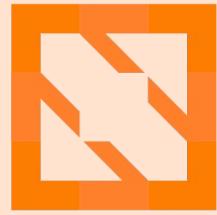




KubeCon



CloudNativeCon

---

Europe 2022

---

WELCOME TO VALENCIA





KubeCon



CloudNativeCon

Europe 2022

# Supporting Long-Lived Pods Using a Simple Kubernetes Webhook

Clement Labbe, Slack



# Supporting Long-Lived Pods



**Clem (he/him)**  
Software Engineer, *Slack*



PromCon  
North America 2021



KubeCon



CloudNativeCon

Europe 2022

# Agenda

Intro

The Problem

- Long Lived Pods

- Nodes Get Killed

Solution

- Minimum Pod Lifespan

- Node Taint Service

- Teaching the Killers

- Service Config to Pod Tolerations

- Simple Admission Webhook

Last Few Words

- Min Lifespan not Max Lifespan

- We Already Had An Admission Webhook



# Intro | Cloud Compute

**16M** Peak Concurrent Users

**45k** EC2 Instances

**162** K8s Clusters

**316** Services Deployed to K8s

**1k** Engineers

**235** Chef Roles



KubeCon



CloudNativeCon

Europe 2022

# Agenda

Intro

## The Problem

Long Lived Pods

Nodes Get Killed

## Solution

Minimum Pod Lifespan

Node Taint Service

Teaching the Killers

Service Config to Pod Tolerations

Simple Admission Webhook

## Last Few Words

Min Lifespan not Max Lifespan

We Already Had An Admission Webhook



KubeCon



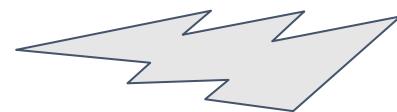
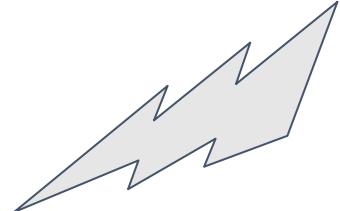
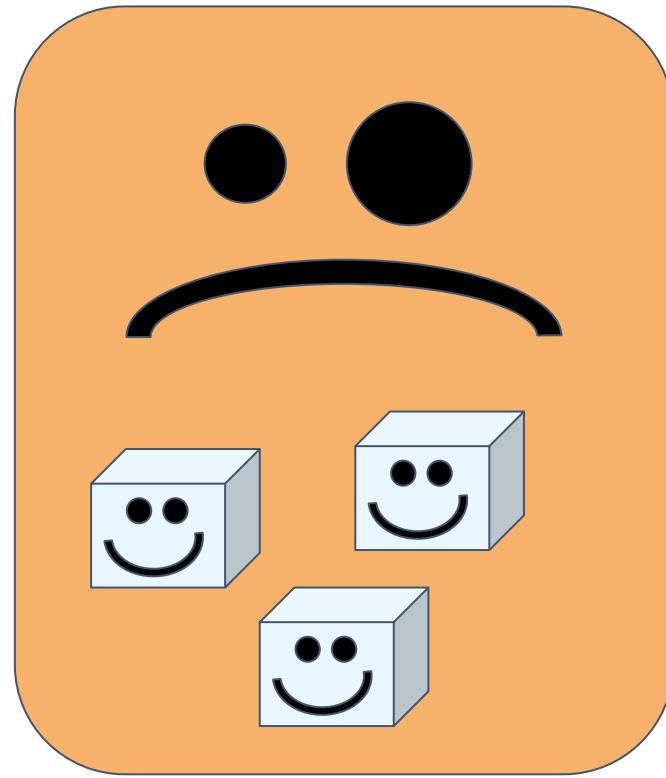
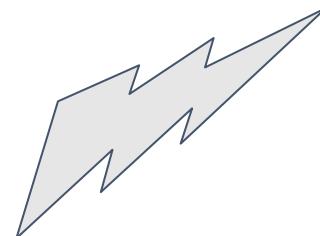
CloudNativeCon

Europe 2022

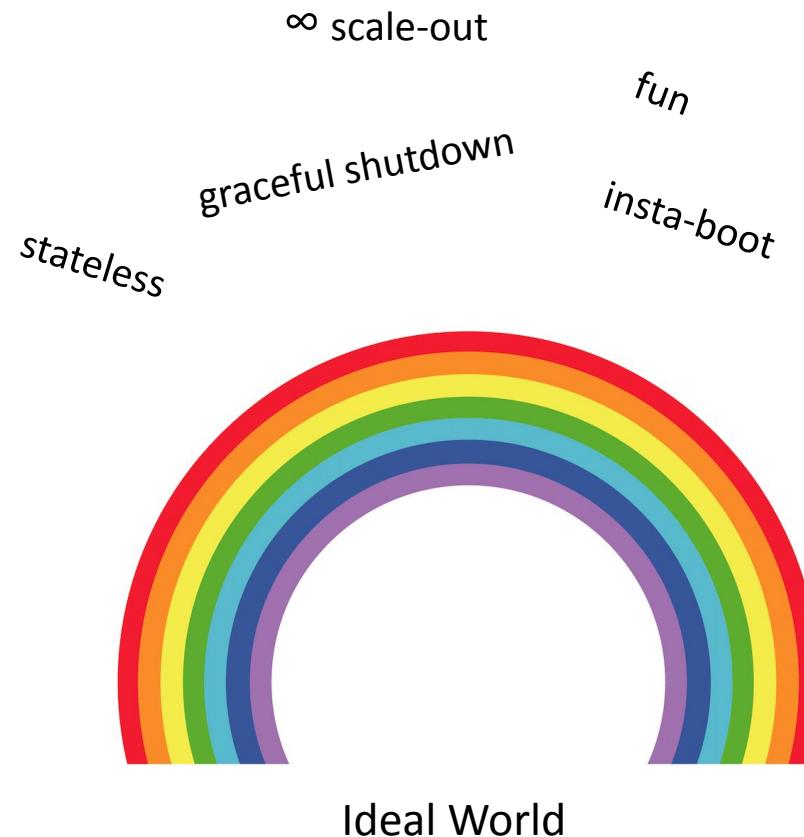
# The Problem

Some pods want a long lifespan...

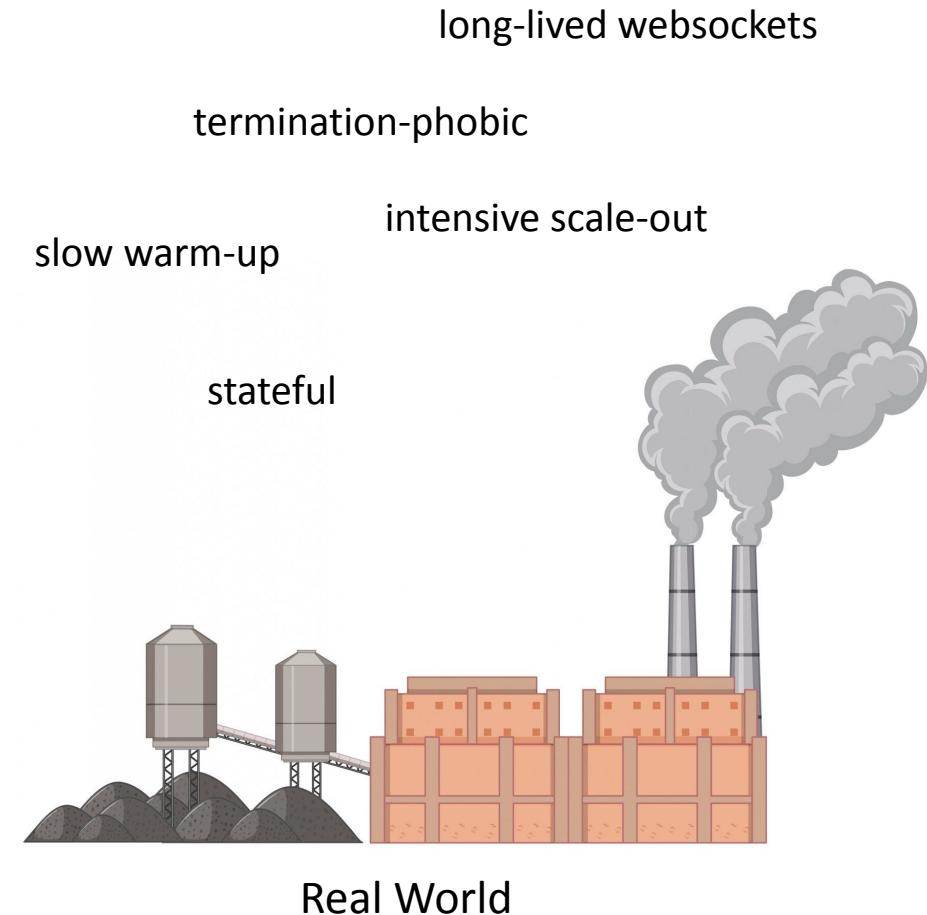
...but nodes get killed



# The Problem | Long Lived Pods



VS





KubeCon



CloudNativeCon

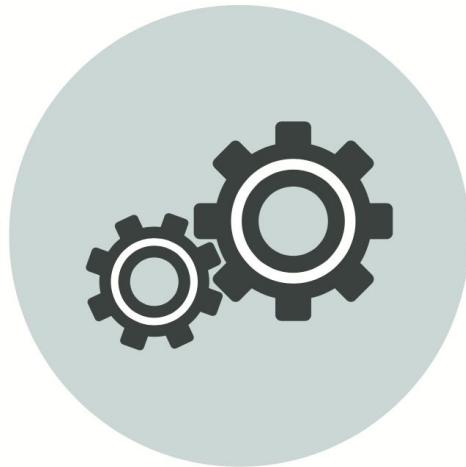
Europe 2022

...we are left with a long tail of unruly ducks



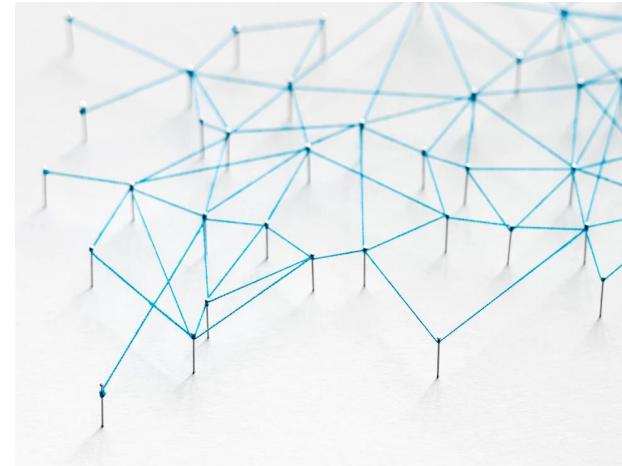
# The Problem | Long Lived Pods

Some examples...



## Batch Jobs

- Re-start work from scratch when killed mid run



## Distributed Caches

- New replicas are slow to warm up
- Pulling data from existing nodes is expensive



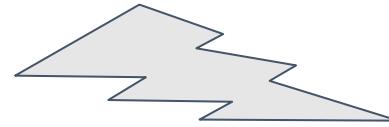
## Jenkins Controller

- Singleton
- Cannot get killed when engineers need it

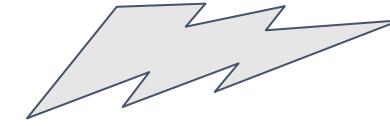
# The Problem | Nodes Get Killed

We control:

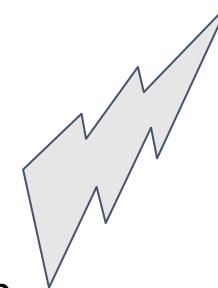
ASG scale-in



Chaos  
Engineering

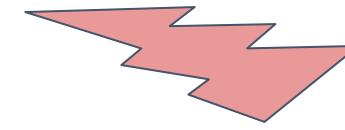
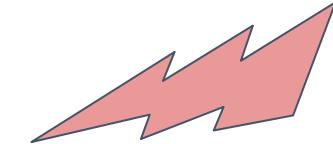


2 weeks  
anniversary stab  
in the back



We don't control:

Emergency patch  
rollout



AWS terminations



KubeCon



CloudNativeCon

Europe 2022

# Agenda

Intro

The Problem

- Long Lived Pods

- Nodes Get Killed

## **The Solution**

- Minimum Pod Lifespan

- Node Taint Service

- Teaching the Killers

- Service Config to Pod Tolerations

- Simple Admission Webhook

Last Few Words

- Min Lifespan not Max Lifespan

- We Already Had An Admission Webhook





KubeCon



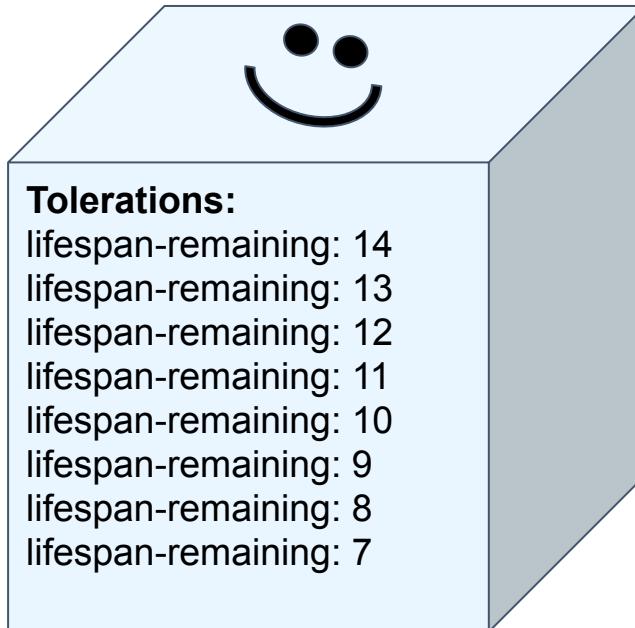
CloudNativeCon

Europe 2022

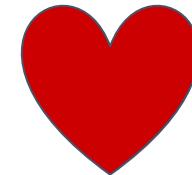
# The Solution | Minimum Pod Lifespan

Matching Pods and Nodes with Tolerations and Taints:

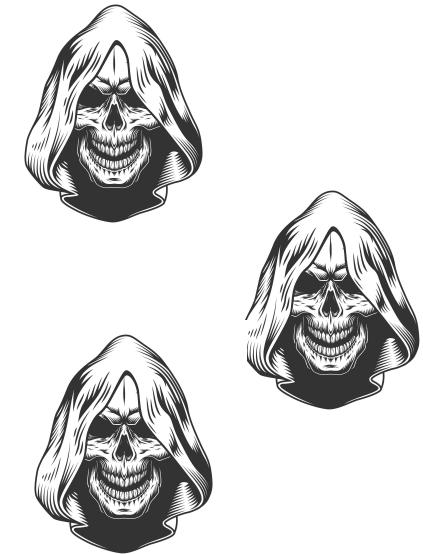
- Nodes live for 14 days



Pod wants a 7 days lifespan



Node is 4 days old:  
 $14 - 4 = 10$





KubeCon

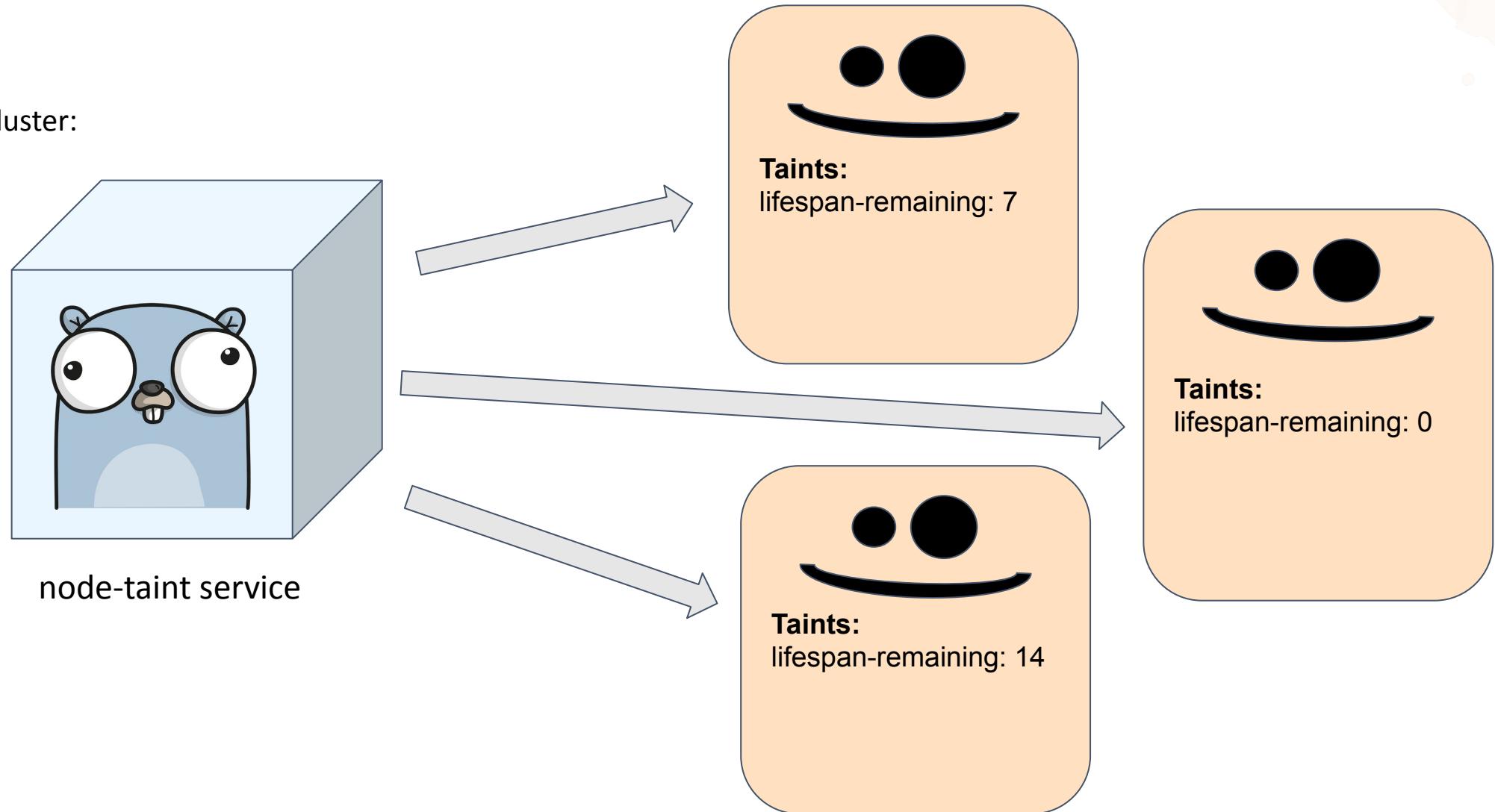


CloudNativeCon

Europe 2022

# Node Taint Service

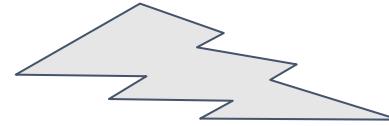
In each k8s cluster:



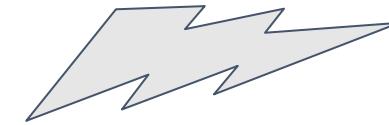
# Teaching the Killers

We control:

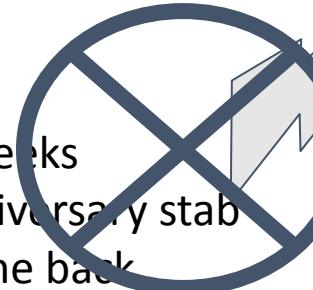
ASG scale-in  
(cluster-autoscaler)



Chaos Engineering  
(kube-test-cluster)

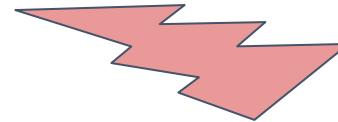
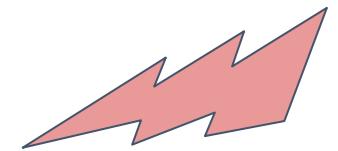


2 weeks  
anniversary stab  
in the back



We don't control:

Emergency patch  
rollout



AWS terminations

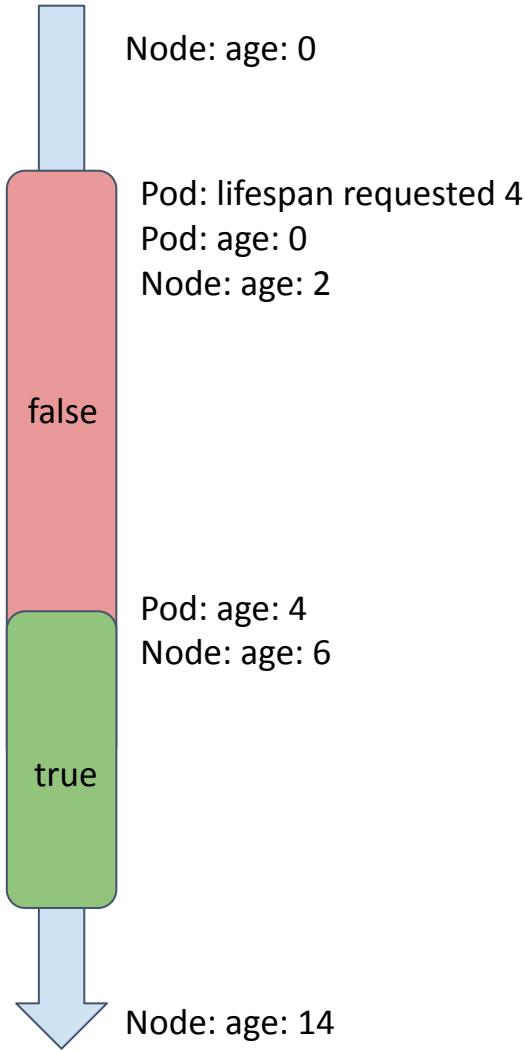
# ASG Scale-In | Cluster Autoscaler



[kubernetes/autoscaler](#)

```
---  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: cluster-autoscaler  
spec:  
  template:  
    command:  
      - ./cluster-autoscaler  
      - --v=4  
      - --stderrthreshold=info  
      - --cloud-provider=aws  
      - --balance-similar-node-groups  
      - --expendable-pods-priority-cutoff=-10  
      - --scale-down-delay-after-add=7200s  
      - --skip-nodes-with-local-storage=false  
      - --ignore-taint=slack.com/lifespan-remaining  
      - --max-total-unready-percentage=100
```

# Chaos Engineering | kube-test-cluster



```
// isLifespanEligible returns a boolean indicating whether the given node can
// be terminated without violating lifespan contracts. In other words, it
// looks at all pods scheduled on the node in question, and returns false if
// any of those pods are younger than their requested lifespan.
//
func isLifespanEligible(log logrus.FieldLogger, client *bedrockClient.Client,
    cluster string, node *corev1.Node) (bool, error) {

    pods, err := client.ListPodsWithLabels(cluster, "", nil)
    if err != nil {
        return false, fmt.Errorf("listing pods: %w", err)
    }

    for _, pod := range pods {
        lifespanStr := pod.Labels[bedrock.LifespanRequestLabel]
        lifespan, _ := strconv.Atoi(lifespanStr)

        podAge := time.Since(pod.CreationTimestamp.Time)
        if podAge < 24*time.Hour*time.Duration(lifespan) {
            log.Debugf("termination would violate min lifespan %d for pod %d days old",
                lifespan, int(podAge.Minutes()/60/24))
            return false, nil
        }
    }

    return true, nil
}
```



KubeCon



CloudNativeCon

Europe 2022

# Agenda

Intro

The Problem

- Long Lived Pods

- Nodes Get Killed

The Solution

- Minimum Pod Lifespan

- Node Taint Service

- Teaching the Killers

## **Service Config to Pod Tolerations**

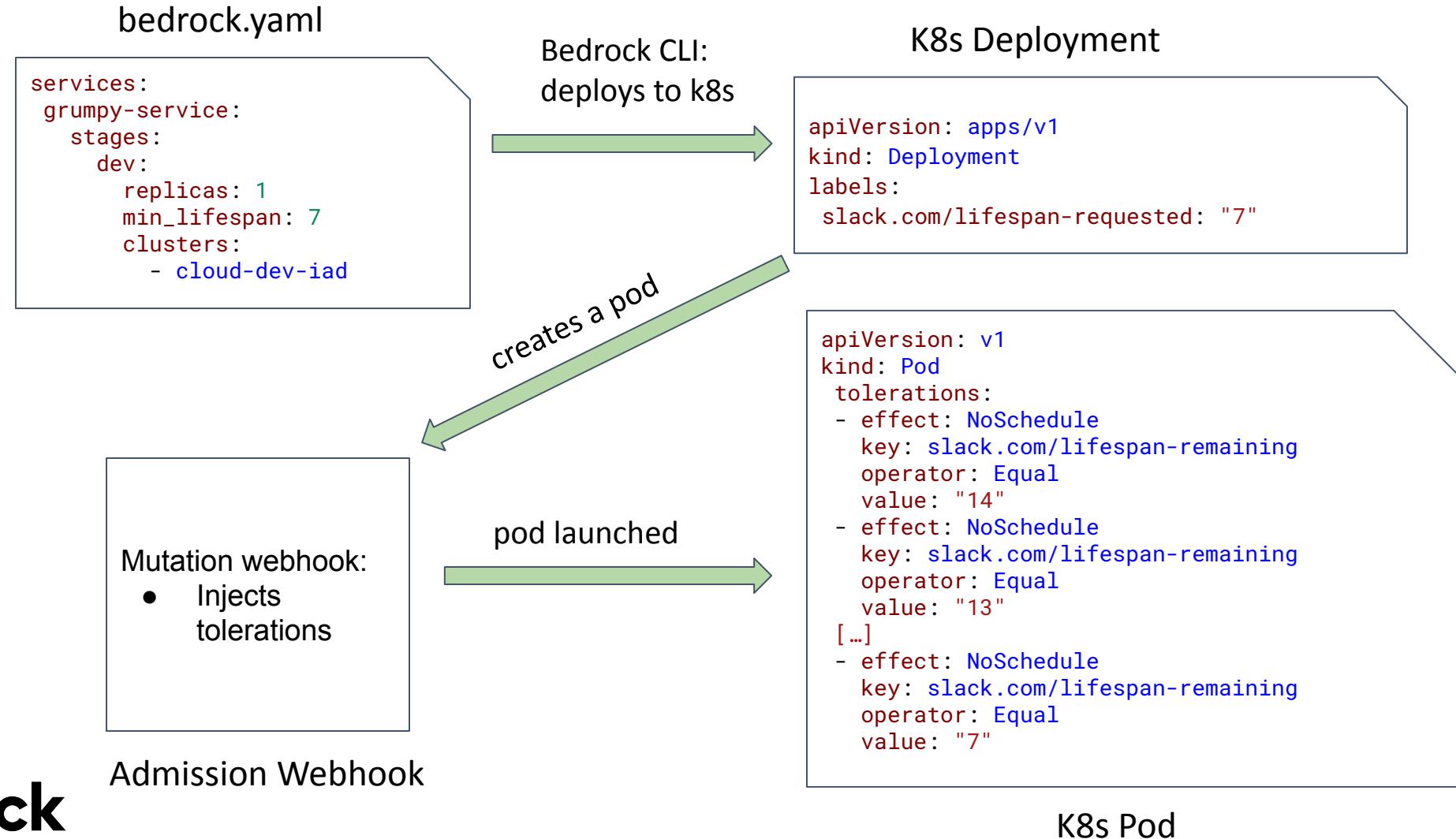
- Simple Admission Webhook

Last Few Words

- Min Lifespan not Max Lifespan

- We Already Had An Admission Webhook

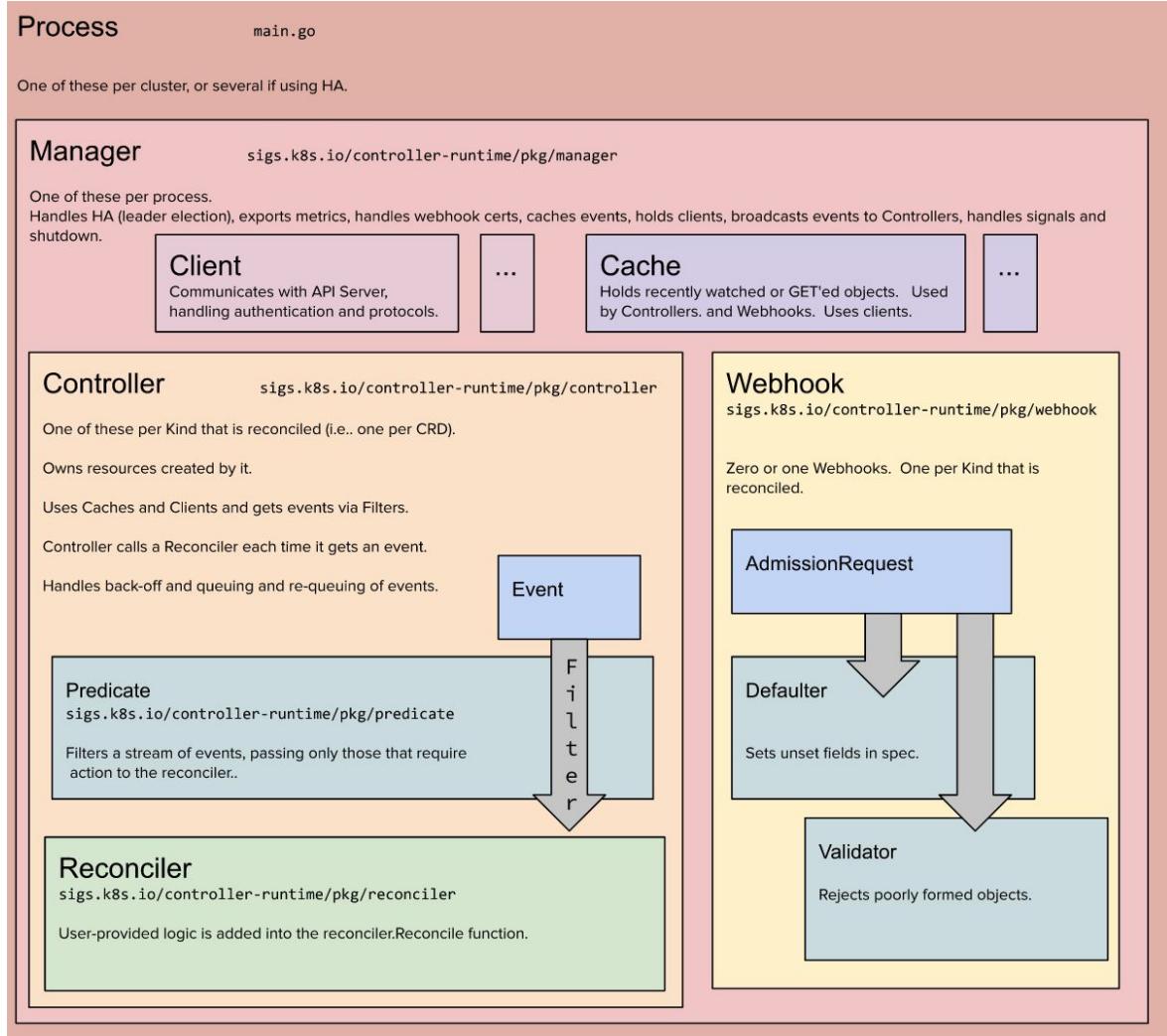
# Service Config to Pod Tolerations



# Simple Admission Webhook



Kubebuilder is a framework for building Kubernetes APIs using custom resource definitions (CRDs).



Architecture Concept Diagram

# Simple Admission Webhook

<https://github.com/slackhq/simple-kubernetes-webhook>

This is a simple [Kubernetes admission webhook](#). It is meant to be used as a validating and mutating admission webhook only and does not support any controller logic. It has been developed as a simple Go web service without using any framework or boilerplate such as kubebuilder.

This project is aimed at illustrating how to build a fully functioning admission webhook in the simplest way possible. Most existing examples found on the web rely on heavy machinery using powerful frameworks, yet fail to illustrate how to implement a lightweight webhook that can do much needed actions such as rejecting a pod for compliance reasons, or inject helpful environment variables.

For readability, this project has been stripped of the usual production items such as: observability instrumentation, release scripts, redundant deployment configurations, etc. As such, it is not meant to use as-is in a production environment. This project is, in fact, a simplified fork of a system used across all Kubernetes production environments at Slack.

## Installation

This project can fully run locally and includes automation to deploy a local Kubernetes cluster (using Kind).

```
kubectl
simple-kubernetes-webhook ✘ main via 🐫 v1.18 took 7s
❯ make cluster

🔧 Creating Kubernetes cluster ...
kind create cluster --config dev/manifests/kind/cluster.yaml
Creating cluster "kind" ...
✓ Ensuring node image (kindest/node:v1.21.1) [!]
✓ Preparing nodes [!]
✓ Writing configuration [!]
✓ Starting control-plane [!]
✓ Installing CNI [!]
✓ Installing StorageClass [!]
Set kubeconfig context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Have a nice day! 🌟

simple-kubernetes-webhook ✘ main via 🐫 v1.18 took 20s
❯ make deploy

📦 Building simple-kubernetes-webhook Docker image ...
docker build -t simple-kubernetes-webhook:latest .
[+] Building 2.5s (12/12) FINISHED
  ⇒ [internal] load build definition from Dockerfile
  ⇒ ⇒ transferring dockerfile: 37B
  ⇒ [internal] load .dockerignore
  ⇒ ⇒ transferring context: 2B
  ⇒ resolve image config for docker.io/docker/dockerfile:experimental
  ⇒ CACHED docker-image://docker.io/docker/dockerfile:experimental@sha256:600e5c62eedff338b
  ⇒ [internal] load metadata for docker.io/library/golang:1.16
  ⇒ [build 1/4] FROM docker.io/library/golang:1.16@sha256:5f6a4662de3efc6d6bb812d02e9de3d86
  ⇒ [internal] load build context
  ⇒ ⇒ transferring context: 7.97kB
```

# Simple Admission Webhook

<https://slack.engineering/simple-kubernetes-webhook/>

The screenshot shows a Slack post from the channel <https://slack.engineering/simple-kubernetes-webhook/>. The post features a large image of a modern, multi-story residential building with a colorful, patterned facade. Below the image is the caption "Photo by Mitchell Luo, Melbourne". The post is titled "A Simple Kubernetes Admission Webhook" and is attributed to Clément Labbe, Senior Software Engineer, Cloud. The post was made 11 minutes ago and written 3 months ago. It includes social sharing icons for Twitter, Facebook, and LinkedIn. The content discusses the need to mutate newly-created pods based on annotations set by users, mentioning KubeBuilder as a framework. It also mentions the creation of a stateless web service to handle POST requests with JSON. A sidebar titled "Most Recent" shows other posts, including one about handling flaky tests at scale and another about stabilizing, modularizing, and modernizing mobile codebases.

11 minutes • Written 3 months ago

While adding a recent feature to our Kubernetes compute platform, we had the need to mutate newly-created pods based on annotations set by users. The mutation needed to follow simple business rules, and didn't need to keep track of any state. Surely there must be a canonical solution to this simple problem? Well, sort of.

There are powerful frameworks like [KubeBuilder](#) which address the many aspects of writing Kubernetes admission controllers. We had simple needs, however, and decided to write our own stateless web service that replies to POST requests with a bit of JSON.

When I first heard about Kubernetes admission controllers a few years ago, it

Most Recent

Handling Flaky Tests at Scale: Auto Detection & Suppression @Arpita Patel

Stabilize, Modularize, Modernize: Scaling Slack's Mobile Codebases @Slack Engineering



KubeCon



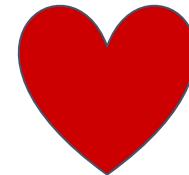
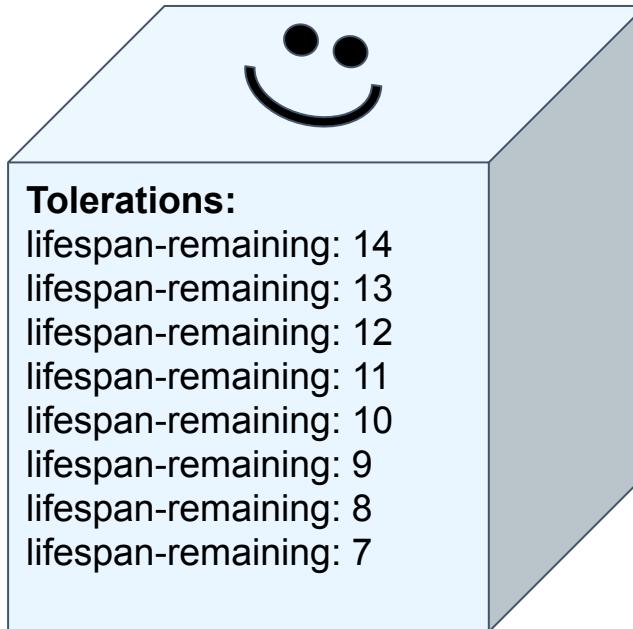
CloudNativeCon

Europe 2022

# The Solution | Minimum Pod Lifespan

Min Lifespan feature at Slack:

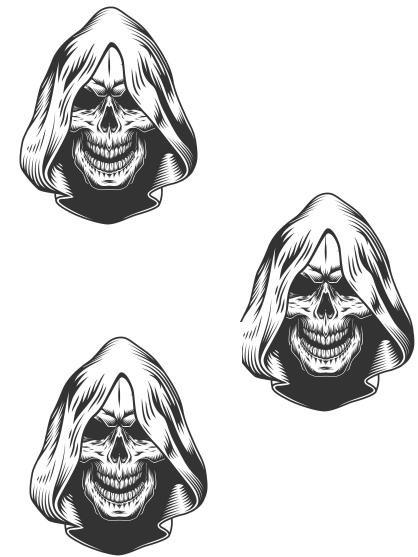
- Used by half a dozen services
- Everyone set max value of 7 days



Pod wants a 7 days lifespan



Node is 4 days old:  
 $14 - 4 = 10$





KubeCon



CloudNativeCon

Europe 2022

# Agenda

Intro

The Problem

- Long Lived Pods

- Nodes Get Killed

The Solution

- Minimum Pod Lifespan

- Node Taint Service

- Teaching the Killers

- Service Config to Pod Tolerations

- Simple Admission Webhook

Last Few Words

- Min Lifespan not Max Lifespan

- We Already Had An Admission Webhook



# Min Lifespan not Max Lifespan



Min lifespan guarantee...

- But no guarantee that the pod will get killed at the expiry date
- Users want control over lifespan dynamics
- eg. Jenkins controller

# We Already Had An Admission Webhook

Existing Admission Webhook:

- Using an ancient Kubebuilder version
- Breaking major version upgrade would require a lot of rework
- Mutates and Validates in-flight (no controllers needed)
- Decided to write the “Simple Webhook”, then migrate old to new
- Migration still ongoing





KubeCon



CloudNativeCon

Europe 2022

# Thanks



Sean Waller



Tricia Bogen



Javier Turegano



KubeCon



CloudNativeCon

Europe 2022

# Credits

Images:

- [Human skull vector created by dgim-studio - www.freepik.com](#)
- [Rainbow cartoon vector created by brgfx - www.freepik.com](#)
- [Pollution vector created by brgfx - www.freepik.com](#)
- [Shield icons created by Freepik - Flaticon](#)
- [Speech icon vector created by zirconicusso - www.freepik.com](#)
- [Village house vector created by brgfx - www.freepik.com](#)
- [Zombie heads created by brgfx - www.freepik.com](#)
- [Appointment booking vector created by freepik - www.freepik.com](#)



KubeCon



CloudNativeCon

Europe 2022

# Thank you! ....Questions?

We're Hiring!

