white duck

Kubernetes 1.22 – What’s new?

Cloud Native Rosenheim Meetup, Oktober 2021

CLOUD NATIVE ROSENHEIM MEET UP

Microsoft Partner

Microsoft

Gold Cloud Platform
Gold DevOps
Silver Application Development
Silver Security
Silver Application Integration

GitHub

Who we are?

Philip Welz (Senior Kubernetes & DevOps Engineer, CKA, CKAD & CKS)

Email:	philip.welz@whiteduck.de
Twitter:	@philip_welz
LinkedIn:	https://www.linkedin.com/in/philip-welz

Dario Brozovic (DevOps Engineer)

Email:	dario.brozovic@whiteduck.de
LinkedIn:	https://www.linkedin.com/in/dariobrozovic

Nico Meisenzahl (Senior Cloud & DevOps Consultant, Cloud & Data Management MVP)

Email:	nico.meisenzahl@whiteduck.de
Twitter:	@nmeisenzahl
LinkedIn:	https://www.linkedin.com/in/nicomeisenzahl

Housekeeping

- this meetup will be streamed on YouTube!
- want to participate?
  - join our Meetup to get access to the Zoom meeting
    - https://www.meetup.com/CloudNative-Rosenheim-Meetup
  - we do also monitor the comments on YouTube

Cloud Native Rosenheim Meetup

- this is our first meetup as Cloud Native Rosenheim Meetup
- we are now a Cloud Native Community Groups member
  - join our Community: https://community.cncf.io/rosenheim
- also follow us on Twitter: @CloudNative_Ro

Agenda

- Kubernetes 1.22 Release
- new features – our pick
- API and feature removals

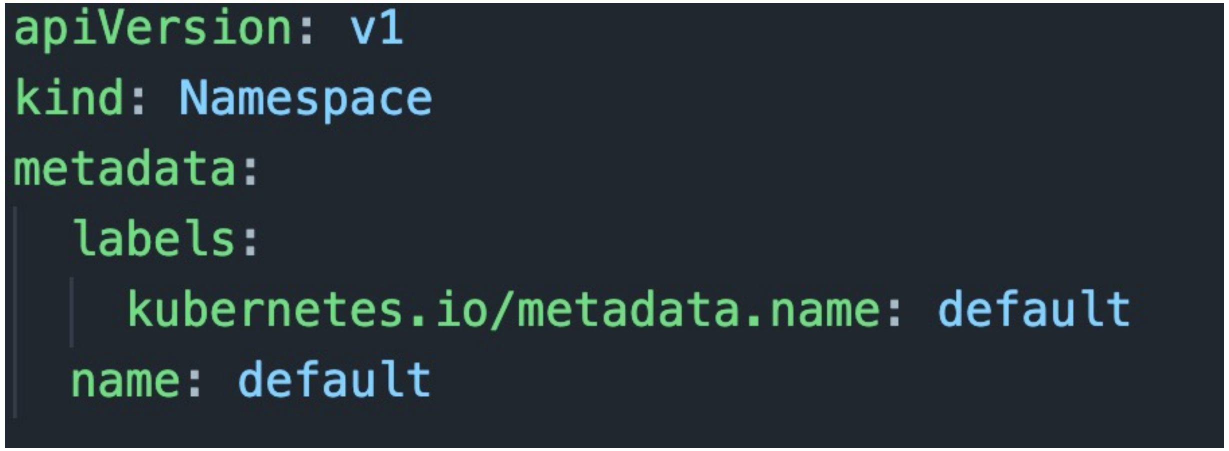
Kubernetes 1.22

- “Reaching New Peaks”
- got released on August 4
  - first longer-cycle release after the change from four to three yearly releases
- this release shifts to security first
- largest release ever with 53 enhancements
  - 13 have graduated to stable
  - 24 are moving to beta
  - 16 are entering alpha
  - 3 features have been deprecated

NEW FEATURES – OUR PICK

#1 Namespace labels

- graduating to Stable
- new label will be added to all namespaces where its value is the namespace name
- this label can be used with any namespace selector



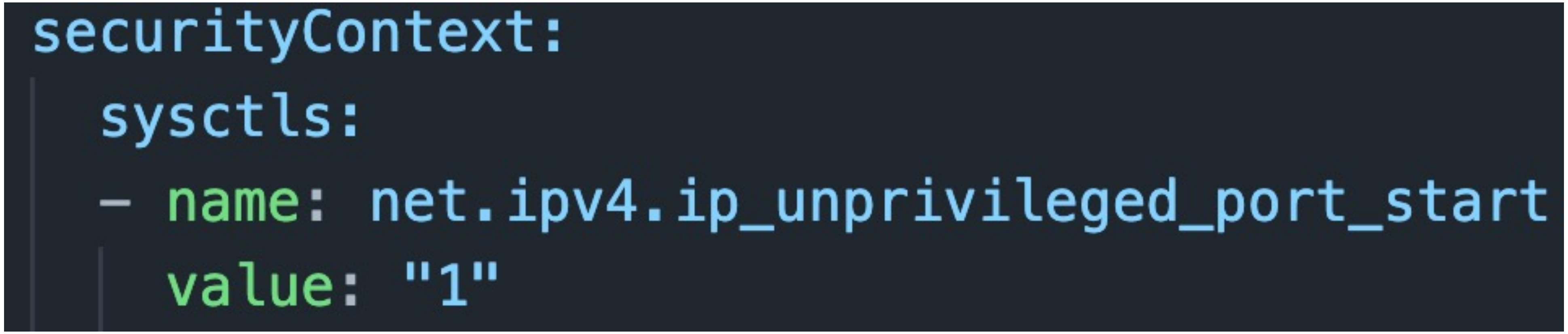
```yaml
apiVersion: v1
kind: Namespace
metadata:
  labels:
    kubernetes.io/metadata.name: default
  name: default
```

#2 Node Swap support

- graduating to Alpha
  - must be enabled via kubelet feature gate
- this enhancement enables Kubernetes workloads to use swap
- Java or Node apps could benefit
- note: global for the whole node, and cannot be configured per workload

#3 Unprivileged Ports

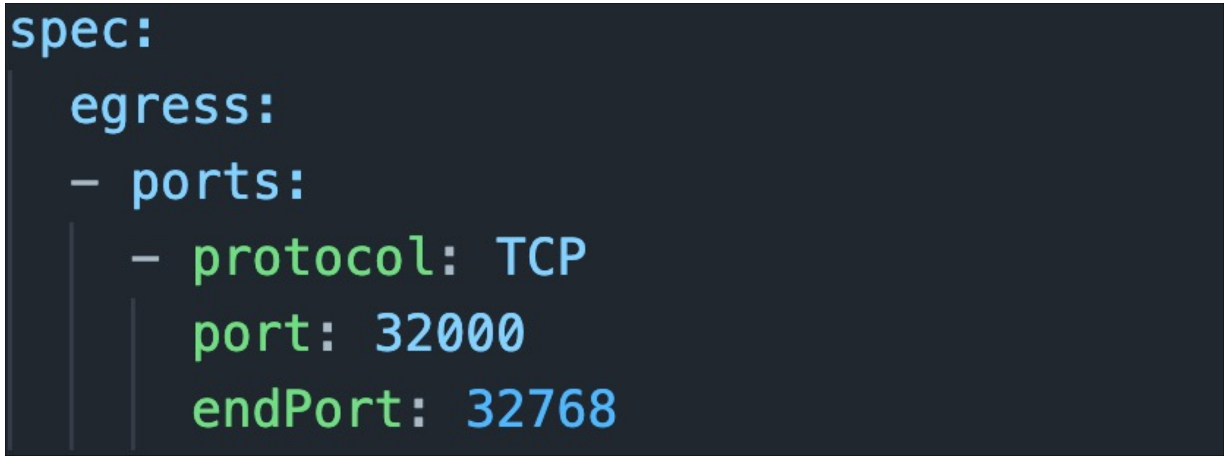
- marked as safe sysctl
- allows containers that are running as unprivileged users to bind low ports



```yaml
securityContext:
  sysctls:
  - name: net.ipv4.ip_unprivileged_port_start
    value: "1"
```

#4 Network Policy Endport

- graduating to Beta
- this enhancement will allow you to define all ports in a NetworkPolicy as a range



```
spec:
  egress:
  - ports:
    - protocol: TCP
      port: 32000
      endPort: 32768
```

#5 Rootless mode

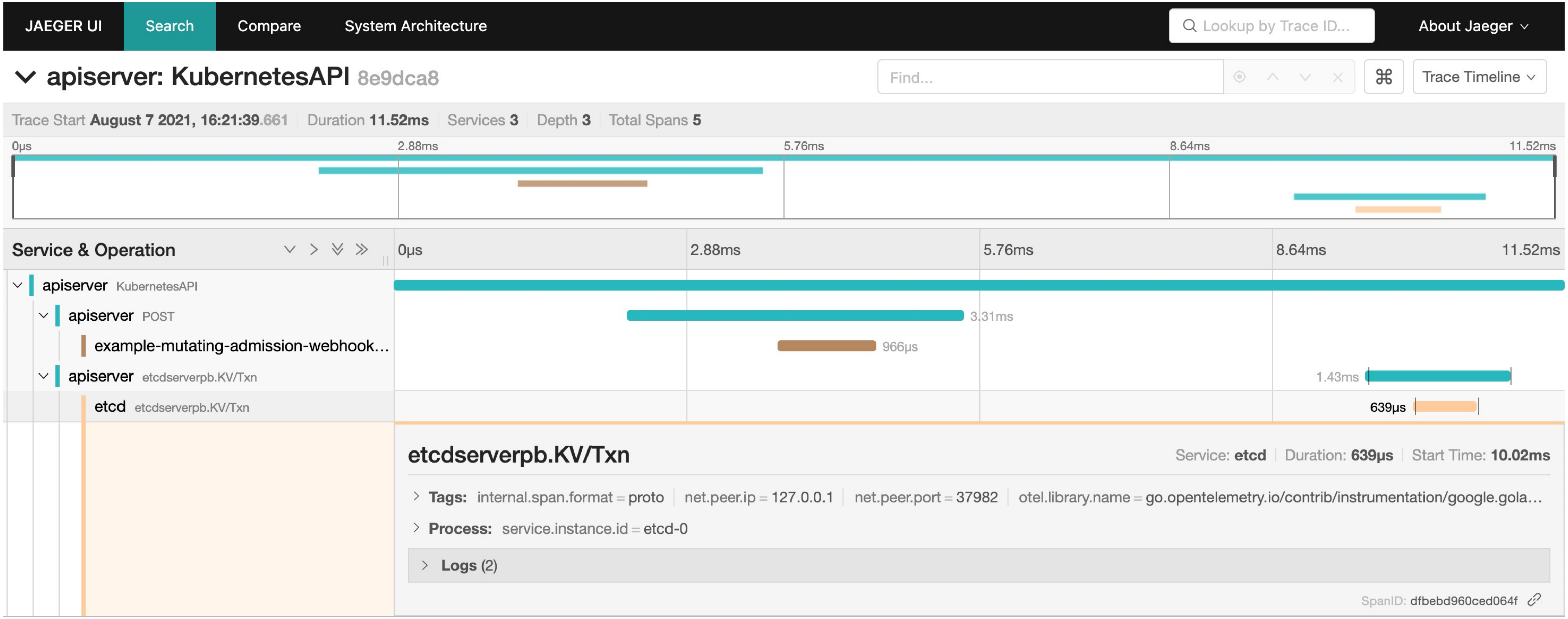
- graduating to Alpha
- this enhancement enables Kubelet to run in a user namespace when the *KubeletInUserNamespace* feature gate is set
- kubeadm: this enhancement also enables to run the control plane as non-root via the *RootlessControlPlane* feature gate

#6 API server tracing

- graduating to Alpha
  - must be enabled via feature gate and --tracing-config-file flag
- this enhancement enables distributed tracing in the Kubernetes API Server
- uses OpenTelemetry format
- ETCD also supports distributed tracing (experimental)

white duck

#6 API server tracing



apiserver: KubernetesAPI 8e9dca8

Trace Start **August 7 2021, 16:21:39**.661 | Duration **11.52ms** | Services **3** | Depth **3** | Total Spans **5**

Service & Operation	Duration
apiserver KubernetesAPI	
apiserver POST	3.31ms
example-mutating-admission-webhook...	966μs
apiserver etcdserverpb.KV/Txn	1.43ms
etcd etcdserverpb.KV/Txn	639μs

etcdserverpb.KV/Txn

Service: **etcd** | Duration: **639μs** | Start Time: **10.02ms**

Tags: internal.span.format = proto | net.peer.ip = 127.0.0.1 | net.peer.port = 37982 | otel.library.name = go.opentelemetry.io/contrib/instrumentation/google.gola...

Process: service.instance.id = etcd-0

Logs (2)

SpanID: dfbebd960ced064f

#7 Pod Security Admission Control

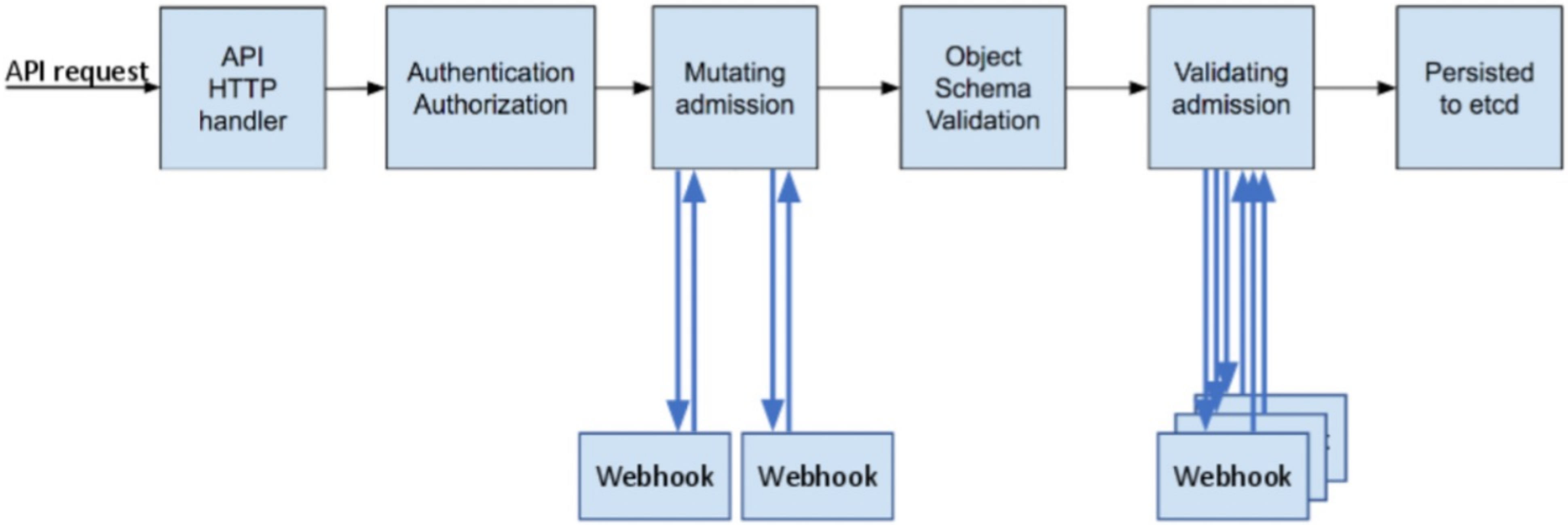
- graduating to Alpha
  - must be explicitly enabled via feature gates on the cluster components
- offers a built-in *Pod Security* admission controller as a successor to PodSecurityPolicies
- but what exactly is admission control?

#7 Pod Security Admission Control

- Admission control:
  - software which is compiled into the kube-apiserver binary
  - intercepts requests to the Kubernetes API server prior to persistence of the object, but after the request is authenticated and authorized
  - Kubernetes supports 37 different admission controllers
    - special controllers:
      - MutatingAdmissionWebhook
      - ValidatingAdmissionWebhook
  - Admission controllers may be "validating", "mutating", or both

#7 Pod Security Admission Control

- Admission control phases:



Admission Controller Phases

#7 PodSecurityPolicy Throwback

- PodSecurityPolicy admission controller which must be enabled via the API Server Flag "--enable-admission-plugins=PodSecurityPolicy"
- target user or pod's serviceaccount; must be authorized via RBAC to use the configured PodSecurityPolicy
- PodSecurityPolicy API and admission controller are deprecated since v1.21 and will be removed from Kubernetes in v1.25

#7 Pod Security Admission Control

- Pod Security Standards
  - define three different *policies* to broadly cover the security spectrum: Privileged, Baseline & Restricted
- enforcement
  - the enforcement can be done at three levels: enforce, audit & warn
  - <enforcement>-version: can be "latest" or specific like *"v1.22"*
- configurable via namespace labels
  or AdmissionConfiguration

#7 Demo Pod Security Admission Control

#7 Pod Security Admission Control

- conclusion: The PSP replacement is less complex than its predecessor but not quite as flexible
- alternatives in the CNCF:
  - Open Policy Agent
  - Kyverno
  - Kubewarden

#8 SeccompDefault

- graduating to Alpha - must be explicitly enabled via kubelet feature gate and turned on via kubelet configuration (or command line)
- with this feature you can enable seccomp to all workloads or to workloads on a specific node, rather than configure it per workload
- but what exactly is seccomp?

#8 Seccomp basics

- secure computing mode: seccomp
- is a Linux kernel mechanism that lets you restrict the system calls a process can use
- reduces the chance that a Linux kernel vulnerability will be exploited
- all container runtimes ship with a default seccomp profile (sane defaults)
- Kubernetes will explicitly set the seccomp profile to *Unconfined* which disables seccomp filtering

#8 SeccompDefault

- the new feature just changes the default seccomp profile from *Unconfined* to *RuntimeDefault*
- profiles:
  - Unconfined: seccomp will not be enabled, which is also the default
  - RuntimeDefault: the container runtimes default profile will be used
  - Localhost: a node local profile will be applied

white duck

#9 Ephemeral Containers

- Alpha feature since v1.16
- what`s new?
  - API changes in v1.22
- eequires
  - kubectl v1.22 since earlier versions use the old API
  - containerd v1.5
  - must be enabled via feature gate in Api-Server, Controller-Manager, Scheduler & Kubelet
  - currently not supported with cri-o
    - https://github.com/cri-o/cri-o/issues/4790

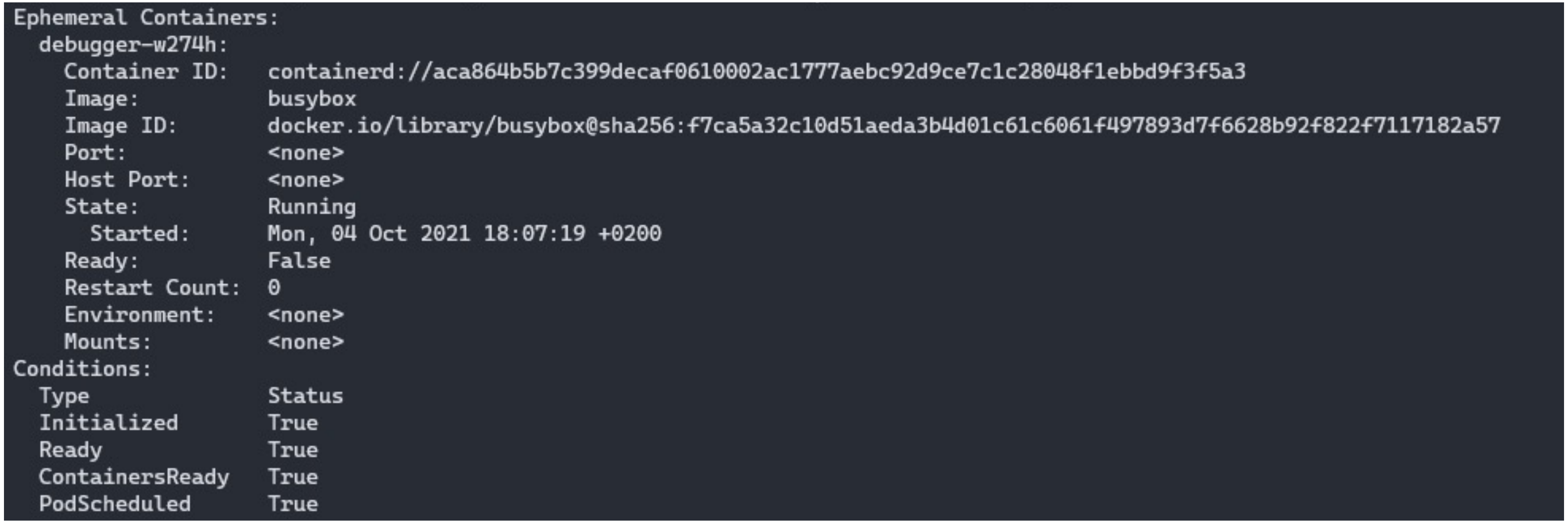
#9 Ephemeral Containers

- what are they used for?

- assumptions
  - Container images should be lightweight as possible
  - Containers running inside of pods only contain application code and required dependencies

- the pod running your application crashes!
  - No shell
  - No debugging tools
    - = no troubleshooting

#9 Demo Ephemeral Containers



```
Ephemeral Containers:
  debugger-w274h:
    Container ID:   containerd://aca864b5b7c399decaf0610002ac1777aebc92d9ce7c1c28048f1ebbd9f3f5a3
    Image:          busybox
    Image ID:       docker.io/library/busybox@sha256:f7ca5a32c10d51aeda3b4d01c61c6061f497893d7f6628b92f822f7117182a57
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Mon, 04 Oct 2021 18:07:19 +0200
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:         <none>
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
```

API AND FEATURE REMOVALS

API and feature removals

- this release removes (not deprecates) APIs and features
  - this happened the last time with 1.16
  - and will also happen with 1.25
- you will need to update your manifests prior the upgrade
- some API groups allow to retrieve or update existing objects with older API versions

Kubernetes Deprecation Policy

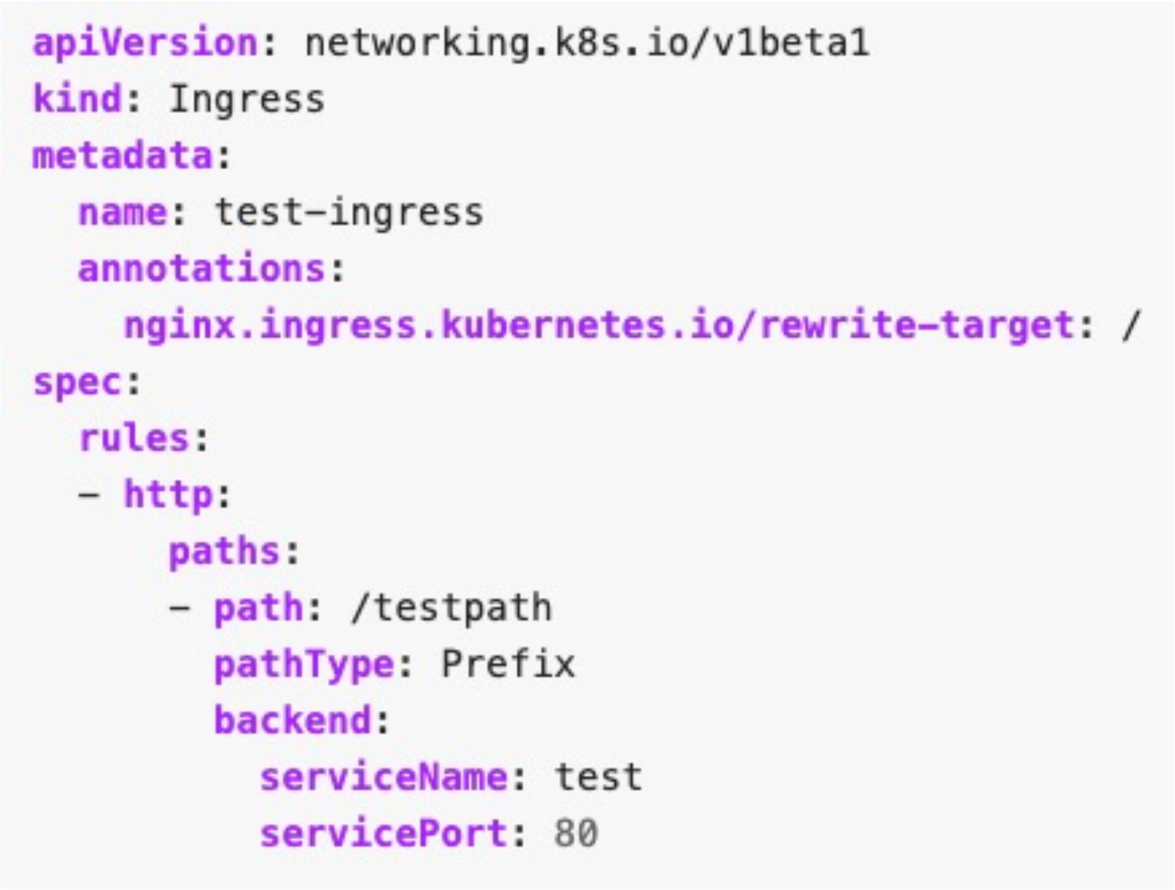
- there are 3 API tracks with different policies!
- GA (generally available, stable), e.g. v1
  - 12 months or 3 releases (whichever is longer)
- Beta (pre-release), e.g. v1beta1
  - 9 months or 3 releases (whichever is longer)
- Alpha (experimental), e.g. v1alpha1
  - 0 releases

white duck

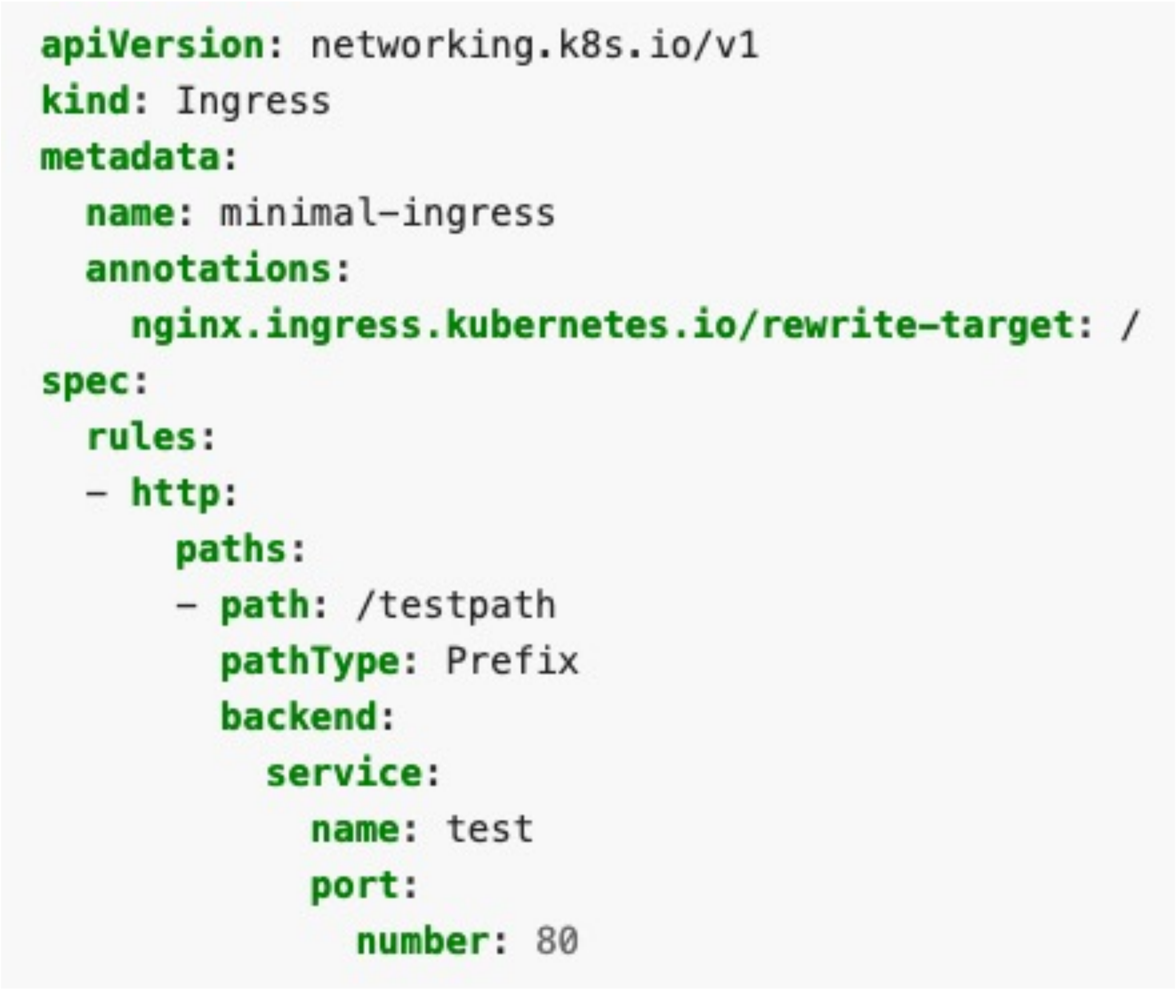
Ingress & IngressClass

- networking.k8s.io/v1 (available since v1.19)
- removals
  - extensions/v1beta1
  - networking.k8s.io/v1beta1
- no notable changes for IngressClass
- notable changes for Ingress
  - many fields are renamed/moves
  - pathType is now required

Ingress



```yaml
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: test-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /testpath
        pathType: Prefix
        backend:
          serviceName: test
          servicePort: 80
```



```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: minimal-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /testpath
        pathType: Prefix
        backend:
          service:
            name: test
            port:
              number: 80
```

Ingress – things to verify

- verify whether you Ingress implementation supports v1
- also migrate your kubernetes.io/ingress.class annotation to the spec.ingressClassName field
- in general: Review all existing Ingress manifests

CustomResourceDefintion

- apiextensions.k8s.io/v1 (available since v1.16)
- removals
  - apiextensions.k8s.io/v1beta1
- notable changes
  - spec.scope no longer defaults to Namespaced
  - multiple fields have been restructured/moved
    - mostly related to the new spec.versions field
- verify whether your third-party tools support v1!

Webhook resources

- relates to MutatingWebhookConfiguration and ValidatingWebhookConfiguration
- admissionregistration.k8s.io/v1 (available since v1.16)
- removals
  - admissionregistration.k8s.io/v1beta1
- notable changes
  - multiple default values have changed
  - webhooks[*].name needs to be unique
- review your third-party tools!

Further removals #1

- SubjectAccessReview: authorization.k8s.io/v1beta1
  - relates to LocalSubjectAccessReview, SelfSubjectAccessReview, SubjectAccessReview
  - authorization.k8s.io/v1 (available since v1.6)
  - notable changes
    - spec.group was renamed to spec.groups
- CertificateSigningRequest: certificates.k8s.io/v1beta1
  - certificates.k8s.io/v1 (available since v1.19)
  - notable changes
    - some/new fields are required now

Further removals #2 (no notable changes)

- RBAC resources: rbac.authorization.k8s.io/v1beta1
  - relates to ClusterRole, ClusterRoleBinding, Role, and RoleBinding
  - rbac.authorization.k8s.io/v1 (available since v1.8)
- Storage resources: storage.k8s.io/v1beta1
  - relates to CSIDriver, CSINode, StorageClass, VolumeAttachment
  - storage.k8s.io/v1 (available since v1.6 to v 1.19)

Further removals #3 (no notable changes)

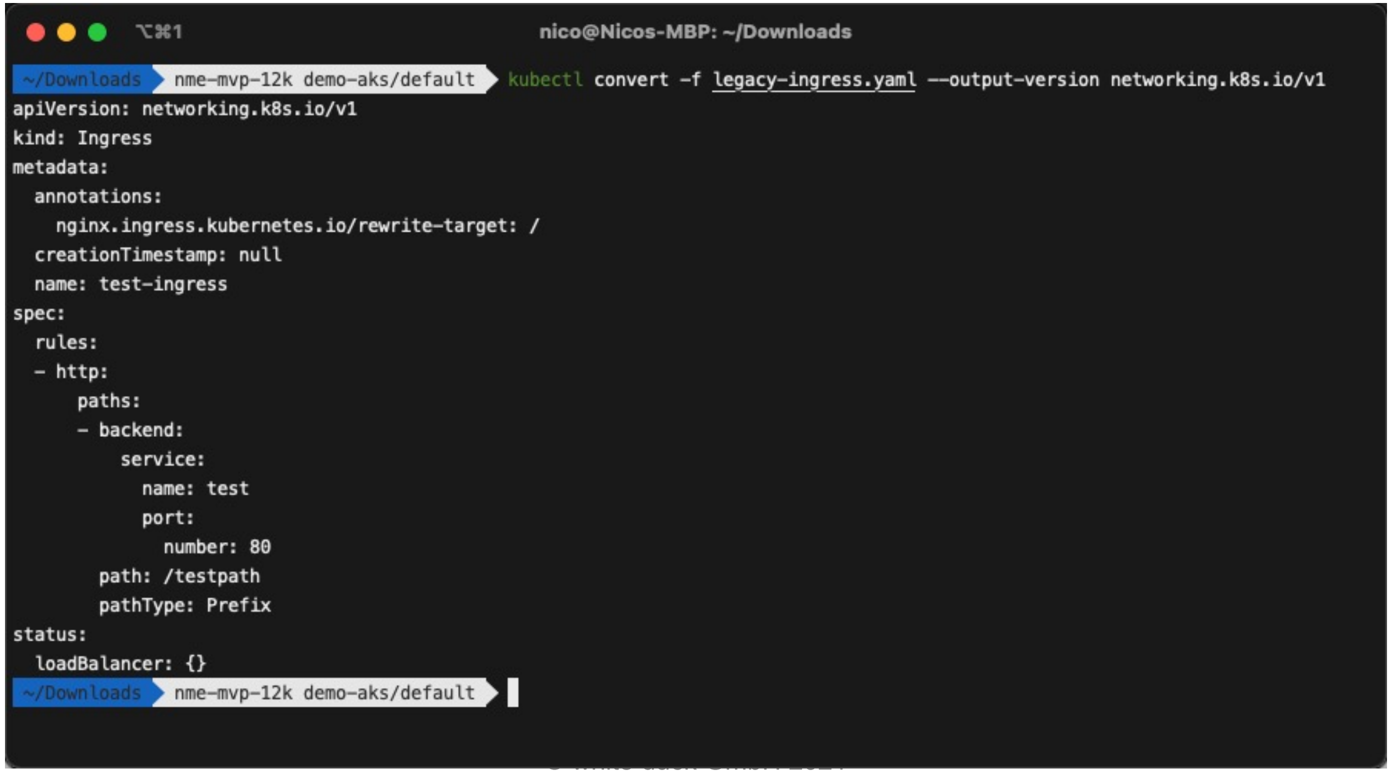
- PriorityClass: scheduling.k8s.io/v1beta1
  - scheduling.k8s.io/v1 (available since v1.14)
- APIService: apiregistration.k8s.io/v1beta1
  - apiregistration.k8s.io/v1 (available since v1.10)
- Lease: coordination.k8s.io/v1beta1
  - coordination.k8s.io/v1 (available since v1.14)

Demo: Verify your API versions

- you can verify your API versions manually, via CI/CD and in-cluster
- tools to use manually & CI/CD
  - https://github.com/doitintl/kube-no-trouble
  - https://github.com/FairwindsOps/pluto
  - https://github.com/yannh/kubeconform
- tools to use in-cluster
  - https://kyverno.io/policies/best-practices/check_deprecated_apis/
- kubectl API warnings (stable since 1.22)

Demo: kubectl convert

- a plugin for kubectl
- helps you upgrading your manifests



```
~/Downloads  nme-mvp-12k demo-aks/default  kubectl convert -f legacy-ingress.yaml --output-version networking.k8s.io/v1
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
  creationTimestamp: null
  name: test-ingress
spec:
  rules:
  - http:
      paths:
      - backend:
          service:
            name: test
            port:
              number: 80
        path: /testpath
        pathType: Prefix
status:
  loadBalancer: {}
~/Downloads  nme-mvp-12k demo-aks/default 
```

Links

- https://github.com/whiteducksoftware/cloud-native-rosenheim-meetup
- https://kubernetes.io/blog/2021/08/04/kubernetes-1-22-release-announcement
- https://kubernetes.io/blog/2021/07/14/upcoming-changes-in-kubernetes-1-22
- https://sysdig.com/blog/kubernetes-1-22-whats-new
- https://blog.aquasec.com/kubernetes-version-1.22-security-features
- https://kubernetes.io/docs/reference/using-api/deprecation-guide
- https://kubernetes.io/docs/reference/using-api/deprecation-policy
- https://kubernetes.io/docs/concepts/security/pod-security-admission/
- https://kubernetes.io/blog/2021/09/03/api-server-tracing/
- https://kubernetes.io/docs/reference/command-line-tools-reference/feature-gates
- https://www.downloadkubernetes.com/ (kubectl convert)

Questions?

Philip Welz (Senior Kubernetes & DevOps Engineer, CKA, CKAD & CKS)

Email:	philip.welz@whiteduck.de
Twitter:	@philip_welz
LinkedIn:	https://www.linkedin.com/in/philip-welz

Dario Brozovic (DevOps Engineer)

Email:	dario.brozovic@whiteduck.de
LinkedIn:	https://www.linkedin.com/in/dariobrozovic

Nico Meisenzahl (Senior Cloud & DevOps Consultant, Cloud & Data Management MVP)

Email:	nico.meisenzahl@whiteduck.de
Twitter:	@nmeisenzahl
LinkedIn:	https://www.linkedin.com/in/nicomeisenzahl