

A short, chaotic, and incomplete
review of custom
Kubernetes autoscalers

Cloud Native Sustainability Week 2024

Hello!

I'm Marta Paciorkowska,
and I wear many hats.

oda



Software Engineer

CNCF TAG ENV volunteer

Green Software Oslo group
co-organizer

Abstract

The Infra team in Oda went on a journey to save ~~money~~ the environment by being smarter about our **Kubernetes** resources.

After taking the first few steps, this is what we found.

oda



Abstract

The Infra team in Oda went on a journey to save ~~money~~ the environment by being smarter about our **Kubernetes** resources.

After taking the first few steps, this is what we found.



oda



Problem(s)

Abstractions add waste

A Pod is yet another layer of abstraction.

Each level comes with operational overhead and potential for waste.

Being efficient becomes extra important.

oda

Docker container

Pod

Kubernetes node

Virtual machine

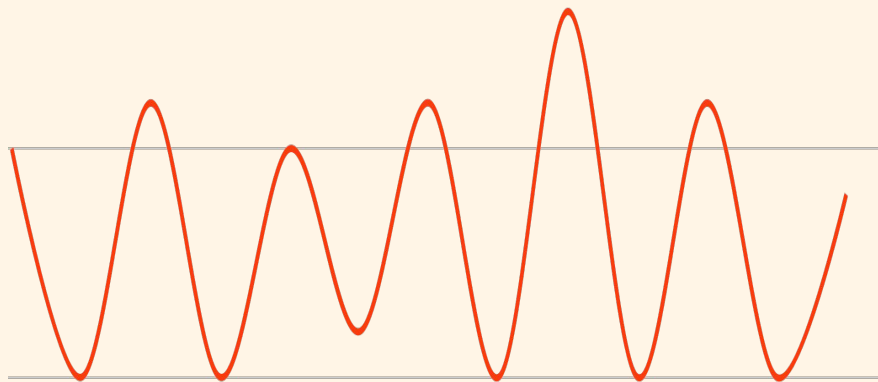
Physical server

Kubernetes daily rhythm

At night, there are mostly ghosts in the machine.

Many workloads follow the same daily usage pattern.

Most applications will run even if they don't have to.

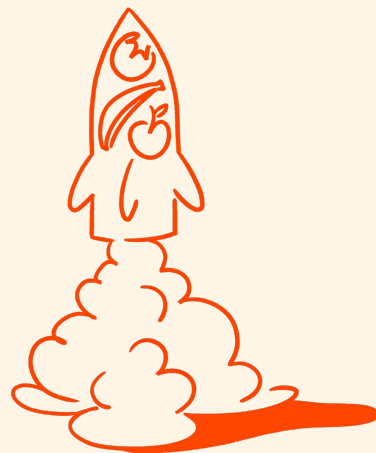


Horizontal Pod Autoscaling

While Kubernetes lets you autoscale workloads using an HPA, it might be too little to achieve optimal efficiency.

Scaling on custom metrics requires a separate external Metrics Server & explicitly exposing each metric.

Impossible to scale down to 0.



Enter... kube-green!



Part of the CNCF landscape.

<https://kube-green.dev/>

Simple: put a namespace to sleep
on **a cron schedule**.

This means *Deployments* and
CronJobs will sleep (*StatefulSets*
are WIP).

Add exceptions (for example via
labels).

oda



kube-green

A Project by Davide Bianchi

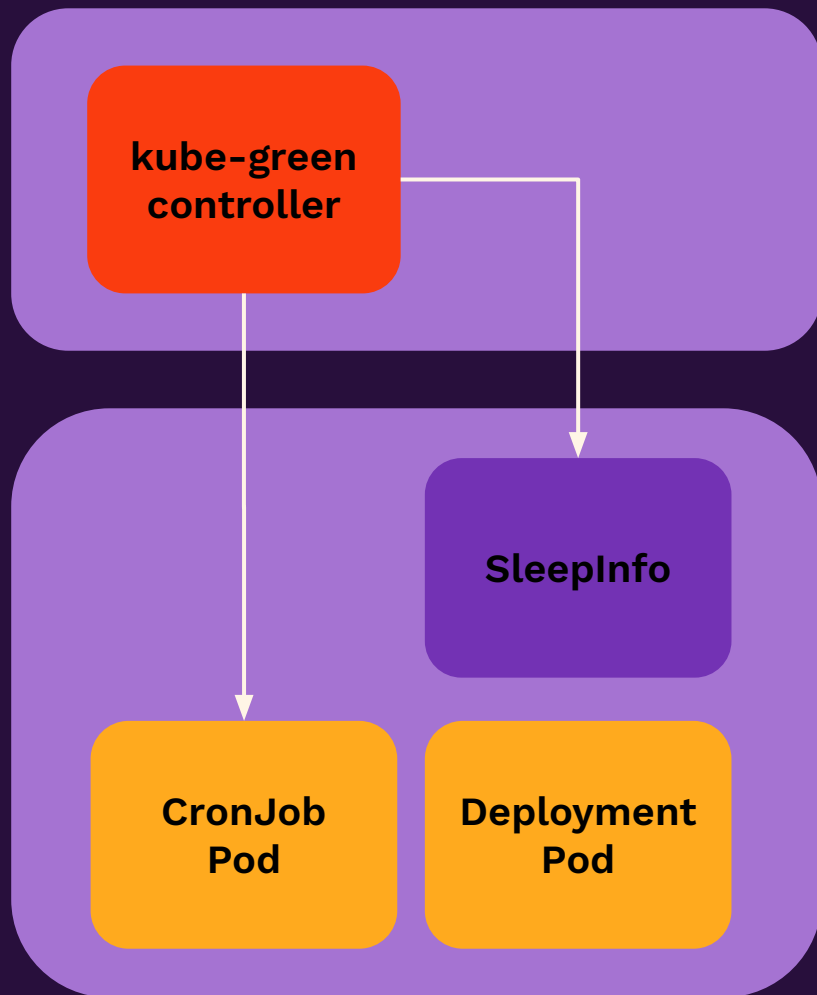
Simplicity is the focus

Kube-green only touches your Pods twice: to tuck them in and to wake them up.

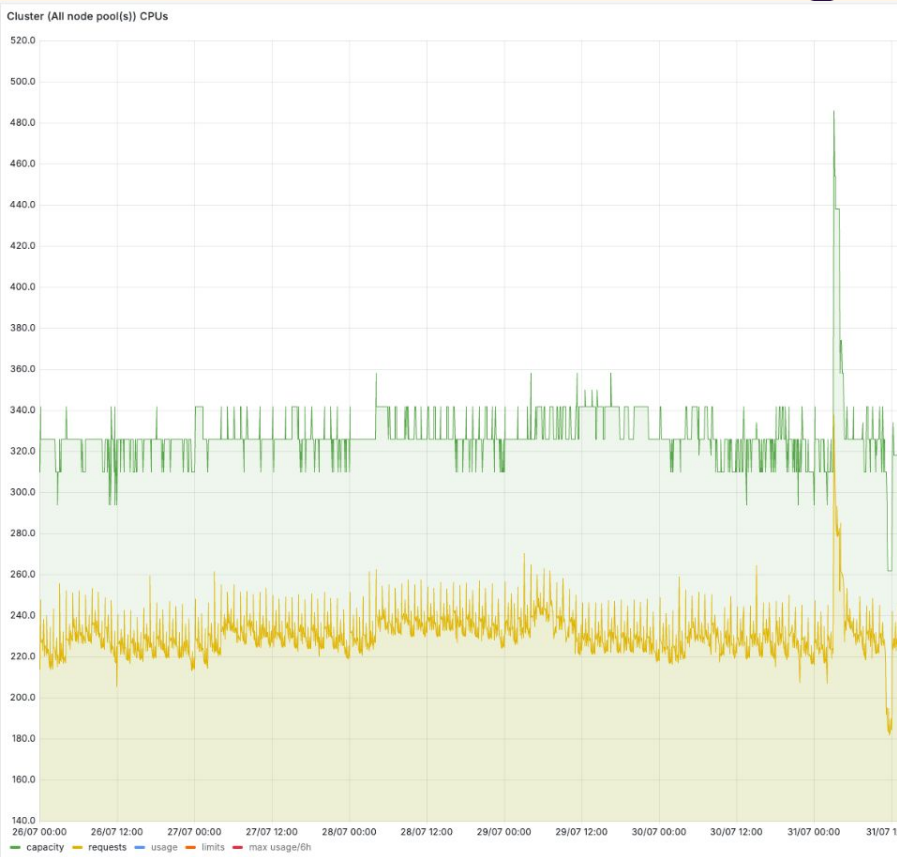
Check out their savings calculator:

<https://kube-green.dev/docs/getting-started/>

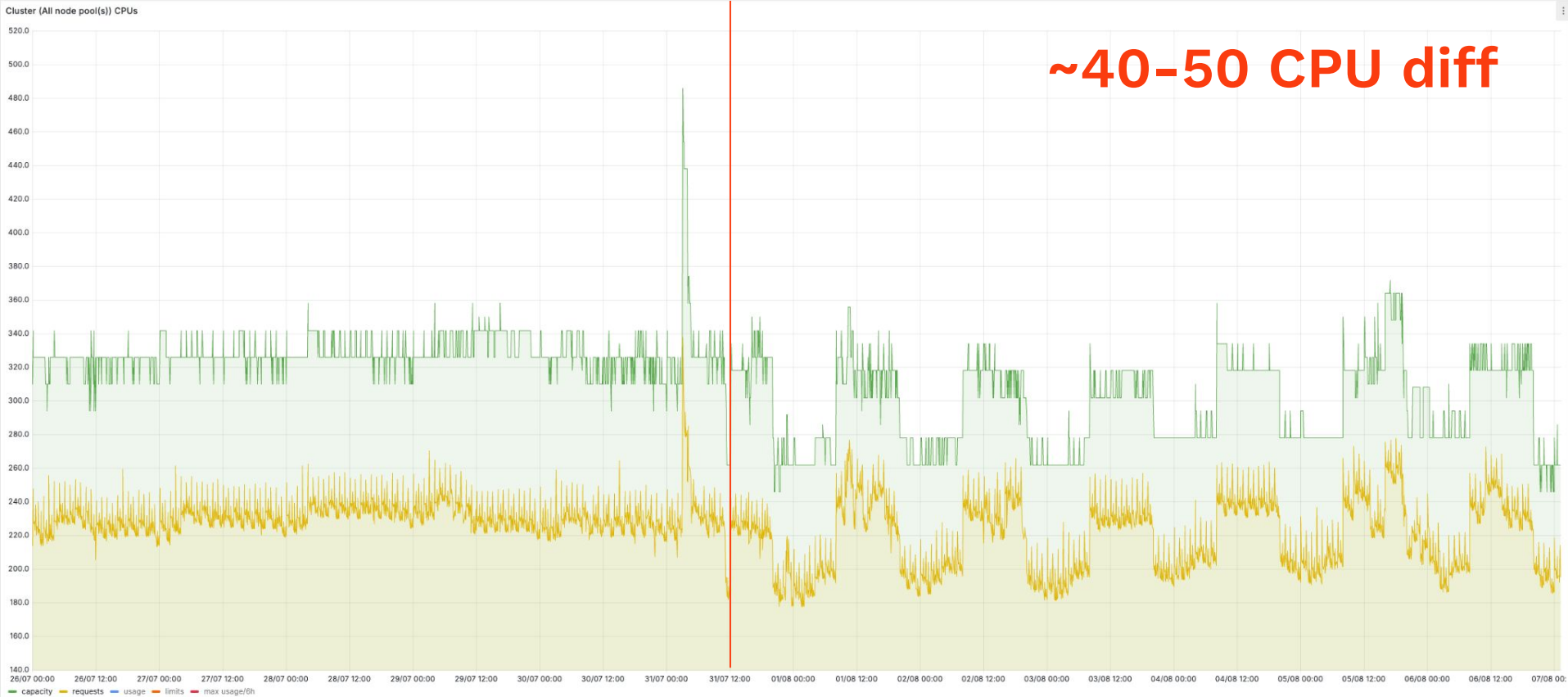
oda



Before & after kube-green



Before & after kube-green



But wait! There's more!



Part of the CNCF landscape.

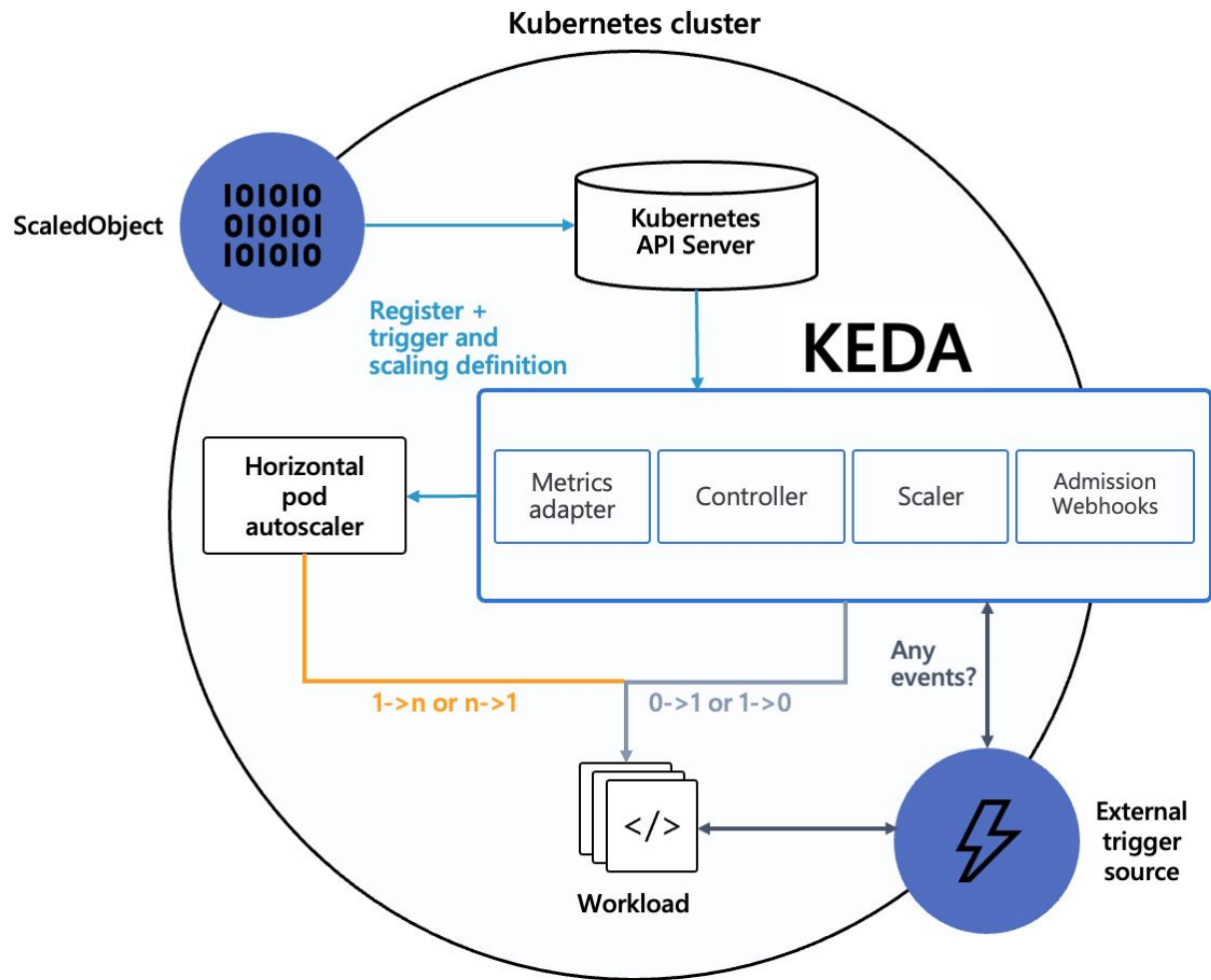
<https://keda.sh>

Scale any* workload using data from inside and outside of your cluster.

Works alongside the HPA.

oda





68 scalers available

Apache kafka

AWS cloudwatch

Cassandra

couchDB

Cron

Cpu and memory metrics

Datadog

Elasticsearch

oda

Github Runner Scaler

GCP Pub/Sub

Loki queries

MSSQL, MySQL, PostgreSQL

OracleDB

Prometheus

RabbitMQ

Redis streams

Quick comparison

kube-green

Easy to get started (deploy
controller, one CRD per
namespace)

Limited use-case? (test clusters)

Limited fine-tuning (whole
namespace*)

Single schedule (no per-day
specifics)

Can see savings fast 🙌

oda

kube-green

Easy to get started (deploy controller, one CRD per namespace)

Limited use-case? (test clusters)

Limited fine-tuning (whole namespace*)

Single schedule (no per-day specifics)

Can see savings fast 🥳

oda

keda

Takes control over the external metric server (might force architectural changes)

Fits many use-cases (test and prod clusters)

Fine-tune extravaganza (schedules, cloud vendor-specific data, integrations)

Really worth it 🥰

What we learned

What our quick & unorganized experiment taught us

It's fine to mix and match tools that do the same thing.

We will use **Keda** as our main tool for HPA.

We'll keep **kube-green** as a command-and-conquer, non-production fix.

oda

What our quick & unorganized experiment taught us

It's fine to mix and match tools that do the same thing.

We will use **Keda** as our main tool for HPA.

We'll keep **kube-green** as a command-and-conquer, non-production fix.

oda

You *can* use financial savings as a proxy for CO₂ savings, **if you know what you're doing.**

Autoscaler-induced savings can really only occur when we **reduce the numbers of Kubernetes nodes**, which makes it a cross-team effort.

Thanks for listening!

Questions?

oda

<https://cloud-native.slack.com>

marta.paciorkowska@oda.com

<https://thatmarta.wordpress.com>

@marta@oslo.town