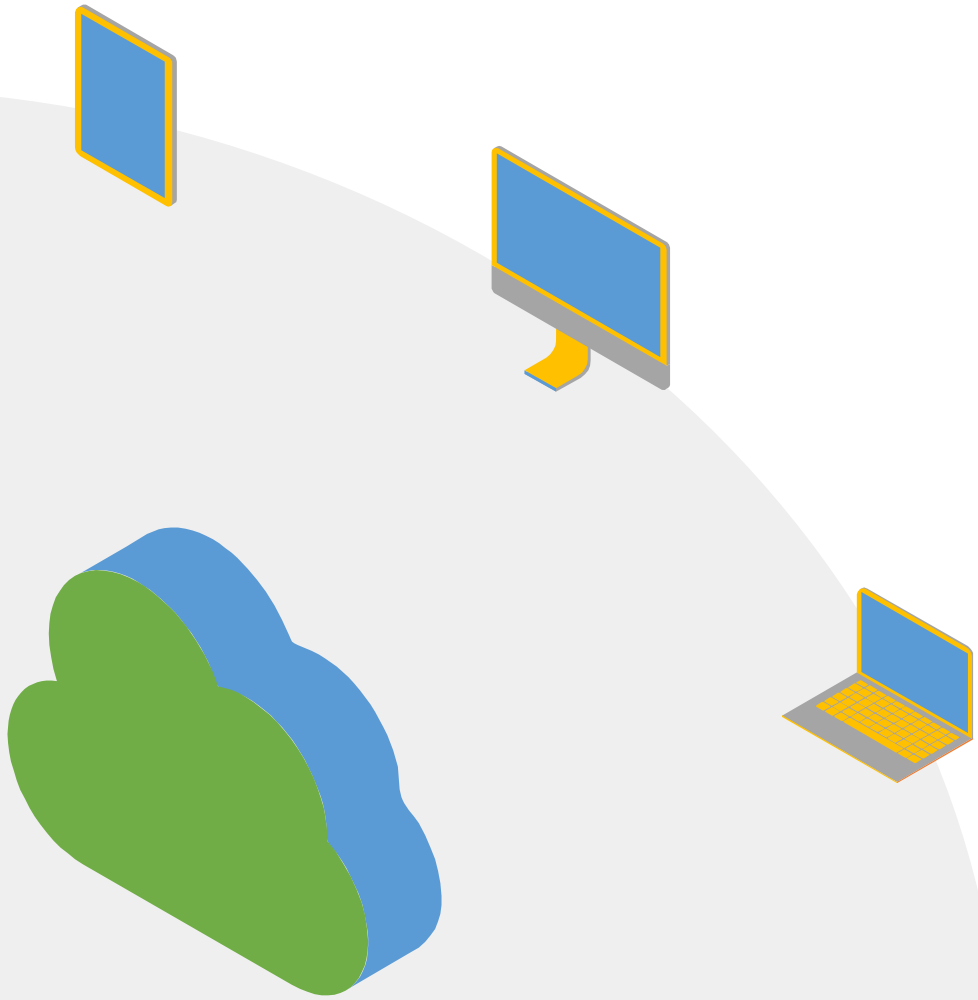


Cloud Demonstration

Instructor:

Xavier A. Tintin

Argo Declarative Workflow

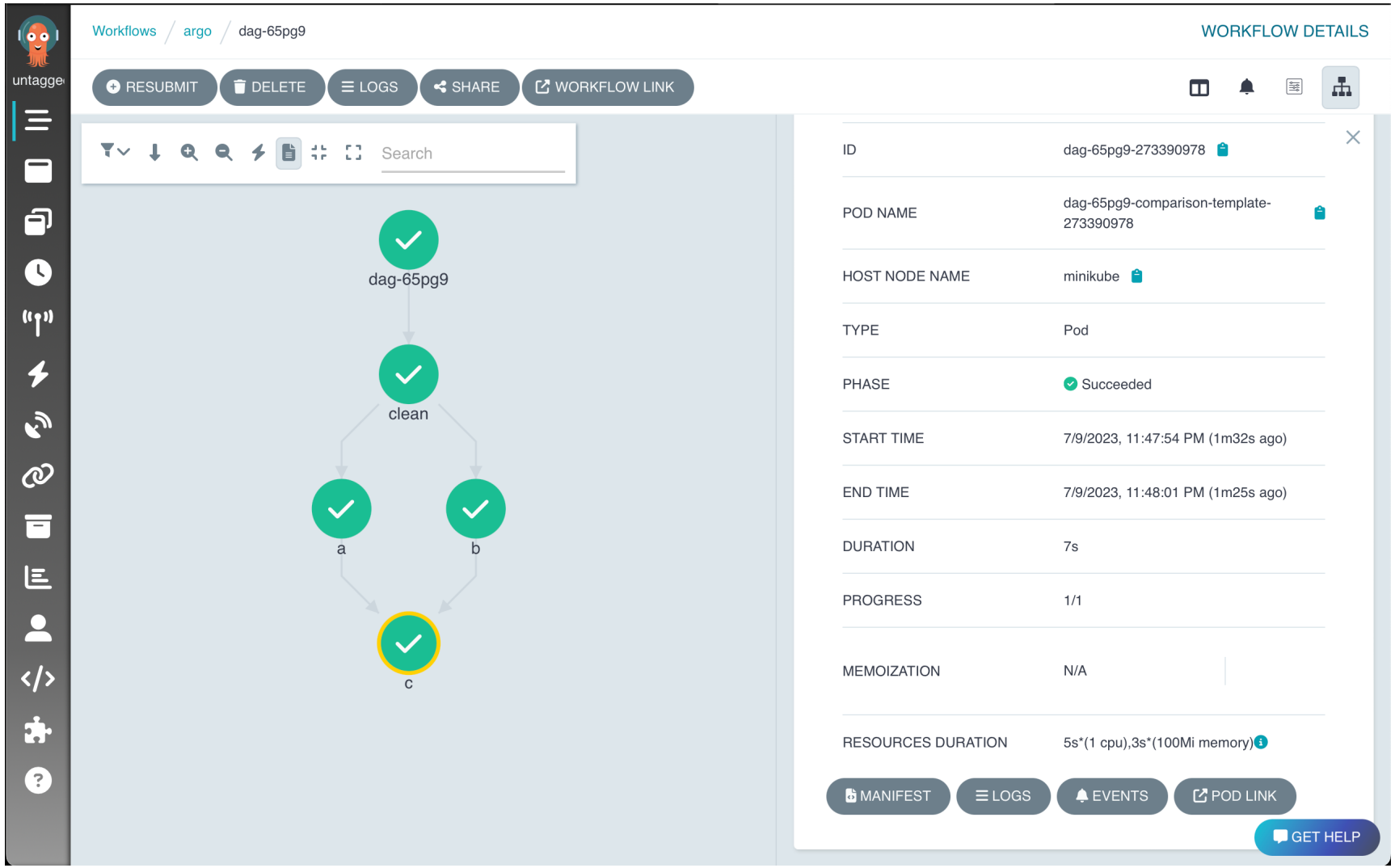
```yaml
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: dag-
spec:
  entrypoint: main
  volumes:
    - name: workdir
      hostPath:
        path: /mnt/vol
        type: DirectoryOrCreate
  templates:
    - name: main
      dag:
        tasks:
          - name: clean
            template: clean-template
          - name: a
            dependencies: [clean]
            template: curl-template
            arguments:
              parameters:
                - name: url
                  value: https://raw.githubusercontent.com/cms-dpoa/cat-hackathon/ma
                - name: output
                  value: number1.txt
          - name: b
            dependencies: [clean]
            template: curl-template
            arguments:
              parameters:
                - name: url
                  value: https://raw.githubusercontent.com/cms-dpoa/cat-hackathon/ma
                - name: output
                  value: number2.txt
          - name: c
            dependencies: [a, b]
            template: comparison-template
            arguments:
              parameters:
                - name: input1
                  value: /mnt/vol/number1.txt
                - name: input2
                  value: /mnt/vol/number2.txt
```

```yaml
    - name: clean-template
      script:
        image: alpine:latest
        command: [sh]
        source: |
          rm -rf /mnt/vol/*

    - name: curl-template
      inputs:
        parameters:
          - name: url
          - name: output
      script:
        image: rootproject/root:latest
        command: [sh]
        source: |
          curl -LO {{inputs.parameters.url}}
          mv {{inputs.parameters.output}} /mnt/vol/
        volumeMounts:
          - name: workdir
            mountPath: /mnt/vol

    - name: comparison-template
      inputs:
        parameters:
          - name: input1
          - name: input2
      script:
        image: alpine:latest
        command: [sh]
        source: |
          number1=$(cat "{{inputs.parameters.input1}}")
          number2=$(cat "{{inputs.parameters.input2}}")
          result=$(($number1 > $number2 ? $number1 : $number2))
          echo $result > /mnt/vol/result.txt
        volumeMounts:
          - name: workdir
            mountPath: /mnt/vol
```

Argo Workflow Analysis




```
→  tmp tree poddata
poddata
├── inputs
│   ├── number1.txt
│   └── number2.txt
└── result.txt

2 directories, 3 files
```

number1.txt

inputs > number1.txt

```
1  42
```

number2.txt

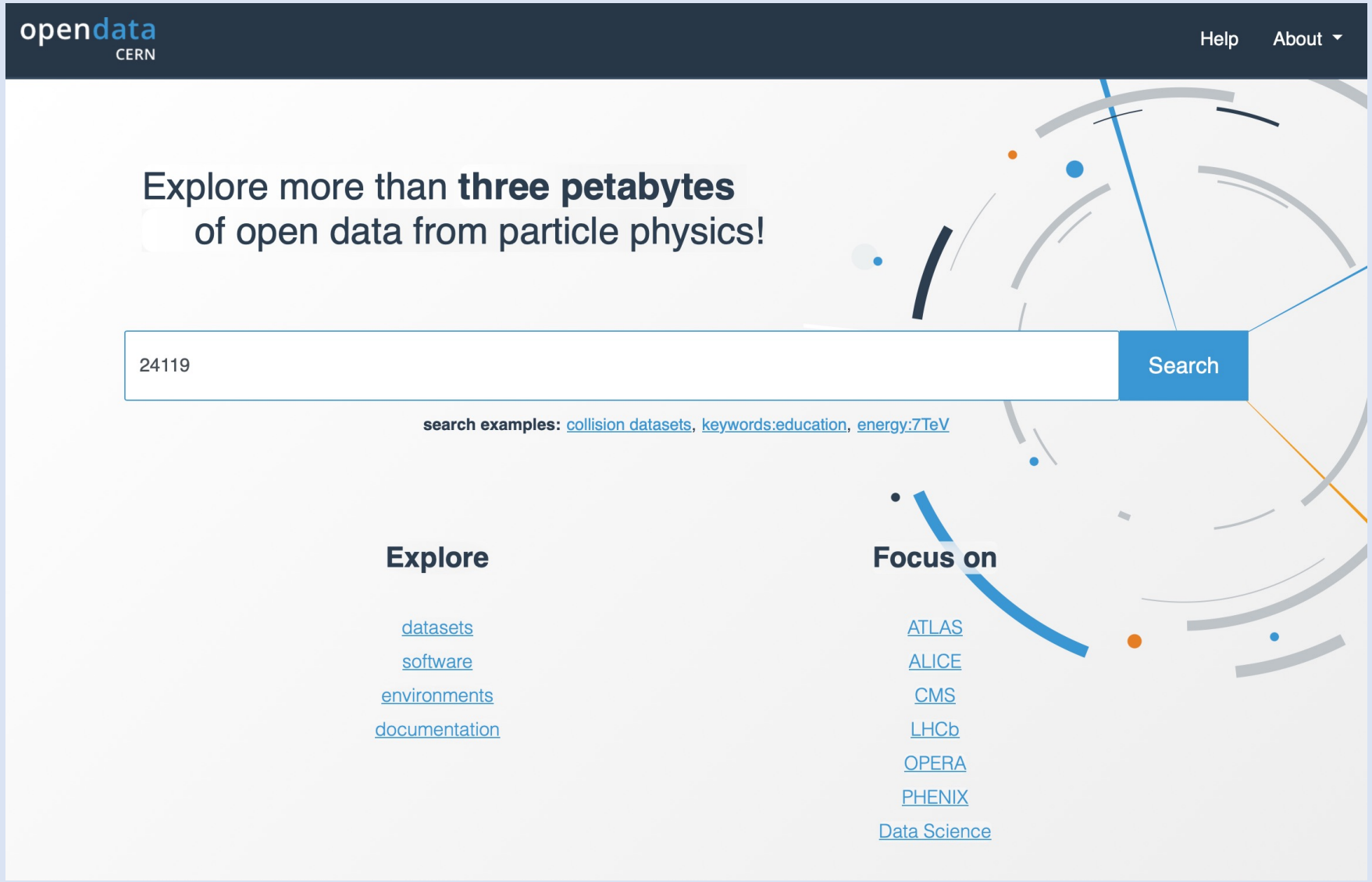
inputs > number2.txt

```
1  333
```

result.txt

result.txt

```
1  333
```


opendata
CERN
Help
About
Explore more than three petabytes
of open data from particle physics!
24119
Search
search examples: collision datasets, keywords:education, energy:7TeV
Explore
datasets
software
environments
documentation
Focus on
ATLAS
ALICE
CMS
LHCb
OPERA
PHENIX
Data Science

opendata CERN

Search

Help About

SingleMuon primary dataset in MINIAOD format from RunD of 2015 (/SingleMuon/Run2015D-16Dec2015-v1/MINIAOD)

/SingleMuon/Run2015D-16Dec2015-v1/MINIAOD, CMS collaboration

Cite as: CMS collaboration (2021). SingleMuon primary dataset in MINIAOD format from RunD of 2015 (/SingleMuon/Run2015D-16Dec2015-v1/MINIAOD). CERN Open Data Portal. DOI:10.7483/OPENDATA.CMS.1LUB.Y1DH

Dataset Collision CMS 13TeV CERN-LHC

Description

SingleMuon primary dataset in MINIAOD format from RunD of 2015. Run period from run number 256630 to 260627.

The list of validated runs, which must be applied to all analyses, either with the full validation or for an analysis requiring only muons, can be found in:

Validated runs, full validation

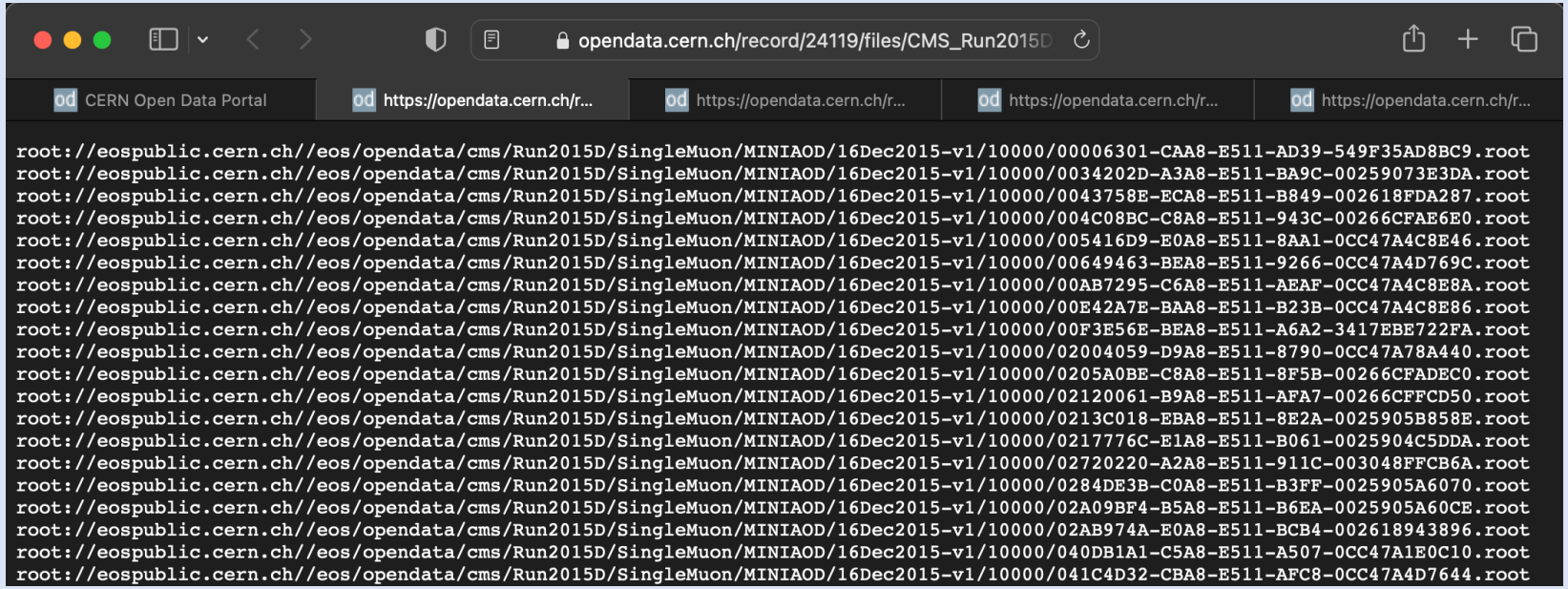
Validated runs, muons only

Related datasets

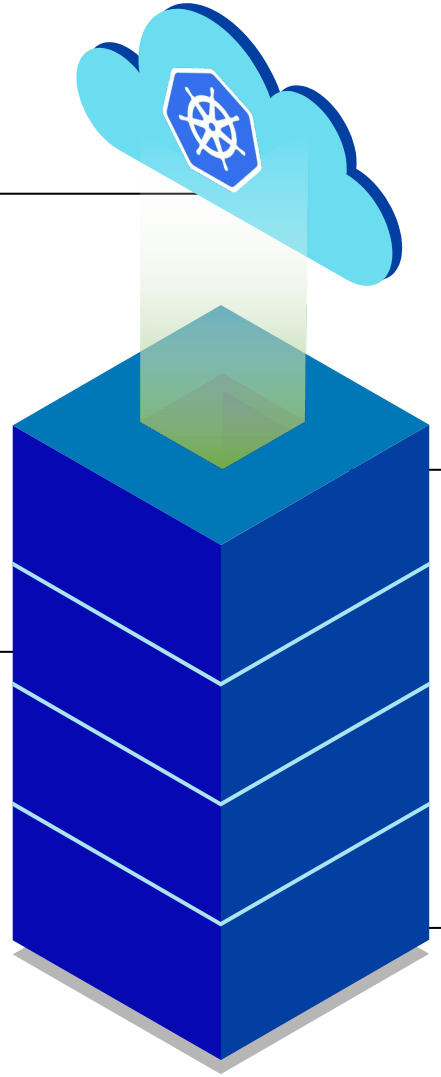
The corresponding AOD dataset:
/SingleMuon/Run2015D-16Dec2015-v1/AOD

File Indexes

Filename	Size	
CMS_Run2015D_SingleMuon_MINIAOD_16Dec2015-v1_10000_file_index.txt	128.8 kB	List Files Download
CMS_Run2015D_SingleMuon_MINIAOD_16Dec2015-v1_10001_file_index.txt	101.6 kB	List Files Download
CMS_Run2015D_SingleMuon_MINIAOD_16Dec2015-v1_20000_file_index.txt	792.0 bytes	List Files Download
CMS_Run2015D_SingleMuon_MINIAOD_16Dec2015-v1_60000_file_index.txt	15.9 kB	List Files Download



Kubernetes Components



Namespace

Kubernetes feature for logical resource separation and management.

Storage Volume

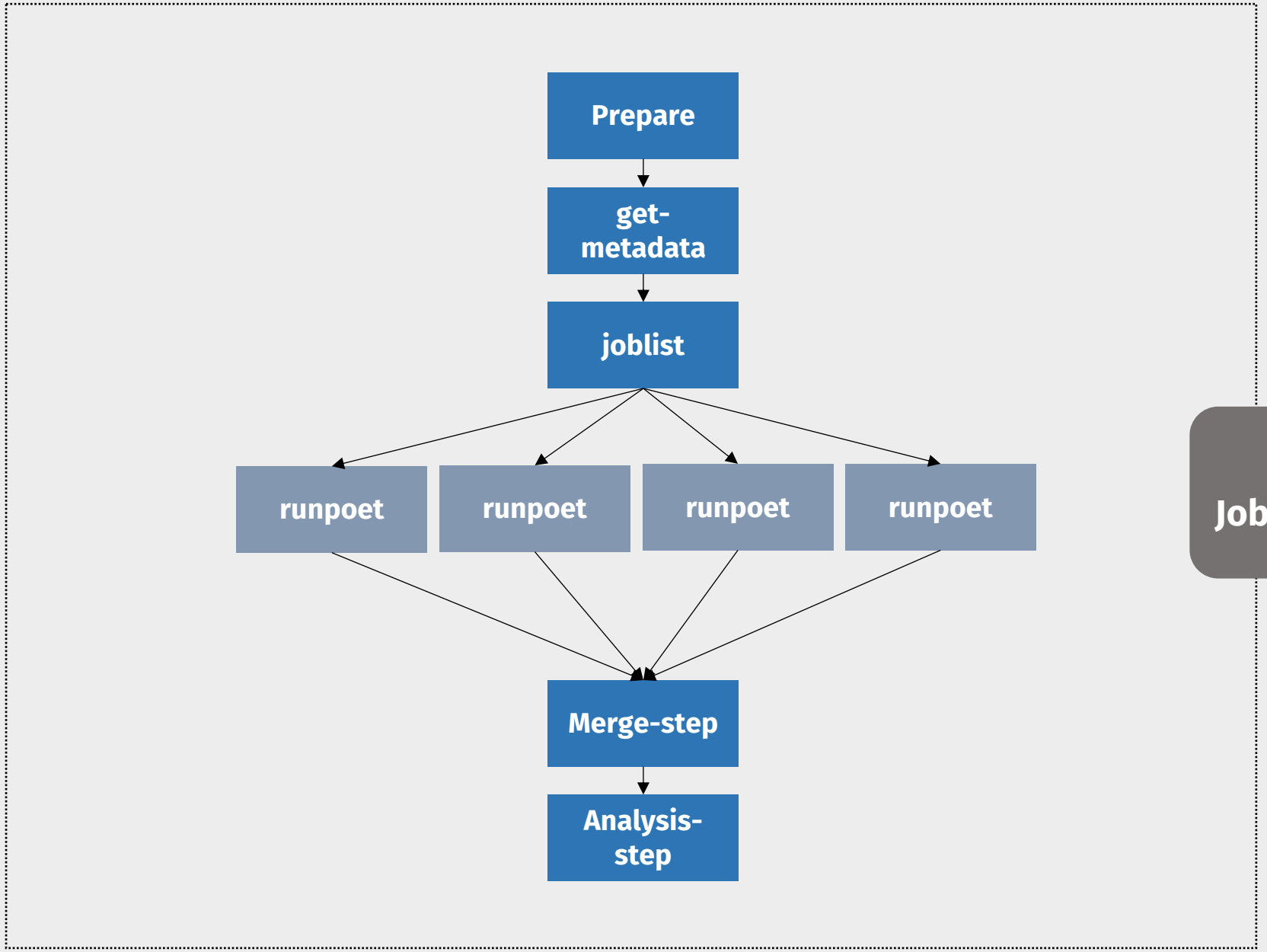
Persistent storage for containers in Kubernetes pods.

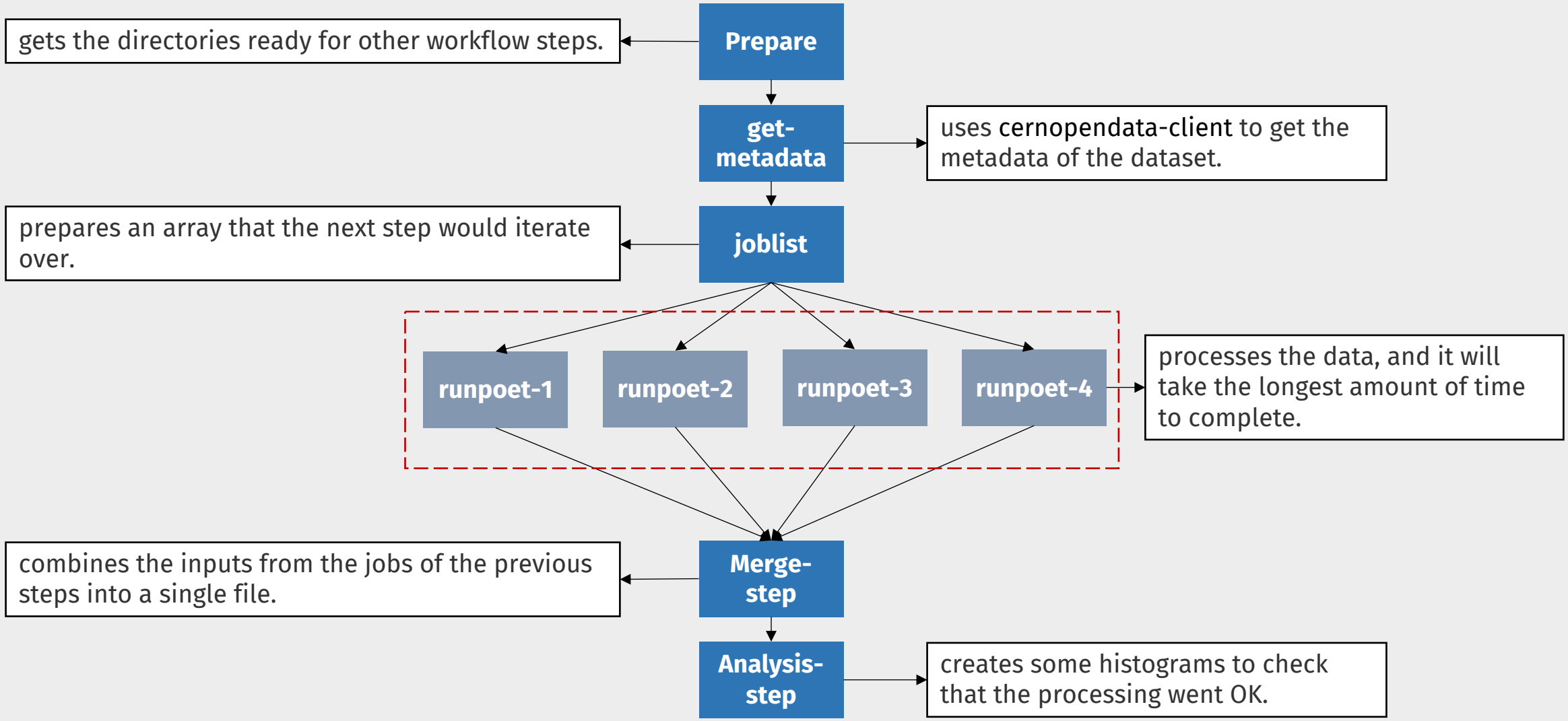
Http Server

Application or service that handles HTTP requests and responses in Kubernetes.

Pod

Is a deployment unit for one or more containers. It enables containers to share resources and network within a cluster.


Prepare
get-metadata
joblist
runpoet
runpoet
runpoet
runpoet
Merge-step
Analysis-step
Argo
Job Controller


gets the directories ready for other workflow steps.
Prepare
get-metadata
uses cernopendata-client to get the metadata of the dataset.
prepares an array that the next step would iterate over.
joblist
runpoet-1
runpoet-2
runpoet-3
runpoet-4
processes the data, and it will take the longest amount of time to complete.
combines the inputs from the jobs of the previous steps into a single file.
Merge-step
Analysis-step
creates some histograms to check that the processing went OK.