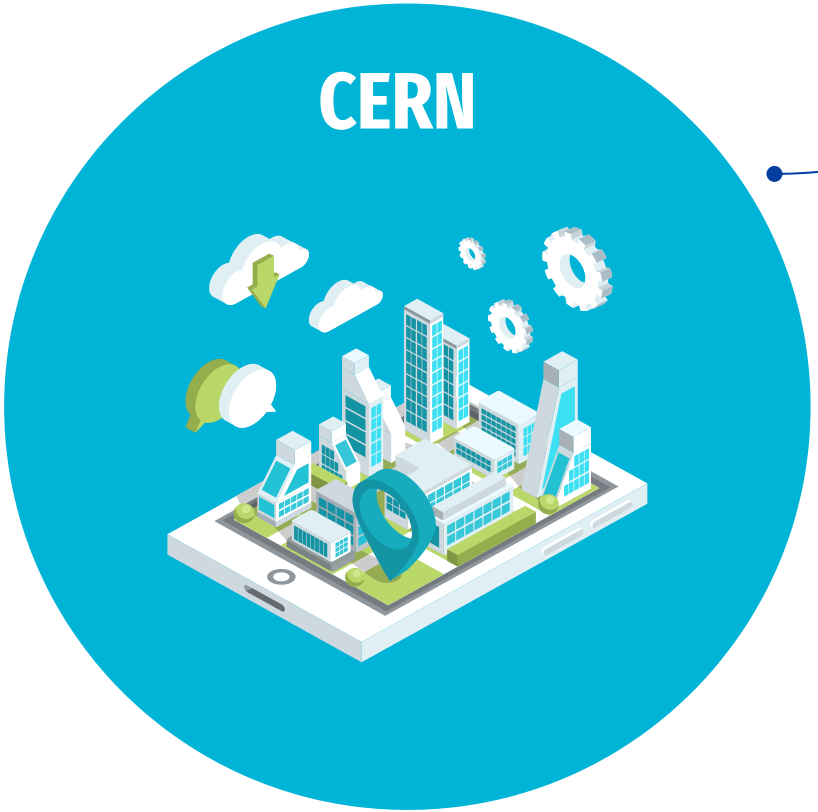
Introduction to Cloud Computing

Instructor:
Romina Trujillo C.

Table of Contents

User- Focused Tutorial



Repository of scientific data

Containing a wealth of knowledge about particle physics experiments and groundbreaking discoveries

Physics analyses

These analyses in the realm of particle physics often involve working with massive volumes of data, spanning several terabytes.

How to access these data?

A team of physicists eager to explore the intricacies of particle physics through the analysis of data from the Large Hadron Collider (LHC) experiments at CERN.

However, they may not have access to these resources at their own institutions.

To overcome this limitation, the **CMS OpenData workshop in 2023** aims to enable researchers to utilize computing resources through public cloud vendors.

What is Cloud Computing?

Provides a flexible and scalable model for delivering computing services over the internet.

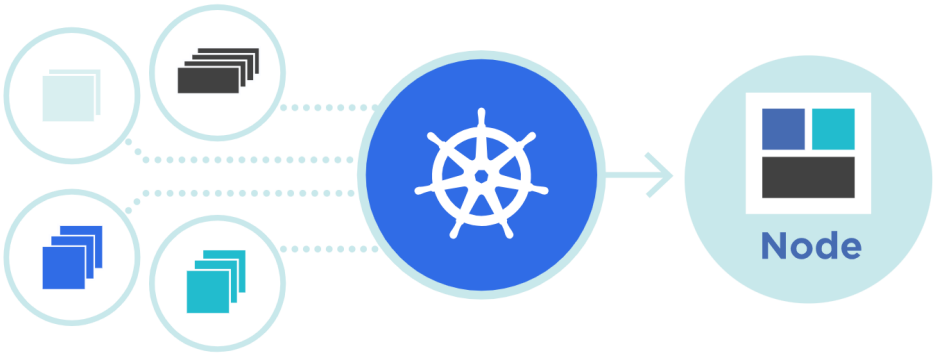
Researchers gain the ability to tap into computing resources without the need for extensive local infrastructure investments.

It enables on-demand access to a wide range of resources, including compute power, storage, databases, and networking capabilities.

Key Advantages of Cloud Computing

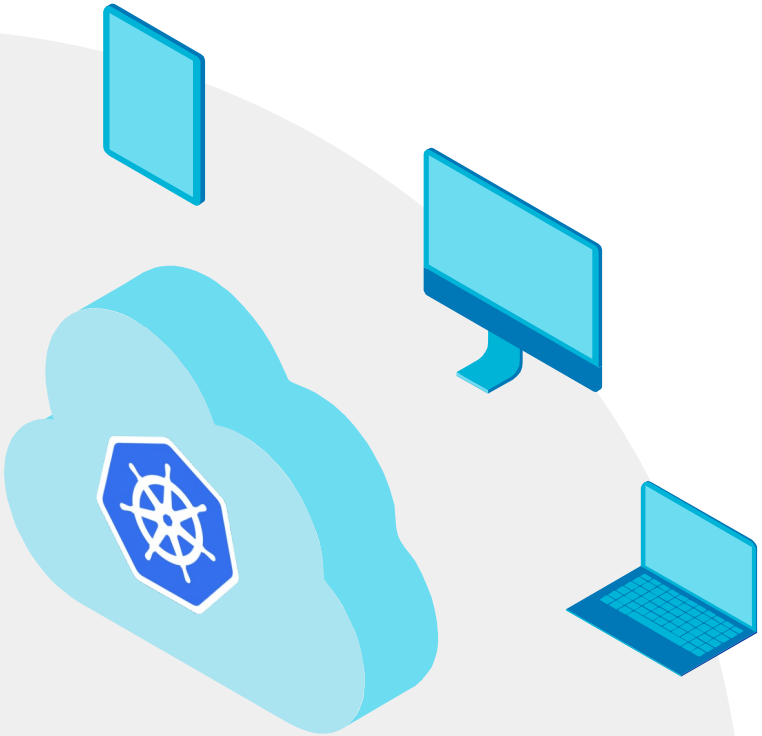
1. Scalability
2. High Availability and Reliability
3. Pay-as-you-go pricing model
4. Accessibility

What is Kubernetes?



Powerful container orchestration platform, for automating deployment, scaling, and management of containerized applications.

Why Kubernetes?



01

Makes possible to process and analyze large-scale datasets without the need for extensive local infrastructure.

02

Efficiently handle the massive volumes of data involved in physics analyses and unlock valuable insights within the CERN Open Data

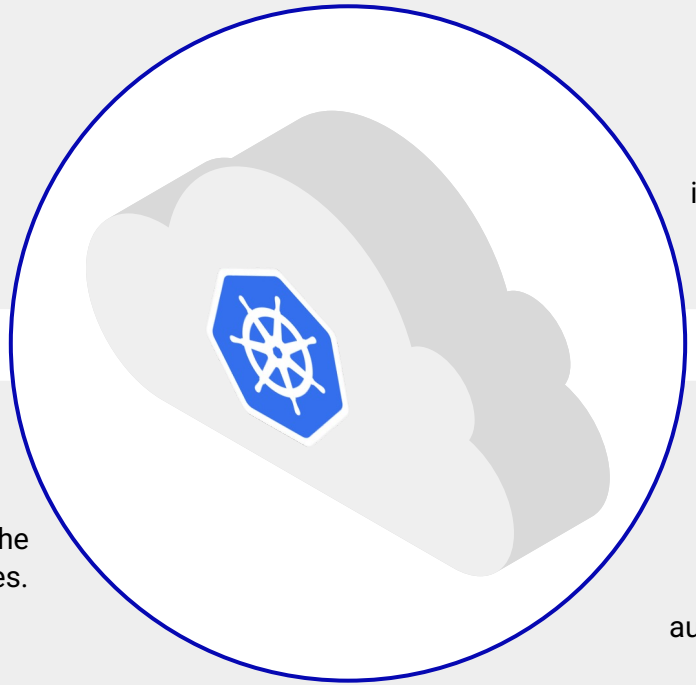
03

Scale computing resources, optimize data processing workflows, and harness the power of distributed computing.

How it works?

1 Containerization

Applications and workflows are encapsulated into containers, which package all the necessary dependencies and configurations.



2 Cluster Creation

A Kubernetes cluster is created, consisting of interconnected nodes (virtual or physical) that run the containerized applications.

3 Deployment and Orchestration

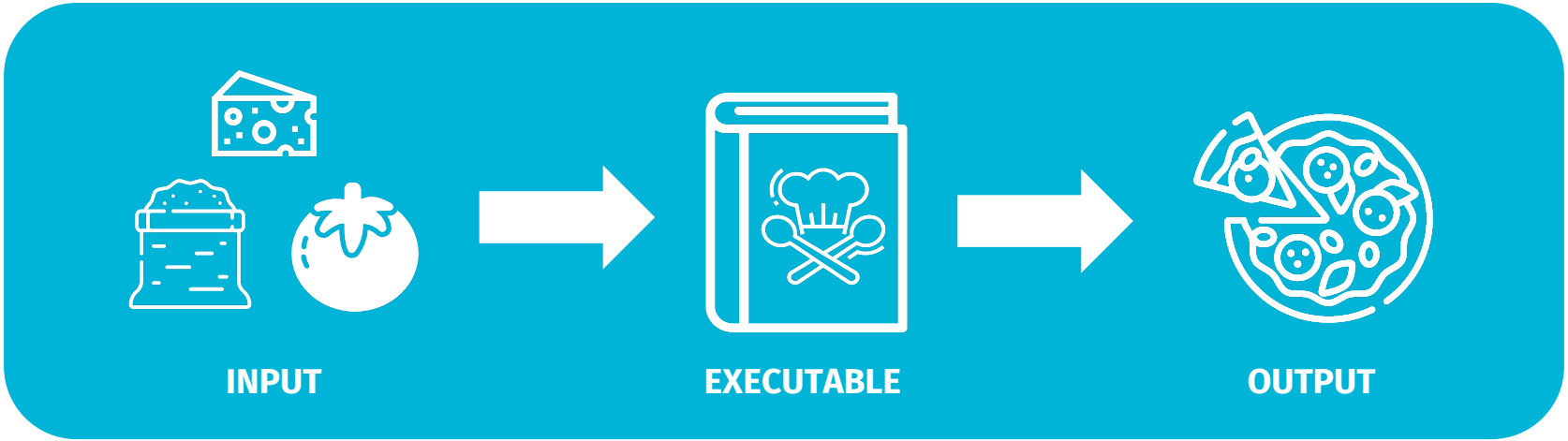
Applications are deployed to the cluster using configuration files. Kubernetes automatically schedules and manages the deployment, ensuring optimal resource utilization and scalability.

4 Monitoring and Self-Healing

Monitors the health of containers and nodes, automatically restarting failed containers and reallocating resources as needed

Compute Tasks

- A single compute task is called a job
- Three main pieces of a job are the input, the executable(program), and the output
- Executable must be runnable from the command line without any interactive input



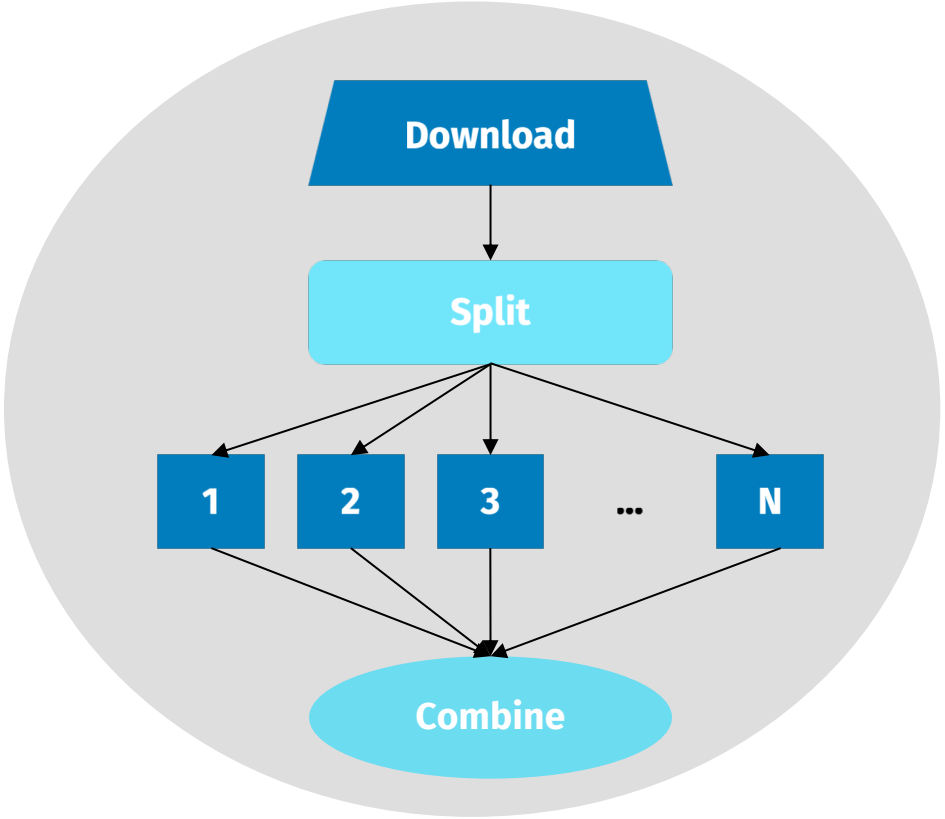
Workflows

Problem

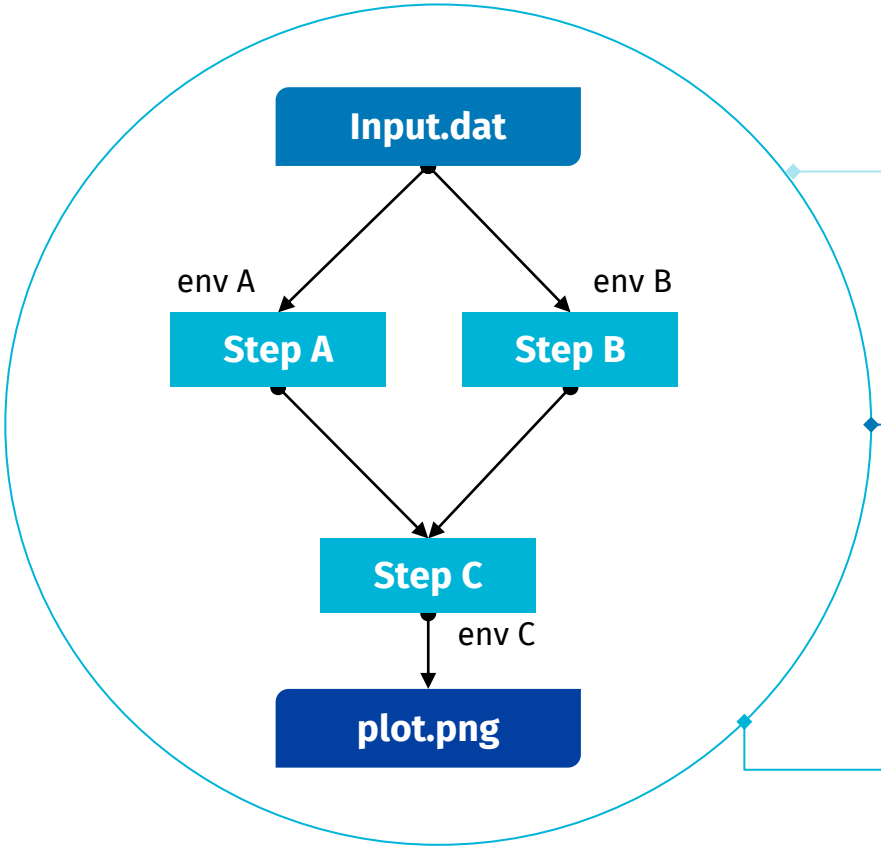
Want to submit jobs in a particular order, with dependencies between groups of jobs

Solution

Compose a Directed Acyclic Graph (DAG)



Directed Acyclic Graph (DAG)

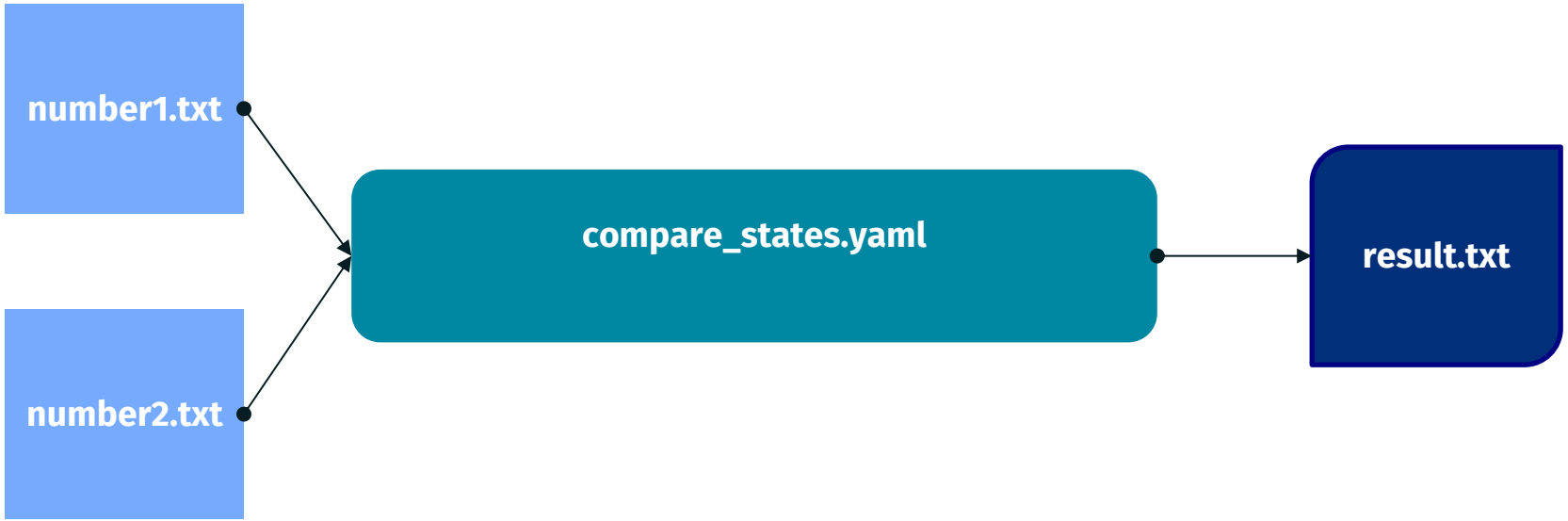


Each graph vertex represents a unit of computation with its inputs and outputs

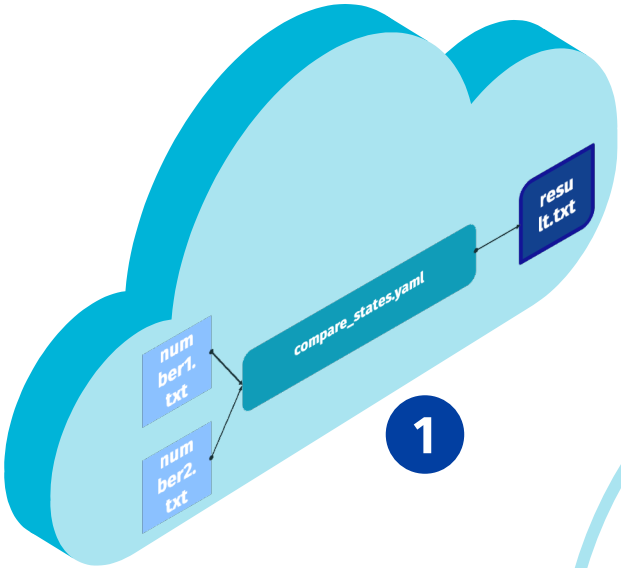
The declarative programming approach is further exemplified by a multi-level job cascading paradigm that was implemented in the Argo workflow specification language.

The focus on declarative rather than imperative programming enables researchers to concentrate on the problem domain at hand without having to think about implementation details such as scalable job orchestration.

Kubernetes Workflow Orchestration - ARGO



Workflow Translation



1

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: dag-
spec:
  entrypoint: main
  volumes:
    - name: workdir
      hostPath:
        path: /mnt/vol
        type: DirectoryOrCreate
  templates:
    - name: main
      dag:
        tasks:
          - name: clean
            template: clean-template
          - name: a
            dependencies: [clean]
            template: curl-template

          - name: b
            dependencies: [clean]
            template: curl-template

          - name: c
            dependencies: [a, b]
            template: comparison-template
```

2

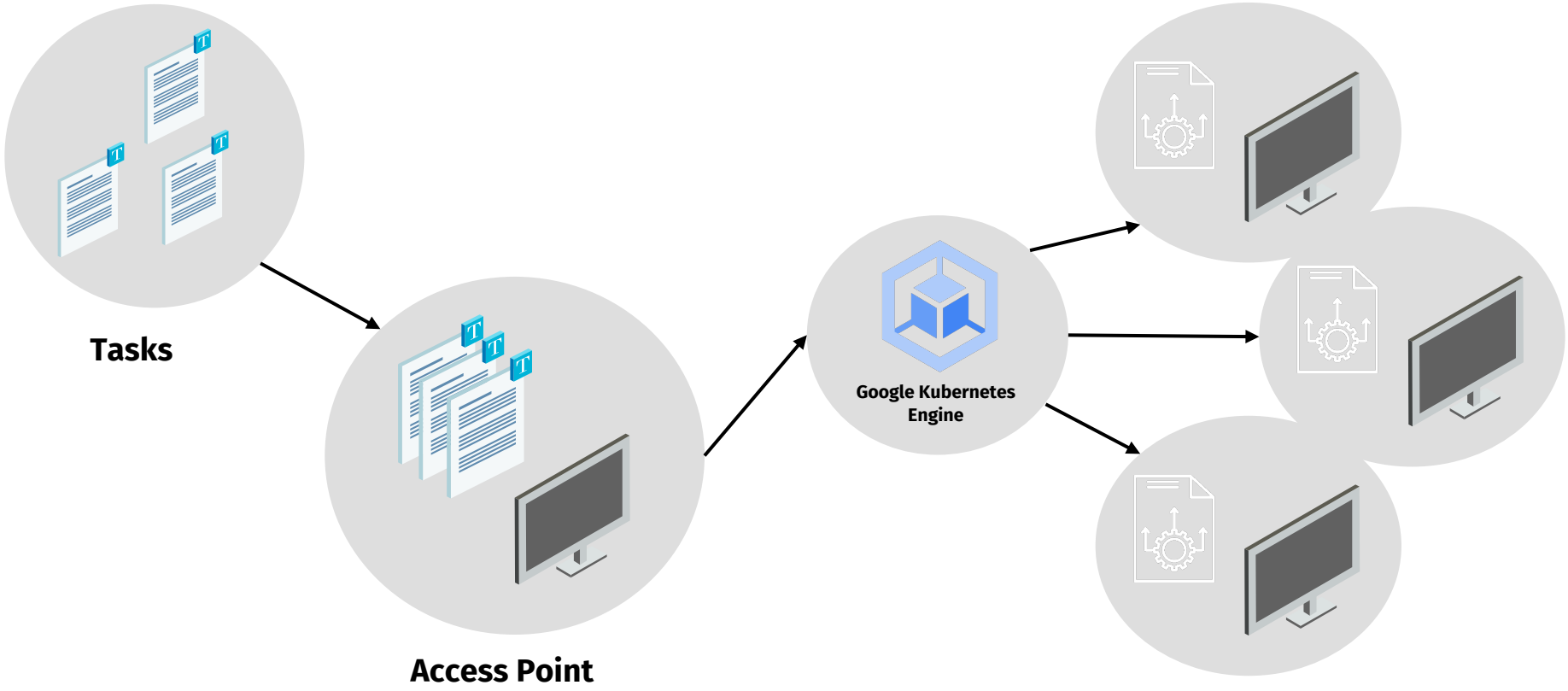
Submit File

Communicates everything about your job(s) to Kubernetes

Google Kubernetes Engine

3

Set Up and Run a Workflow on the Cloud



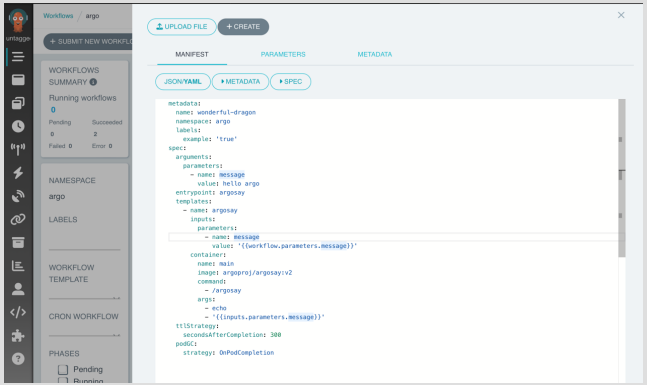
Declarative workflow run analysis workflows on Kubernetes

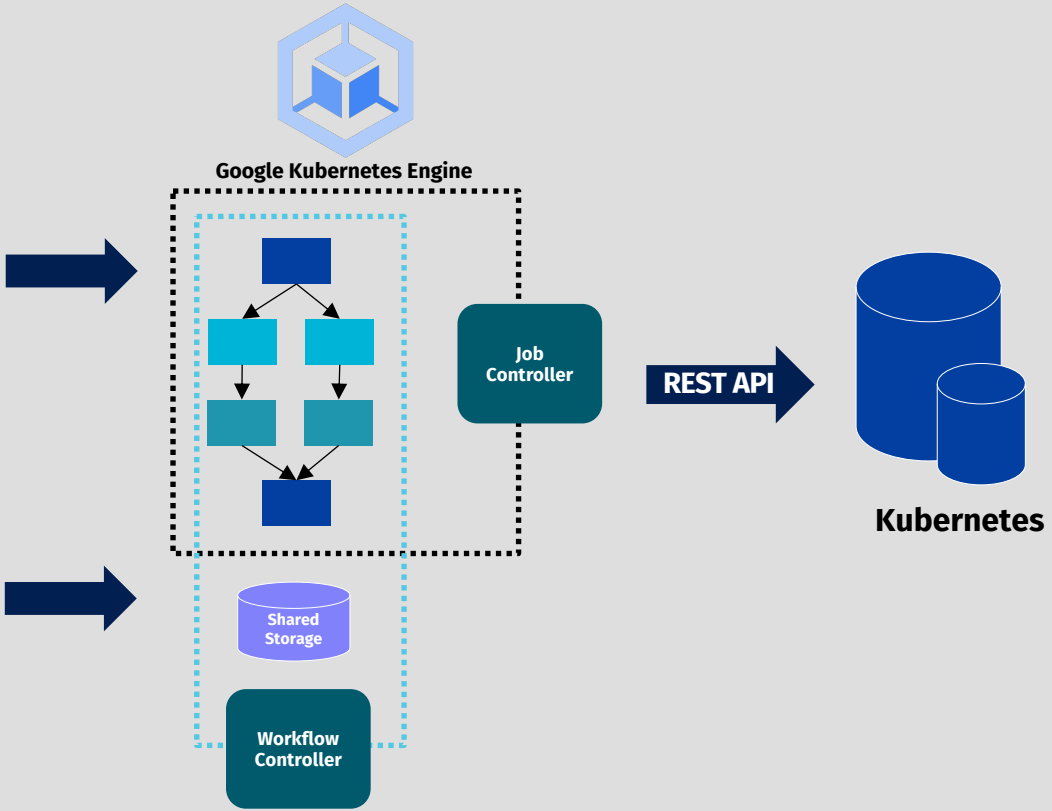
CLI UI

```
● → Minikube argo submit submit-file.yaml -n argo
Name:                dag-7df5v
Namespace:           argo
ServiceAccount:      unset (will run with the default ServiceAccount)
Status:              Succeeded
Conditions:
 PodRunning          False
 Completed           True
Created:             Sun Jul 09 23:35:55 +0200 (1 minute ago)
Started:             Sun Jul 09 23:35:55 +0200 (1 minute ago)
Finished:            Sun Jul 09 23:36:56 +0200 (now)
Duration:            1 minute 1 seconds
Progress:            4/4
ResourcesDuration:   24s*(1 cpu),13s*(100Mi memory)

STEP          TEMPLATE             PODNAME                                  DURATION  MESSAGE
 ✓ dag-7df5v  main
 ├─✓ clean    clean-template       dag-7df5v-clean-template-996255253       9s
 ├─✓ a        curl-template        dag-7df5v-curl-template-3536282375       12s
 ├─✓ b        curl-template        dag-7df5v-curl-template-3553059904       11s
 └─✓ c        comparison-template  dag-7df5v-comparison-template-3569837613  8s
```

Web UI





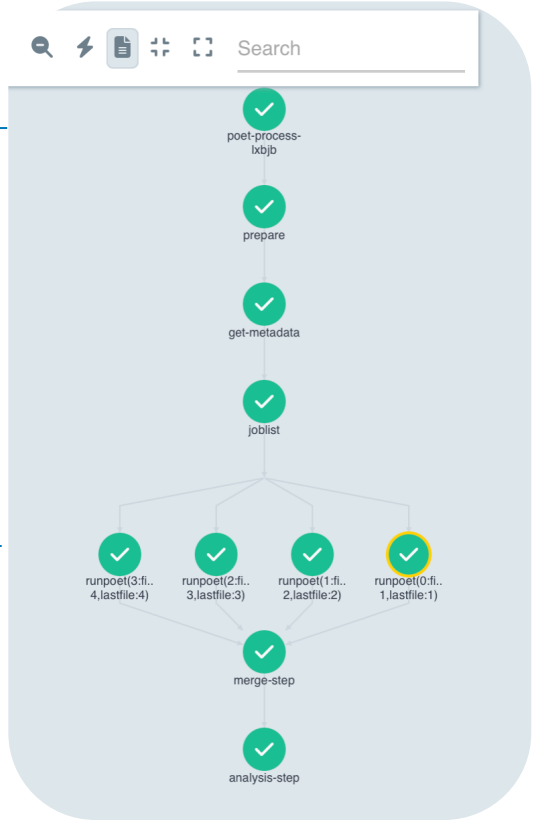
Submitting and Monitoring Jobs

To submit a job

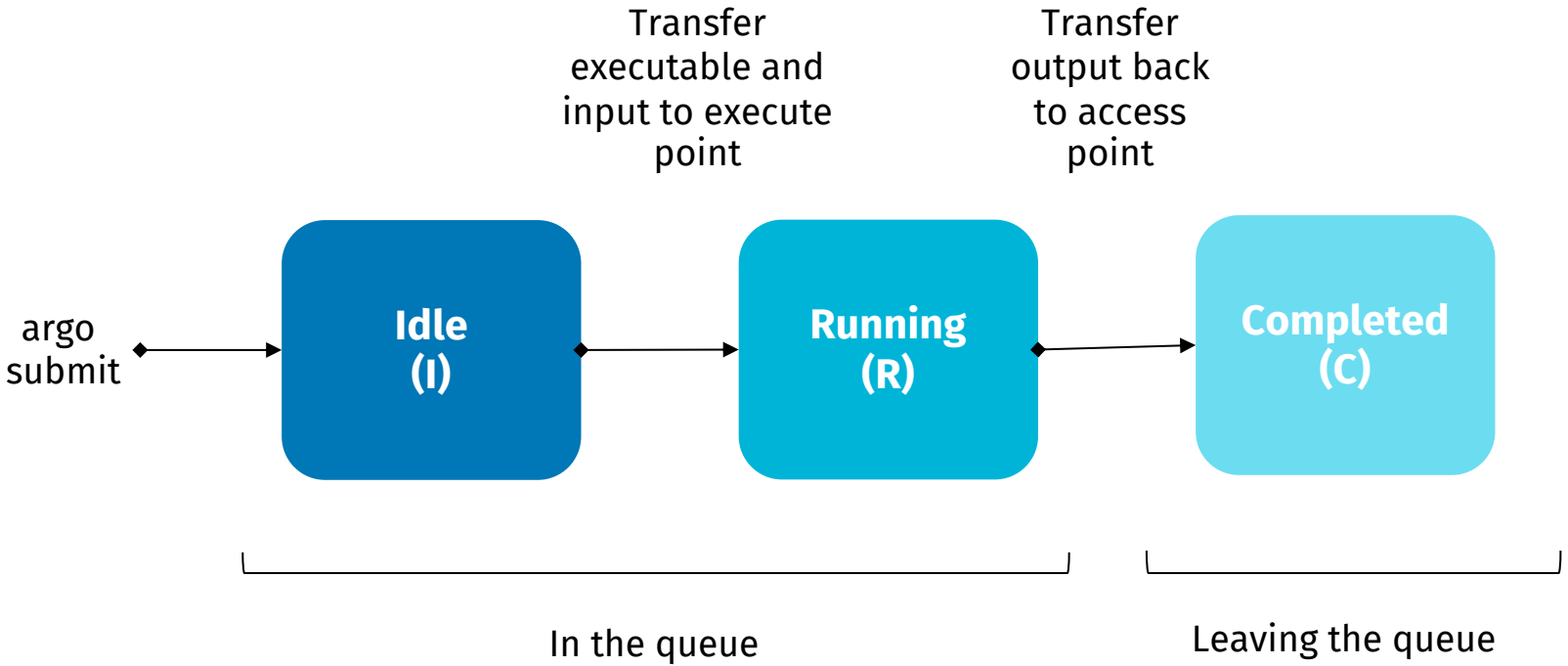
$ argo submit dag-workflow.yaml -n argo

To monitor your workflow

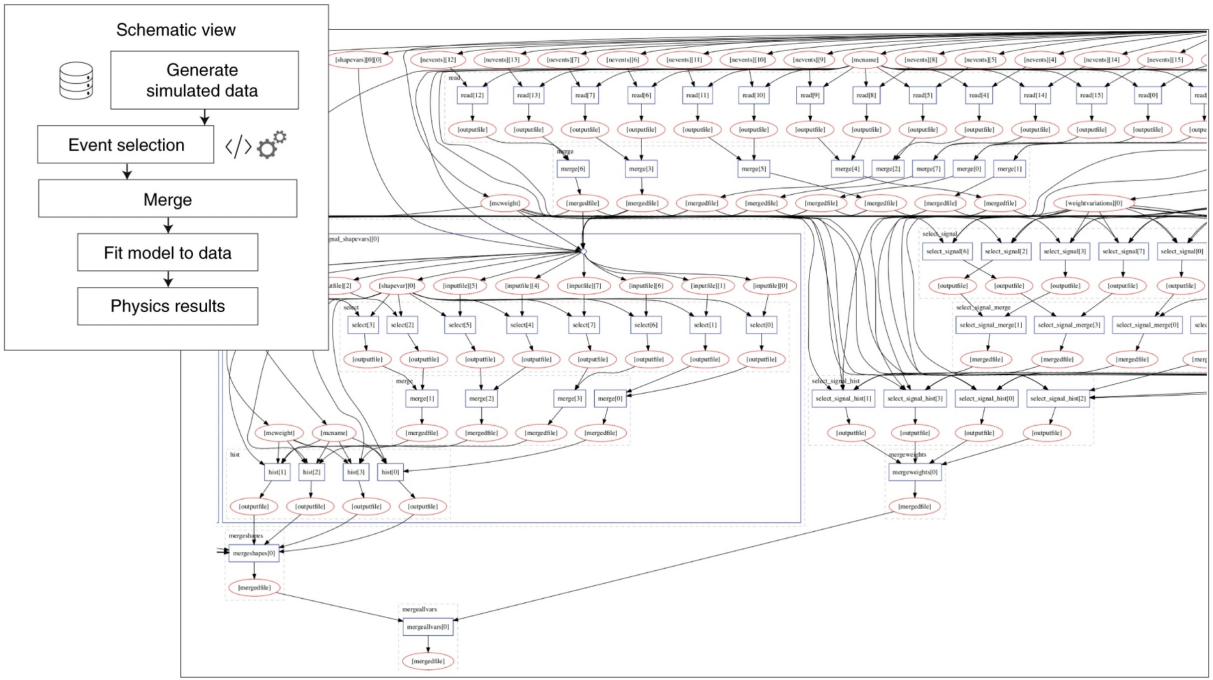
Use tools like Argo GUI

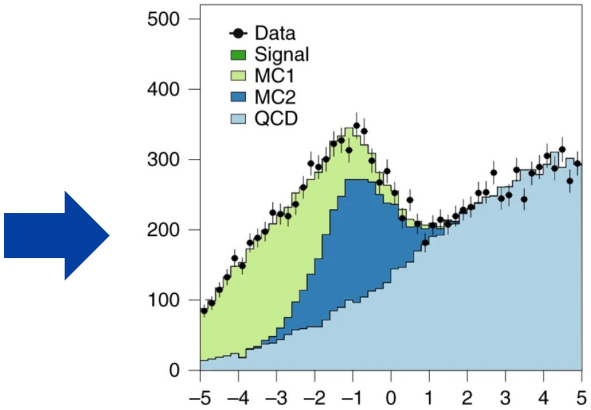


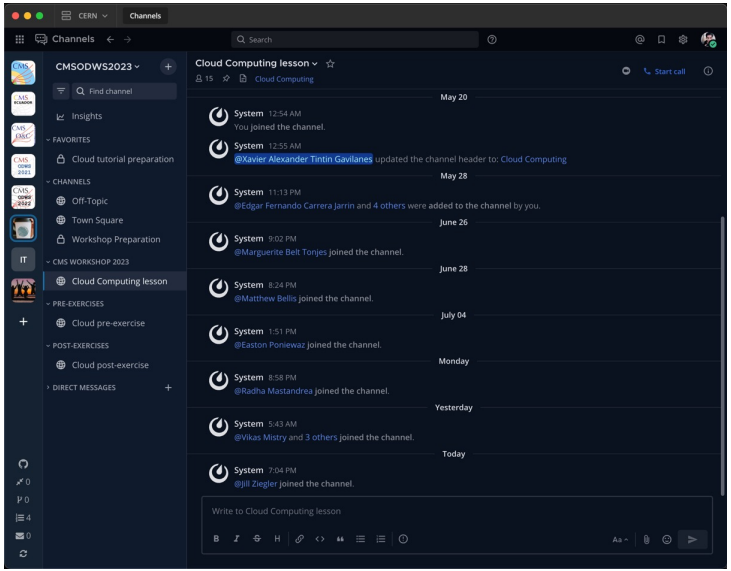
Argo Workflow States



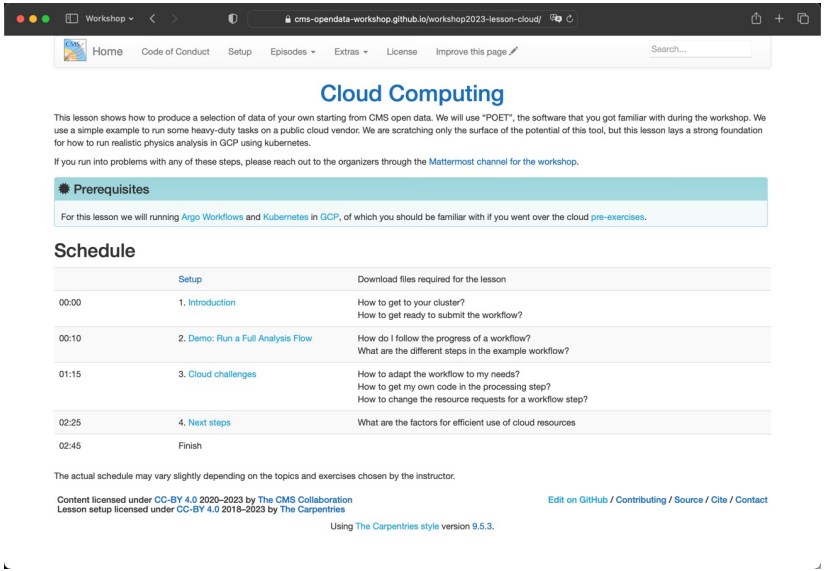
Example of a complex computational workflow







Mattermost: Cloud Lesson Channel



ODWS Cloud Lesson