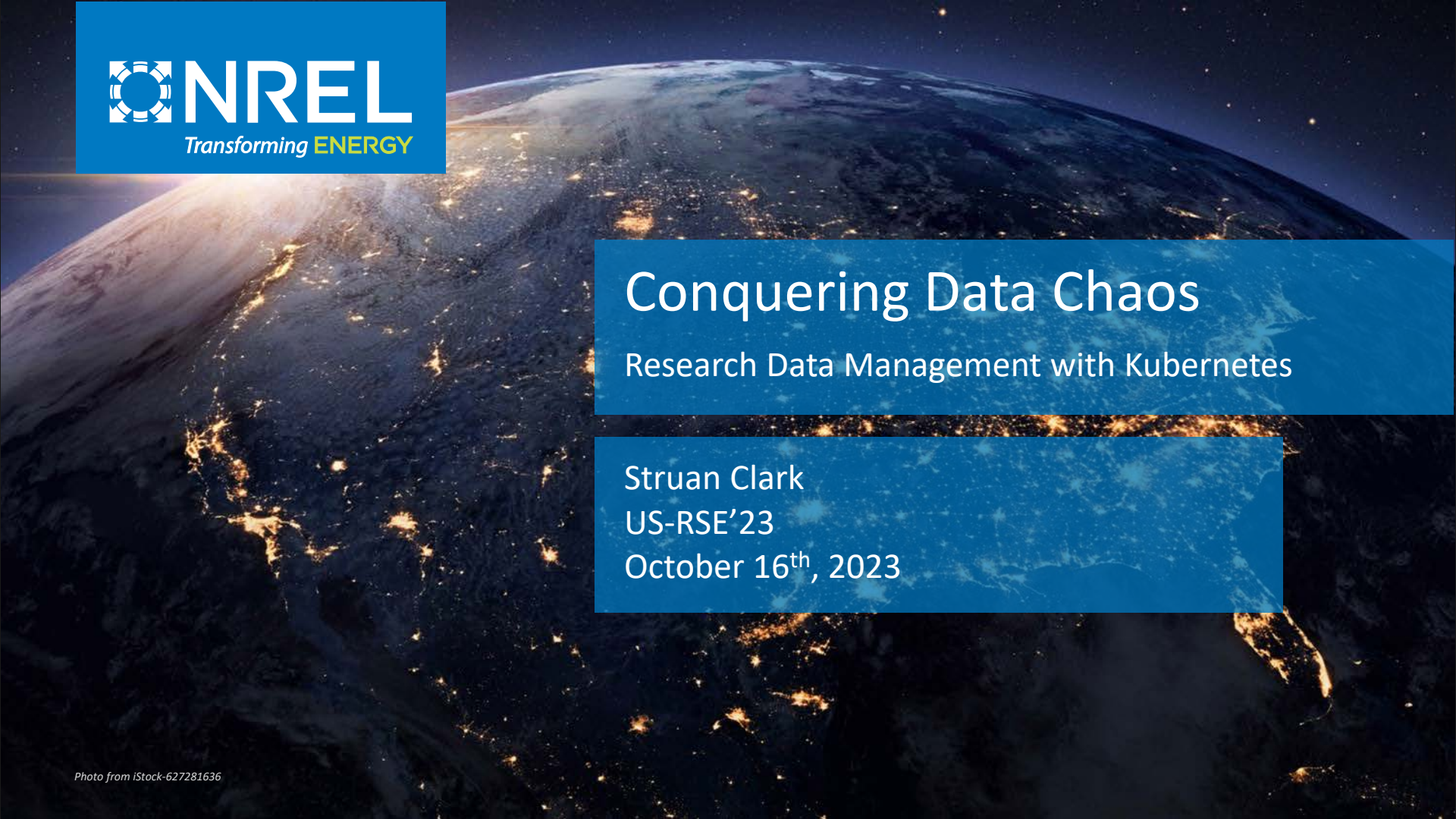


Conquering Data Chaos

Research Data Management with Kubernetes

Struan Clark

US-RSE'23

October 16th, 2023

Photo from iStock-627281636

Outline

1. Data Challenges
2. Kubernetes Primer
3. Our Solution & Stack
4. Questions

My Path to RSE

- Senior Backend Engineer and Data Architect in NREL's Data, Analysis, and Visualization (DAV) group
  - 2021-today: Data architect at NREL
  - 2011-2021: Data analysis at small defense contractor
  - 2007-2011: B.S. in Computer Engineering from Virginia Tech

- Generalist - worked across wide variety of projects and fulfilled several different roles

1 Data Challenges

2 Kubernetes Primer

3 Our Solution & Stack

4 Questions

Types of Data in DAV

- High-Performance Computing (HPC) data
  - Job queue
  - System and facility power and usage
- Experimental data from NREL lab facilities
  - Hydrogen storage equipment
  - Renewable energy experiments
- External datasets for visualization
  - Power grid
  - Traffic flow

Data Challenge #1

“We have a lot of different scripts running to collect data. Some write data to files, others to a one-off database.”

Data Challenge #2

“I have a bunch of big data files and some code I run on my laptop that I use to analyze and visualize it.”

Our Solution

- Build out shared infrastructure to store data
- Provide some common patterns to get data into this infrastructure, leverage existing code where possible
- Once data is there, provide a generic way to access it so users can reuse their client code

Our Solution

- Provide a collaborative frontend that allows users to do basic querying and plotting of their data
- Keep the infrastructure flexible so we can adapt as data needs change
- **Kubernetes helps with all of these!**

1 Data Challenges

2 Kubernetes Primer

3 Our Solution & Stack

4 Questions

What is Kubernetes?

- Platform for managing containerized workloads
  - Evolved from an internal Google platform named Borg
  - Open-sourced as Kubernetes in 2014

- Infrastructure-as-Code (IaC)
  - Almost always managed using the Helm tool
  - Similar to services provided by public clouds
  - Concept of servers as “cattle” not “pets”

Kubernetes Features

- Automated rollouts and rollbacks
- Service discovery and load balancing
- Storage orchestration
- Self-healing infrastructure
- Batch execution (in addition to services)
  - This is very useful for ETL (Extract, Transform, Load) processes!
- Horizontal scaling
- Secret and configuration management

1 Data Challenges

2 Kubernetes Primer

3 Our Solution & Stack

4 Questions

Publicly Available “Data Engines”

- Streaming data platform – Redpanda
  - Apache Kafka-compatible API
  - Better performance and simpler setup than “true” Kafka
- Time-series data storage – Apache Druid
  - High-performance for very large data quantities
  - Native ingestion support via Kafka API

Publicly Available “Data Engines”

- Relational database – PostgreSQL
  - Not in Kubernetes, but integrates with services that are
- Dashboarding and plotting – Apache Superset
  - Connectors for Kafka, Druid, PostgreSQL, and more
  - Users can view and collaborate on data immediately

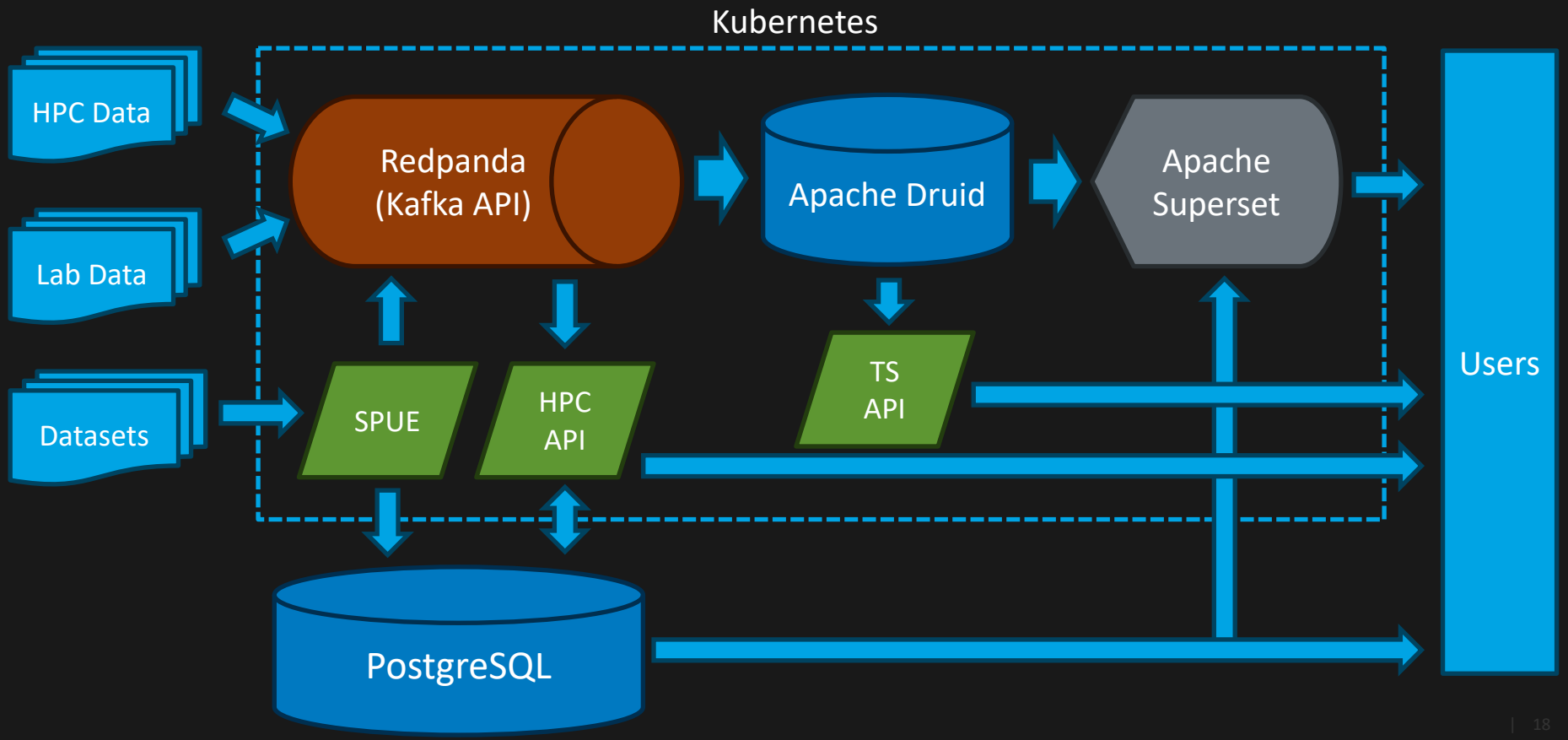
DAV Applications

- HPC Monitoring API – Deployment/Service
  - Integrates with data engines to both load HPC data into PostgreSQL and provide convenient user access to HPC data
- Time-series (TS) API – Deployment/Service
  - Integrates with Druid to provide convenient user access to loaded time-series data

DAV Applications

- SPUE (Simple Python Utility for ETL) – Job
  - Provides solid connectors for writing data to Kafka (and by extension Druid) and PostgreSQL
  - Allows “drop-in” of existing file parsing or API consuming code with minimal modifications

DAV Data Systems Architecture



DevOps Configuration

- Data Engines
  - Redpanda and Superset are deployed using public Helm charts with DAV-specific configuration overlayed on top
  - Druid is deployed using a heavily customized Helm chart based on the public Druid chart

DevOps Configuration

- DAV Applications
  - HPC API, TS API, and SPUE are containerized using Docker Compose and are deployed using Helm charts
  - The APIs run as long-lived Kubernetes Deployments, whereas SPUE is deployed as a Kubernetes Job or CronJob

- **We can deploy most pieces of our stack with 1-click**

Lessons Learned

1. Encapsulate as much of the deployment logic as possible in Helm charts or other IaC tools

2. Make sure you know your application requirements
   - Some software is more optimized than others for a Kubernetes environments

Lessons Learned

3. Understand the actual hardware backing the cluster
   - This is important for anything more complicated than a basic web app or API

4. Kubernetes is not an island
   - Make sure to think about storage and networking (especially DNS and SSL certificates)

1 Data Challenges

2 Kubernetes Primer

3 Our Solution & Stack

4 Questions

Thank you! Questions?

www.nrel.gov

NREL/PR-2C00-87807

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by U.S. Department of Energy Office of Energy Efficiency and Renewable Energy. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

Photo from iStock-627281636

NREL
Transforming ENERGY