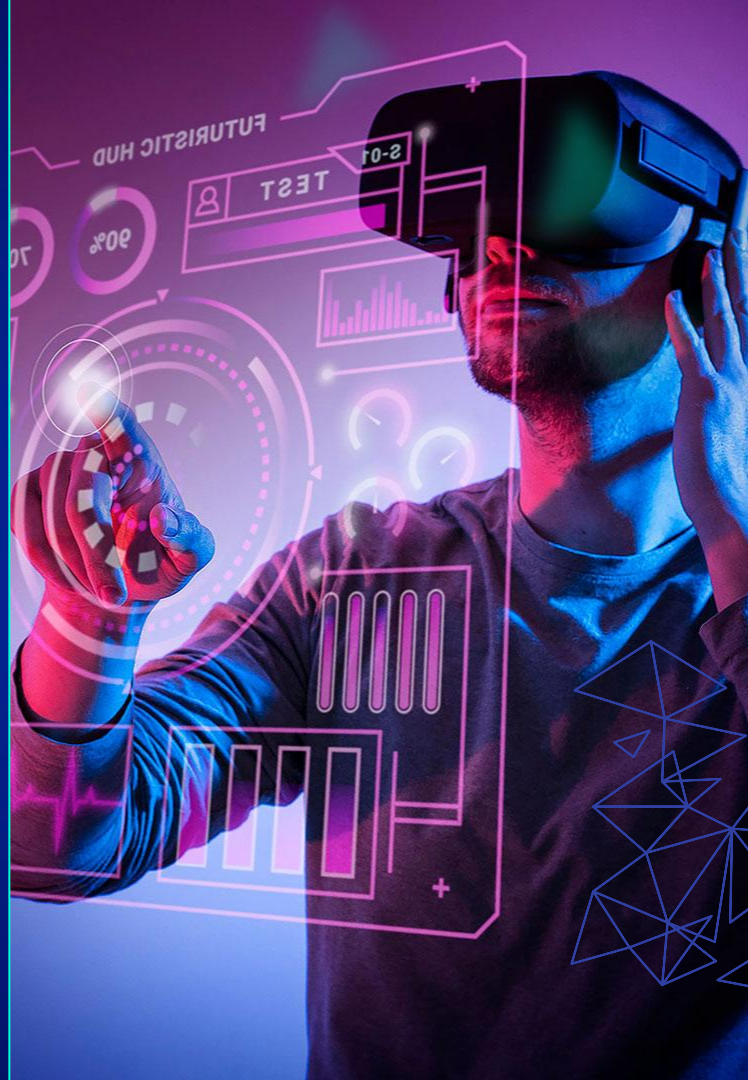
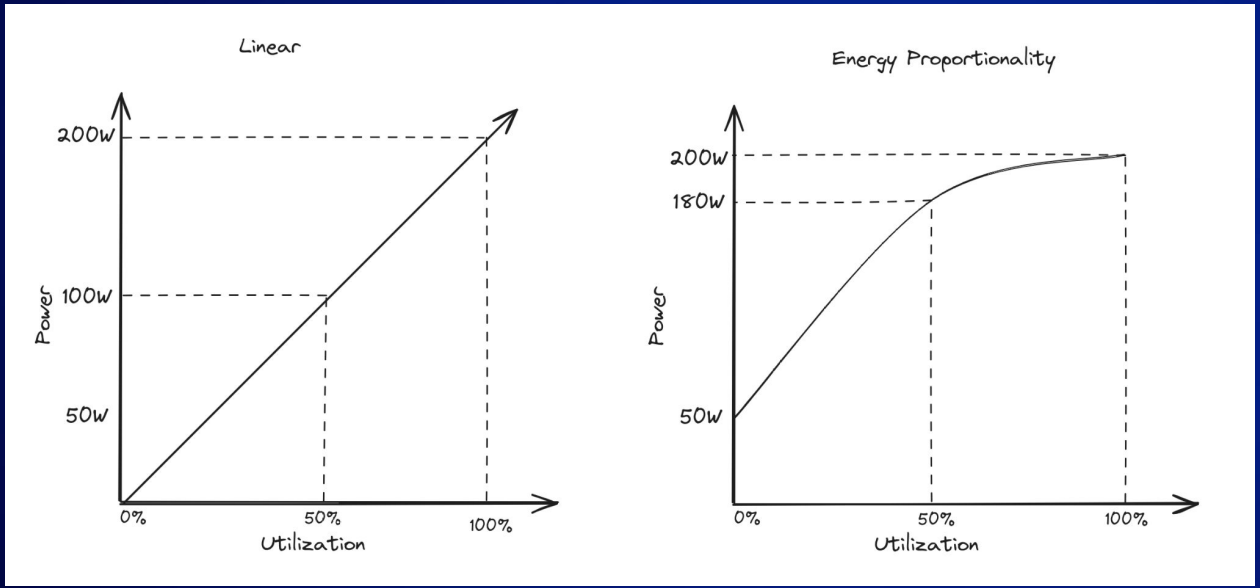
Green Cloud Computing

Max Körbächer
Founder & Cloud Native Advisor @ Liquid Reply
Co-Chair CNCF TAG Environmental Sustainability | CNCF Ambassador | LF Europe Advisory Board

Why cloud platforms are/can be the key?



Energy Proportionality



The more compute resources you consume the more energy efficient it becomes.

In other words:
if you don’t use your resources at maximum you are harmful to the environment.

REDUCE

REMOVE

REPLACE

REFACTOR

REPLATFORM

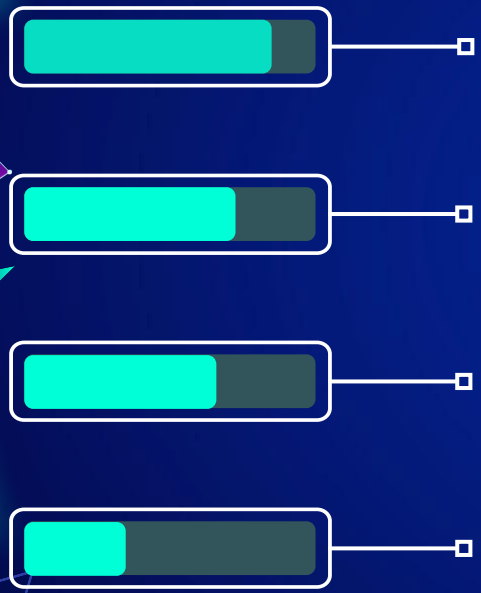
RELOCATE

Maximize the Consumption and Utilization

Cut down everything existing as much as we can

Increase the utilization of what we have to its best performance

Optimization Strategies



Scale, reduce & rightsize

Change hardware, cloud & location

Adjust systems architecture

Optimize Software & build process

LOCATION

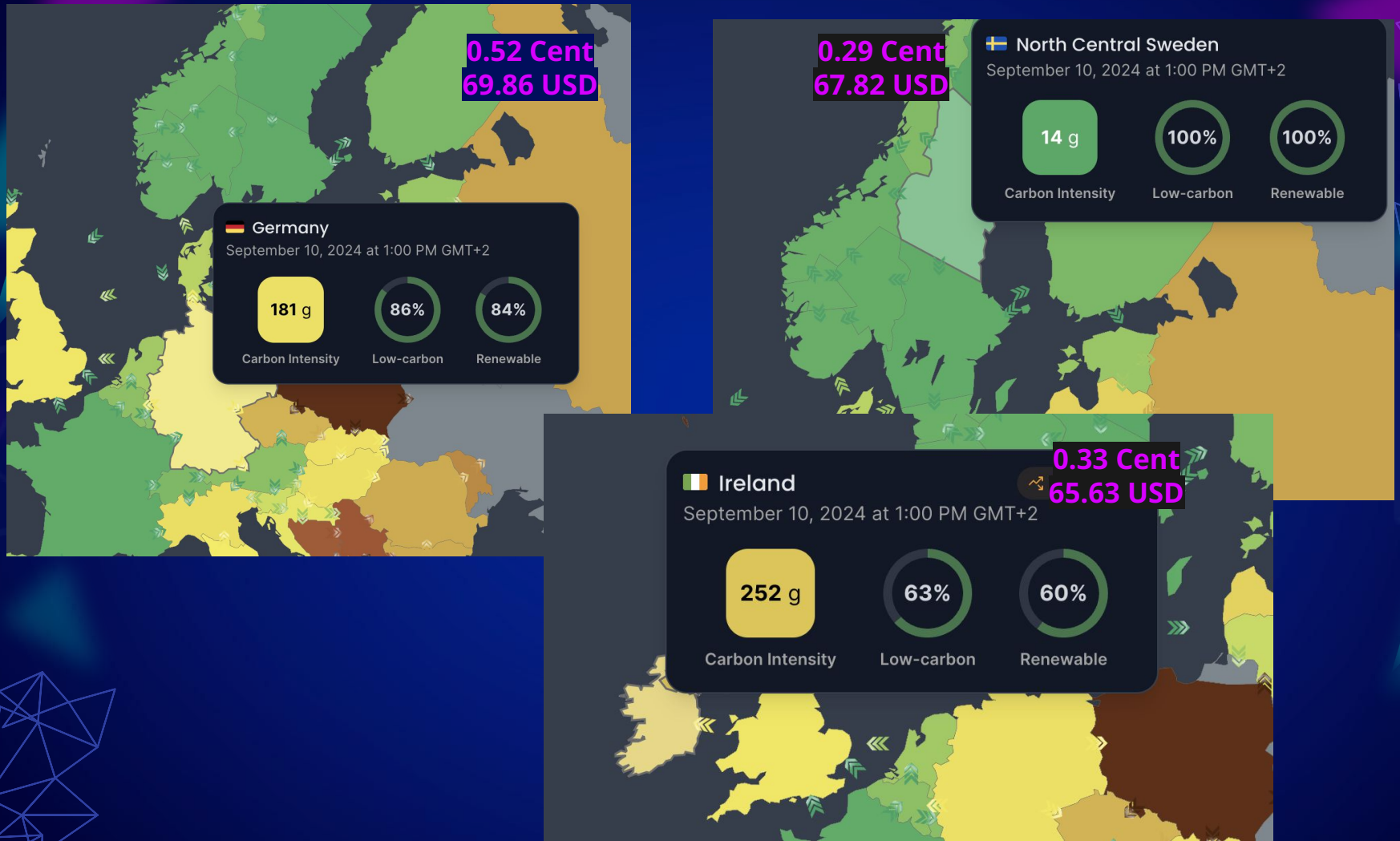
Cloud Provider Insights

Item	SCOPE \ (relative to provider)	AZURE	GCP	AWS
Life cycle stage				
Extraction	3	YES	YES	NO
Manufacture	3	YES	YES	NO
Usage	1 & 2	YES	YES	YES
End of life	3	YES	NO	NO
Item				
Building	3	NO	YES	NO
IT Equipment	2 & 3	YES	YES	YES
Overhead	2 & 3	YES	YES	?
Employee commutes	3	NO	YES	NO
Fugitive emissions from HVAC system coolants.	1	?	NO	?
Impacts of IP traffic	Depending on the type of infra	YES	NO	?
On-site combustible fossil energy source	1	?	YES	YES
Idle Resource Impacts	2 & 3	YES	YES	?
Impacts of internal services	2 & 3	YES	YES	?
Methodology of the carbon intensity of electricity				
Line loss	2	?	NO	?
Manufacture of energy infrastructure	2	?	YES	?
Location Based	2	Indirectly	YES	NO
Market Based	2	YES	YES	YES

Source: https://boavizta.org/en/blog/calculettes-carbone-clouds-providers

- **Location-based**: The location-based method calculates the emissions related to electricity consumption according to the electricity mix of the region where consumption takes place.
- **Market-based**: The market-based method calculates the emissions related to electricity consumption based on the electricity purchased by the consumer. Some players propose a dual reporting - location-based & market-based.

Always prefer **location-based** methods. When both figures are reported, those evaluated with a location-based method must be given precedence.


0.52 Cent
69.86 USD
Germany
September 10, 2024 at 1:00 PM GMT+2
181 g
Carbon Intensity
86%
Low-carbon
84%
Renewable
0.29 Cent
67.82 USD
North Central Sweden
September 10, 2024 at 1:00 PM GMT+2
14 g
Carbon Intensity
100%
Low-carbon
100%
Renewable
Ireland
0.33 Cent
65.63 USD
September 10, 2024 at 1:00 PM GMT+2
252 g
Carbon Intensity
63%
Low-carbon
60%
Renewable

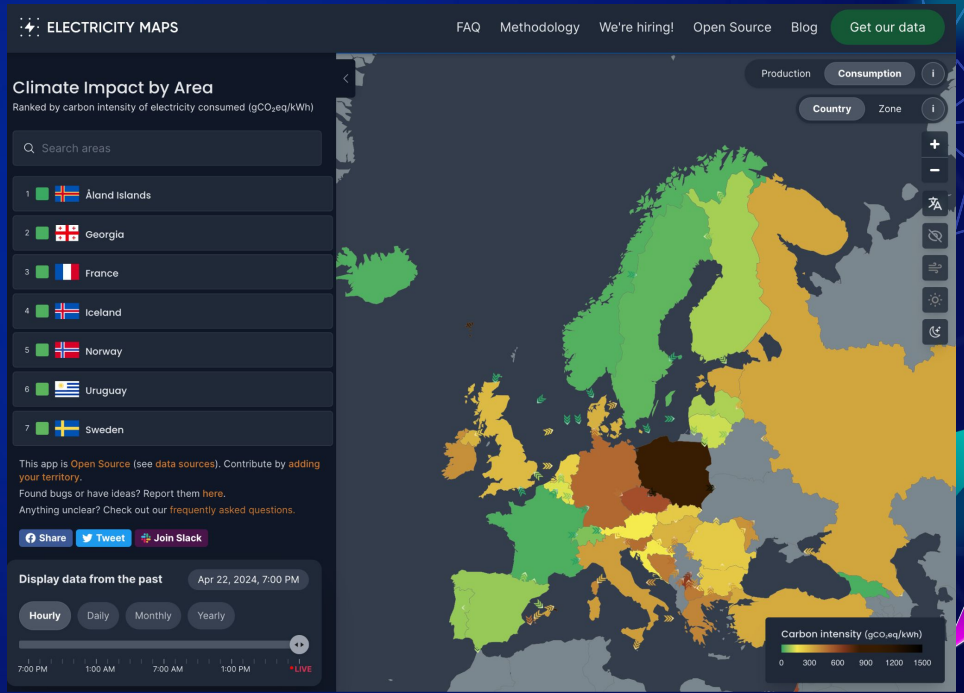
Better Locations

Let's have a look!

Deliver platforms in areas that tend to be greener:

1. Norway, Iceland
2. Sweden, Switzerland, Spain, Portugal, France
3. Netherlands, Belgium, Austria

Difference:
Germany 466g vs Sweden 36g **-92%**
Netherlands 266g vs France 31g **-88%**
Ireland 384g vs Switzerland 65g **-83%**

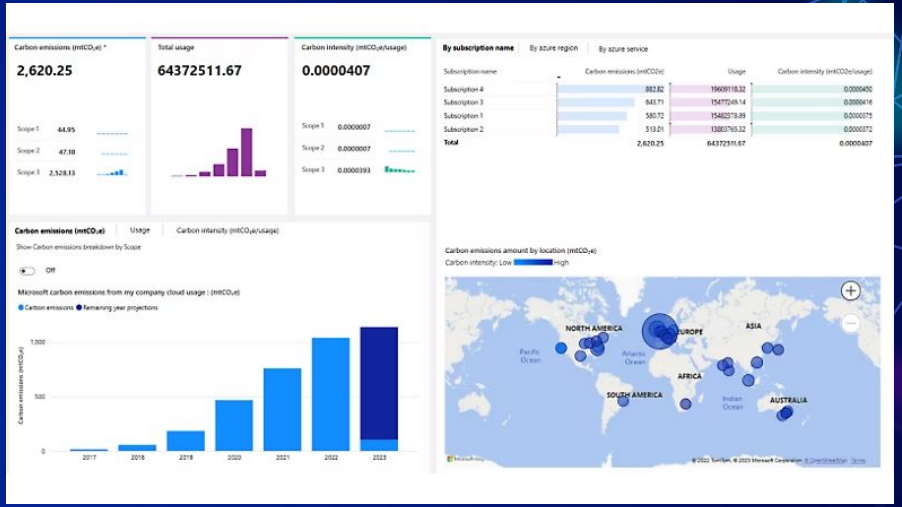


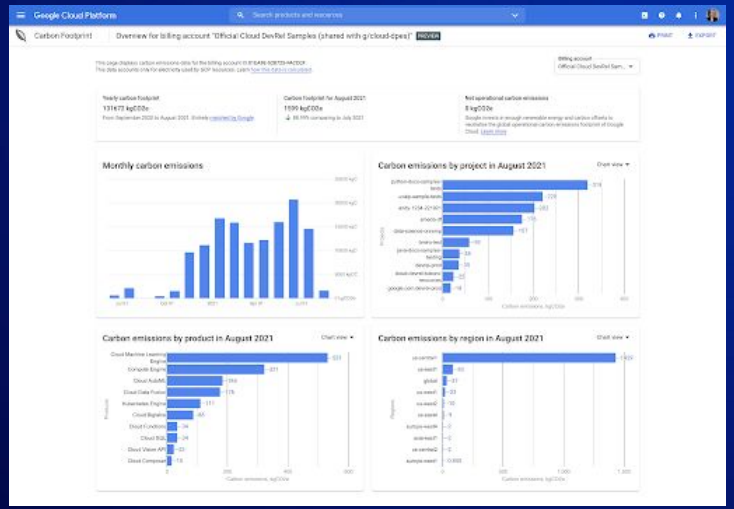
TRANSPARENCY

Cloud Provider Insights

CSP dashboards are good to know but often using data from previous years as a base and mix it with current power consumption.

Should be used to report overall CO2e output and to optimize in the long run.

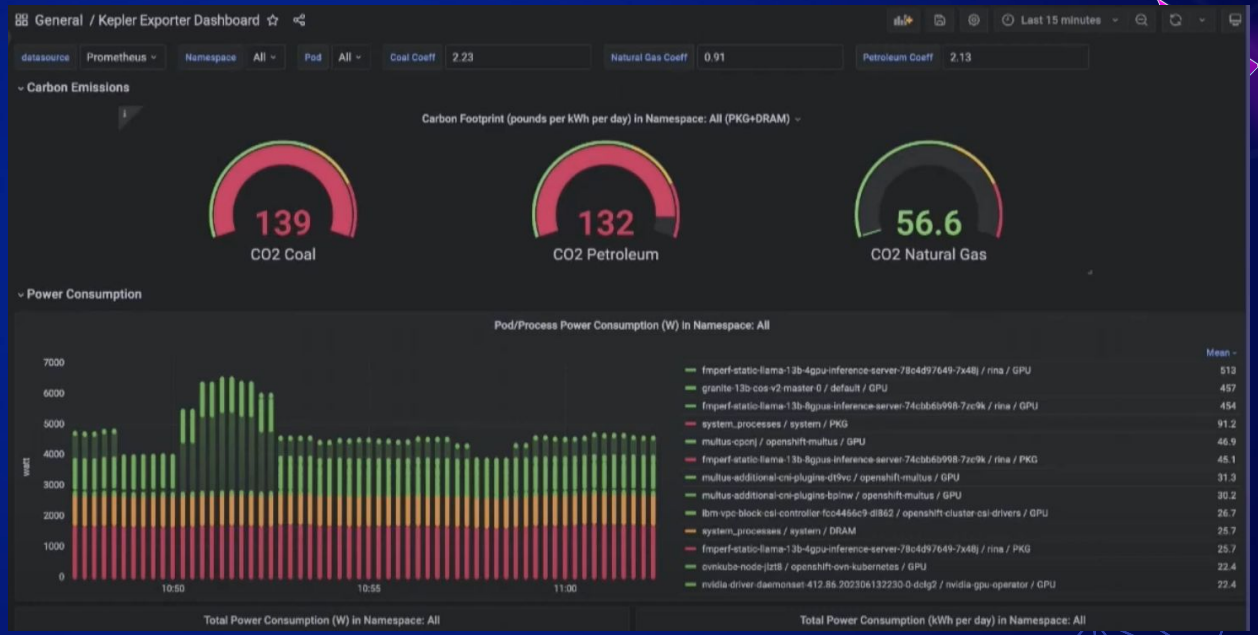




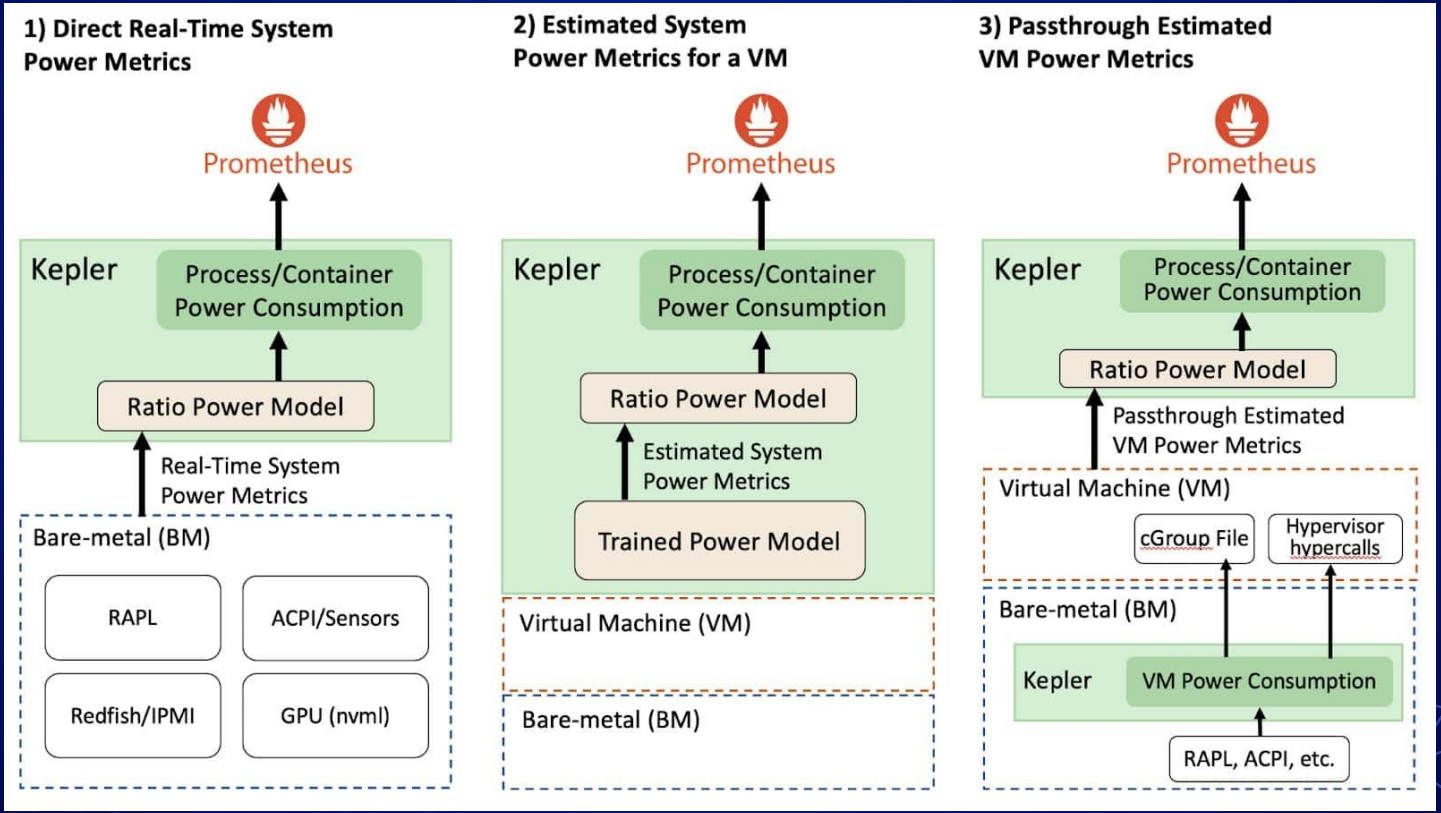
Kepler & co

Kepler (Kubernetes-based Efficient Power Level Exporter) is a Prometheus exporter. It uses eBPF to probe CPU performance counters and Linux kernel tracepoints.

These data and stats from cgroup and sysfs can then be fed into ML models to estimate energy consumption by Pods.



Kepler Deep Dive



CNCF TAG Environmental Sustainability

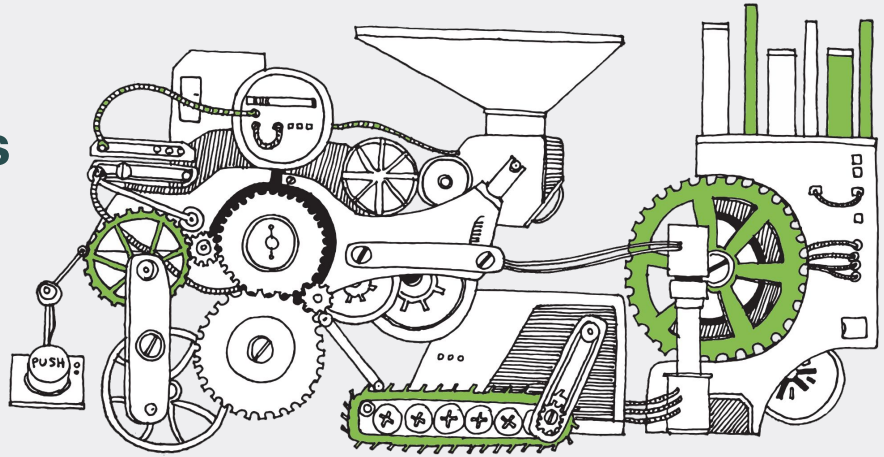
Working Group Green Reviews

Pipeline

- Equinix infrastructure
- GitOps (CAPI, Ansible)
- GitHub Actions with tests
- k6 benchmark tests

Collaboration with CNCF Projects

- Falco



Metrics

- Standard metrics
- Energy metrics (Kepler)
- GSF’s Software Carbon Intensity specification

ACTION

A Million Ways

HPA

Cluster Auto Scaling - Descaling

ARM vs Intel

Limits & Requests

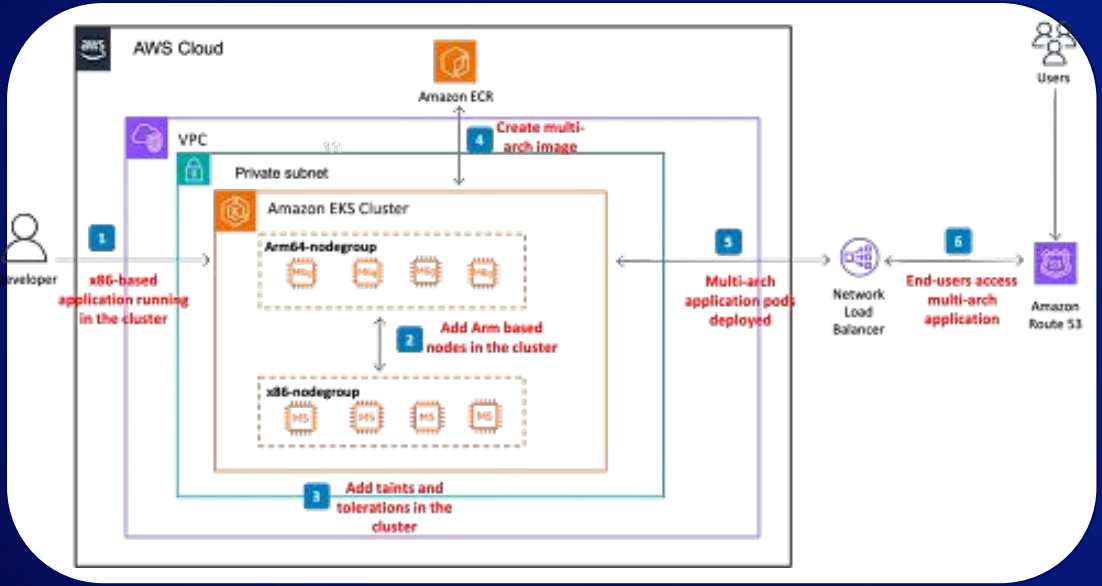
VPA

Custom Scheduler

Reduce Container Size

Efficient Infrastructure

Multi Architecture Infrastructure




```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: multi-arch-cluster
  region: us-east-1

nodeGroups:
  - name: x86-node-group
    instanceType: m5.large
    desiredCapacity: 2
    volumeSize: 80
  - name: arm64-node-group
    instanceType: m6g.large
    desiredCapacity: 2
    volumeSize: 80
```

Efficient Infrastructure

Multi Architecture Deployment

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: arm-deployment
  labels:
    app: hello
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello
      tier: web
  template:
    metadata:
      labels:
        app: hello
        tier: web
    spec:
      containers:
      - name: hello
        image: <your-docker-repo-path>/multi-arch-demo:arm64
        imagePullPolicy: Always
        ports:
          - containerPort: 8080
        env:
          - name: NODE_NAME
            valueFrom:
              fieldRef:
                fieldPath: spec.nodeName
          - name: POD_NAME
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
        resources:
          requests:
            cpu: 300m
      nodeSelector:
        kubernetes.io/arch: arm64
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: multi-arch-deployment
  labels:
    app: hello
spec:
  replicas: 6
  selector:
    matchLabels:
      app: hello
      tier: web
  template:
    metadata:
      labels:
        app: hello
        tier: web
    spec:
      containers:
      - name: hello
        image: <your-docker-repo-path>/multi-arch:latest
        imagePullPolicy: Always
        ports:
          - containerPort: 8080
        env:
          - name: NODE_NAME
            valueFrom:
              fieldRef:
                fieldPath: spec.nodeName
          - name: POD_NAME
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
        resources:
          requests:
            cpu: 300m
```

Kube Green + Karpenter

Karpenter

- Search for pods that the Kubernetes scheduler has marked as unschedulable
- Evaluation of the resources requested by the pods (resource requirements, node selections, affinities, tolerances and topology constraints)
- Deploying nodes that meet the requirements of the pods
- Removing the nodes when they are no longer needed

```
cat <<EOF | kubectl apply -f -
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  requirements:
    - key: karpenter.sh/capacity-type
      operator: In
      values: ["spot"]
  limits:
    resources:
      cpu: 1000
  providerRef:
    name: default
  ttlSecondsAfterEmpty: 30
---
apiVersion: karpenter.k8s.aws/v1alpha1
kind: AWSNodeTemplate
metadata:
  name: default
spec:
  subnetSelector:
    karpenter.sh/discovery: ${CLUSTER_NAME}
  securityGroupSelector:
    karpenter.sh/discovery: ${CLUSTER_NAME}
EOF
```

Kube Green + Karpenter

Kube Green

- Schedules Pods and kills them
- Suspend CronJobs

```yaml
apiVersion: kube-green.com/v1alpha1
kind: SleepInfo
metadata:
  name: example
spec:
  weekdays: "1-5"
  sleepAt: "20:00"
  wakeUpAt: "08:00"
  timeZone: "Europe/Berlin"
  suspendDeployments: true
  suspendCronJobs: false
  excludeRef:
    - apiVersion: "apps/v1"
      kind:       Deployment
      name:       api-gateway
```

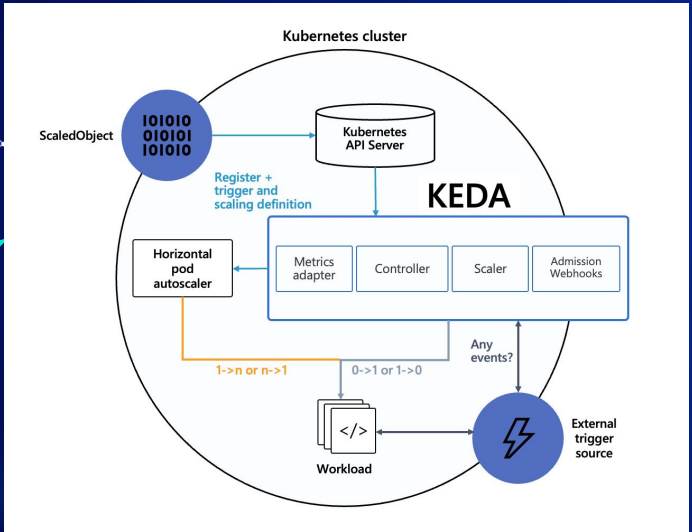
Kube Green with Karpenter

- Use kube-green to shut down Off Office Hours pods.
- Karpenter then dynamically optimizes the remaining resources and removes nodes.

- Real Use Case: Karpenter shuts down 7 of 9 Nodes (c6id.2xlarge) for additional 98h saving per week.

Saves 0.5 tons of CO2e

Event Driven Auto Scaler



With Keda your users are enabled to scale up and down their workload based on events.

This is especially useful when you are in a market where the demand also changes over time.

The less resources you use and the more often you increase density per node the better is your utilization.

An alternative is the combination of kube-green & karpenter. KG shuts down pods time based, while karpenter removes empty nodes.

A Keda + karpenter combination is possible too.

Event Driven Auto Scaler on CO2e steroids


```yaml
apiVersion: carbonaware.kubernetes.azure.com/v1alpha1
kind: CarbonAwareKedaScaler
metadata:
  name: carbon-aware-word-processor-scaler
spec:
  kedaTarget: scaledobjects.keda.sh        # can be used for ScaledObjects & ScaledJobs
  kedaTargetRef:
    name: word-processor-scaler
    namespace: default
  carbonIntensityForecastDataSource:       # carbon intensity forecast data source
    mockCarbonForecast: false              # [OPTIONAL] use mock carbon forecast data
    localConfigMap:                        # [OPTIONAL] use configmap for carbon forecast data
      name: carbon-intensity
      namespace: kube-system
      key: data
  maxReplicasByCarbonIntensity:            # array of carbon intensity values in ascending ord
    - carbonIntensityThreshold: 437        # when carbon intensity is 437 or below
      maxReplicas: 110                     # do more
    - carbonIntensityThreshold: 504        # when carbon intensity is >437 and <=504
      maxReplicas: 60
    - carbonIntensityThreshold: 571        # when carbon intensity is >504 and <=571 (and beyo
      maxReplicas: 10                      # do less
  ecoModeOff:                              # [OPTIONAL] settings to override carbon awareness;
    maxReplicas: 100                       # when carbon awareness is disabled, use this value
    carbonIntensityDuration:               # [OPTIONAL] disable carbon awareness when carbon i
      carbonIntensityThreshold: 555        # when carbon intensity is equal to or above this v
      overrideEcoAfterDurationInMins: 45   # if carbon intensity is high for this many hours d
    customSchedule:                        # [OPTIONAL] disable carbon awareness during specif
      - startTime: "2023-04-28T16:45:00Z"  # start time in UTC
        endTime: "2023-04-28T17:00:59Z"    # end time in UTC
    recurringSchedule:                     # [OPTIONAL] disable carbon awareness during specif
      - "* 23 * * 1-5"                     # disable every weekday from 11pm to 12am UTC
```

Container Caching

- if you omit the imagePullPolicy field, and you specify the digest for the container image, the imagePullPolicy is automatically set to IfNotPresent.
- if you omit the imagePullPolicy field, and the tag for the container image is :latest, imagePullPolicy is automatically set to Always;
- if you omit the imagePullPolicy field, and you don't specify the tag for the container image, imagePullPolicy is automatically set to Always;
- if you omit the imagePullPolicy field, and you specify the tag for the container image that isn't :latest, the imagePullPolicy is automatically set to IfNotPresent.

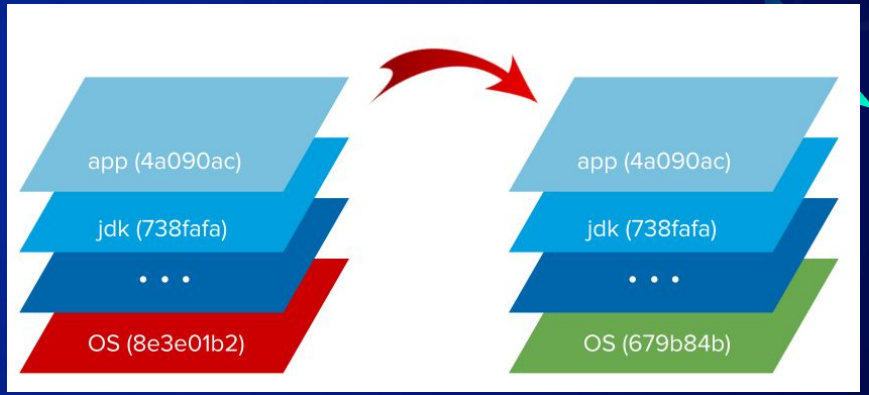

```
apiVersion: v1
kind: Pod
metadata:
  name: if-not-present-policy
spec:
  containers:
  - name: demo-container
    image: demo/image
    imagePullPolicy: IfNotPresent
```

Container Build

Consider Buildpacks

Instead of using plain container build files, consider build packs.

If the base image changes but none of the layers above it, then buildpacks are ideal.



Other options?

Serverless

Suitable but only if your underlying infrastructure is fast too.

Difficulties in optimizing the utilization except you include it into steady workload to fill up the last 10% of capacity.

WASM

Reduces the footprint in size and speed of startup.

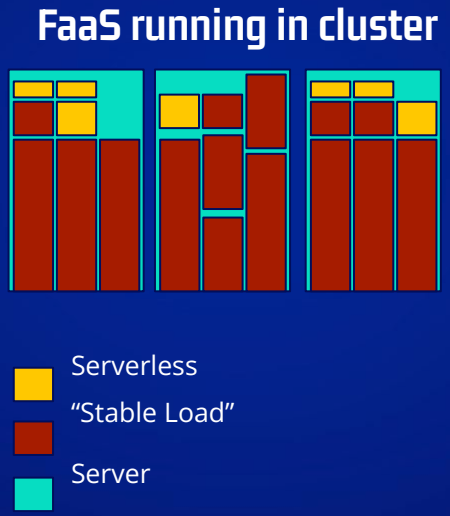
Doesn't have an effect on compute efficiency.

Self Service

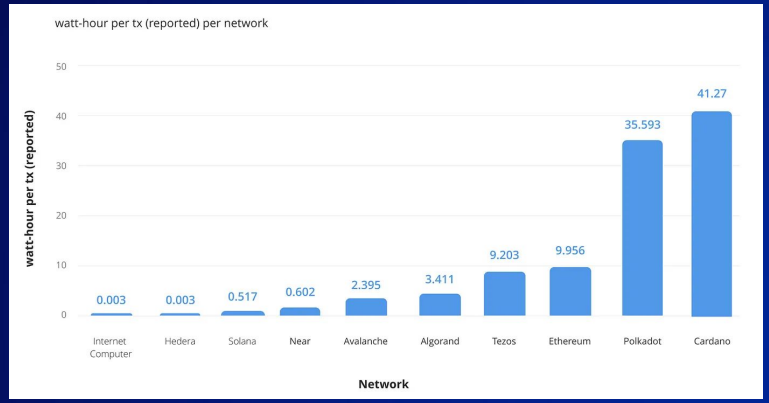
Provide a service catalog of optimized deployments with guides that have sustainability in mind.

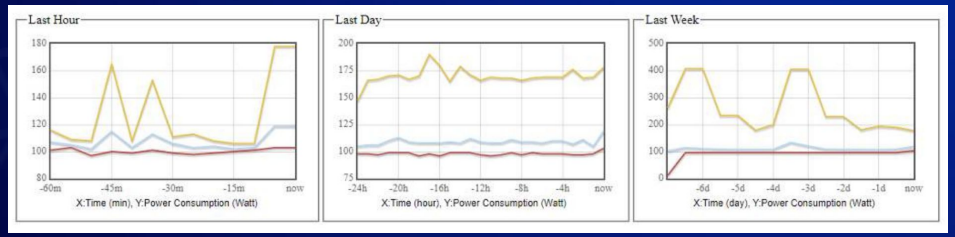
Users tend to use the default, so make the default great!

Strategic Approach to “Fill-up-the-Gaps”



Cloud is so 2000, let's do Web3!





Yellow: max., Red: min., Blue: Average.

- ICP - 0.003 Wh/tx
- Hedera - 0.003 Wh/tx
- Visa - between 1.5-3Wh/tx
- Mastercard - around 0.7 Wh/tx
- Ethereum - between 0.8 - 14.7 Wh

The Power of Platform Engineering

Transparency

Opportunities

Show the effectiveness of the actions taken.

Making changes visible.

Enable the self-driven activities to find the right options.

Correlate a change with an effect.

Education = Awareness

Create Awareness

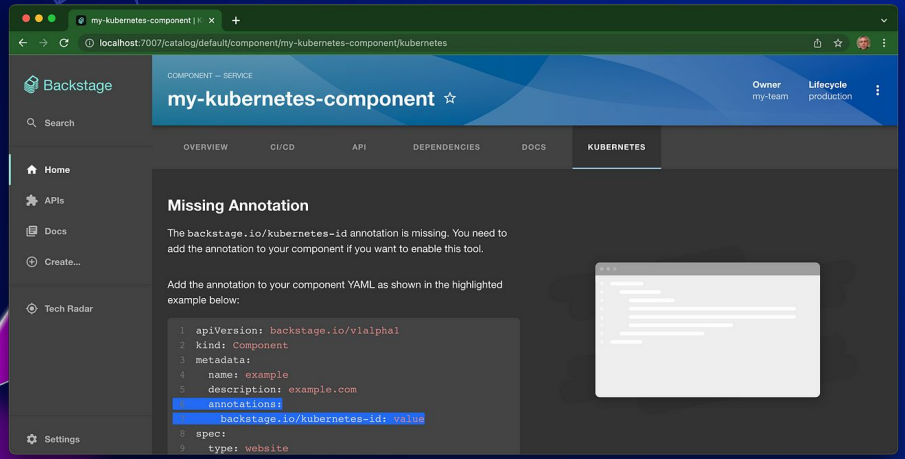
It’s on us to share and build an understanding of why we shouldn’t treat cloud resources as infinite.

Educate Actively

Provide demos and best practices that are already with a sustainable architecture and configuration in mind.

As Platform Engineers we have the opportunity to share this knowledge and enforce it into the community.

Use Backstage as Multiplicator



Backstage is a perfect multiplication gainer through its integrated documentation and references.

Provided templates might be used more often then self developed stand alone items.

What about AI?

IT’s ON YOU

Create Opportunities

Efficient Infrastructure

Provide options in hosting, e.g. with Kubernetes to provide different node groups.
ARM tend to be better in performance, price and energy consumption.

Dynamic App Management

Apps should scale by default, but often require the right surrounding.

Serverless platforms, event driven autoscaling, de-scheduling and reduction are needed implementations.

“Better” Locations

Enable other regions and countries, document their CO2e efficiency and guide users to those options.

Steps to Take

Reducing Data

Est. >90% stored data is untouched

Increase Utilization

Only at max. Used servers are good servers (old or new)

Optimize Processes

(I'm not sorry, but) you are not Google. Spotify or co

Define Sustainable Architecture

More lightweight, flexible, polyglot, robust and humanity friendly

Have a strategy & no blind optimization

Don't swap your compute randomly around

Provide Platforms to ease things

You have to implement the possibilities to make one use them

Thank You For Your Time

Max Körbächer

Founder & Cloud Native Advisor
CNCF Ambassador, LF Europe Advisory Board

REPLY
LIQUID

← **Let's connect, LinkedIn**

Join my Newsletter, Zeitgeist of Bytes →