JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN

de.NBI
GERMAN NETWORK FOR BIOINFORMATICS INFRASTRUCTURE

CLOUD USER 24 MEETING

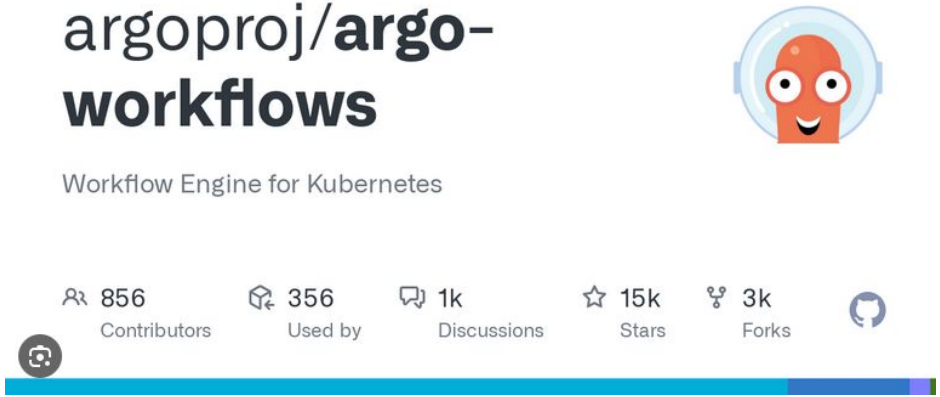
Advanced Kubernetes
Day 1: Workflows on K8s

Sebastian Beyvers | Bioinformatics & Systems Biology | JLU Giessen
sebastian.beyvers@cb.jlug.de

04.12.2024

Cloud Native Workflow Engines

- Kubernetes native workflow engine
- Uses custom resources and the “operator pattern”
- Containerized steps
- DAG definition via yamls
- Built-in scripting support
- Can create other K8s resources
- Data science focused Python library (hera)



Simple example

Containers

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: hello-world-  # Name of this Workflow
spec:
  entrypoint: hello-world     # Defines "hello-world" as the "main" template
  templates:
  - name: hello-world         # Defines the "hello-world" template
    container:
      image: busybox
      command: [echo]
      args: ["hello world"]   # This template runs "echo" in the "busybox" image w
```

Scripts

```yaml
- name: gen-random-int
  script:
    image: python:alpine3.6
    command: [python]
    source: |
      import random
      i = random.randint(1, 100)
      print(i)
```

```yaml
- name: k8s-owner-reference
  resource:
    action: create
    manifest: |
      apiVersion: v1
      kind: ConfigMap
      metadata:
        generateName: owned-eg-
      data:
        some: value
```

Resources

Templates

Step based workflows

```yaml
- name: hello-hello-hello
  steps:
  - - name: step1
      template: prepare-data
  - - name: step2a
      template: run-data-first-half
    - name: step2b
      template: run-data-second-half
```

DAG based workflow

```yaml
- name: diamond
  dag:
    tasks:
    - name: A
      template: echo
    - name: B
      dependencies: [A]
      template: echo
    - name: C
      dependencies: [A]
      template: echo
    - name: D
      dependencies: [B, C]
      template: echo
```

Nextflow on Kubernetes: Old way

- Shared volume needed for nextflow work directory
- All steps must be a container
- Easiest to spawn from inside the cluster

```
process {
    executor = 'k8s'
}

k8s {
    storageClaimName = 'vol-claim'
    storageMountPath = '/mount/path'
    storageSubPath = '/my-data'
}
```

Nextflow config

Nextflow on Kubernetes

- New:
  - Uses a fuse s3 adapter
    - Needs *privileged* permissions / or a custom controller
  - requires S3 configuration
  - Uses “weave” containers that can be automatically built based on conda environments

```
wave {
    enabled = true
}

fusion {
    enabled = true
}

process {
    executor = 'k8s'
}

k8s {
    context = '<YOUR K8S CONFIGURATION CONTEXT>'
    namespace = '<YOUR K8S NAMESPACE>'
    serviceAccount = '<YOUR K8S SERVICE ACCOUNT>'
}
```

Hands on !

Snakemake on Kubernetes

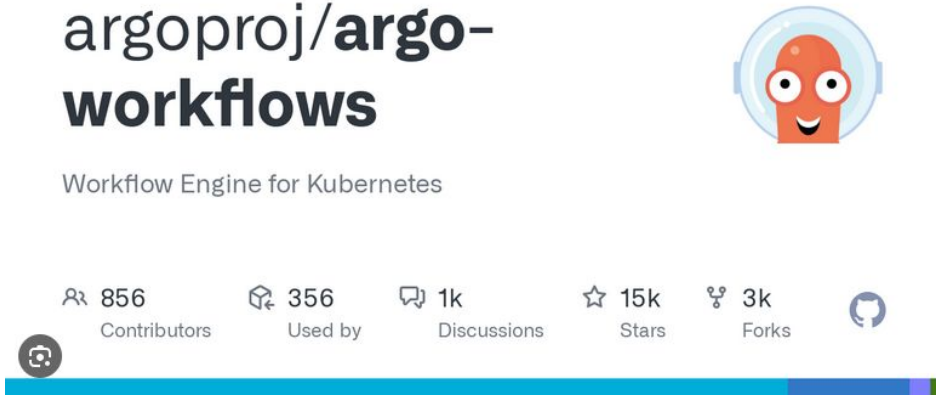
- Needs an external storage setup (like S3)
- Preferably needs pre-configured workflows from git
- All intermediate results will be put into S3 storage
- Built-in conda support to create environments on the fly
- Unfortunately: Very sparsely documented

snakemake

Hands on !

A full service demonstration

- Kubernetes native workflow engine
- Uses custom resources and the “operator pattern”
- Containerized steps
- DAG definition via yamls
- Built-in scripting support
- Can create other K8s resources
- Data science focused Python library (hera)



Questions ?