Exploring best practices for sustainability of Kubernetes clusters and Kubernetes-hosted workloads

Marta Paciorkowska && Kristina Devochko

Who are we?

Kristina Devochko

Platform Engineer @Tietoevry

CNCF TAG ENV Tech Lead

Green Software Oslo group organizer

Marta Paciorkowska

Software Engineer @Oda

CNCF TAG ENV Project Co-Lead

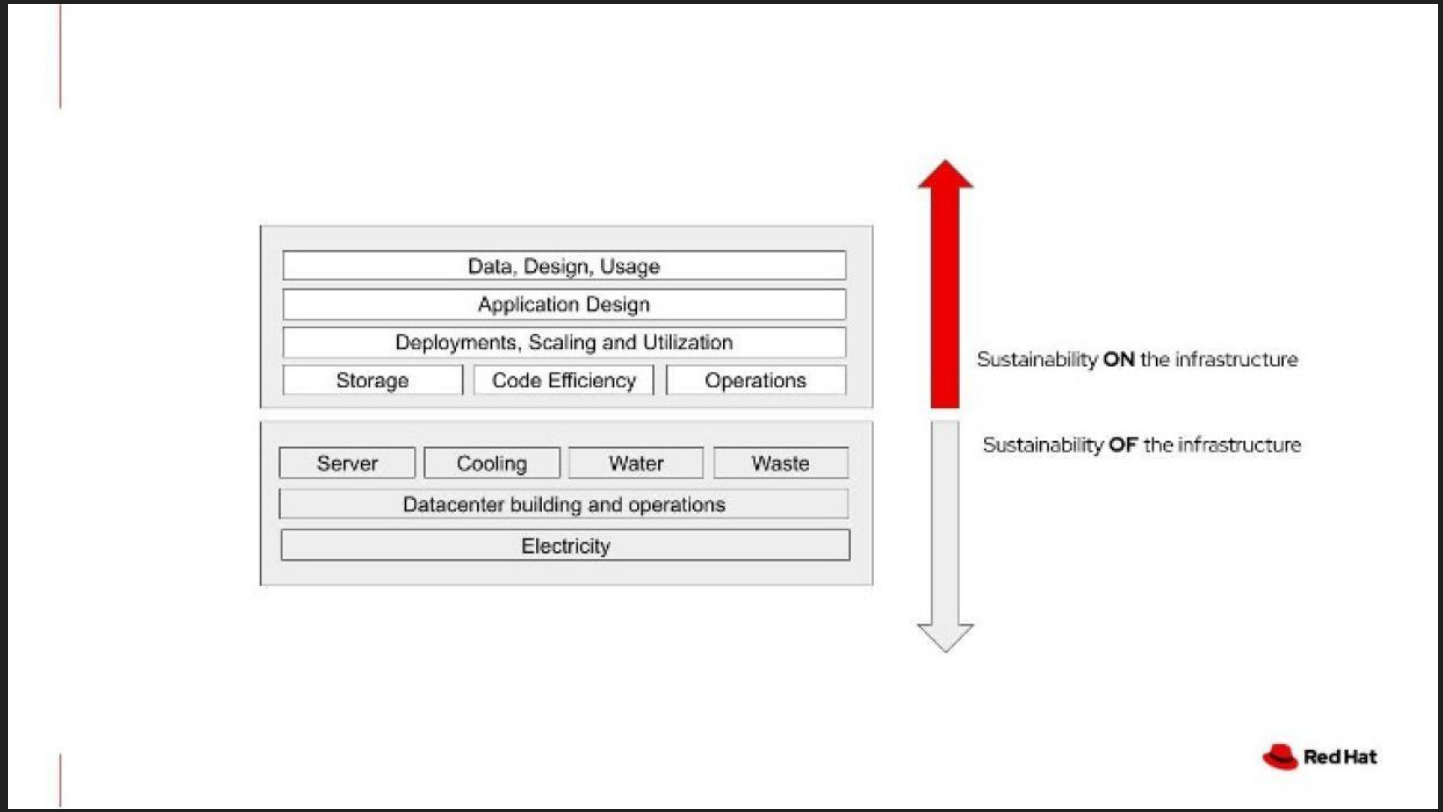
Green Software Oslo group organizer

What you'll learn today

- ★ Why is sustainability relevant?
- ★ How we think about it within Kubernetes🌱
- ★ What can **YOU** do?
- ★ Where to learn more and get involved!

Why is sustainability in context of Kubernetes relevant to you?

It’s a shared responsibility model



Why should you care about sustainability in Kubernetes?

If you work directly with Kubernetes (f.ex. as a K8s admin, platform/cloud engineer ++):

- ➢ You can directly impact efficiency and sustainability of the platforms you build and clusters you configure;
- ➢ You can guide, educate and collaborate with developers on making their applications greener, and eliminate unneeded applications;
- ➢ You can help developers understand where to improve the apps with help of monitoring and measurement tooling;

If you work indirectly with Kubernetes (f.ex. as a developer of apps hosted in K8s):

- ➢ Efficiency and sustainability of hosting platforms is directly affected by how resource-intensive your application is;
- ➢ You can guide, educate and collaborate with the tech platform team(s) on making platforms/clusters greener;
- ➢ You can impact development routines to help platform teams clean up unneeded workloads and effectivize hosting of actively used workloads;

How do we think about sustainability in context of Kubernetes?🌱

♻ Reduce, Reuse, Recycle ♻

A few concepts

- Reduce
- Refuse
- Resize
- Reschedule
- Repeat
- Repair

Reduce

Run what you **truly** need

Over a third of CPUs for cloud native applications are never used

37% 37% of CPUs for cloud-native applications are never used, on average.

46% the total impact of rightsizing and a cost-effective selection of VMs amounts to 46% in dollar terms.

60% the savings reach a substantial 60% in dollar terms by adding spot instances to applicable workloads.

The State of Kubernetes Report: Overprovisioning in Real-Life Containerized Applications

cast.ai/the-state-of-kubernetes-overprovisioning

cast

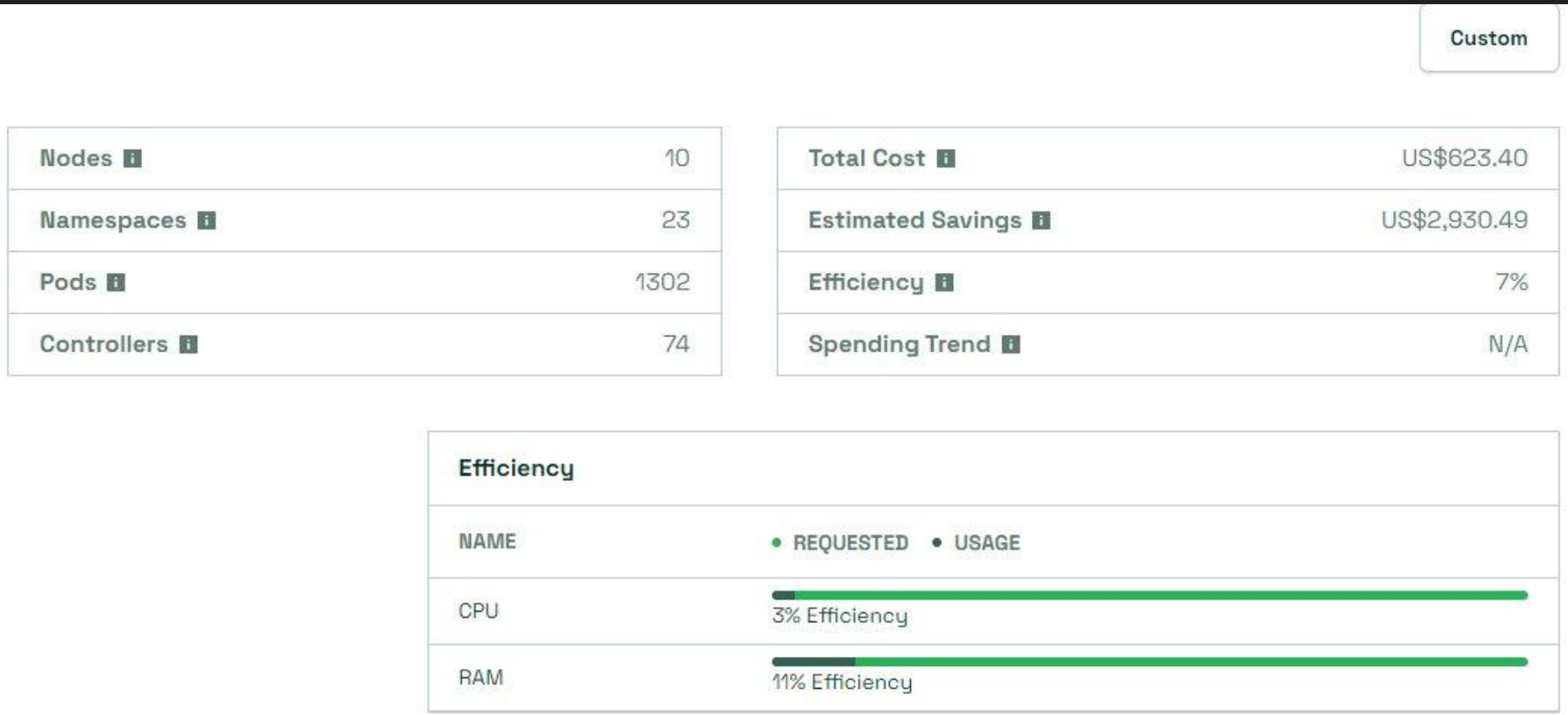
The trend appears unlikely to change in the near future given the widening gap between provisioned and requested CPUs between 2022 and 2023 (37% versus 43%). As more companies adopt Kubernetes, cloud waste will likely continue to grow.

https://cast.ai/kubernetes-cost-benchmark

Reduce

Run what you **truly** need

- Eliminate waste. Kill “zombies”. Turn off when you can. Execute regular “Dugnad”.
- Do I need this service? Do I need X replicas of this service? Can I scale this service to zero when there’s no traffic? Can I avoid overprovisioning?
- Application refactoring may be needed.
- Monitor and get help from tooling (f.ex. OpenCost, Kubecost, Cast AI, ++)



Screenshot from Kubecost tool: https://www.kubecost.com

Nodes with underutilized CPU & memory

Nodes with low memory and CPU utilization are candidates for being turned down or resized. The following nodes have sustained usage below 25% in both categories. Your cluster has enough resource availability to support turning these nodes down.

Maximum CPU/RAM Request Utilization (60%)

Node	Node Checks	Pod Checks	Recommendation	
akswinpool00003o	Passed	Passed	Safe to drain. Save $246.87 / mo.	↓
akswinpool00003i	Passed	Passed	Safe to drain. Save $246.87 / mo.	↓
akswinpool00003s	Passed	Passed	Safe to drain. Save $246.87 / mo.	↓
akswinpool00003n	Passed	Passed	Safe to drain. Save $246.87 / mo.	↓
aks-nodepool1-16599594-vmss000000	Failed	Failed	Do not drain	↓
akswinpool00003p	Failed	Passed	Do not drain	↓
akswinpool00003r	Failed	Passed	Do not drain	↓

Screenshot from Kubecost tool: https://www.kubecost.com

Refuse

You can say “no” to Pods

You can deny Pods from being scheduled on Kubernetes!

- Readiness Gates (K8s 1.30)
- kyverno

Refuse

You can say “no” to Pods

Use-case #1: you offer node pools tailored to specific workloads.

- you want Pods to always run on tailored nodes.

Refuse

You can say “no” to Pods

What are the benefits?

- potentially better node/resource utilization

What are the drawbacks?

- will only fit particular use-cases.

Refuse

You can say “no” to Pods

Use-case #2: You want to support better utilization from the start.

- you want Pods etc. to always set resource requests/limits

Refuse

You can say “no” to Pods

What are the benefits?

- potentially better node/resource utilization

What are the drawbacks?

- setting requests & limits isn’t magic - the numbers need to make sense.

Resize

Adjust capacity as the need/load changes

- Needed capacity isn't a static, one-time decision.
- Do you reeeeeally need to overprovision resources?
- Think carefully about reserved capacity..
- Spot instances aren't as scary as they seem!
  - Example story: https://itnext.io/running-production-workloads-in-eks-using-spot-instances-fc6808a7b462
- Use tooling to help with resizing.

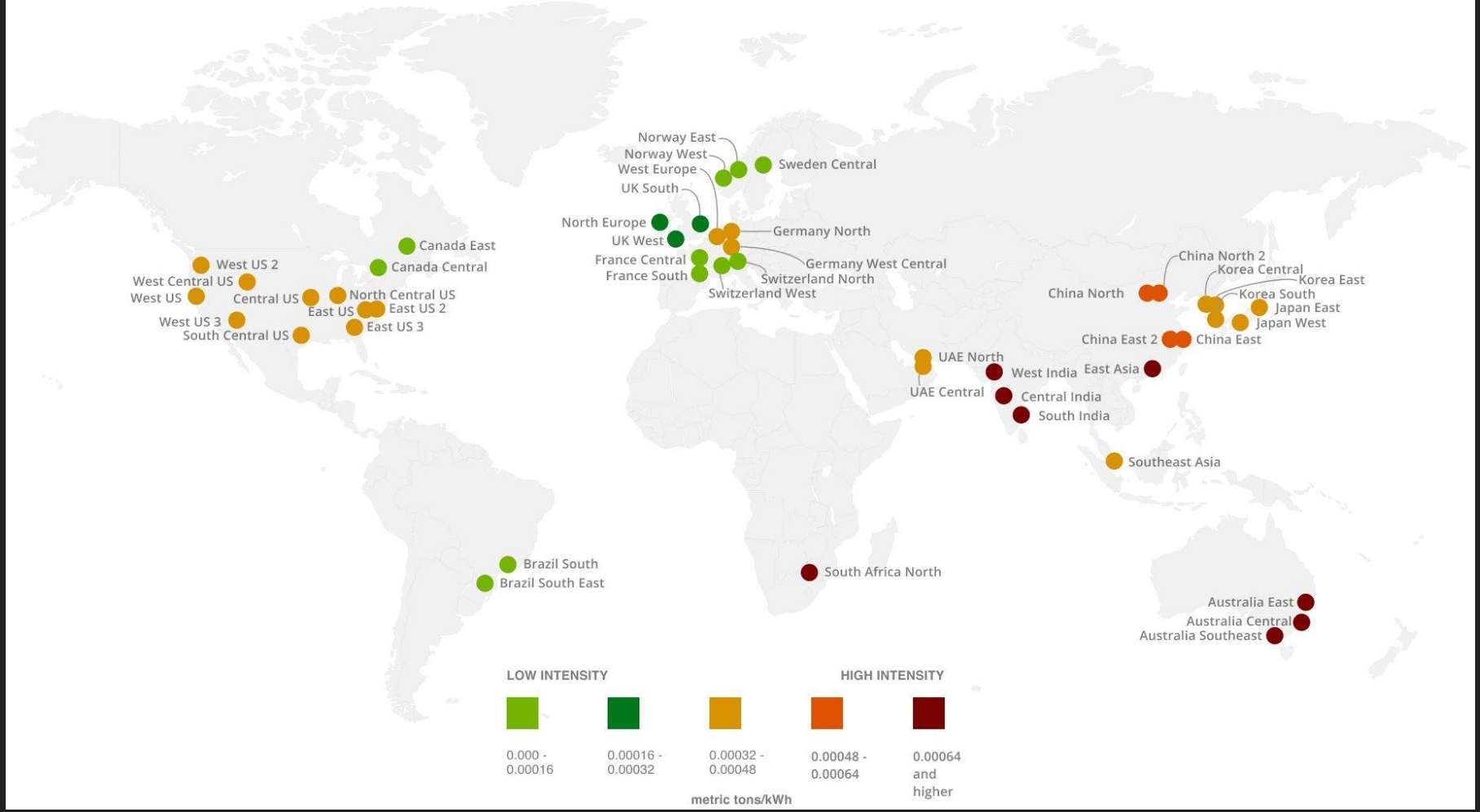
	CURRENT	RECOMMENDATION: COMPLEX	RECOMMENDATION: SIMPLE
Total cost	US$2,850.79/mo	US$421.50/mo	US$398.58/mo
Savings		US$2,429.29 (85.2%)	US$2,452.21 (86%)
Node count	9	4	3
CPU	64 VCPUs	14 VCPUs	12 VCPUs
CPU utilization	25.5% utilized	70.6% utilized	71.7% utilized
RAM	424 GB	33 GB	48 GB
RAM utilization	5.9% utilized	44.8% utilized	26.6% utilized
Instance breakdown	7 DS13 v2 x86 VCPUs: 8 VCPUs ea. RAM: 56 RAM (GB) ea. Cost: n/a 2 D4s v3 x86 VCPUs: 4 VCPUs ea. RAM: 16 RAM (GB) ea. Cost: n/a	2 B1ls x86 VCPUs: 1 VCPUs ea. RAM: 0.5 RAM (GB) ea. Cost: US$4.16/mo ea. 1 F8s v2 x86 VCPUs: 8 VCPUs ea. RAM: 16 RAM (GB) ea. Cost: US$280.32/mo ea. 1 B4ms x86	3 B4ms x86 VCPUs: 4 VCPUs ea. RAM: 16 RAM (GB) ea. Cost: US$132.86/mo ea.

Screenshot from Kubecost tool: https://www.kubecost.com

Reschedule

Can it run when the demand is less and where the energy is greener?

- Demand shifting:
  - Temporal shifting (time)
  - Spacial shifting (place)
- Be critical about the needs of your application.
- Tooling may help:
  - Cloud Carbon Footprint
  - KEDA carbon-aware operator, kube-green, Karpenter, ++



Screenshot from Cloud Carbon Footprint tool: https://www.cloudcarbonfootprint.org

Repeat

Should you run continuously?

Consider a workload

requesting

100MB of RAM 24/24h

vs

200MB of RAM 2/24h

Deployment

CronJob

Consider a workload

requesting

100MB of RAM 24/24h

vs

200MB of RAM 2/24h

Repeat

Should you run continuously?

Good use-cases:

- exports, batch jobs, database backups, ...

Bad use-case:

- API servers

Repeat

Should you run continuously?

What are the benefits?

- More predictable usage patterns

What are the drawbacks?

- Potentially time-intensive

Repair

The cost of broken workloads

- Workloads can be **visibly** or **invisibly** broken.
- Broken workloads reserve and use resources.

Repair

The cost of broken workloads

- Built-in K8s mechanisms
  - restartPolicy
  - liveness and health checks
- User-defined alerting
  - Pods that don't heal
  - idle workloads that misbehave

Measure

An example from the TAG ENV Green Reviews Working Group

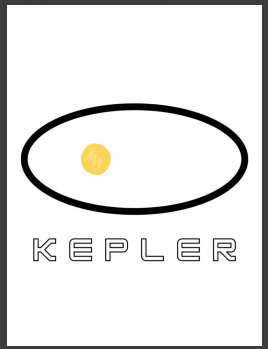
What does Green Reviews WG do?

- Provide metrics, guidelines, guidance, and processes for measuring and improving the sustainability footprint of CNCF projects.
- Leverage existing tools, libraries, and frameworks for conducting CNCF project sustainability footprint reviews whenever possible.
- Be a point of contact for project maintainers to understand the outcome of the projects' reviews.
- For more information and dashboard access: https://github.com/cncf-tags/green-reviews-tooling

Carbon emitted per kWh of energy, gCO2/kWh

Carbon emitted through the hardware that the software is running on

SCI = ((E * I) + M) per R



Energy consumed by software in kWh

Functional Unit; this is how software scales, for example per user or per device

What can **YOU** do?

A lot to choose from!

We have some suggestions😀

- Get involved/Build awareness in your organization
- Get involved with the community:
  - CNCF TAG ENV
  - Green Software Foundation
  - Green Web Foundation
  - Green Software Oslo
  - ++
- Share your learnings

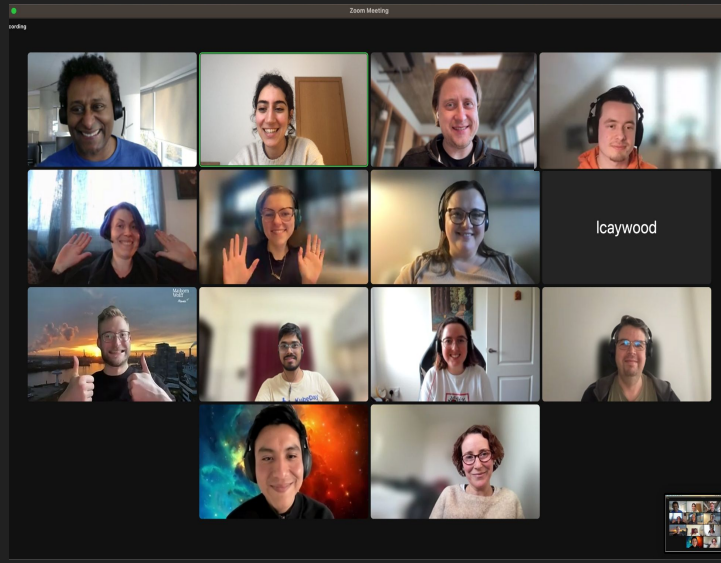
CNCF TAG Environmental Sustainability

- https://tag-env-sustainability.cncf.io
- Tech & non-tech contributions are welcome!
- Cloud Native Sustainability Week 2024 is coming to Oslo in October!💚

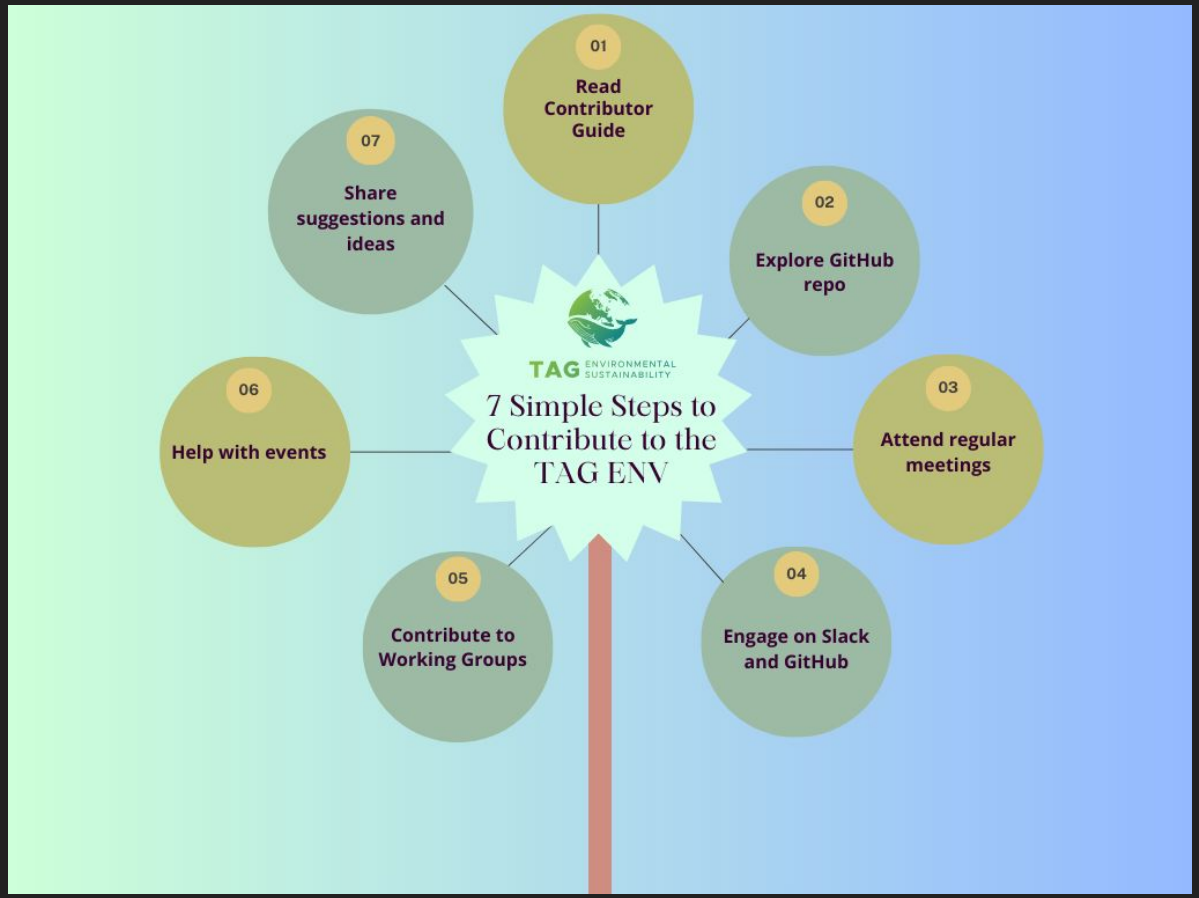
TAG ENVIRONMENTAL SUSTAINABILITY

Join us!💚

- TAG ENV is diverse, inclusive, driven.
- **1000+** contributors across Slack, LinkedIn, and X.
- **70+** contributors across TAG ENV's GitHub repos.
- **2** active Working Groups and **6** projects in planning/development.
- First-ever Cloud Native Sustainability Week 2023 with **3000+** participants across **17** countries and **4** continents.



Many ways you can contribute!



Want to know more?

Thank you!