Static Pods Are Weird

Iain Smart, AmberWolf

Cloud Native and Kubernetes Edinburgh

May 2025

Whoami

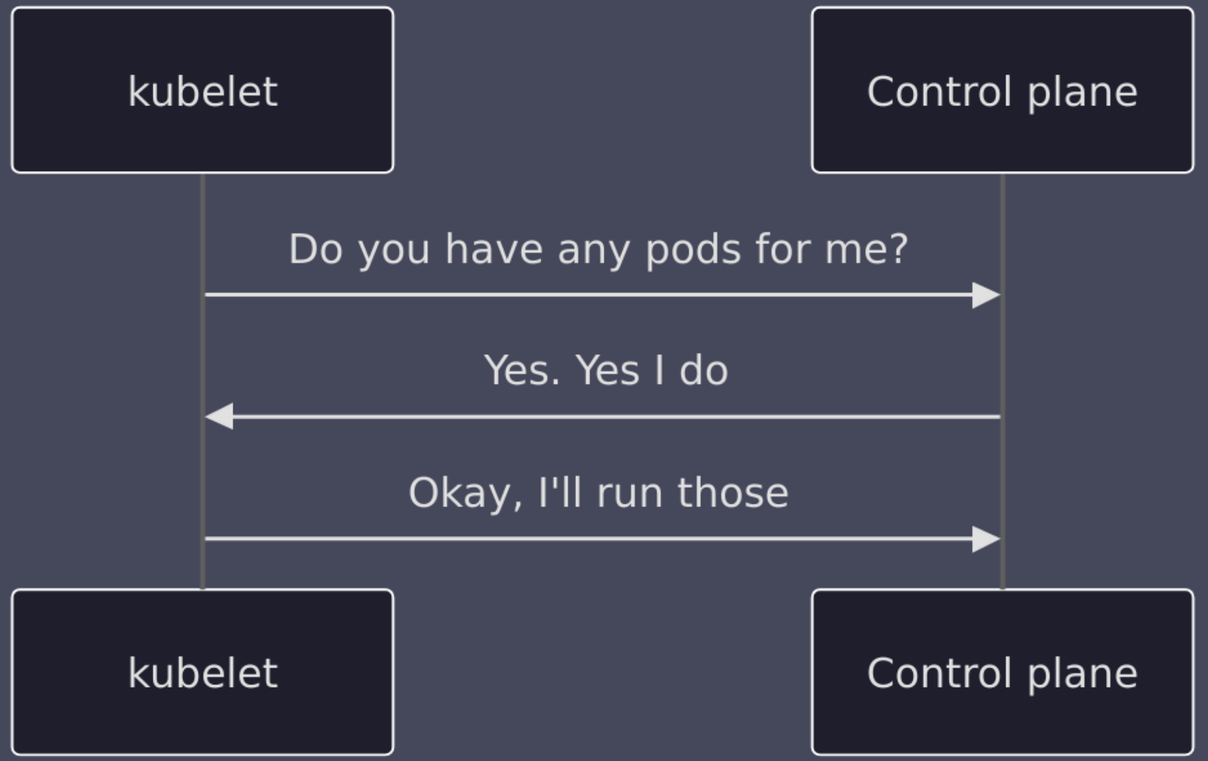
- Principal Consultant at AmberWolf
- Cloud Native Offensive Security
- K8S SIG-Security Member / Third Party Audit Subproject Lead

Agenda

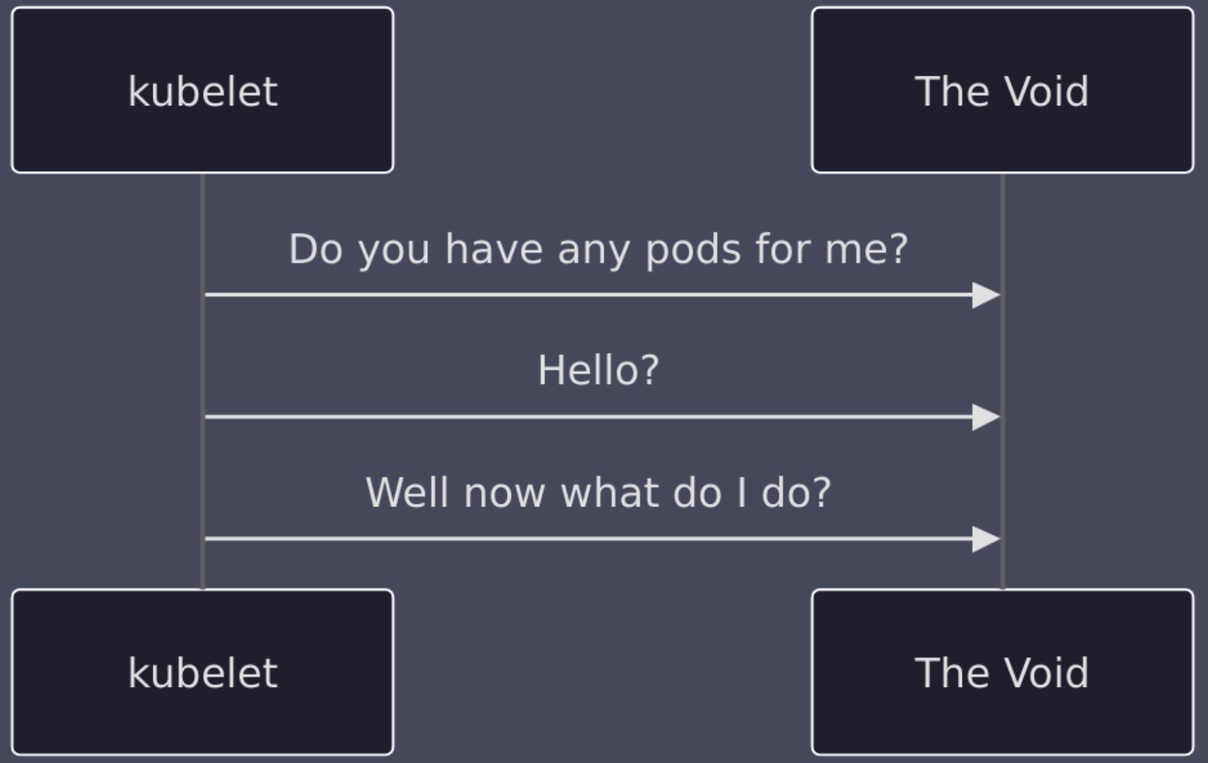
- What are Static Pods?
- Why are these interesting?

Bootstrapping

Kubernetes often runs the control plane as pods


kubelet
Control plane
Do you have any pods for me?
Yes. Yes I do
Okay, I'll run those
kubelet
Control plane

But the control plane determines which pods run

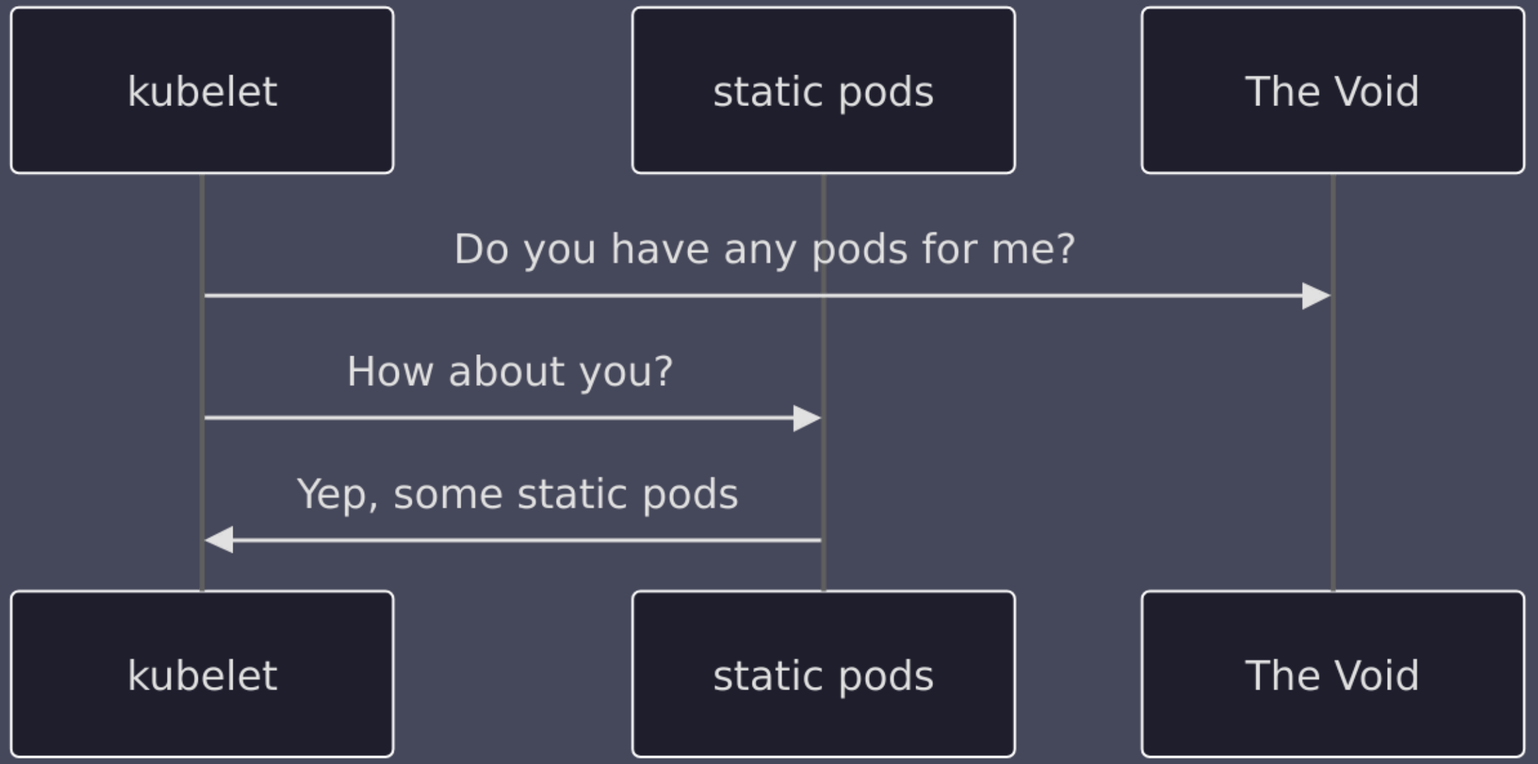

kubelet
The Void
Do you have any pods for me?
Hello?
Well now what do I do?
kubelet
The Void

So how do we make the control plane pods run?

Static Pods

```
# root@kind-control-plane:/# cat /var/lib/kubelet/config.yaml
apiVersion: kubelet.config.k8s.io/v1beta1
[...]
staticPodPath: /etc/kubernetes/manifests
[...]
```

```
root@kind-control-plane:/etc/kubernetes/manifests# ls -la
total 28
drwxr-xr-x 1 root root 4096 Oct 11 14:05 .
drwxr-xr-x 1 root root 4096 Oct 11 14:05 ..
-rw------- 1 root root 2406 Oct 11 14:05 etcd.yaml
-rw------- 1 root root 3896 Oct 11 14:05 kube-apiserver.yaml
-rw------- 1 root root 3428 Oct 11 14:05 kube-controller-manager.yaml
-rw------- 1 root root 1463 Oct 11 14:05 kube-scheduler.yaml
```


kubelet
static pods
The Void
Do you have any pods for me?
How about you?
Yep, some static pods
kubelet
static pods
The Void


kubelet
static pods
API Server
Do you have any pods for me?
How about you?
Yep, some static pods
Now I have these other pods
kubelet
static pods
API Server

Pods are running. How do we see them?

```
❯ kubectl get pods -n kube-system
NAME                                          READY   STATUS    RESTARTS   AGE
coredns-6f6b679f8f-9r7j9                      1/1     Running   0          27s
coredns-6f6b679f8f-qf4vz                      1/1     Running   0          27s
etcd-kind-control-plane                       1/1     Running   0          33s
kindnet-v5gcj                                 1/1     Running   0          27s
kube-apiserver-kind-control-plane             1/1     Running   0          33s
kube-controller-manager-kind-control-plane    1/1     Running   0          32s
kube-proxy-ltn69                              1/1     Running   0          27s
kube-scheduler-kind-control-plane             1/1     Running   0          32s
```

These are pods, and we can restrict pods

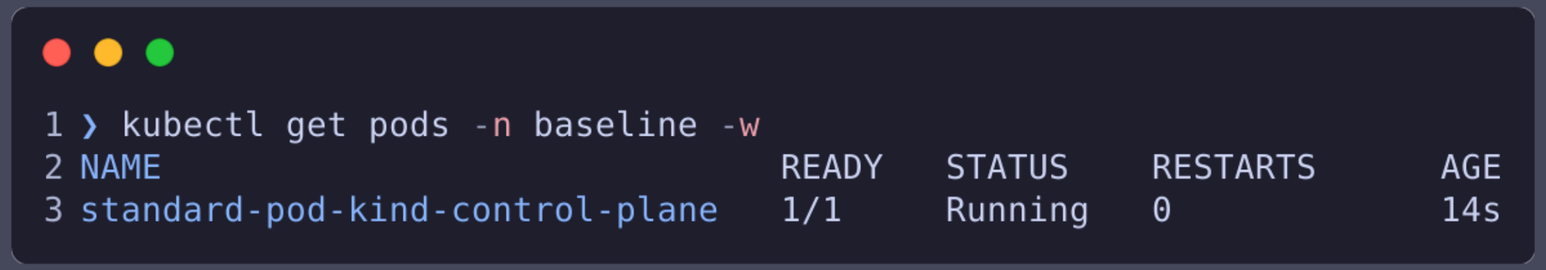
Admission Control

- Node Authorization
- Pod Security Standards
- ValidatingAdmissionPolicy
- Third Party

Let's try it out

```
❯ cat pod-standard.yaml
apiVersion: v1
kind: Pod
metadata:
  name: standard-pod
  namespace: baseline
spec:
  containers:
  - image: nginx
    name: standard-pod
```

```
❯ cat pod-privileged.yaml
apiVersion: v1
kind: Pod
metadata:
  name: privileged-pod
  namespace: baseline
spec:
  containers:
  - image: nginx
    name: privileged-pod
    securityContext:
      privileged: true
```

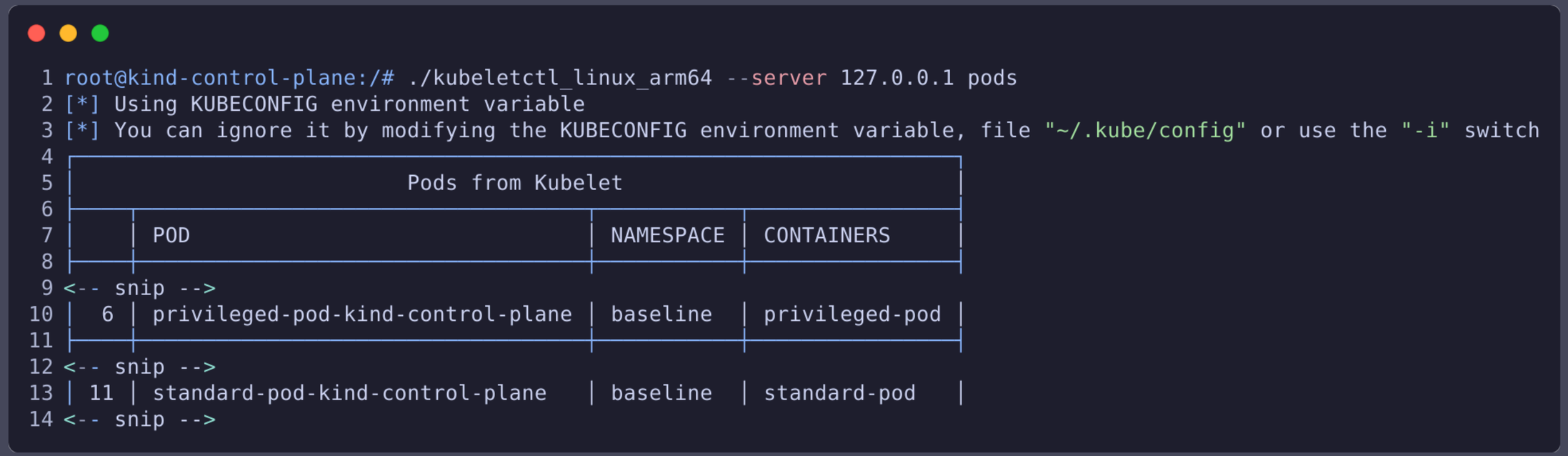

1 ❯ kubectl get pods -n baseline -w
2 NAME READY STATUS RESTARTS AGE
3 standard-pod-kind-control-plane 1/1 Running 0 14s

```
root@kind-control-plane:/# journalctl -u kubelet | grep standard-pod
Oct 12 13:38:16 kind-control-plane kubelet[693]: I1012 13:38:16.210457     693 pod_startup_latency_tracker.go:104] "Observed pod startup
duration" pod="baseline/standard-pod-kind-control-plane" podStartSLOduration=18.210431419 podStartE2EDuration="18.210431419s"
podCreationTimestamp="2024-10-12 13:37:58 +0000 UTC" firstStartedPulling="0001-01-01 00:00:00 +0000 UTC" lastFinishedPulling="0001-01-01
00:00:00 +0000 UTC" observedRunningTime="2024-10-12 13:38:16.210392461 +0000 UTC m=+55.347144485" watchObservedRunningTime="2024-10-12
13:38:16.210431419 +0000 UTC m=+55.347183443"
```

```
root@kind-control-plane:/# journalctl -u kubelet | grep privileged-pod
Oct 12 13:38:00 kind-control-plane kubelet[693]: E1012 13:38:00.645679     693 kubelet.go:1915] "Failed creating a mirror pod for" err="pods
\"privileged-pod-kind-control-plane\" is forbidden: violates PodSecurity \"baseline:latest\": privileged (container \"privileged-pod\" must
not set securityContext.privileged=true)" pod="baseline/privileged-pod-kind-control-plane"
```

```
root@kind-control-plane:/# crictl ps
CONTAINER     IMAGE         CREATED            STATE   NAME           ATTEMPT POD ID        POD
cfb34cd8921c4 048e090385966 About a minute ago Running privileged-pod 0       0057552e19e0f privileged-pod-kind-control-plane
fc9e700421fe6 048e090385966 About a minute ago Running standard-pod   0       c145bd43367e8 standard-pod-kind-control-plane
<-- snip -->
```

```
root@kind-control-plane:/etc/kubernetes/manifests# crictl pods
POD ID        CREATED            STATE NAME                                NAMESPACE ATTEMPT RUNTIME
0057552e19e0f About a minute ago Ready privileged-pod-kind-control-plane  baseline  0       (default)
c145bd43367e8 About a minute ago Ready standard-pod-kind-control-plane    baseline  0       (default)
```


root@kind-control-plane:/# ./kubeletctl_linux_arm64 --server 127.0.0.1 pods
[*] Using KUBECONFIG environment variable
[*] You can ignore it by modifying the KUBECONFIG environment variable, file "~/.kube/config" or use the "-i" switch
Pods from Kubelet
POD
NAMESPACE
CONTAINERS
<-- snip -->
6
privileged-pod-kind-control-plane
baseline
privileged-pod
<-- snip -->
11
standard-pod-kind-control-plane
baseline
standard-pod
<-- snip -->

Non-existent Namespaces

```
Oct 12 21:44:29 kind-control-plane kubelet[693]: E1012 21:44:29.888186     693 kubelet.go:1915] "Failed creating a mirror pod for"
err="namespaces \"fake-ns\" not found" pod="fake-ns/fake-ns-pod-kind-control-plane"
```

```
root@kind-control-plane:/# crictl ps
CONTAINER     IMAGE         CREATED         STATE   NAME        ATTEMPT POD ID         POD
78c94040ffd50 048e090385966 2 minutes ago   Running fake-ns-pod 0       89d585212ff8b fake-ns-pod-kind-control-plane
```

Uses

Questions

- iain@iainsmart.co.uk
- @smarticu5{.bsky.social}
- https://www.linkedin.com/in/iainsmart/

References

Blog Post this was based on

GitHub Manifests

Walls Within Walls: What if Your Attacker Knows Parkour? - Tim Allclair & Greg Castle, Google

Node Authorization

Pod Security Standards