Kubernetes on EDGE DAY

Feel the Breeze

The High-Energy, Fan-Powered Guide to Sustainable Kubernetes

David Pech & Petr Rais

Kubernetes On Edge Day 2026
Salt Lake City, Utah · November 9

Who



David Pech

wrike

GOLDEN Kubestronaut



Petr Rais

sluno

How did you get here?

One-way to Salt Lake City · illustrative routes · per person



An illustrative **5 W always-on pod** uses 43.8 kWh/year.
At Salt Lake City's grid factor: **~25 kg CO_2e/year**. Now count your replicas.

EPA 2025, Table 10: travel bars show kg CO_2; rail uses a US average, not UTA measurements. Approximate distances; flights exclude non-CO_2 effects.
Pod: assumed 5 W × 8,760 h × 568 gCO_2e/kWh (Google, 2025); IT energy only.

Even hyperscalers hit the wall

The constraint is increasingly physical · IEA update, 2026

485 TWh

Global data-centre electricity use in 2025

950 TWh

Projected for 2030
~3% of global demand

+17%

Data-centre electricity demand growth in 2025

Grid connections, transformers, permits.
IEA reports tighter bottlenecks and growing local opposition.

US developers are turning to onsite gas.
Reliable supply can require 30–70% more generation capacity than the data-centre load.

Our platform lever: **reduce wasted capacity and make suitable workloads flexible.**

IEA — Key Questions on Energy and AI (2026) · 2030 values are projections, not measured consumption. CC BY 4.0.

Kubernetes that fits in your pocket

k0s on a Raspberry Pi 4 · ARM64 · 2 GB RAM

Small edge, familiar tools

- k0s runs the Kubernetes control plane + worker
- Same Kubernetes API and `kubectl`
- Argo CD at HQ manages the remote cluster
- Prometheus scrapes edge telemetry

Rebuild from declared state

- Flash Ubuntu and apply cloud-init
- Install k0s and register the Pi at HQ
- Argo CD syncs the declared workloads
- Restore device credentials separately

> Keep the edge lean: put **Argo CD, Prometheus and Grafana at HQ**.
> Measure the actual memory and power budget on your hardware.

Demo configuration: our feel-the-breeze project repository · `pi/` + `hq/`

One HQ, N edge clusters



ArgoCD remote cluster feature

- Register edge cluster with a ServiceAccount token
- HQ ArgoCD manages apps on the remote k0s node
- No ArgoCD agent on the edge — HQ reaches out
- `ApplicationSet` fans out one app to N clusters

HQ does the heavy lifting

- Central services amortized across N sites
- HQ Prometheus scrapes edge via API-server proxy
- Suspend eligible **workloads** off-hours
- 1 site or 1000 sites: **one git repo**

Project Kepler

Kubernetes-based Efficient Power Level Exporter · CNCF Sandbox

Current: v0.12.0 · Sep 22, 2026

- Hardware energy → workload attribution → Prometheus
- CPU-time attribution using **procfs + sysfs**
- RAPL; experimental hwmon, GPU and Redfish support
- No eBPF capability required in the current architecture

What changed since the old deck?

- The eBPF/model-server design is **legacy (≤0.9)**
- v0.12 adds ARM64 images, hwmon fallback and MIG attribution via DCGM
- Per-pod watts are **attributed**, not a dedicated physical meter

> **This demo uses the older Helm chart 0.5.10.** On our Pi, Kepler values are model estimates; the USB tester measures solar input. These are different signals.

Kepler v0.12.0 release · *Current documentation* · *Legacy architecture* · *verified November 4, 2026.*

Watts are a signal. Work is the denominator.

01 / MEASURE

W → kWh

Integrate power over time. State which hardware boundary your sensor covers.

02 / ATTRIBUTE

Node → workload

Label measured inputs and estimated shares. Keep idle power visible.

03 / COMPARE

Energy per useful result

Compare the same job, request volume and SLO before and after.

Operational emissions = kWh × grid intensity

For a fuller view: **SCI = (operational + embodied emissions) / unit of work.**

Green Software Foundation — Software Carbon Intensity specification · Kepler power attribution.
To reduce node baseline power, consolidate workloads and scale down infrastructure.

Move the work, keep the deadline

A worked example · hypothetical grid signals, same completed job

RUN NOW

50 kg CO_2e

100 kWh × 500 gCO_2e/kWh

→

RUN IN A CLEANER WINDOW

10 kg CO_2e

100 kWh × 100 gCO_2e/kWh

80% less operational carbon in this example, with the same energy and output.

Good candidates
Batch analytics, CI, media transcoding, non-urgent backups.

Keep the constraints explicit
Deadline, queue age, SLO, data locality and signal freshness.

Illustrative arithmetic, not a measured saving. Account for restart, transfer and waiting overhead.
Carbon Aware KEDA Operator can cap maximum replicas using an external carbon-intensity forecast; KEDA still determines demand.

CNCF Environmental Sustainability

The former TAG now continues as a Technical Community Group (TCG)

Join the community

- Energy and carbon metrics; tooling such as Kepler
- Share measurements, patterns and case studies
- Cloud Native Sustainability Month
- Slack: **#tcg-environmental-sustainability**

Green Reviews: a new home

Now a subproject of **TAG Operational Resilience**.

Help CNCF projects assess their resource and energy footprint through structured, repeatable reviews.

> Bring one repeatable experiment. Share the workload, hardware, method and results.

Environmental Sustainability TCG · Green Reviews / TAG Operational Resilience · Current meetings · verified November 4, 2026.



Kubernetes
on EDGE DAY

Demo time!

The solar + fan control loop

A physical signal makes the scaling decision visible

SUNLIGHT

Solar panel

↓ USB power
FNB48P tester

→

RASPBERRY PI

Python / pyUSB

Measured solar watts
`solar_generation_watts`

→

HQ CLUSTER

Prometheus → KEDA

Scale controller pods

↓ local LAN commands

LED: follows solar availability
Controller scales 0 ↔ 1 with measured input power.

Fan: runs until blackout
Controller stays at 1 in sun/cloud; blackout scales it to 0.

A **renewable-availability demo**. A production carbon-aware loop needs a grid-intensity signal and workload constraints.

Meter → Pi API → HQ Prometheus/KEDA → controller → Tuya over LAN. Solar input is measured; Kepler on this Pi is estimated.

Key takeaways

1. **K8s at the small edge is mature.**
   k0s + ArgoCD-from-HQ + shared Prometheus = a real fleet pattern, not a toy.

2. **Power is observable now.**
   Kepler attributes energy to workloads. Verify the sensor coverage and label estimates.

3. **The next scaling axis is carbon.**
   Move flexible work to cleaner hours or regions, while meeting the same deadline and SLO.

Same VM, different region

`n4-standard-8` / year · 8 vCPU, 32 GB · illustrative energy model



Six continents · approximate region locations

	Region / 2025 grid factor	kg CO_2e / yr
1	**Paris** europe-west9 · 16 g/kWh	~12
2	**São Paulo** southamerica-east1 · 73 g/kWh	~56
3	**Sydney** australia-southeast1 · 468 g/kWh	~361
4	**Salt Lake City** us-west3 · 568 g/kWh	~438
5	**Mumbai** asia-south1 · 598 g/kWh	~461
6	**Johannesburg** africa-south1 · 641 g/kWh	~494

~40× spread between Paris and Johannesburg under identical energy assumptions. Check latency, capacity and data residency before moving.

Assumed *80 W IT × 8,760 h × PUE 1.10 = 770.88 kWh/year. Not measured N4 power.*
Google region data, 2025: location-based grid averages, not Google's CFE% or your billed footprint. Excludes embodied hardware, storage and network. Map: Natural Earth.

Try it this week

1. Start with one supported node

```
helm install kepler \
  oci://quay.io/sustainable_computing_io/charts/kepler \
  --namespace kepler --create-namespace
```

Check sensor support and the emitted metric names before building a dashboard. Pin chart and image versions after validation.

2. Pick one flexible workload

Record its energy, completed work and SLO. Test one cleaner time window or an eligible lower-carbon region.

3. Compare before / after

- kWh per completed job
- Operational CO_2e per job
- Runtime and deadline met
- Node idle time and baseline power

> A repeatable measurement beats a hardware-spec comparison.

Kepler quick start · This is the current upstream installation path; our Pi demo keeps its existing legacy chart.

Thank you

Questions? Come *feel the breeze* on stage.

David Pech

github.com/depeche-io

Petr Rais

github.com/petrrais

Kubernetes On Edge Day 2026
Salt Lake City, Utah · November 9

#K8sEdgeDay · #FeelTheBreeze

Demo repository + slides



Kubernetes
on EDGE DAY

Backup Video

If the demo gods are not on our side