# OS-Climate Data Commons Overview

OS-C

# OS-Climate & Global Data Commons

OS-Climate applies a community-based open-source approach to solve data & analytics challenges required for investment to achieve Paris Climate Accord goals

## OPEN SOURCE COMMUNITY

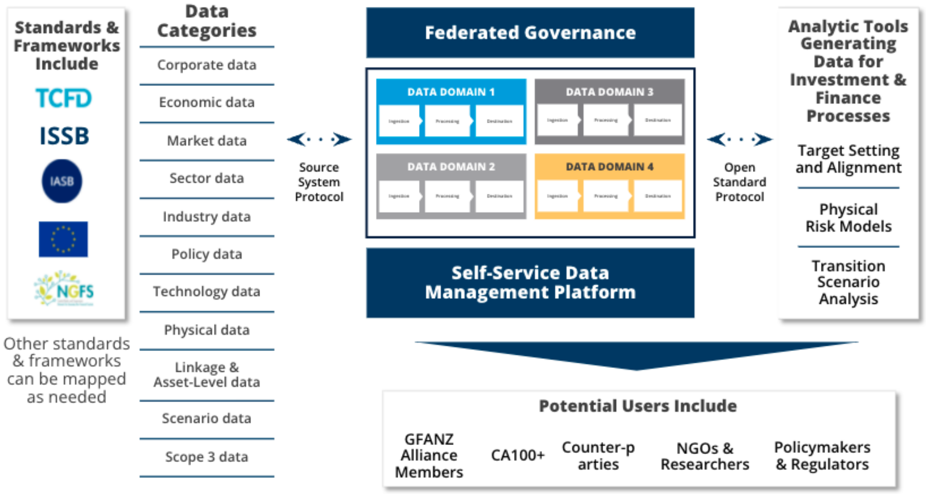
- Governance, licensing, and collaboration structures enabling stakeholders to share cost, intellectual property, and effort.
- Joint projects for new data, modelling, standards, and supporting technology

## GLOBAL DATA COMMONS

- Curated access to library of public and private sources, for both transition and physical risk/opportunity
- More accurate corporate historical and forward-looking climate & ESG metrics as a public good

## ANALYTICS TOOLS

- Integrate climate-related risk and opportunity into decisions by investors, financial institutions, regulators, etc
- Scenario analysis and alignment tools for climate change risk, physical risk and transition risk



**Standards & Frameworks Include**

TCFD

ISSB

IASB

NGFS

Other standards & frameworks can be mapped as needed

**Data Categories**

- Corporate data
- Economic data
- Market data
- Sector data
- Industry data
- Policy data
- Technology data
- Physical data
- Linkage & Asset-Level data
- Scenario data
- Scope 3 data

Source System Protocol

**Federated Governance**

DATA DOMAIN 1: Ingestion, Processing, Destination

DATA DOMAIN 3: Ingestion, Processing, Destination

DATA DOMAIN 2: Ingestion, Processing, Destination

DATA DOMAIN 4: Ingestion, Processing, Destination

**Self-Service Data Management Platform**

Open Standard Protocol

**Analytic Tools Generating Data for Investment & Finance Processes**

- Target Setting and Alignment
- Physical Risk Models
- Transition Scenario Analysis

**Potential Users Include**

GFANZ Alliance Members | CA100+ | Counter-parties | NGOs & Researchers | Policymakers & Regulators

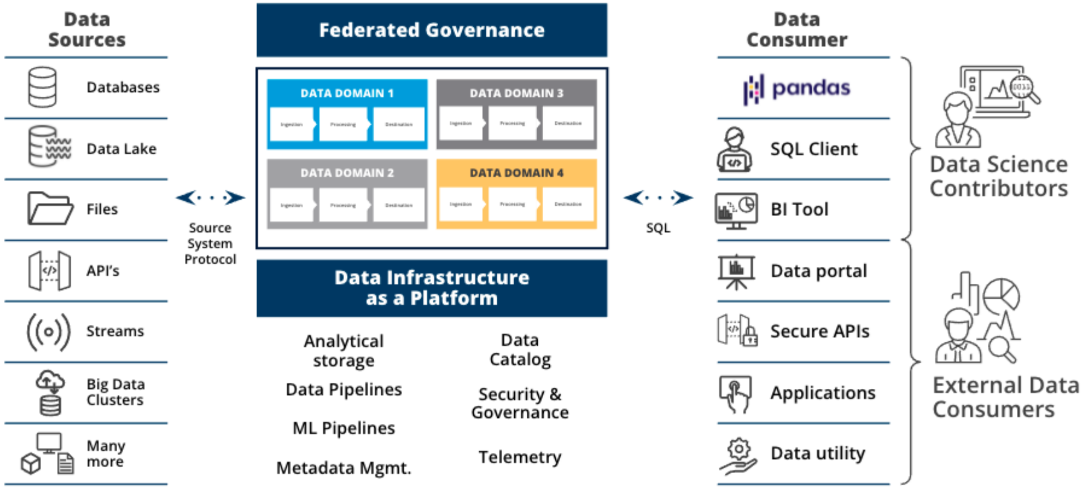
# A Data Mesh provides faster and broader access to climate data

| | Data Warehouse | Data Lake and Lakehouse | Data mesh |
|---|---|---|---|
| **Single point of access to all your data** | ✓ ETL required | ✓ ELT required | ✓ Query federation |
| **Cost-effective scaling and elasticity** | ✗ Cost grows with data retained | ✓ Separation of storage and compute | ✓ Separation of storage and compute |
| **Easily access new and existing data** | ✗ ETL required | ✗ ELT required | ✓ No data movement required |
| **Global security and compliance** | ✗ Limited, by region | ✗ Limited, by region | ✓ Global / hybrid / multi-cloud data access |
| **ANSI SQL interface** | ✗ Specialized skills required (Spark, python, etc) | ✗ Specialized skills required (Spark, python, etc) | ✓ SQL-based interface |

OS-C

# OS-Climate Data Commons Architecture Overview

Moving away from centralized data monoliths by adopting a distributed data mesh approach



**Self-service data infrastructure**

Standardized self-service infrastructure and tooling for creating, maintaining and managing data products

**Decentralized data product ownership**

Domain data product owners are responsible for all capabilities within a given domain, including discoverability, understandability, quality and security of the data

**Federated governance**

Common operating standards around data / metadata / data lineage management, quality assurance, security and compliance policies

# Data Commons: Based on Open Data Hub and Operate First

Upstream code enhanced with operational excellence

### Open Data Hub

Community driven upstream meta-project demonstrating AI/ML platform on Red Hat OpenShift comprised of open source projects

### Operate First (https://www.operate-first.cloud/)

Incorporate operational experience into Open Data Hub - operating software and services in the Open for our community members

### OS-Climate Data Commons

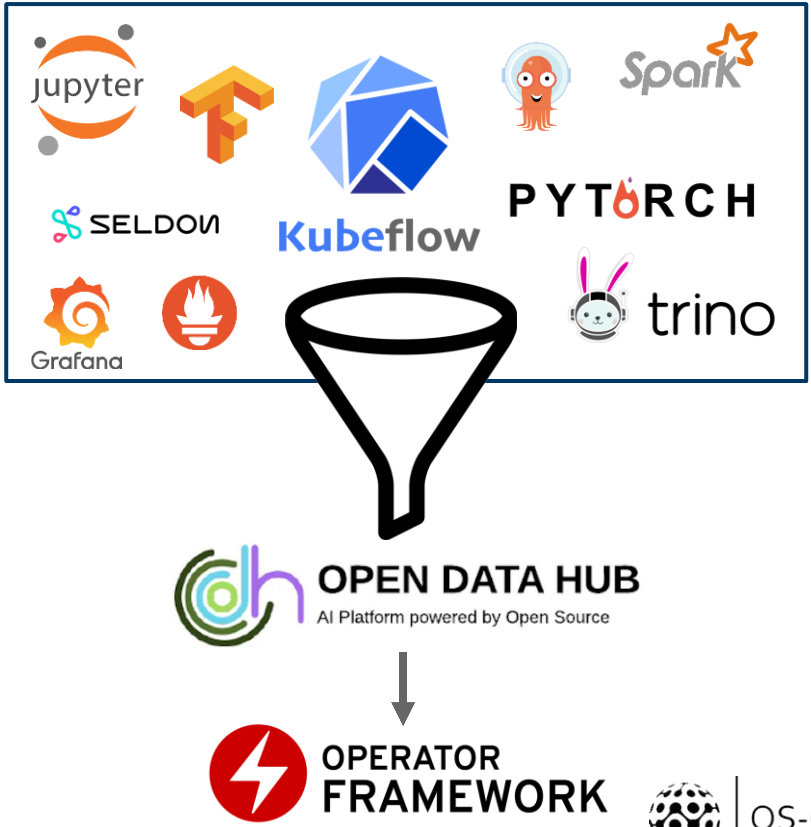
Data science platform based on Open Data Hub and delivered as a cloud service on Red Hat OpenShift  on any public or private cloud provider

OS-C

# Open Data Hub: an open source ML architecture blueprint

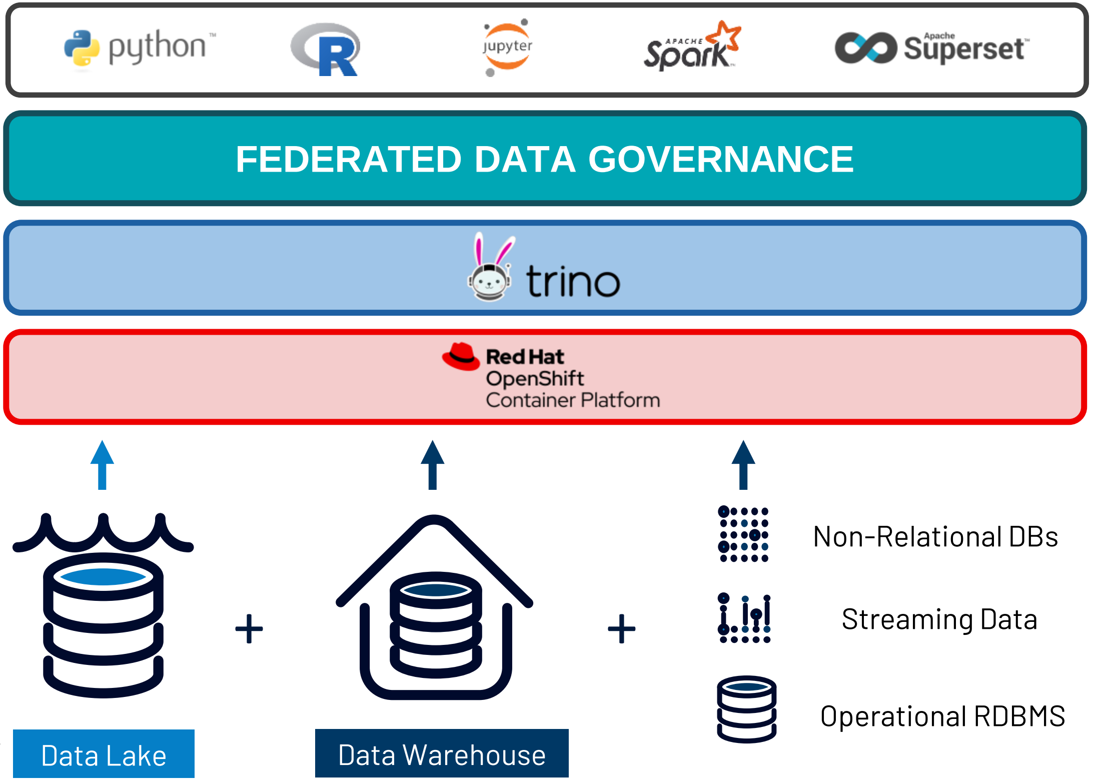
## GOALS

- Provide an end-to-end AI/ML platform leveraging Open Source components
- One stop easy operator deployment on Enterprise Kubernetes
- Provide Tools for each stage of the data science process and for all AI/ML user personas including:
  - Development tools for Data Scientists
  - ELT tools used by Data Engineers
  - Monitoring tools for model and services used by DevOps
- Act as a “glue” for a rich collection of open source data science projects, which also have enterprise offering



# OS-Climate Data Commons Technical Approach

## Building a blueprint for a distributed data platform on top of  Open Data Hub



FEDERATED DATA GOVERNANCE

trino

Red Hat OpenShift Container Platform

Non-Relational DBs

Streaming Data

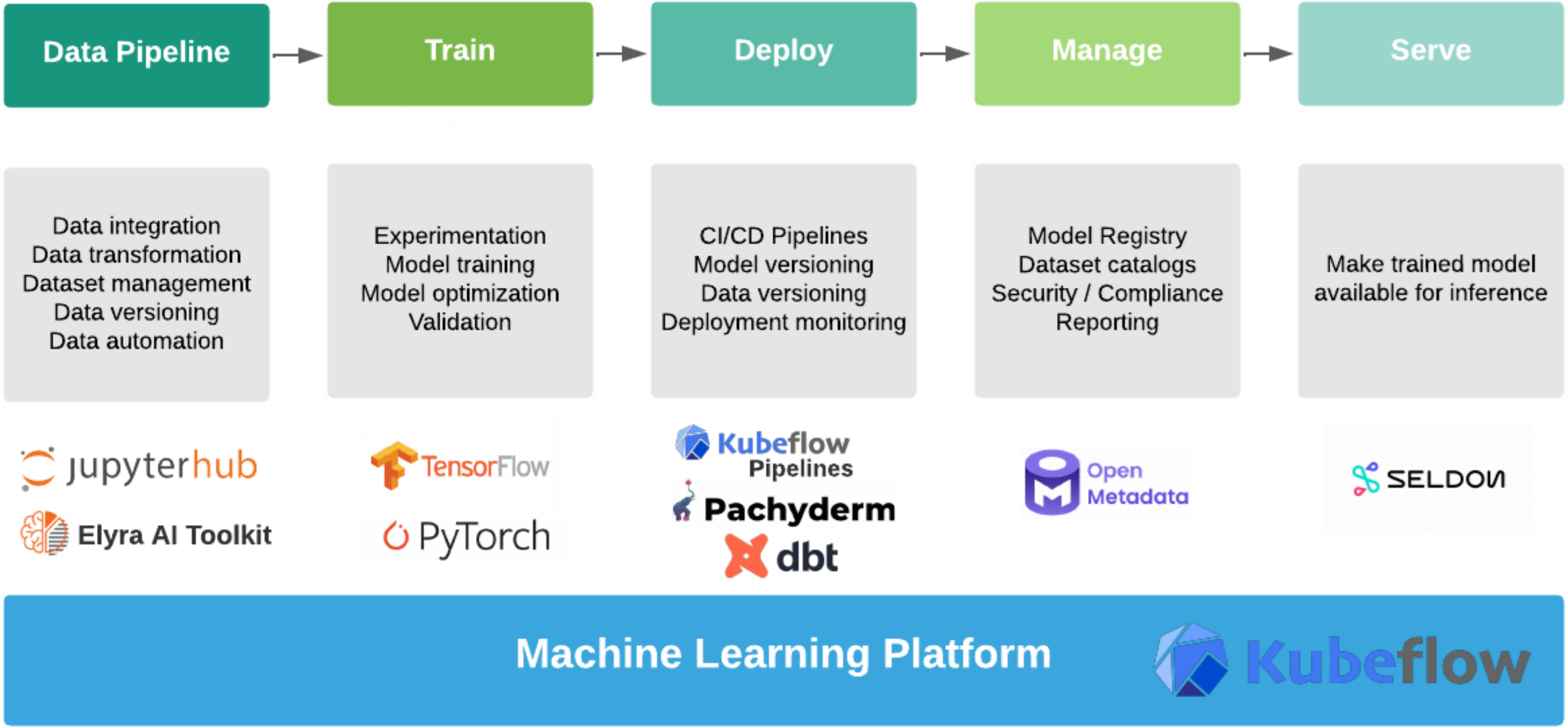
Operational RDBMS

Data Lake + Data Warehouse +

**Major capabilities being added to support the Federated Data Governance Layer:**

- Open Table Format for big data analytics, handling partitioning and time-travel / rollback (Apache Iceberg)
- Automated data versioning and data lineage (Pachyderm)
- Metadata management, data discovery, data quality, observability (OpenMetadata)
- Fine-grained Role Based Access Control (RBAC) with row-level and column-level permissions (Trino)
- Data Protection (IBM Fybrik + OPA)

# Architecture: Data Pipelines



**Data Pipeline** → **Train** → **Deploy** → **Manage** → **Serve**

Data Pipeline:
Data integration
Data transformation
Dataset management
Data versioning
Data automation

Train:
Experimentation
Model training
Model optimization
Validation

Deploy:
CI/CD Pipelines
Model versioning
Data versioning
Deployment monitoring

Manage:
Model Registry
Dataset catalogs
Security / Compliance
Reporting

Serve:
Make trained model available for inference

jupyterhub
Elyra AI Toolkit

TensorFlow
PyTorch

Kubeflow Pipelines
Pachyderm
dbt

Open Metadata

SELDON

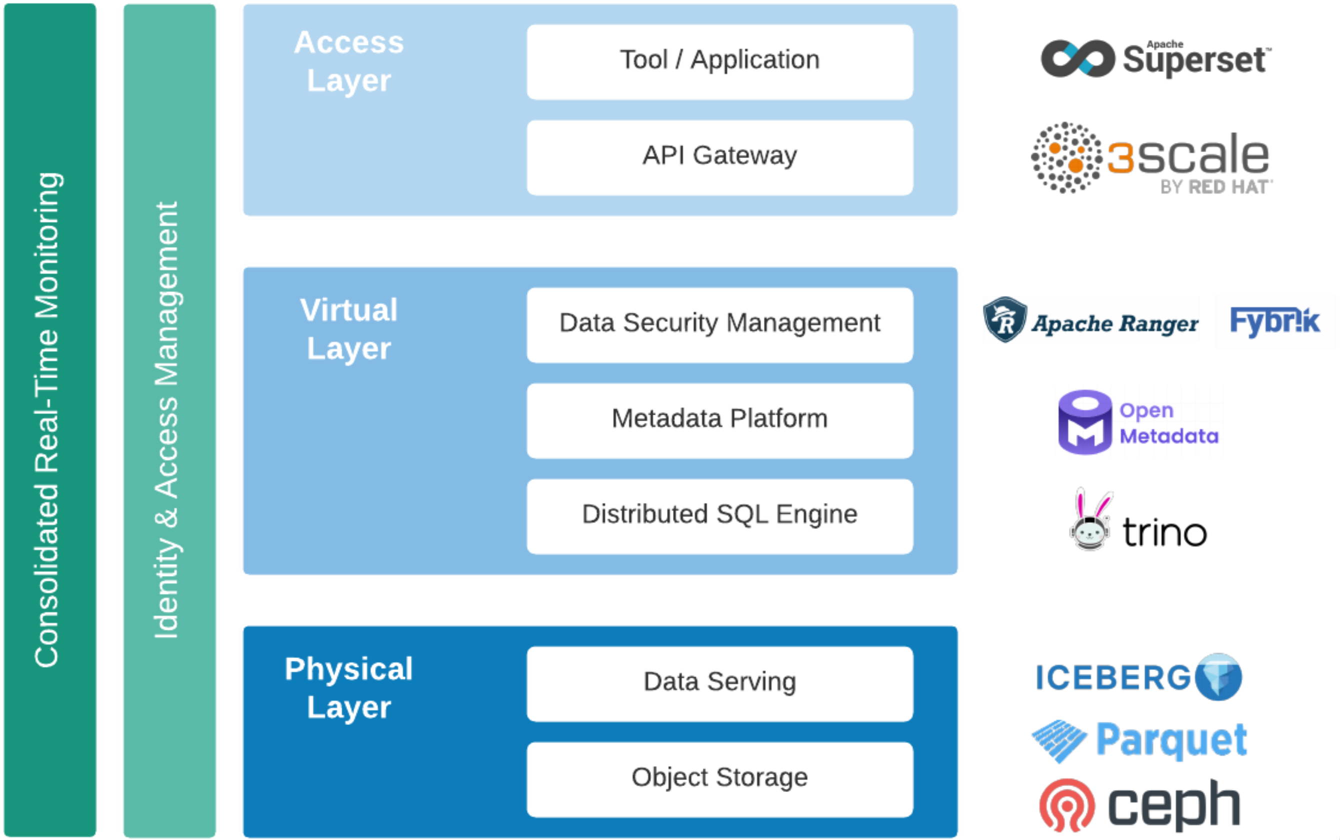
**Machine Learning Platform** Kubeflow

# Data Science Platform Roadmap

| Layer | Role | What we have now | Roadmap |
|---|---|---|---|
| Data Pipeline | Data integration, transformation, versioning, automation and overall management. | Build and manage end-to-end data pipelines on Jupyter notebooks provided as a service. Elyra provides visual pipeline editor and batch management with notebook and python scripts, as well as version control with Github. | Data testing, documentation, and profiling (great_expectations) |
| Train | Model training, optimization and validation. | Training of ML models via any available training operator in Kubeflow. | |
| Deploy | CI/CD pipelines development and management, model, experiment and data versioning. | Pipeline automation via Elyra / Kubeflow / Airflow. Data-driven pipelines, data versioning (Pachyderm), data lineage (DBT). | Improvements in templating / automation of metadata ingestion and management. |
| Manage | Data and metadata catalogs for data sets / pipelines / models. Security and compliance management. | Dataset metadata is managed into a data catalogue (OpenMetadata) and refreshed / versioned automatically. | Integration of metadata for security and compliance (Apache Ranger / Fybrik). |
| Serve | Make trained model available for inference to tool / application. | Kubeflow supports KFServing and Seldon Core by default. It has not been tested / documented. | |

OS-C

# Architecture: Data Management



Consolidated Real-Time Monitoring

Identity & Access Management

Access Layer
- Tool / Application
- API Gateway

Virtual Layer
- Data Security Management
- Metadata Platform
- Distributed SQL Engine

Physical Layer
- Data Serving
- Object Storage

# Data Access Management

| Layer | Role | What we have now | Roadmap |
|---|---|---|---|
| Object Storage | Secure access to data source for ingestion | Currently based on S3 with secret-based access controls.<br>Proprietary data sits on standalone bucket.<br>Secrets are only provided for ingestion pipelines developers. | Automatic secret retrieval by automated ingestion pipeline for production pipelines. Container storage implementation and review of data ingestion so it is not infrastructure-specific (boto3). |
| Data Serving | Manage data and schema versioning automatically with ACID transactions. | anage data and schema evolution automatically. Time travel enables reproducible queries that use exactly the same table snapshot, or lets users easily examine changes. Version rollback allows users to quickly correct problems by resetting tables to a good state. | |
| Distributed SQL Query Engine | Centralized data access and analytics with query federation. Authentication and data access controls for all data queries. | Integration with GitHub SSO via temporary JTW. Access management by catalog, schema (source / pipeline), table (data set), column (data elements). Can filter by row and mask data. integration with data catalogue (OpenMetadata). | Support of complex data types such as GeoTIFF. |
| Metadata Platform | Data schema & metadata management, lineage at the dataset level, data catalogue browse and search. | Data schema and metadata management, data catalogue. | Data compliance management at metadata level. |
| Data Security Management | Enable, monitor and manage data security across the platform (Admin GUI with authorization management and audit). | NA | Centralized management and monitoring of access at query engine level.<br>Data security management at metadata level. |
| API Gateway | Enforce policies which control security aspects such as the authentication, authorization of services acquiring data for external applications / tools. | NA | API Gateway with distributed data management by data owner. |

OS-C

# Demonstration Scope

Enabling real-time energy consumption and carbon emissions reporting through integration of Kepler with CO2 emissions statistics

ELECTRICITY MAPS

CO2 emission intensity

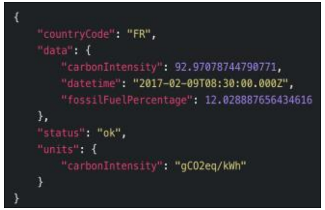
Kepler

Power measurement

Data Center PUE

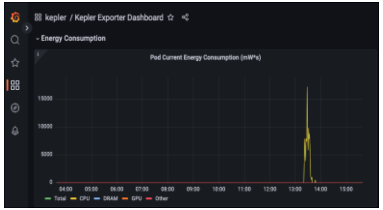
Data center power efficiency

CO2 footprint at Pod, Container, Service level



```
{
    "countryCode": "FR",
    "data": {
        "carbonIntensity": 92.97078744790771,
        "datetime": "2017-02-09T08:30:00.000Z",
        "fossilFuelPercentage": 12.028887656434616
    },
    "status": "ok",
    "units": {
        "carbonIntensity": "gCO2eq/kWh"
    }
}
```

+

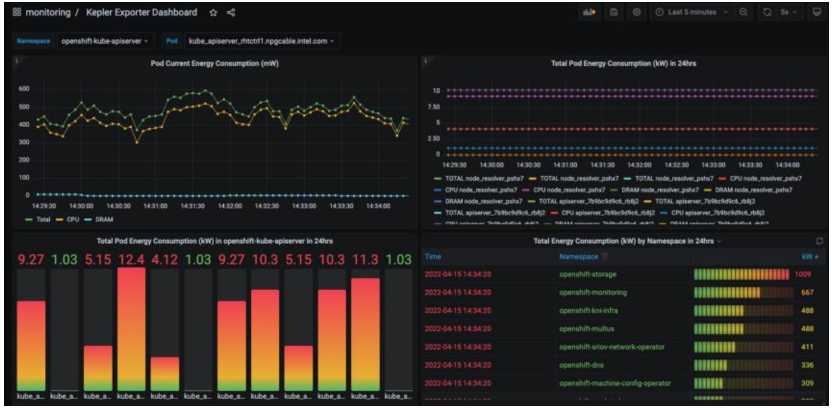


YouTube

https://www.youtube.com/watch?v=dd-6rqp_qlA

OS-C

# Kubernetes Efficient Power Level Exporter (Kepler)



Namespace: openshift-kube-apiserver

Pod: kube_apiserver_rhtctrl1.npgcable.intel.com

Total Energy Consumption (kW) by Namespace in 24hrs

| Time | Namespace | kW |
|---|---|---|
| 2022-04-15 14:34:20 | openshift-storage | 1009 |
| 2022-04-15 14:34:20 | openshift-monitoring | 667 |
| 2022-04-15 14:34:20 | openshift-kni-infra | 488 |
| 2022-04-15 14:34:20 | openshift-multus | 488 |
| 2022-04-15 14:34:20 | openshift-sriov-network-operator | 411 |
| 2022-04-15 14:34:20 | openshift-dns | 336 |
| 2022-04-15 14:34:20 | openshift-machine-config-operator | 309 |

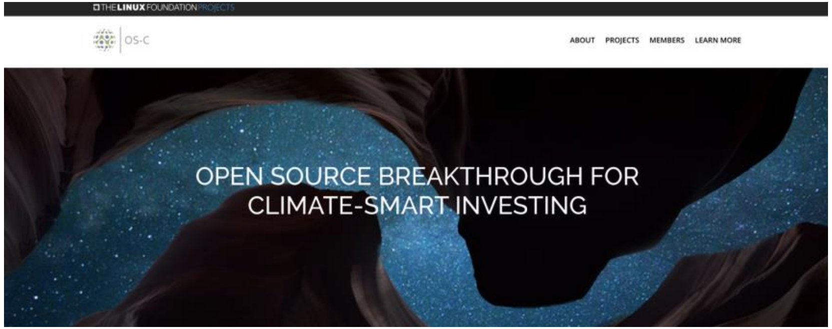
$$\text{Energy}_{\text{Pod}} = \sum \text{(CPU / Memory / GPU / cgroupfs / hwmon Stats)}$$

Kepler uses uses eBPF to probe energy related system stats and exports as Prometheus metrics.

- Red Hat, IBM, Intel are major contributors
- Measures K8s node energy usage thru processor Running Average Power Limit (RAPL) interfaces
- Current support x86_64, being extended to support ARM64 and S390 platforms
- Estimates pod energy usage from node usage, including CPU / GPU / RAM, leveraging ML models for the estimation
- More accurate than existing dashboards such as CCF / Scaphandre which report energy consumption based on CPU time
- Project applied for CNCF Sandbox

# Next Steps: Find Out More

- Open Data Hub community page provides a Get Started guide at https://opendatahub.io/
- OS-Climate Data Commons Architecture: https://github.com/os-climate/os_c_data_commons
- OS-Climate Data Commons article: https://towardsdatascience.com/making-climate-data-easy-to-find-use-and-share-5190a0926407
- Kepler GitHub: https://github.com/sustainable-computing-io/kepler


THE LINUX FOUNDATION PROJECTS
OS-C
ABOUT
PROJECTS
MEMBERS
LEARN MORE
OPEN SOURCE BREAKTHROUGH FOR CLIMATE-SMART INVESTING