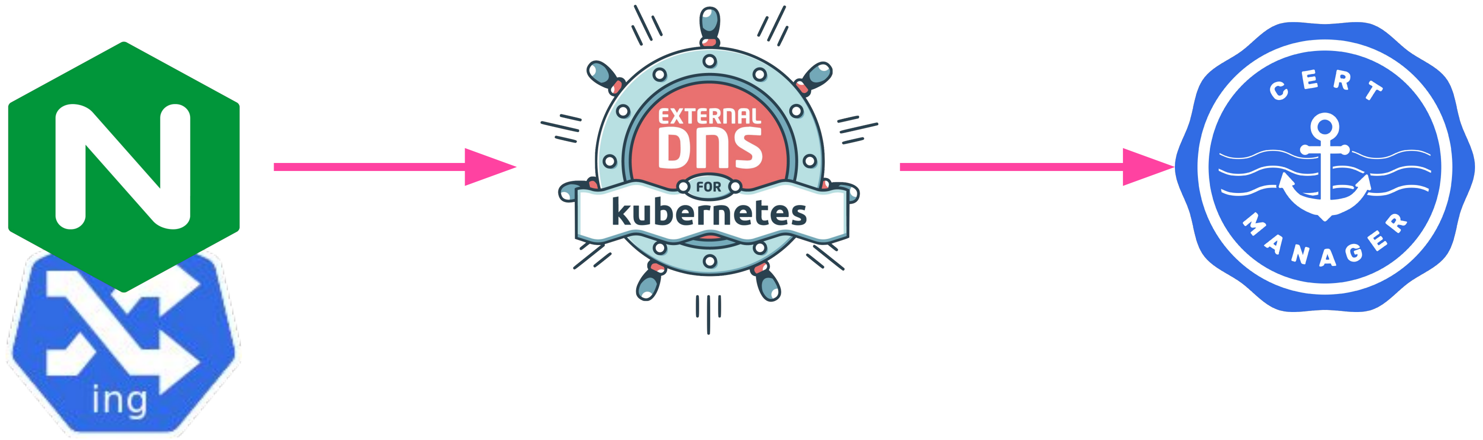
https://github.com/syseleven/containerday-workshops/tree/main/700-cert-manager

external-dns, ingress-nginx & cert-manager in action

1. create ingress resource

2. create DNS record

3. request certificate

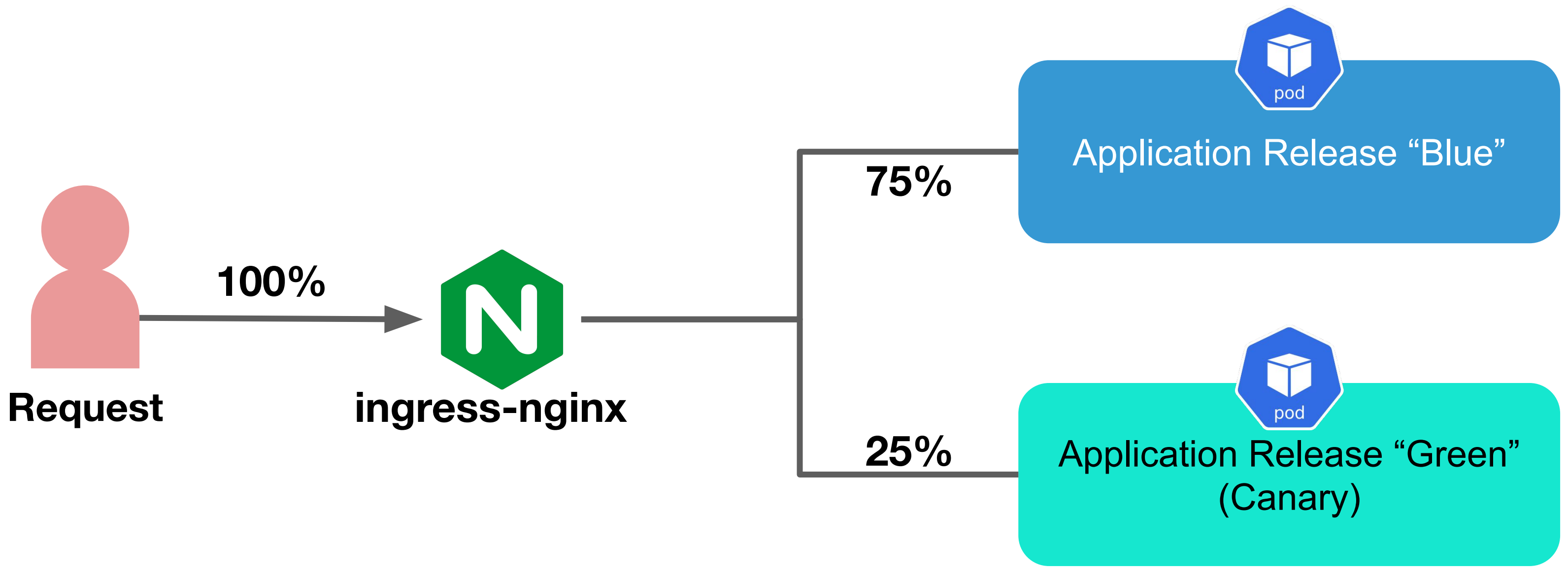


Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/800-deployment-with-ingress-and-tls

canary deployments
in action

Canary deployments


100%
Request
ingress-nginx
75%
25%
pod
Application Release “Blue”
pod
Application Release “Green” (Canary)

Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/810-canary-deployments

one Helmfile to rule them all

Tooling: Helm

Anatomy of a Helm Chart

Chart.yaml

- Release Name
- Chart Version
- App Version

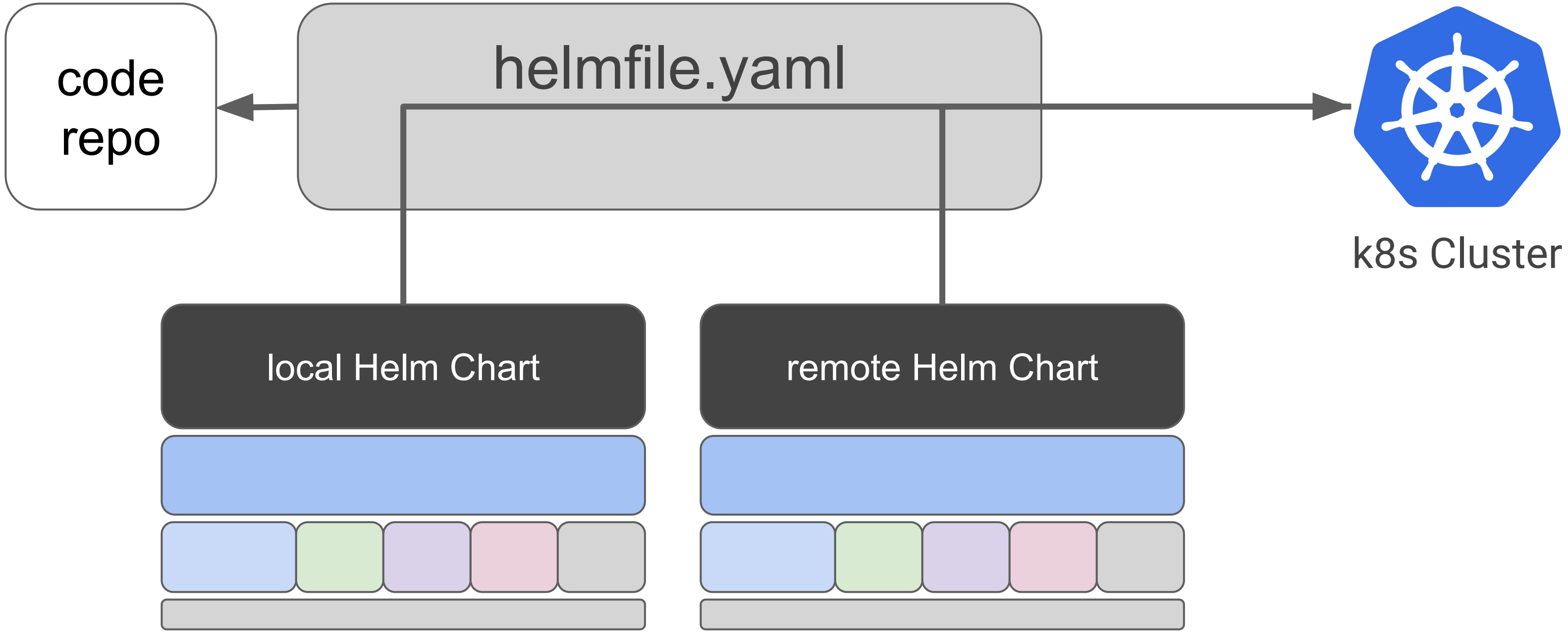
values.yaml

- default config parameters

Deployment | Secret | Config Map | Volume | Ingress

… other templates

Tooling: Helmfile



Hands-on

https://github.com/syseleven/containerday-workshops/tree/main/900-helmfile

reactive & supportive measures

Observability, Monitoring & Alerting

Backup & Recovery

IaaC, automation, continuous deployment

SysEleven

Questions?

Thank you!

syseleven.de/trial-metakube