



Kube-Oddities

The Quirks That Keep Kubernetes Interesting

**KubeCon + CloudNativeCon Europe
March 25th 2026**

Hi 🙋 I'm Marcus Noble!

I'm a platform engineer at **monzo**

I'm a CNCF Ambassador  and I run a monthly newsletter at [CloudNative.Now](https://cloudnative.now)

7+ years experience with cloud native and Kubernetes



Hi 🙋 I'm Márk Sági-Kazár!

↪ Don't try to pronounce it

I'm (also) a CNCF Ambassador  and I organize the Hungarian Cloud Native community (meetup and KCD).

Look out for KCD Budapest later this year ↪

I like building and breaking stuff, hoping that it helps others.
My latest project is [Kubernetes the Very Hard Way](#).

Kube-Oddities

Over our combined many years of working with Kubernetes we've seen some ***weird*** things that resulted in confusion and frustration.

We're here to share some of those with y'all to help you avoid the same.

Whether you're a Kubernetes veteran from the pre-CRD days or you're just getting started with *kubectrl* we've got something for you!

Kube-Oddities

Over our combined many years of working with Kubernetes we've seen some ***weird*** things that resulted in confusion and frustration.

We're here to share some of those with y'all to help you avoid the same.

Whether you're a Kubernetes veteran from the pre-CRD days or you're just getting started with ~~kubect!~~ we've got something for you!

↪ *Kube-Control*

Kube-Oddities

Over our combined many years of working with Kubernetes we've seen some ***weird*** things that resulted in confusion and frustration.

We're here to share some of those with y'all to help you avoid the same.

Whether you're a Kubernetes veteran from the pre-CRD days or you're just getting started with ~~kubect!~~ we've got something for you!

↪ ~~Kube Control~~
↪ Kube-C-T-L

Kube-Oddities

Over our combined many years of working with Kubernetes we've seen some ***weird*** things that resulted in confusion and frustration.

We're here to share some of those with y'all to help you avoid the same.

Whether you're a Kubernetes veteran from the pre-CRD days or you're just getting started with ~~kubect!~~ we've got something for you!

↪ ~~Kube Control~~

↪ ~~Kube C T L~~

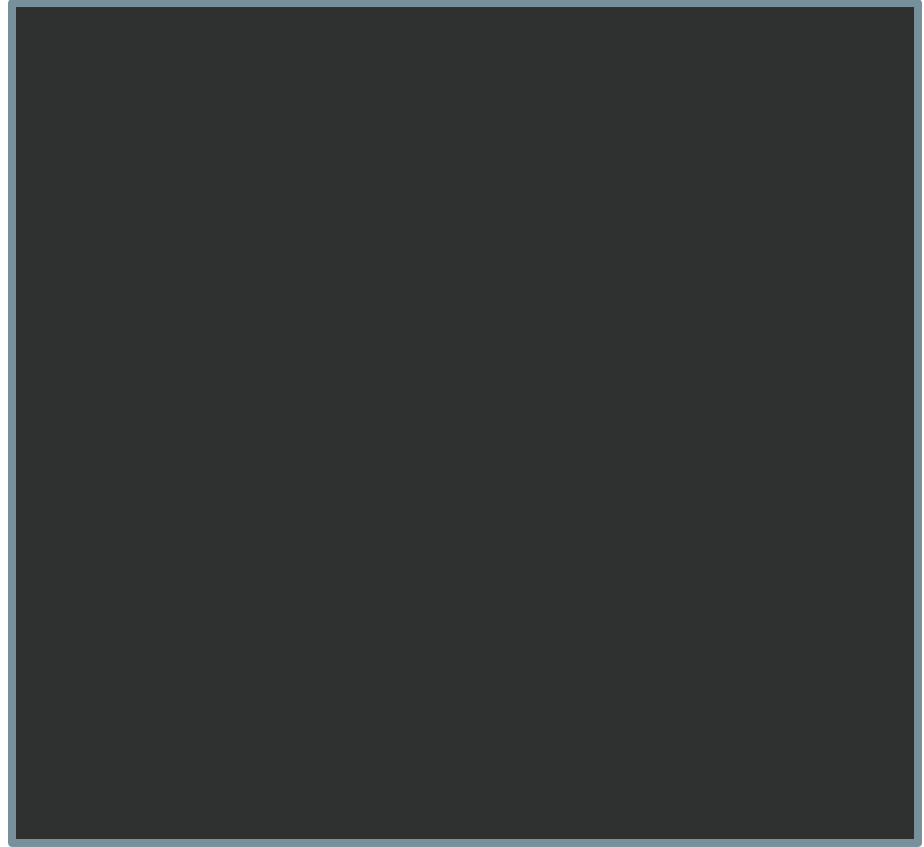
↪ Kube-Cuddle!!!

The background of the image is a repeating pattern of various geometric shapes in light gray, including squares, triangles, circles, and wavy lines, set against a white background. The word "Pods" is centered in a bold, dark blue font.

Pods

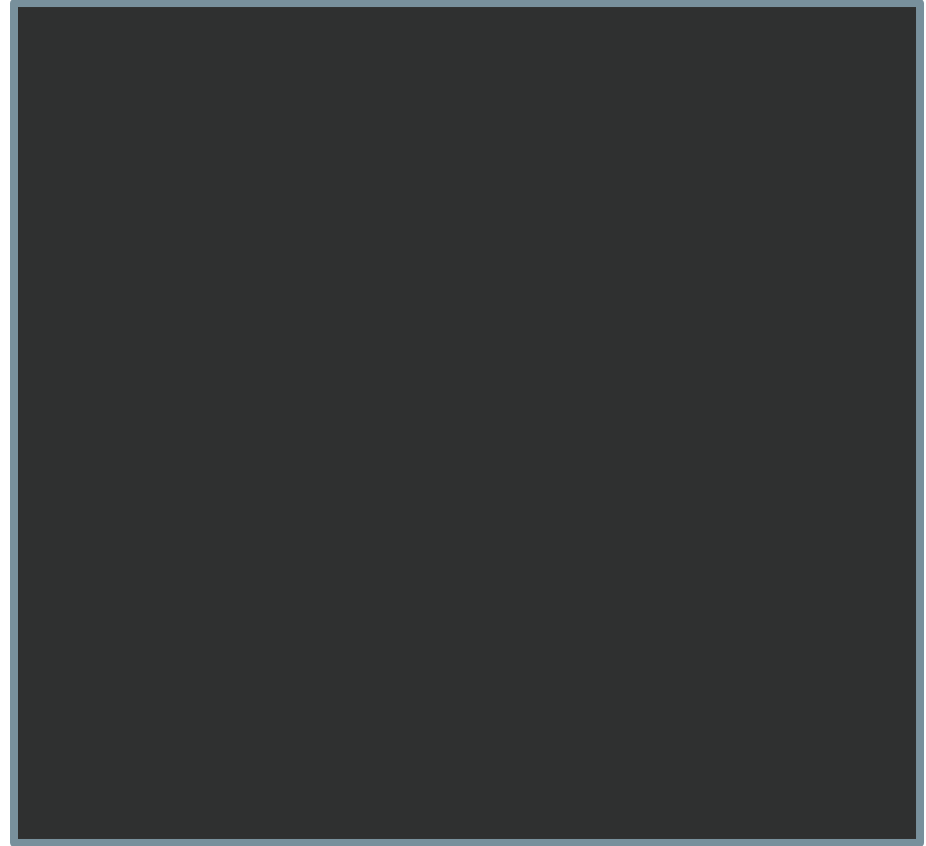
Sidecar Containers

- Secondary containers that run within a Pod alongside your app container



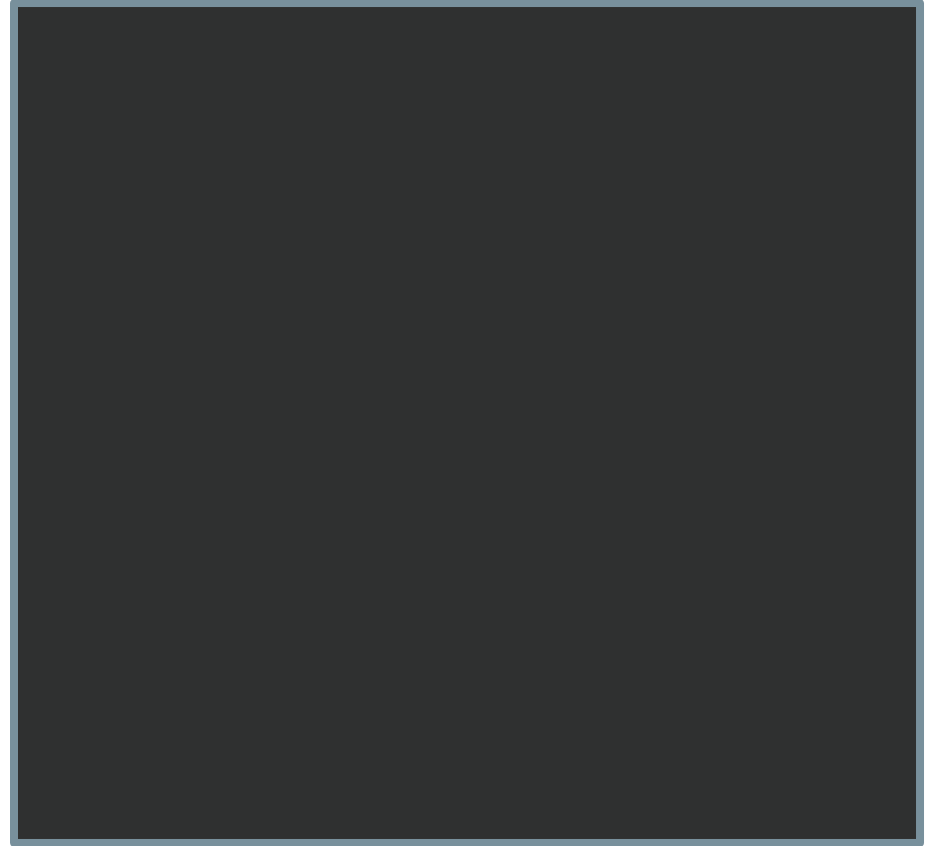
Sidecar Containers

- Secondary containers that run within a Pod alongside your app container
- Concept popularised by Istio service mesh



Sidecar Containers

- Secondary containers that run within a Pod alongside your app container
- Concept popularised by Istio service mesh
- Officially supported, and enabled by default, from Kubernetes v1.29



Sidecar Containers

- Secondary containers that run within a Pod alongside your app container
- Concept popularised by Istio service mesh
- Officially supported, and enabled by default, from Kubernetes v1.29

```
apiVersion: v1
kind: Pod
metadata:
  name: "example-pod"
spec:
  initContainers:
    - name: iptables-setup
      image: alpine
```

Sidecar Containers

- Secondary containers that run within a Pod alongside your app container
- Concept popularised by Istio service mesh
- Officially supported, and enabled by default, from Kubernetes v1.29

```
apiVersion: v1
kind: Pod
metadata:
  name: "example-pod"
spec:
  initContainers:
    - name: iptables-setup
      image: alpine
  containers:
    - name: application
      image: my-super-app:latest
```

Sidecar Containers

- Secondary containers that run within a Pod alongside your app container
- Concept popularised by Istio service mesh
- Officially supported, and enabled by default, from Kubernetes v1.29

```
apiVersion: v1
kind: Pod
metadata:
  name: "example-pod"
spec:
  initContainers:
    - name: iptables-setup
      image: alpine
  containers:
    - name: application
      image: my-super-app:latest
  sidecarContainers:
    - name: service-mesh
      image: istio:latest
```

Sidecar Containers

- Secondary containers that run within a Pod alongside your app container
- Concept popularised by Istio service mesh
- Officially supported, and enabled by default, from Kubernetes v1.29

```
apiVersion: v1
kind: Pod
metadata:
  name: "example-pod"
spec:
  initContainers:
    - name: iptables-setup
      image: alpine
  containers:
    - name: application
      image: my-super-app:latest
  sidecarContainers:
    - name: istio-mesh
      image: istio/mesh:latest
```



Sidecar Containers

- Secondary containers that run within a Pod alongside your app container
- Concept popularised by Istio service mesh
- Officially supported, and enabled by default, from Kubernetes v1.29

```
apiVersion: v1
kind: Pod
metadata:
  name: "example-pod"
spec:
  initContainers:
    - name: iptables-setup
      image: alpine
    - name: service-mesh
      image: istio
      restartPolicy: Always
  containers: ↶ This is the "magic"
    - name: application
      image: my-super-app:latest
```

Image tag & SHA

- Recommended best practice - use image SHA!

```
apiVersion: v1
kind: Pod
metadata:
  name: "tiny-pod"
spec:
  containers:
  - name: "nginx"
    image: "nginx:1.25.1@sha256:9d6b58feebd2db...2072c9496"
    imagePullPolicy: "Always"
```

Image tag & SHA

- Recommended best practice - use image SHA!
- SHA-based image tag ensure exactly the same image is used each time, even if tag it overwritten

```
apiVersion: v1
kind: Pod
metadata:
  name: "tiny-pod"
spec:
  containers:
  - name: "nginx"
    image: "nginx:1.25.1@sha256:9d6b58feebd2db...2072c9496"
    imagePullPolicy: "Always"
```

Image tag & SHA

- Recommended best practice - use image SHA!
- SHA-based image tag ensure exactly the same image is used each time, even if tag it overwritten
- If SHA is used, the tag is completely ignored and may no longer match the SHA! ⚠
 - Be careful with automated dependency updaters - make sure the sha is also updated!

```
apiVersion: v1
kind: Pod
metadata:
  name: "tiny-pod"
spec:
  containers:
  - name: "nginx"
    image: "nginx:1.25.1@sha256:9d6b58feebd2db...2072c9496"
    imagePullPolicy: "Always"
```

Image tag & SHA

- Recommended best practice - use image SHA!
- SHA-based image tag ensure exactly the same image is used each time, even if tag it overwritten
- If SHA is used, the tag is completely ignored and may no longer match the SHA! ⚠
 - Be careful with automated dependency updaters - make sure the sha is also updated!

```
apiVersion: v1
kind: Pod
metadata:
  name: "tiny-pod"
spec:
```

```
  containers:
```

```
  - name: "nginx"
```

↪ *Meaningless!!!*

```
    image: "nginx:1.25.1@sha256:9d6b58feebd2db...2072c9496"
```

```
    imagePullPolicy: "Always"
```

↪ *This is actually 1.25.2* 😱



Let's play a game!



Spot the mistake:

Pod Volume Edition

Volumes

```
apiVersion: v1
kind: Pod
metadata:
  name: "demo-pod"
spec:
  containers:
    - name: demo
      image: nginx
  volumes:
    - name: config-vol
      configMap:
        name: demo-html
```

```
apiVersion: v1
kind: Pod
metadata:
  name: "demo-pod"
spec:
  containers:
    - name: demo
      image: nginx
  volumes:
    - name: secret-vol
      secret:
        name: demo-html
```

Volumes

```
apiVersion: v1
kind: Pod
metadata:
  name: "demo-pod"
spec:
  containers:
    - name: demo
      image: nginx
  volumes:
    - name: config-vol
      configMap:
        name: demo-html
```

↪ Why not configMapName?!

```
apiVersion: v1
kind: Pod
metadata:
  name: "demo-pod"
spec:
  containers:
    - name: demo
      image: nginx
  volumes:
    - name: secret-vol
      secret:
        secretName: demo-html
```

WHY?!?! ↪

The background of the slide is a light gray field filled with a repeating pattern of various geometric shapes. These shapes include squares, triangles, circles, and wavy lines, some of which are solid and others are outlines. The pattern is dense and covers the entire background.

Networking

spec.hostname

You can control the hostname that is set in a Pod with the `hostname` and `subdomain` (among other) properties.

The Pod will then see this as its hostname:

```
/# hostname -A  
demo.example.default.svc.cluster.local
```

 *Namespace*

 *Even though this is a pod*

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: demo-pod  
  namespace: default  
spec:  
  hostname: demo  
  subdomain: example  
  containers:  
    - name: demo  
      image: nginx
```

spec.hostname

You can control the hostname that is set in a Pod with the `hostname` and `subdomain` (among other) properties.

The Pod will then see this as its hostname:

```
/# hostname -A  
demo.example.default.svc.cluster.local
```

Even though this is a pod

Namespace

Great! We can configure how our Pods can be reached, right?

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: demo-pod  
  namespace: default  
spec:  
  hostname: demo  
  subdomain: example  
  containers:  
    - name: demo  
      image: nginx
```

spec.hostname

You can control the hostname that is set in a Pod with the `hostname` and `subdomain` (among other) properties.

The Pod will then see this as its hostname:

```
/# hostname -A  
demo.example.default.svc.cluster.local
```

Even though this is a pod

Namespace

Great! We can configure how our Pods can be reached, right?



```
apiVersion: v1  
kind: Pod  
metadata:  
  name: demo-pod  
  namespace: default  
spec:  
  hostname: demo  
  subdomain: example  
  containers:  
    - name: demo  
      image: nginx
```

spec.hostname

```
bash-5.0# nslookup demo.example.default.svc.cluster.local
Server:      10.96.0.10
Address:     10.96.0.10#53

** server can't find demo.example.default.svc.cluster.local: NXDOMAIN
```

spec.hostname

```
bash-5.0# nslookup demo.example.default.svc.cluster.local
```

```
Server:      10.96.0.10
```

```
Address:     10.96.0.10#53
```

```
** server can't find demo.example.default.svc.cluster.local: NXDOMAIN
```

```
bash-5.0# nslookup demo-pod.default.svc.cluster.local
```

spec.hostname

```
bash-5.0# nslookup demo.example.default.svc.cluster.local
```

```
Server:      10.96.0.10
```

```
Address:     10.96.0.10#53
```

```
** server can't find demo.example.default.svc.cluster.local: NXDOMAIN
```

```
bash-5.0# nslookup demo-pod.default.svc.cluster.local
```

```
Server:      10.96.0.10
```

```
Address:     10.96.0.10#53
```

```
** server can't find demo-pod.default.svc.cluster.local: NXDOMAIN
```

 *Yeah, sorry, Pods can't actually have DNS assigned to them*

Headless Services

```
apiVersion: v1
kind: Service
metadata:
  name: headless
spec:
  clusterIP: None
  selector:
    app: web-server
  ports:
    - port: 80
```

Ok, Pods can kinda get DNS thanks to “Headless Services”.

These services sit in front of pods and expose each as a DNS entry.

```
bash-5.0# nslookup headless.default.svc.cluster.local
Server:          10.96.0.10
Address:         10.96.0.10#53

Name:   headless.loki.svc.cluster.local
Address: 10.244.7.90
Name:   headless.loki.svc.cluster.local
Address: 10.244.4.67
Name:   headless.loki.svc.cluster.local
Address: 10.244.5.247
```

Headless Services

```
apiVersion: v1
kind: Service
metadata:
  name: headless
spec:
  clusterIP: None
  selector:
    app: web-server
  ports:
    - port: 80
```

Ok, Pods can kinda get DNS thanks to “Headless Services”.

These services sit in front of pods and expose each as a DNS entry.

```
bash-5.0# nslookup pod-0.headless.default.svc.cluster.local
Server:          10.96.0.10
Address:         10.96.0.10#53

Name:   pod-0.headless.loki.svc.cluster.local
Address: 10.244.7.90
```

↪ Each individual Pod also gets their own DNS
(pod-0 is the Pod name here)



Let's play another game!



Guess the default!

DNS Policy Edition

DNS Policy

```
apiVersion: v1
kind: Pod
metadata:
  name: "demo-pod"
spec:
  dnsPolicy: ???
  containers:
    - name: demo
      image: nginx
```

- a) Default
- b) ClusterFirst
- c) ClusterFirstWithHostNet
- d) None

↪ *All are valid values*

DNS Policy

```
apiVersion: v1
kind: Pod
metadata:
  name: "demo-pod"
spec:
  dnsPolicy: ClusterFirst
  containers:
    - name: demo
      image: nginx
```

↪ Obviously not the default! 😬

~~a) Default~~

b) ClusterFirst

~~c) ClusterFirstWithHostNet~~

~~d) None~~

“Default” inherits the host nodes DNS resolution config with no knowledge of Kubernetes services and endpoints.

The background of the image is a light gray field filled with a repeating pattern of various geometric shapes. These shapes include squares, triangles, circles, and wavy lines, some of which are solid gray and others are outlined in a lighter gray. The pattern is dense and covers the entire background.

Security

Irrevocable Credentials

What would you say if I told you it was possible to create credentials in Kubernetes that were impossible to rotate or revoke?

Irrevocable Credentials

What would you say if I told you it was possible to create credentials in Kubernetes that were impossible to rotate or revoke?



Irrevocable Credentials

Let me introduce the token request API!

```
kubectl create token node-controller -n kube-system
```

 *ServiceAccount name*

Irrevocable Credentials

Let me introduce the token request API!

```
kubectl create token node-controller -n kube-system
```

Congratulations! You now have a JWT auth token associated with the node-controller Service Account that cannot be revoked!

Irrevocable Credentials

```
kubectl auth whoami
ATTRIBUTE      VALUE
Username       system:serviceaccount:kube-system:node-controller
UID            bf9b4829-fdb8-41ca-a9c9-dfd2eb7a417a
Groups         [system:serviceaccounts system:serviceaccounts:kube-system system:authenticated]

kubectl auth can-i get pods
yes

kubectl auth can-i delete pods
yes

kubectl auth can-i create pods
no

kubectl auth can-i delete nodes
yes
```

Irrevocable Credentials

The only way to remove the credentials...

```
kubectl delete serviceaccount -n kube-system node-controller
```

The only alternative is to wait for the credentials to expire. Hopefully you've set sensible defaults.

```
--service-account-max-token-expiration: "24h"
```

RBAC Escalate

Any Kubernetes user that has the permission to create Roles / ClusterRoles cannot (thankfully!) use this ability to escalate their permissions by creating a new Role with more permissions than they currently have.

Kubernetes says NO! 🙅

RBAC Escalate

Any Kubernetes user that has the permission to create Roles / ClusterRoles cannot (thankfully!) use this ability to escalate their permissions by creating a new Role with more permissions than they currently have.

Kubernetes says NO! 🙅

Unless.... 😅

RBAC Escalate

Let me introduce you to the ESCALATE verb.

```
- apiGroups:
  - 'rbac.authorization.k8s.io'
  resources:
  - '*'
  verbs:
  - '*'
```

↪ Includes the ESCALATE verb, oh no! 😱

If you happen to have this you've got superpowers!

You can now edit your own (Cluster)Roles with whatever you want!

Admission Policy Blind Spot

- RBAC = additive only permissions
- Admission controllers = more complex permissions

While admissions controllers (webhooks and policies) are often used to remove specific permissions granted via RBAC - e.g. a DELETE permission could be further restricted to resources with a specific naming pattern - there is a huge blind spot.

Admission Policy Blind Spot

- RBAC = additive only permissions
- Admission controllers = more complex permissions

While admissions controllers (webhooks and policies) are often used to remove specific permissions granted via RBAC - e.g. a DELETE permission could be further restricted to resources with a specific naming pattern - there is a huge blind spot.

Admissions controllers cannot restrict themselves! Any API calls related to admission webhooks or admission policies skip over the admission controller phase!

Admission Policy Blind Spot

```
// IsExemptAdmissionConfigurationResource determines if an admission.Attributes object is describing
// the admission of a ValidatingWebhookConfiguration or a MutatingWebhookConfiguration
// or a ValidatingAdmissionPolicy or a ValidatingAdmissionPolicyBinding
func IsExemptAdmissionConfigurationResource(attr admission.Attributes) bool {
    gvk := attr.GetKind()
    if gvk.Group == "admissionregistration.k8s.io" {
        if gvk.Kind == "ValidatingWebhookConfiguration" || gvk.Kind == "MutatingWebhookConfiguration"
            || gvk.Kind == "ValidatingAdmissionPolicy" || gvk.Kind == "ValidatingAdmissionPolicyBinding"
            || gvk.Kind == "MutatingAdmissionPolicy" || gvk.Kind == "MutatingAdmissionPolicyBinding" {
            return true
        }
    }
    return false
}
```

Taken from: <https://github.com/kubernetes/kubernetes/blob/master/staging/src/k8s.io/apiserver/pkg/admission/plugin/webhook/predicates/rules/rules.go>

The background of the slide is a light gray field filled with a repeating pattern of various geometric shapes. These shapes include squares, triangles, circles, and wavy lines, some of which are solid and others are outlines. The pattern is dense and covers the entire background.

Operations

crictl pods != kubectl get pods

↪ *CRI-Cuddle!!!*

crictl pods != kubectl get pods

crictl pods != kubectl get pods

- containerd has “Pods”, these are not the same as Kubernetes “Pods”

```
/ # crictl pods --latest --output yaml
items:
- annotations:
    kubernetes.io/config.seen: "2026-03-03T13:32:20.327405939Z"
    kubernetes.io/config.source: api
  createdAt: "1772544742252447431"
  id: 40188e7730d6bba51f2674662d66a118396ca18931d633219bf5f93edc056f41
  labels:
    app.kubernetes.io/managed-by: kubectl-debug
    io.kubernetes.pod.name: node-debugger-talos-192-168-1-168-m8zgz
    io.kubernetes.pod.namespace: default
    io.kubernetes.pod.uid: 43de432d-29e3-43c6-a726-b06859b93a48
  metadata:
    attempt: 0
    name: node-debugger-talos-192-168-1-168-m8zgz
    namespace: default
    uid: 43de432d-29e3-43c6-a726-b06859b93a48
  runtimeHandler: ""
  state: SANDBOX_READY
```

Static Manifests

- All nodes can define Pods that will run on itself
- Managed by the Kubelet, not Kubernetes api-server
- Kubelet attempts to create a “mirror pod” on the api-server as a read-only view
- Cannot be edited or deleted via Kubernetes API
- Can create *hidden pods*

Static Manifests

- All nodes can defined Pods that will run on itself
- Managed by the Kubelet, not Kubernetes api-server
- Kubelet attempts to create a “mirror pod” on the api-server as a read-only view
- Cannot be edited or deleted via Kubernetes API
- Can create hidden pods

↪ Wait! What?!

Static Manifests

- All nodes can defined Pods that will run on itself
- Managed by the Kubelet, not Kubernetes api-server
- Kubelet attempts to create a “mirror pod” on the api-server as a read-only view
- Cannot be edited or deleted via Kubernetes API
- Can create hidden pods

```
$> kubectl get pods -n not-a-namespace
No resources found in not-a-namespace namespace.
```

```
$> kubectl get namespace not-a-namespace
Error from server (NotFound): namespaces "not-a-namespace" not found
```

```
apiVersion: v1
kind: Pod
metadata:
  name: "demo-pod"
  namespace: "not-a-namespace"
spec:
  dnsPolicy: ClusterFirst
  containers:
    - name: demo
      image: nginx
```

Kubelet Standalone Mode

- Useful for running Pods on single compute instances
- Leverage the Kubelet to run containers with a subset of Kubernetes features including:
 - Pods
 - initContainers
 - CNI
 - Health checks (probes)
- Limited to what is possible with static manifests (so no ConfigMaps or Volumes)
- Access kubelet via localhost REST endpoint or [cyberark/kubeletctl](https://github.com/cyberark/kubeletctl) for a more kubectl-like feel

Kubelet Standalone Mode

- Useful for running Pods on single compute instances
- Leverage the Kubelet to run containers with a subset of Kubernetes features including:
 - Pods
 - initContainers
 - CNI
 - Health checks (probes)
- Limited to what is possible with static manifests (so no ConfigMaps or Volumes)
- Access kubelet via localhost REST endpoint or [cyberark/kubeletctl](https://github.com/cyberark/kubeletctl) for a more kubectl-like feel

If you've enjoyed this talk...

Check out Marcus' **Pod Deep Dive: The Interesting Bits** talk



https://youtu.be/E_r56x92KZw

Check out Mark's **Kubernetes the Very Hard Way** on iximiuz Labs

<https://labs.iximiuz.com/courses/kubernetes-the-very-hard-way-0cbfd997>



Slides and resources
available at:

<https://go-get.link/kubecon-eu-26>



Contact Marcus at:



MarcusNoble.com



@averagemarcus.bsky.social



@Marcus@k8s.social

Contact Márk at:



sagikazarmark.com



@sagikazarmark.com



Feedback and
suggestions welcome:

<https://talkpulse.app/feedback/28XKGMMT>

Thank you