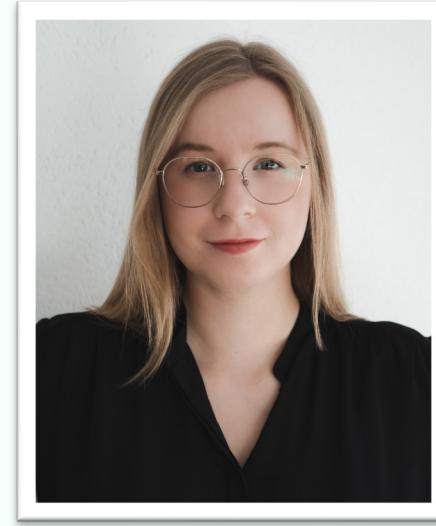


# How Kubernetes Optimization can combat climate change

Annie Talvasto

# Annie Talvasto

- CMO at VSHN
- CNCF Ambassador
- Azure MVP
- Kubernetes & CNCF meetup co-organizer
- Startup coach
- Co-host of Cloudgossip podcast - [cloudgossip.net](http://cloudgossip.net)



@AnnieTalvasto

Welcome to  
the session!

What will you  
learn?



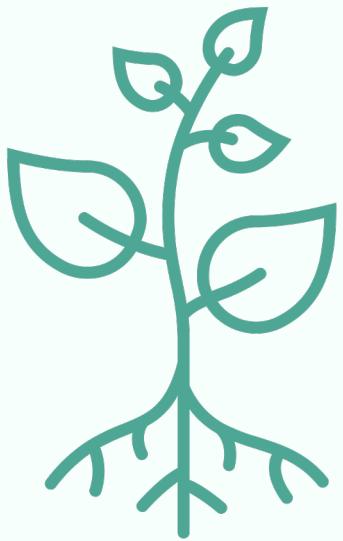
@AnnieTalvasto

# Agenda

- Introduction
- Principles of Green Software
- Principles in Practice: Microservices
- Open Source Projects: KEDA
- Resources



@AnnieTalvasto

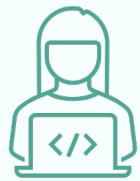


People care  
about  
**sustainability –**  
but what about  
at work?



@AnnieTalvasto

# Sustainability & software development



ICT industry consistently adds 2 to 6% of emissions each year since 2007 – the same as airline industry.



Modern technologies require more and more compute power.



World is going to experience irreversible changes if climate change remains unchecked.



@AnnieTalvasto

# KUBERNETES HAS CROSSED THE ADOPTION CHASM TO BECOME A MAINSTREAM GLOBAL TECHNOLOGY

According to CNCF's respondents, **96%** of organizations are either using or evaluating Kubernetes – a record high since our surveys began in 2016. Particularly interesting is the regional adoption of Kubernetes in production, with emerging technology hub Africa (73%) jumping ahead of

other more established tech centers including Europe (69%) and North America (55%). Additionally, 93% of respondents are currently using, or planning to use, containers in production, echoing 92% in our [2020 survey](#).

---

**96%** OF ORGANIZATIONS ARE EITHER USING OR EVALUATING KUBERNETES

---

## ARE YOU USING KUBERNETES?

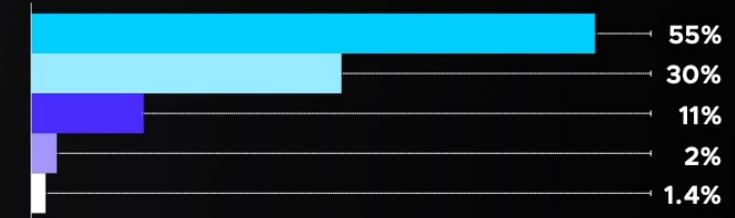
### AFRICA



### AUSTRALIA & OCENIA



### N. AMERICA



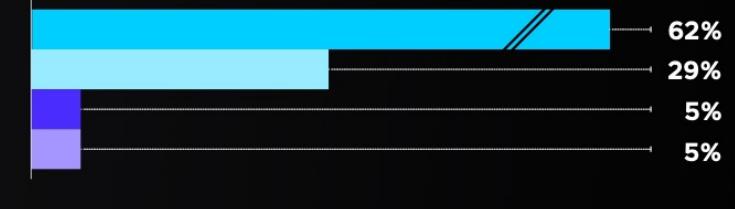
### ASIA



### EUROPE



### S. & C. AMERICA



Search

- ▶ Home
- ▶ Getting started
- ▼ Concepts
  - ▶ Overview
  - ▶ Cluster Architecture
  - ▶ Containers
  - ▶ Workloads
  - ▶ Services, Load Balancing, and Networking
  - ▶ Storage
- ▼ Configuration
  - Configuration Best Practices**
    - ConfigMaps
    - Secrets
    - Resource Management for Pods and Containers
    - Organizing Cluster Access Using kubeconfig Files
    - Resource Management for Windows nodes

# Configuration Best Practices

This document highlights and consolidates configuration best practices that are introduced throughout the user guide, Getting Started documentation, and examples.

This is a living document. If you think of something that is not on this list but might be useful to others, please don't hesitate to file an issue or submit a PR.

[!\[\]\(17acf1afa8cdf0b67c53d4865a5ed469\_img.jpg\) Edit this page](#)[!\[\]\(e8fb589d58dad1692debababa5e928b6\_img.jpg\) Create child page](#)[!\[\]\(f95dab70c751fda7d824b8b03650f7aa\_img.jpg\) Create an issue](#)[!\[\]\(e1c624d4757f08486e89482c18364c17\_img.jpg\) Print entire section](#)

General Configuration Tips

"Naked" Pods versus ReplicaSets, Deployments, and Jobs

Services

Using Labels

Using kubectl

## General Configuration Tips

- When defining configurations, specify the latest stable API version.
- Configuration files should be stored in version control before being pushed to the cluster. This allows you to quickly roll back a configuration change if necessary. It also aids cluster re-creation and restoration.
- Write your configuration files using YAML rather than JSON. Though these formats can be used interchangeably in almost all scenarios, YAML tends to be more user-friendly.
- Group related objects into a single file whenever it makes sense. One file is often easier to manage than several. See the [guestbook-all-in-one.yaml](#) file as an example of this syntax.
- Note also that many `kubectl` commands can be called on a directory. For example, you can call `kubectl apply` on a directory of config files.
- Don't specify default values unnecessarily: simple, minimal configuration will make errors less likely.
- Put object descriptions in annotations, to allow better introspection.

## "Naked" Pods versus ReplicaSets, Deployments, and Jobs

# Principles of Green Software



@AnnieTalvasto



13 Oct - 11 Nov



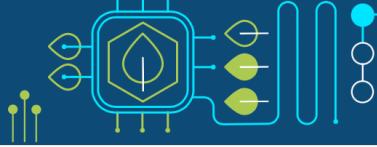
Carbon  
Hack 22

by



Green  
Software  
Foundation

Register Now



Green  
Software  
Foundation

About Working Groups Projects Resources Articles



JOIN US

We are building a trusted ecosystem of people,  
standards, tooling and best practices for

## GREEN SOFTWARE

Sign up to our newsletter...

Sign up



# Principles of Green software

- Carbon: Build applications that are carbon efficient.



@AnnieTalvasto

# Principles of Green software

- Carbon: Build applications that are carbon efficient.
- Electricity: Build applications that are energy efficient.



@AnnieTalvasto

# Principles of Green software

- Carbon: Build applications that are carbon efficient.
- Electricity: Build applications that are energy efficient.
- Carbon Intensity: Consume electricity with the lowest carbon intensity.



@AnnieTalvasto

# Principles of Green software

- Carbon: Build applications that are carbon efficient.
- Electricity: Build applications that are energy efficient.
- Carbon Intensity: Consume electricity with the lowest carbon intensity.
- Embodied Carbon: Build applications that are hardware efficient.



@AnnieTalvasto

# Principles of Green software

- Carbon: Build applications that are carbon efficient.
- Electricity: Build applications that are energy efficient.
- Carbon Intensity: Consume electricity with the lowest carbon intensity.
- Embodied Carbon: Build applications that are hardware efficient.
- Energy Proportionality: Maximize the energy efficiency of hardware.



@AnnieTalvasto

# Principles of Green software

- Carbon: Build applications that are carbon efficient.
- Electricity: Build applications that are energy efficient.
- Carbon Intensity: Consume electricity with the lowest carbon intensity.
- Embodied Carbon: Build applications that are hardware efficient.
- Energy Proportionality: Maximize the energy efficiency of hardware.
- Networking: Reduce the amount of data and distance it must travel across the network.



# Principles of Green software

- Carbon: Build applications that are carbon efficient.
- Electricity: Build applications that are energy efficient.
- Carbon Intensity: Consume electricity with the lowest carbon intensity.
- Embodied Carbon: Build applications that are hardware efficient.
- Energy Proportionality: Maximize the energy efficiency of hardware.
- Networking: Reduce the amount of data and distance it must travel across the network.
- Demand Shaping: Build carbon-aware applications.



@AnnieTalvasto

# Principles of Green software

- Carbon: Build applications that are carbon efficient.
- Electricity: Build applications that are energy efficient.
- Carbon Intensity: Consume electricity with the lowest carbon intensity.
- Embodied Carbon: Build applications that are hardware efficient.
- Energy Proportionality: Maximize the energy efficiency of hardware.
- Networking: Reduce the amount of data and distance it must travel across the network.
- Demand Shaping: Build carbon-aware applications.
- Measurement & Optimization: Focus on step-by-step optimizations that increase the overall carbon efficiency.



# Principles of Green software

- Carbon: Build applications that are carbon efficient.
- Electricity: Build applications that are energy efficient.
- Carbon Intensity: Consume electricity with the lowest carbon intensity.
- Embodied Carbon: Build applications that are hardware efficient.
- Energy Proportionality: Maximize the energy efficiency of hardware.
- Networking: Reduce the amount of data and distance it must travel across the network.
- Demand Shaping: Build carbon-aware applications.
- Measurement & Optimization: Focus on step-by-step optimizations that increase the overall carbon efficiency.



@AnnieTalvasto

# Too many things? Focus on these:

- Carbon efficiency
  - Energy efficiency
  - Hardware efficiency
  - Carbon awareness



@AnnieTalvasto

# Energy



@AnnieTalvasto

# Location sifting



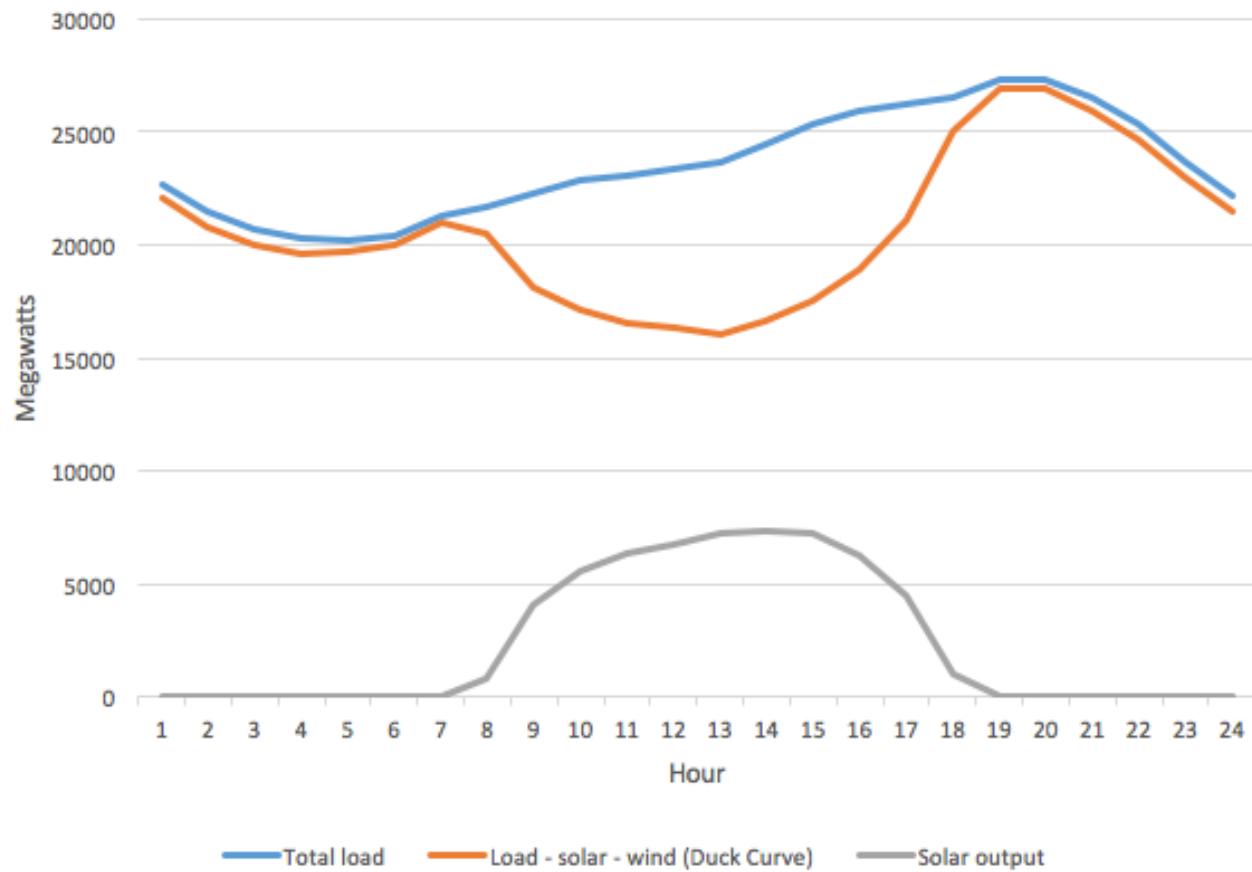
@AnnieTalvasto

# Varies by location



# Time sifting

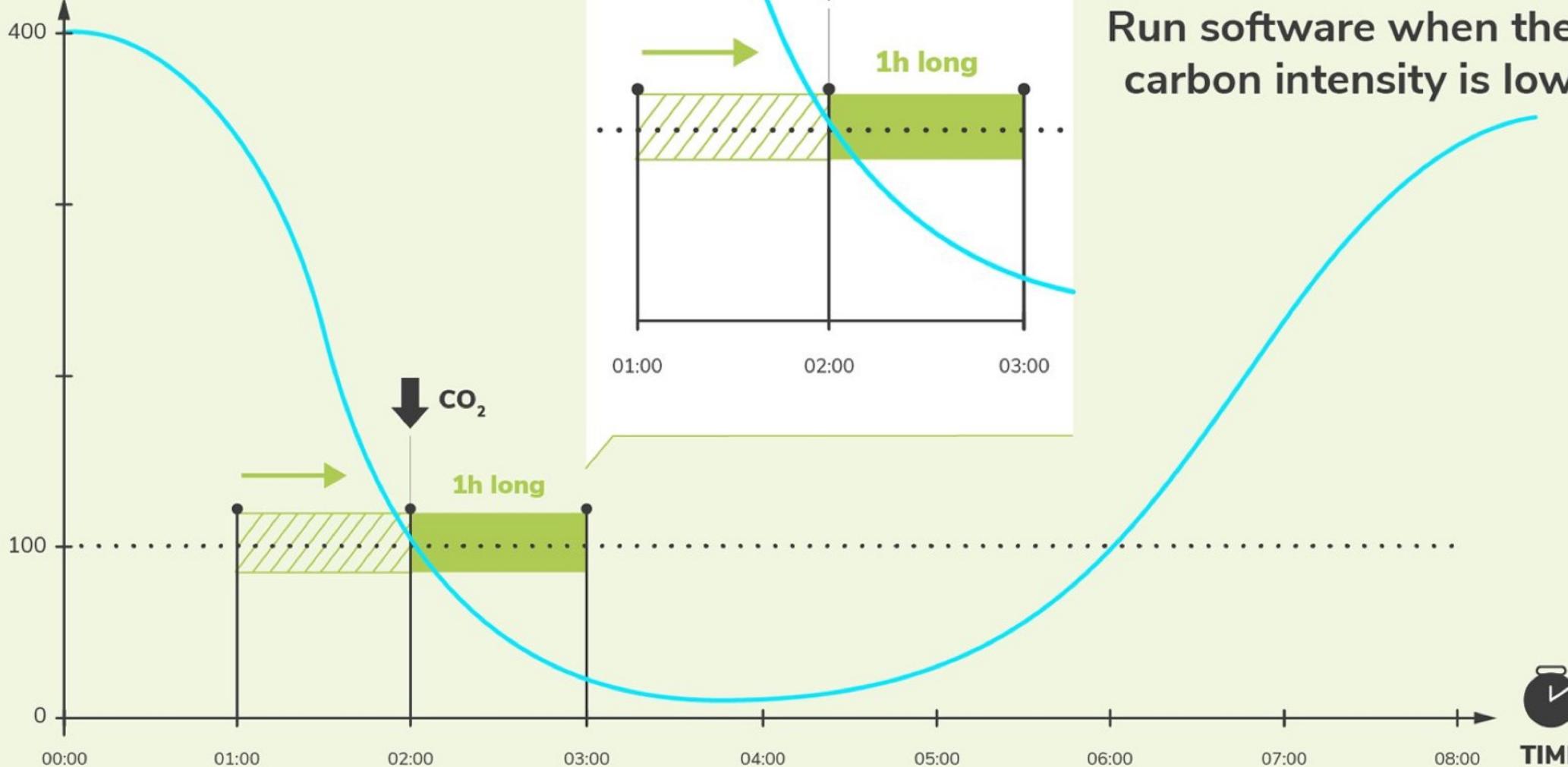
California hourly electric load vs.  
load less solar and wind (Duck Curve)  
for October 22, 2016



@AnnieTalvasto



## CARBON INTENSITY



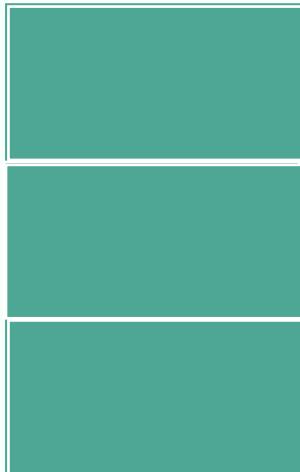
Run software when the carbon intensity is low

# Demand shaping



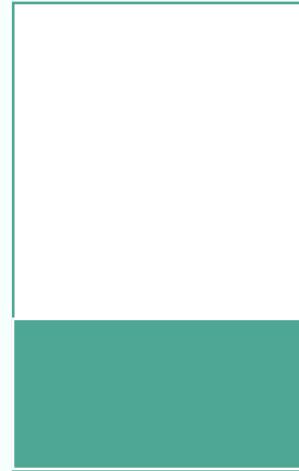
@AnnieTalvasto

# Increasing utilization



100%.

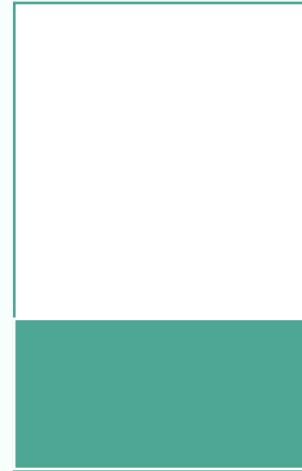
vs.



30%



30%

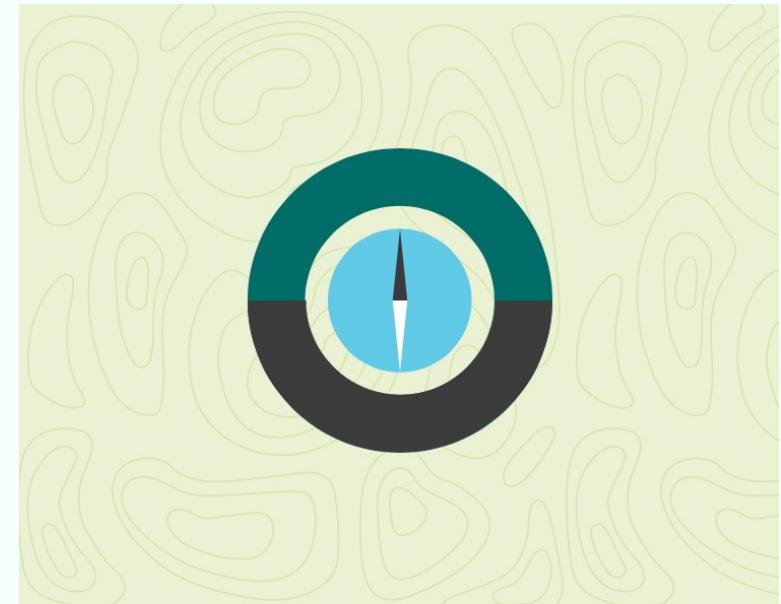


30%



@AnnieTalvasto

# Software Carbon Intensity – SCI



## **Software Carbon Intensity (SCI) Specification**

The Software Carbon Intensity (SCI) Specification defines a methodology for calculating the rate of carbon emissions for a software system. The purpose is to help users and developers make informed choices about which tools, approaches, architectures, and services they use in the future. It is a score rather than a total; lower numbers are better than higher numbers, and reaching 0 is impossible.

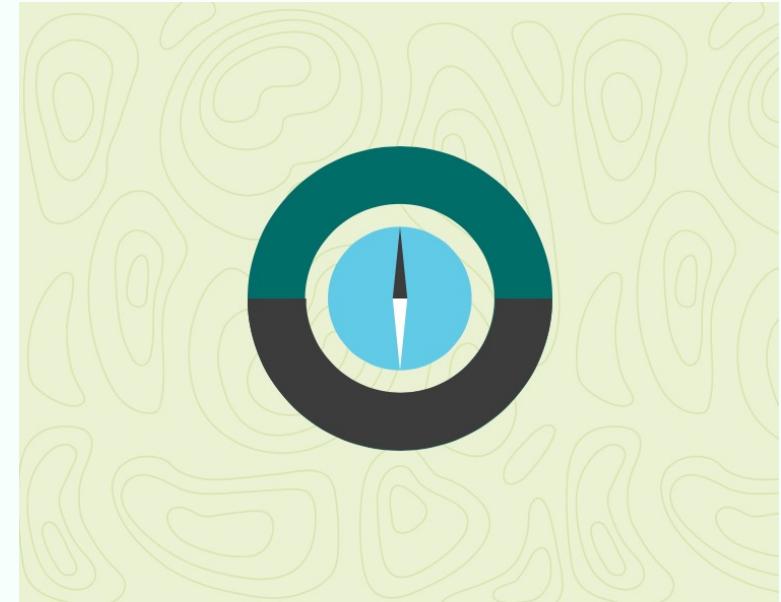


@AnnieTalvasto

# Software Carbon Intensity – SCI

Deep dive in GSF GH:

[https://github.com/Green-Software-Foundation/software\\_carbon\\_intensity/blob/main/Software\\_Carbon\\_Intensity/Software\\_Carbon\\_Intensity\\_Specification.md](https://github.com/Green-Software-Foundation/software_carbon_intensity/blob/main/Software_Carbon_Intensity/Software_Carbon_Intensity_Specification.md)



## **Software Carbon Intensity (SCI) Specification**

The Software Carbon Intensity (SCI) Specification defines a methodology for calculating the rate of carbon emissions for a software system. The purpose is to help users and developers make informed choices about which tools, approaches, architectures, and services they use in the future. It is a score rather than a total; lower numbers are better than higher numbers, and reaching 0 is impossible.



@AnnieTalvasto

# Principles of Green Software in Practice: Microservices



@AnnieTalvasto

# Microservices example (or Web Queue Worker)

- Optimize network traffic
- Increase your compute utilization
- Reduce your number of Microservices
- Optimize your database
- Understand your latency limitations



@AnnieTalvasto

# Optimize Your Network Traffic

- Traffic & architecture
- Caching headers
- CDN (Content delivery network)
- Reduce the size and optimize your bundles and static assets.
- Compression and decompression for data



@AnnieTalvasto

# Increase Your Compute Utilization

- Update workload distribution and compute resources - use less resources at a higher utilization.
- Smaller virtual machines
- PaaS
- Auto-scaling or burst capabilities
- Physical tiers & logical layers



@AnnieTalvasto

# Reduce the Number of Microservices

- Microservices architecture
- Combining services
- If two or more microservices are highly coupled, consider co-locating to reduce network congestion and latency
- Use Regions with a lower carbon intensity
- Use languages and technology stacks that optimize the efficiency of a specific microservices function.



@AnnieTalvasto

# Energy Efficiency across Programming Languages

## Energy Efficiency across Programming Languages

How Do Energy, Time, and Memory Relate?

Rui Pereira  
HASLab/INESC TEC  
Universidade do Minho, Portugal  
[rui.pereira@di.uminho.pt](mailto:rui.pereira@di.uminho.pt)

Marco Couto  
HASLab/INESC TEC  
Universidade do Minho, Portugal  
[marco.l.couto@inesctec.pt](mailto:marco.l.couto@inesctec.pt)

Francisco Ribeiro, Rui Rua  
HASLab/INESC TEC  
Universidade do Minho, Portugal  
[fribeiro@di.uminho.pt](mailto:fribeiro@di.uminho.pt)  
[rrua@di.uminho.pt](mailto:rrua@di.uminho.pt)

Jácome Cunha  
NOVA LINCS, DI, FCT  
Univ. Nova de Lisboa, Portugal  
[jacome@fct.unl.pt](mailto:jacome@fct.unl.pt)

João Paulo Fernandes  
Release/LISP, CISUC  
Universidade de Coimbra, Portugal  
[jpf@dei.uc.pt](mailto:jpf@dei.uc.pt)

João Saraiva  
HASLab/INESC TEC  
Universidade do Minho, Portugal  
[saraiva@di.uminho.pt](mailto:saraiva@di.uminho.pt)

	Energy		Mb
(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Go	1.05
(c) C++	1.34	(c) C	1.17
(c) Ada	1.70	(c) Fortran	1.24
(v) Java	1.98	(c) C++	1.34
(c) Pascal	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Rust	1.54
(v) Lisp	2.27	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Haskell	2.45
(c) Fortran	2.52	(i) PHP	2.57
(c) Swift	2.79	(c) Swift	2.71
(c) Haskell	3.10	(i) Python	2.80
(v) C#	3.14	(c) Ocaml	2.82
(c) Go	3.23	(v) C#	2.85
(i) Dart	3.83	(i) Hack	3.34
(v) F#	4.13	(v) Racket	3.52
(i) JavaScript	4.45	(i) Ruby	3.97
(v) Racket	7.91	(c) Chapel	4.00
(i) TypeScript	21.50	(v) F#	4.25
(i) Hack	24.02	(i) JavaScript	4.59
(i) PHP	29.30	(i) TypeScript	4.69
(v) Erlang	42.23	(v) Java	6.01
(i) Lua	45.98	(i) Perl	6.62
(i) JRuby	46.54	(i) Lua	6.72
(i) Ruby	69.91	(v) Erlang	7.20
(i) Python	75.88	(i) Dart	8.64
(i) Perl	79.58	(i) JRuby	19.84



@AnnieTalvasto

# Energy Efficiency across Programming Languages

## Energy Efficiency across Programming Languages

How Do Energy, Time, and Memory Relate?

Rui Pereira  
HASLab/INESC TEC  
Universidade do Minho, Portugal  
ruipereira@di.uminho.pt

Jácome Cunha  
NOVA LINCS, DI, FCT  
Univ. Nova de Lisboa, Portugal  
jacome@fct.unl.pt

Marco Couto  
HASLab/INESC TEC  
Universidade do Minho, Portugal  
marco.l.couto@inesctec.pt

João Paulo Fernandes  
Release/LISP, CISUC  
Universidade de Coimbra, Portugal  
jpf@dei.uc.pt

Francisco Ribeiro, Rui Rua  
HASLab/INESC TEC  
Universidade do Minho, Portugal  
fribeiro@di.uminho.pt  
rrua@di.uminho.pt

João Saraiva  
HASLab/INESC TEC  
Universidade do Minho, Portugal  
saraiva@di.uminho.pt

	Energy	Mb	
(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Go	1.05
(c) C++	1.34	(c) C	1.17
(c) Ada	1.70	(c) Fortran	1.24
(v) Java	1.98	(c) C++	1.34
(c) Pascal	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Rust	1.54
(v) Lisp	2.27	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Haskell	2.45
(c) Fortran	2.52	(i) PHP	2.57
(c) Swift	2.79	(c) Swift	2.71
(c) Haskell	3.10	(i) Python	2.80
(v) C#	3.14	(c) Ocaml	2.82
(c) Go	3.23	(v) C#	2.85
(i) Dart	3.83	(i) Hack	3.34
(v) F#	4.13	(v) Racket	3.52
(i) JavaScript	4.45	(i) Ruby	3.97
(v) Racket	7.91	(c) Chapel	4.00
(i) TypeScript	21.50	(v) F#	4.25
(i) Hack	24.02	(i) JavaScript	4.59
(i) PHP	29.30	(i) TypeScript	4.69
(v) Erlang	42.23	(v) Java	6.01
(i) Lua	45.98	(i) Perl	6.62
(i) JRuby	46.54	(i) Lua	6.72
(i) Ruby	69.91	(v) Erlang	7.20
(i) Python	75.88	(i) Dart	8.64
(i) Perl	79.58	(i) JRuby	19.84



@AnnieTalvasto

# Optimize your Database

- Choose your database well
- Ensure you are using the best database for interacting with your data set
  - If no easy solution - redundant copies
- Index
- Evaluate and optimize your queries
- Database cache



@AnnieTalvasto

# Understand your latency limits

- Think about what you actually need for your application
- Request/response cycle
- Worker processes vs web processes
- Worker processes in a region with lower carbon intensity.
- Delay worker process to run when the carbon intensity is the lowest



@AnnieTalvasto

# Autoscaling



@AnnieTalvasto

# What is Keda?

- Default Kubernetes Scaling is not well suited for event driven applications.
- **Keda:** Event driven scale controlling that can run inside any kubernetes cluster -> Can monitor rate of the events to pre-emptively act before CPU is affected.
- You can install it into new or existing clusters.
- Provides 30+ built-in scalers, but you can build your own
  - Support for external scaler, external push or Metrics API



@AnnieTalvasto

Demo

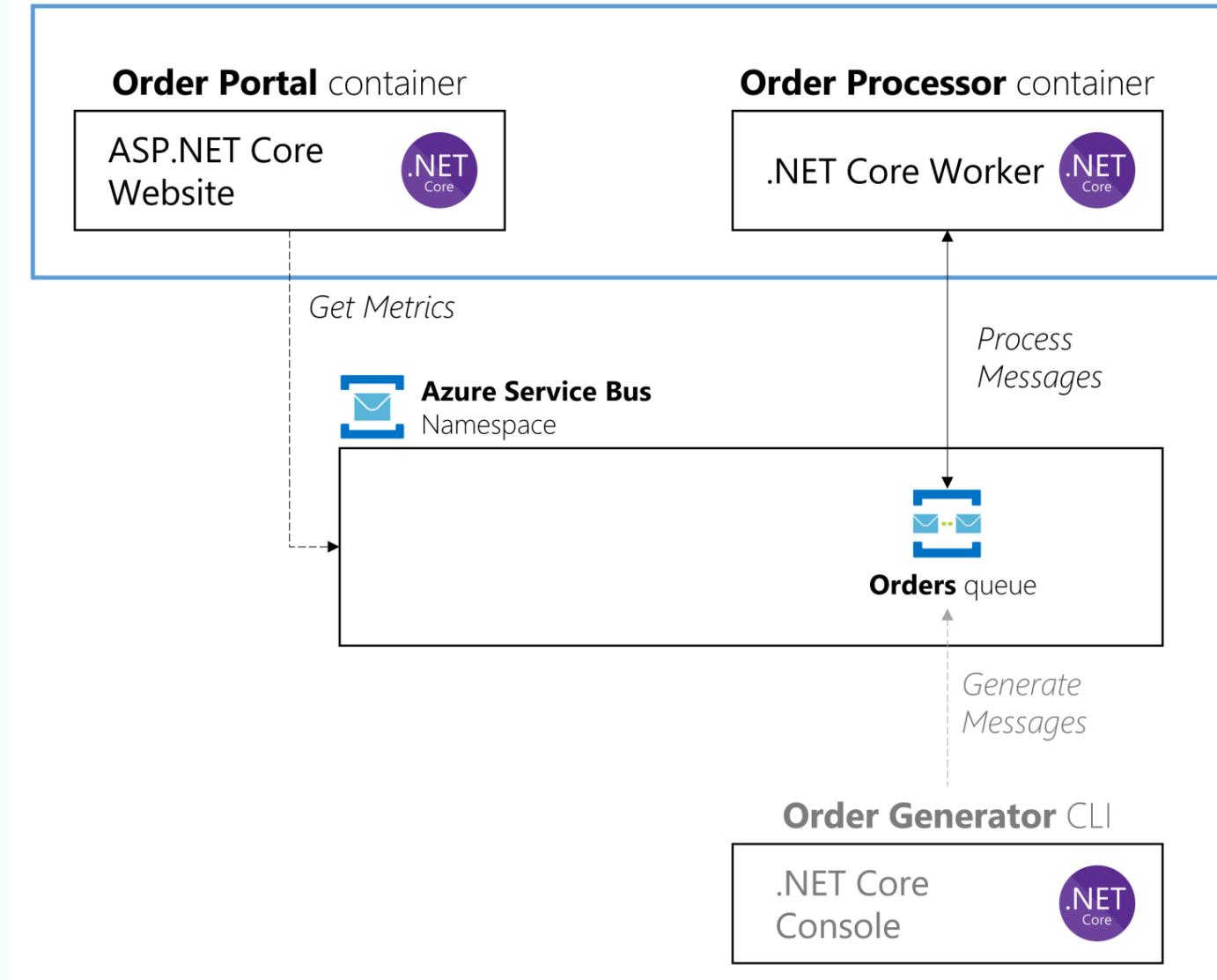
KEDA:  
Scaling .NET Core  
worker with Azure  
Service Bus



@AnnieTalvasto



# kubernetes



@AnnieTalvasto



KubeCon



CloudNativeCon

Europe 2023

# Sponsored Keynote: Building a Sustainable, Carbon-Aware Cloud: Scale Workloads and Reduce Emissions

Jorge Palma, Principal PM Lead, Microsoft Azure

0:00 / 5:44



Sponsored Keynote: Building a Sustainable, Carbon-Aware Cloud: Scale Workloads and Re... Jorge Palma



CNCF [Cloud Native Computing Foundation]

100K subscribers

Subscribe



6



Share

Clip

Save

...

244 views 7 days ago

Sponsored Keynote: Building a Sustainable, Carbon-Aware Cloud: Scale Workloads and Reduce Emissions - Jorge Palma, Principal PM Lead, Microsoft Azure

When we think about sustainability in the technology space, we know that reducing emissions is essential, even as we face greater demand to build scalable appli Show more



@AnnieTalvasto

BLOG / COMMUNITY POST

# CNCF WG Environmental Sustainability

By **Max Körbächer + Leonard Pahlke**

May 31, 2022

*Community post by **Max Körbächer**, Co-Founder of Liquid Reply, and **Leonard Pahlke**, Consultant at Liquid Reply*

We are pleased to announce that we have established a new working group for environmental sustainability. Our mission is to promote sustainability awareness and develop a culture within the CNCF landscape to establish sustainability best practices and standards.

Environmental sustainability is a pressing issue for humanity, and with simultaneous digitization, software is playing an increasingly important role. Making software ecologically sustainable is becoming more and more vital.

Data Centers, and therefore every compute resource accessible, are currently using 2% of the world's energy. This is conservatively expected to grow to hold 1-2% in the next couple of years, up to 12% by 2040. The growth in energy for computing is outpacing the global growth in energy production. The contributing factors include an explosion in data, the emergence of energy-intensive workloads such as AI and, the flattening of Moore's law. The energy consumption

# Greenkube

Product ▾ Solutions ▾ Open Source ▾ Pricing

Search Sign in Sign up

kube-green / kube-green Public

Code Issues 15 Pull requests 7 Discussions Actions Projects 1 Security Insights

main 10 branches 8 tags Go to file Code

davidebianchi Merge pull request #233 from kube-green/dependabot/github... 6cb0c3b last week 446 commits

.github chore(deps): bump azure/setup-kubectl from 3.1 to 3.2 2 weeks ago

api/v1alpha1 Merge pull request #226 from kube-green/feat/kubernetes-e2e-test 3 weeks ago

bundle upgrade tools 3 months ago

config Merge pull request #226 from kube-green/feat/kubernetes-e2e-test 3 weeks ago

controllers Merge pull request #226 from kube-green/feat/kubernetes-e2e-test 3 weeks ago

hack fix: upgrade image in bundle 3 months ago

logo update readme 3 months ago

testdata fix: change testdata last year

tests/e2e added e2e tests for cronjob suspension - part two last month

.dockerignore Initial commit 2 years ago

.gitignore Initial commit 2 years ago

CHANGELOG.md docs: CHANGELOG now links to releases 5 months ago

CONTRIBUTING.md docs: add adopters 2 months ago

Dockerfile chore(deps): bump golang from 1.19.4 to 1.19.5 2 weeks ago

LICENSE Initial commit 2 years ago

Makefile e2e tests 2 months ago

README.md

Notifications Fork 29 Star 389

About

A K8s operator to reduce CO2 footprint of your clusters

kube-green.dev

kubernetes resources k8s

cloud-native hacktoberfest downscale

green-software

Readme

MIT license

389 stars

3 watching

29 forks

Releases 5

v0.4.1 Latest on Oct 5, 2022

+ 4 releases

Packages

No packages published

Contributors



@AnnieTalvasto

Getting Started

How it works

Install

Configuration

Real use case >

Tutorials >

Advanced >

API Reference

FAQ

Adopters

## How many CO2 is produced by pod?

This calculations are based on the following assumptions:

- Emissions for cloud server using 100% green electricity: **160 Kg CO2eq / year and server** (from [goclimate.com](#) )
- Cluster node of 2 cpu. We approximate **1 node is 1 server**
- **15 pods per node**

With this assumption, the mean consumption of CO2 per pod in a year is  $160 / 15 = \mathbf{11 \text{ Kg CO2eq / year per pod.}}$

### CO2 Calculator

CO2 per pods per year (kg CO2eq)

Total number of pods

Total pods when kube-green active

Hour of sleep per week

### Results

Total (Kg CO2eq/week)

without kube-green: 21

**with kube-green: 5**

Difference: -16

[Edit this page](#)

Previous  
[« API Reference](#)

Next  
[Adopters »](#)



@AnnieTalvasto

I have a couple of nodes with low utilization, but they are not scaled down. Why?

[How many CO2 is produced by pod?](#)

# More projects

The image shows two side-by-side screenshots of GitHub project pages.

**Top Screenshot (sustainable-computing-io / kepler):** This page displays the 'Code' tab for the 'kepler' repository. It shows 12 branches and 4 tags. The main branch has 839 commits. Key contributors listed include sustainable-computing-bot, github, automation/presubmit-tests, bpfseries/perf\_event, build, cmd, data, doc, e2e, enhancements, and grafana-dashboards. The repository is described as "Kepler (Kubernetes-based Efficient Power Level Exporter) uses eBPF to probe energy related system stats and exports as Prometheus metrics".

**Bottom Screenshot (Green-Software-Foundation / carbon-aware-sdk):** This page displays the 'Code' tab for the 'carbon-aware-sdk' repository. It shows 13 branches and 1 tag. The dev branch has 725 commits. Key contributors listed include vauhngknight, .devcontainer, .github, .vscode, docs, images, samples, scripts/package, src, .gitignore, CONTRIBUTING.md, GettingStarted.md, LICENSE, README.md, action.yml, and entrypoint.sh. The repository is described as "Carbon-Aware SDK".

This screenshot shows the GitHub page for the 'carbon-aware-scaling-poc' repository under the 'sustainable-computing-io' organization.

**Repository Overview:** The repository has 7 branches and 0 tags. The main branch was last updated on Oct 20, 2022, with 16 commits. Contributors listed include rootfs, carbon-intensity-exporter, demo, deploy, docs, forecast/train\_pipeline, images, schemas, src, .gitignore, LICENSE, and README.md. The repository is described as "No description, website, or topics provided".

**README.md Content:** The README.md file contains the following text:  
Carbon Aware Scaling Poc  
  
This project is to integrate CO2 emission intensity data into KEDA scaling decision.

**Project Metrics:** The repository has 2 stars, 5 watching, and 2 forks.

**Contributors:** 5 contributors are listed with their profile icons.

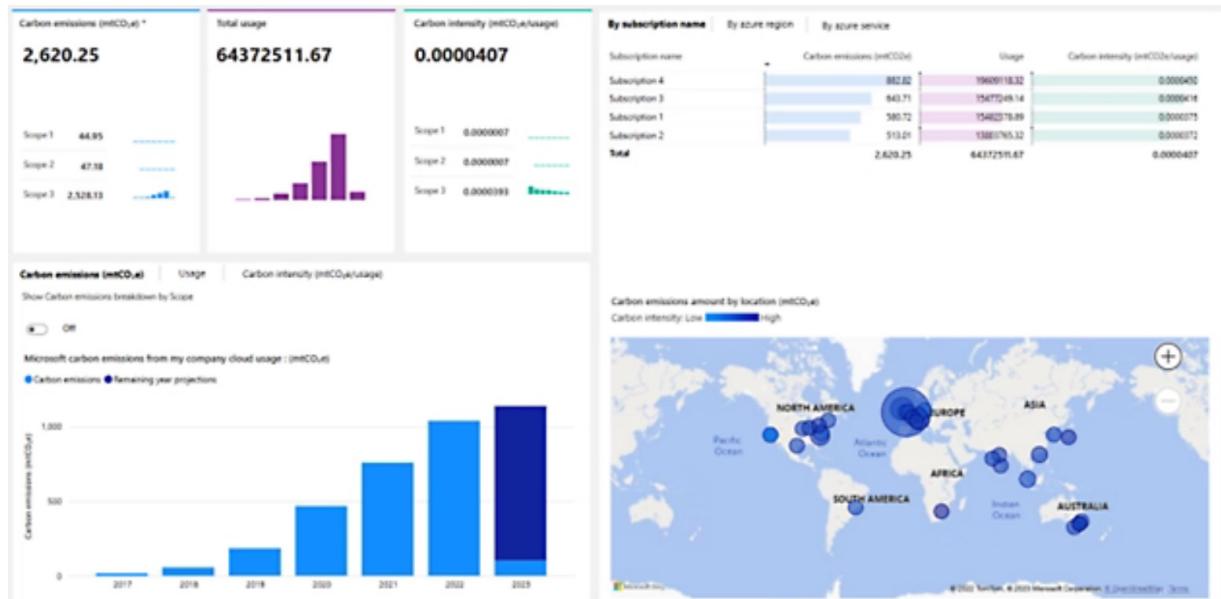
**Languages:** The project is primarily written in Python (90.9%) and Shell (7.5%).



@AnnieTalvasto

# Gain transparency into the carbon impact of your cloud usage

- Track direct and indirect greenhouse gas emissions related to your cloud usage.
- Understand your emissions avoided over time through our datacenter efficiency.
- Easily share your findings, shown in CO<sub>2</sub>-equivalent metric tons, through cloud data export.
- **New!** Access your Azure emissions data with the [Microsoft Cloud for Sustainability API \(Preview\)](#).



**Microsoft Power BI Emissions Impact Dashboard for Microsoft 365**

File Export Chat in Teams Get insights Subscribe

Overview Carbon Intensity Emissions Savings GHG Preparation Report Usage Report Calculation Methodology Learn More Legal Information Opt Out

Estimate emissions already avoided by migrating to the Microsoft cloud

Currently displaying data for the following tenant: DEMO - Please connect with a valid Microsoft 365 admin profile

Select the characteristics of your on-premises alternative

Date: All Efficiency Scale: Low Medium High Renewable energy purchases: 0%

Value	Description
12932.361	mTCO <sub>2</sub> e from on-premises alternative
99.23%	Carbon emissions saved (mTCO <sub>2</sub> e)
-11734.659	mTCO <sub>2</sub> e saved from Microsoft efficiencies
12832.462	Carbon emissions saved (mTCO <sub>2</sub> e)
-1097.803	mTCO <sub>2</sub> e saved from Microsoft renewable energy purchases
31375370.276	Carbon emissions saved in driven distance
99.899	mTCO <sub>2</sub> e from switch to Exchange Online and SharePoint Online*

How to select your on-premises characteristics

**Efficiency**

This calculation estimates emissions that result from your use of Exchange Online and SharePoint Online services, savings relative to purchase of those same services at low medium, and high efficiency on-premises deployments, and the renewable energy projects in which Microsoft invests.

**Low efficiency** (Physical servers and direct attached storage in a small internet data center (0-1,000 square feet))

**Medium efficiency** (Physical servers and virtual servers and shared, dedicated storage in a mid internet data center (1,000-20,000 square feet))

**High efficiency** (Virtualized servers and dedicated storage in a high-end internet data center (> 20,000 square feet))

The estimated emissions for your switch to the Microsoft cloud include Scope 1 and Scope 2 emissions only. The figure accounts for Microsoft's renewable energy power purchases and includes energy used to transmit data over the internet.

**Renewable energy purchases**: Specify a percentage of renewable energy purchases used on your on-premises datacenter. If your on-premises datacenter resides in multiple geographies, please specify an average of the geographic based on power consumption.

\*This figure is based on Scope 1 and 2 emissions from your organization's usage of Exchange Online and SharePoint Online. Other Microsoft 365 applications are not currently included.

## Estimate emissions that you've avoided

Calculate the emissions your organization has already avoided by moving on-premises workloads to the Microsoft Cloud.

## Carbon Footprint

Benefits

Key features

Customers

What's new

Documentation

Pricing

Partners

Take the next step

# Carbon Footprint

Measure, report, and reduce your cloud carbon emissions.

Try it for free

- ✓ Include gross carbon emissions data in reports and disclosures
- ✓ Visualize carbon insights via dashboards and charts
- ✓ Reduce the gross emissions of cloud applications and infrastructure

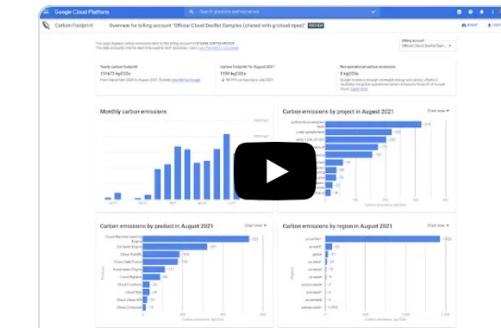
### BENEFITS

#### Accurately measure your gross carbon footprint

View the gross, location-based emissions that derive from your Google Cloud usage, providing transparency into emissions associated with your cloud applications. Your [net operational emissions are zero](#).

#### Track the emissions profile of cloud projects

Monitor your gross cloud emissions over time by project, product, and region—giving IT teams and developers metrics that can help them improve their carbon footprint.



### VIDEO

Watch this session and learn how to measure carbon emissions on Google Cloud

8.02

#### Share detailed methodology with reviewers

Our detailed calculation methodology is [published](#) so that reviewers and reporting teams can verify that their emissions data meets [GHG Protocol](#).

Reminder:

Use best practices  
from other fields  
creatively and as  
fits!

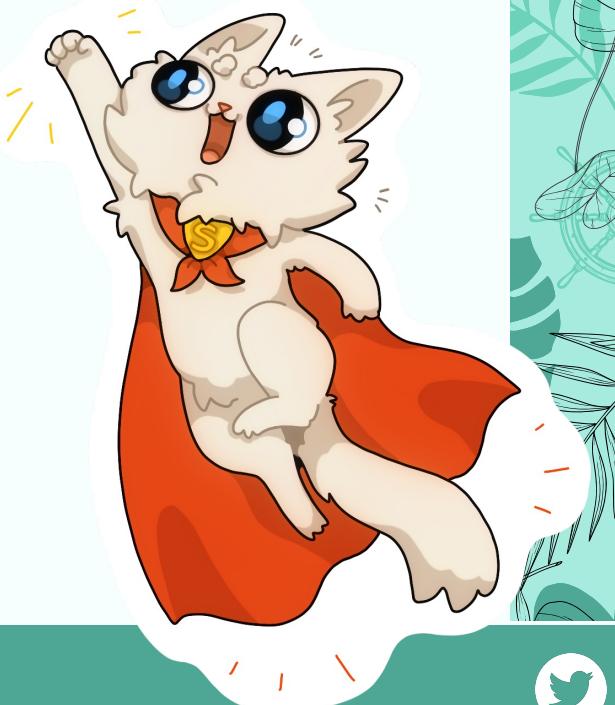


@AnnieTalvasto

# Benefits

- Organizational improvements
- Improved performance
- Cost savings
- Sustainability goals

Saving the world !



@AnnieTalvasto

# Wrap up

- Introduction
- Principles of Green Software
- Principles in Practice: Microservices
- CNCF Open Source Projects: KEDA
- Resources



@AnnieTalvasto

# Resources

## Slides & materials:

- [Github.com/annietalvasto](https://github.com/annietalvasto)



@AnnieTalvasto

# Resources

- **Green Software Foundation**

- <https://greensoftware.foundation>
- Youtube, Podcast, blog etc
- A lot of upcoming pro

- **CNCF Environmental Sustainability TAG**

- A lot of projects upcoming: How to Get Involved in CNCF Environmental Sustainability TAG - Marlow Weston & Huamin Chen -  
<https://www.youtube.com/watch?v=XFZZ9YfOyl8>
- Slack channel (CNCF Slack): <https://cloud-native.slack.com/archives/C03F270PDU6>
- High level view: <https://github.com/orgs/cncf/projects/10/views/1>
- Best practices: <https://github.com/cncf/tag-env-sustainability/issues/54>



@AnnieTalvasto

# Resources

- Lightning Talk: Green(Ing) CI/CD: A Sustainability Journey with GitOps - Niki Manoledaki, Weaveworks (Kepler, Flux)
  - [https://www.youtube.com/watch?v=rgvHbG3U\\_c4&t](https://www.youtube.com/watch?v=rgvHbG3U_c4&t)
- Using kube-green with Argo CD
  - <https://akuity.io/blog/argo-cd-kube-green/>
- Carbon Aware Kubernetes article
  - <https://greensoftware.foundation/articles/carbon-aware-kubernetes>
- Green software patterns
  - <https://patterns.greensoftware.foundation/catalog/cloud/>



@AnnieTalvasto

# Conclusion



@AnnieTalvasto

Thank you!

when you remember to bring your reusable bag to the grocery store



@AnnieTalvasto