aws

INNOVATE

DATA AND AI/ML EDITION

22 February 2023

aws INNOVATE
DATA AND AI/ML EDITION

Sustainable and scalable machine learning with Amazon EKS and Argo Workflows

Mitch Beaumont

Principal Solutions Architect
Amazon Web Services

Agenda

- Introduction to Amazon EKS
- Quick overview to machine learning pipelines
- An introduction to Argo Workflows
- Installing Argo Workflow
- Creating workflows using Argo
- Wrap-up

Introduction to Amazon EKS

We heard from customers that running Kubernetes increased their operational overhead

Running and scaling Kubernetes is difficult and requires significant investment

Securing Kubernetes increased the operational overhead of running applications

Applications needed a native way to integrate with other AWS services securely and reliably

All of this extra work was leading to a lot of undifferentiated heavy lifting

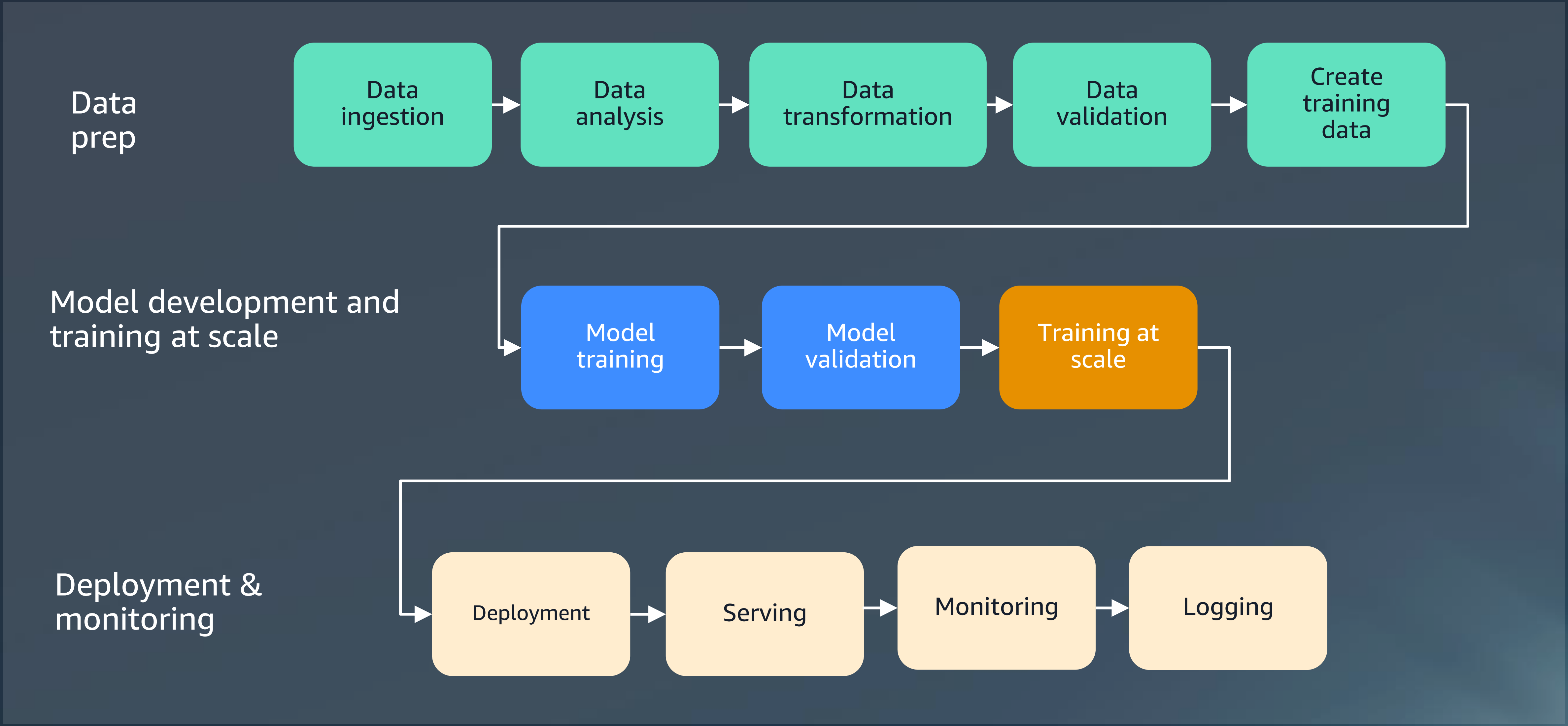
Introduction to Amazon EKS

Amazon Elastic Kubernetes Service (Amazon EKS)

- ✓ Amazon EKS runs vanilla Kubernetes. Amazon EKS is upstream and certified conformant version of Kubernetes (with backported security fixes)
- ✓ Amazon EKS supports four versions of Kubernetes, giving customers time to test and roll out upgrades
- ✓ Amazon EKS provides a managed Kubernetes experience for performant, reliable, and secure Kubernetes
- ✓ Amazon EKS makes Kubernetes operations, administration, and management simple and boring

Amazon EKS enables you to build reliable, stable, and secure applications in any environment

What is a machine learning pipelines?



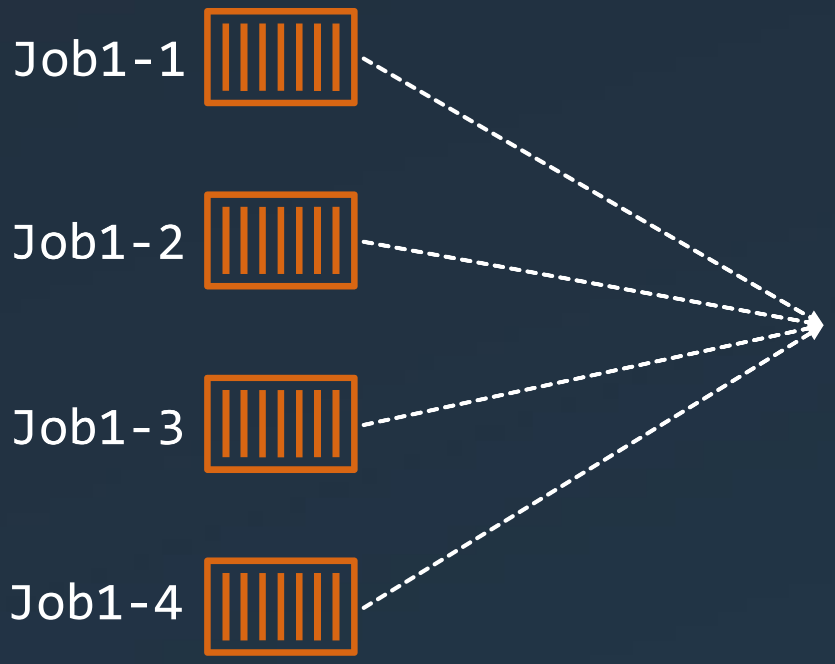
What are Argo Workflows?

- Define and executes complex container-based workflows on Kubernetes
- Workflows are a collection of tasks, executed in a specific order, with the ability to retry tasks on failure, chain tasks in parallel and specify dependencies between different tasks
- Implemented a as Kubernetes CRD, or custom resource definitions) and is built on top of Kubernetes primitives like pods and containers

Doesn’t Kubernetes have Jobs?

Amazon **EKS**

```
kubectl apply -f whalesay-hello-k8s-job.yaml
```

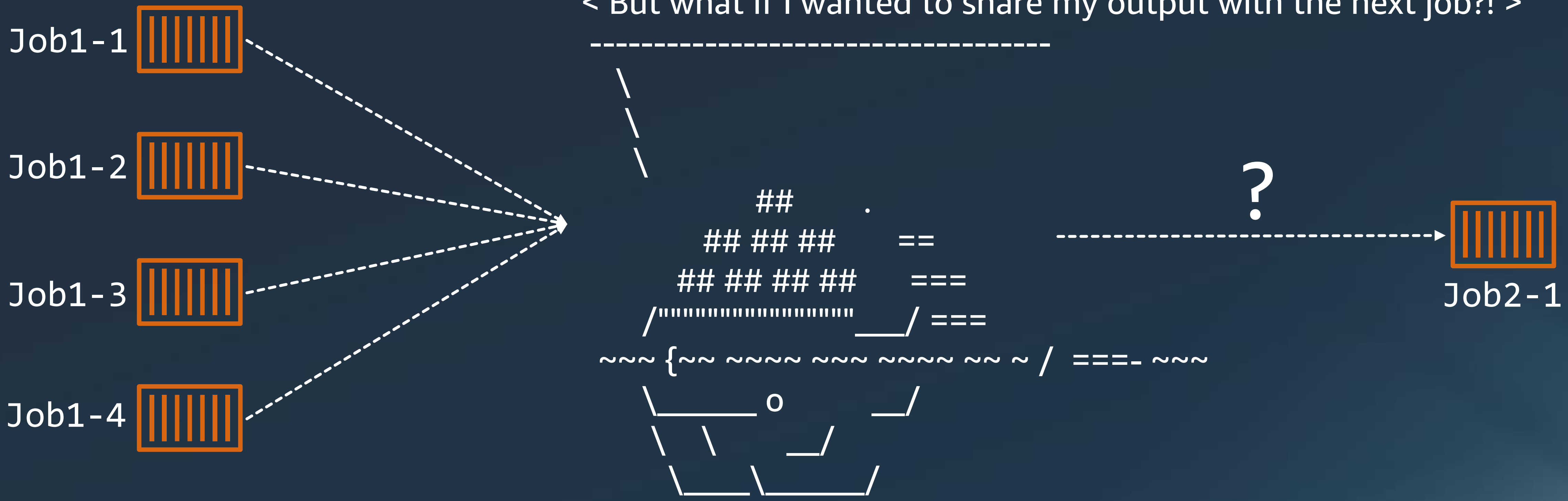


```
 _________________________________________
< Hello from a K8s job running on AWS! >
 -----------------------------------------
    \
     \
      \
                    ##        .
              ## ## ##       ==
           ## ## ## ##      ===
       /""""""""""""""""___/ ===
  ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
       \______ o          __/
        \    \        __/
          \____\______/
```

Doesn’t Kubernetes have Jobs?

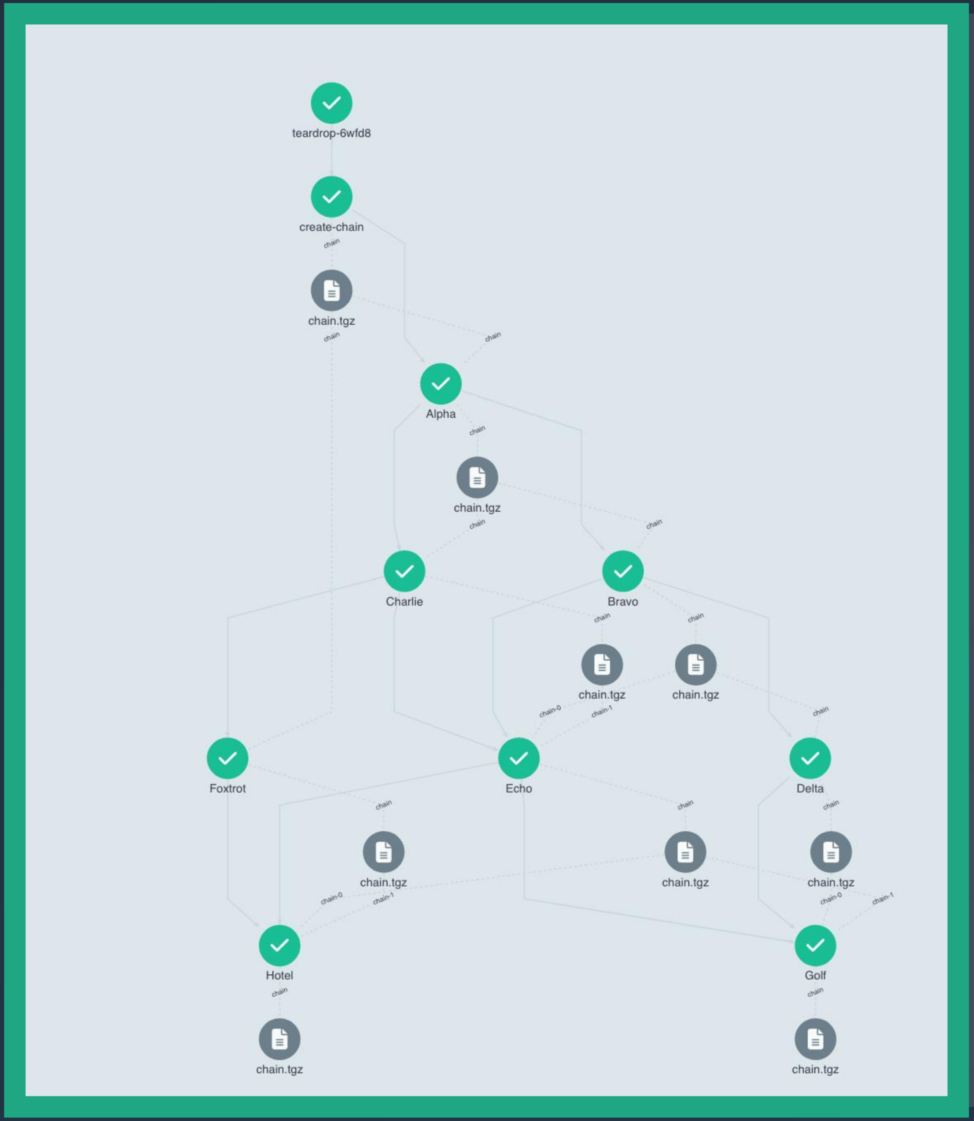
Amazon Elastic Kubernetes Service (Amazon EKS)

```
kubectl apply –f whalesay-hello-k8s-job.yaml
```



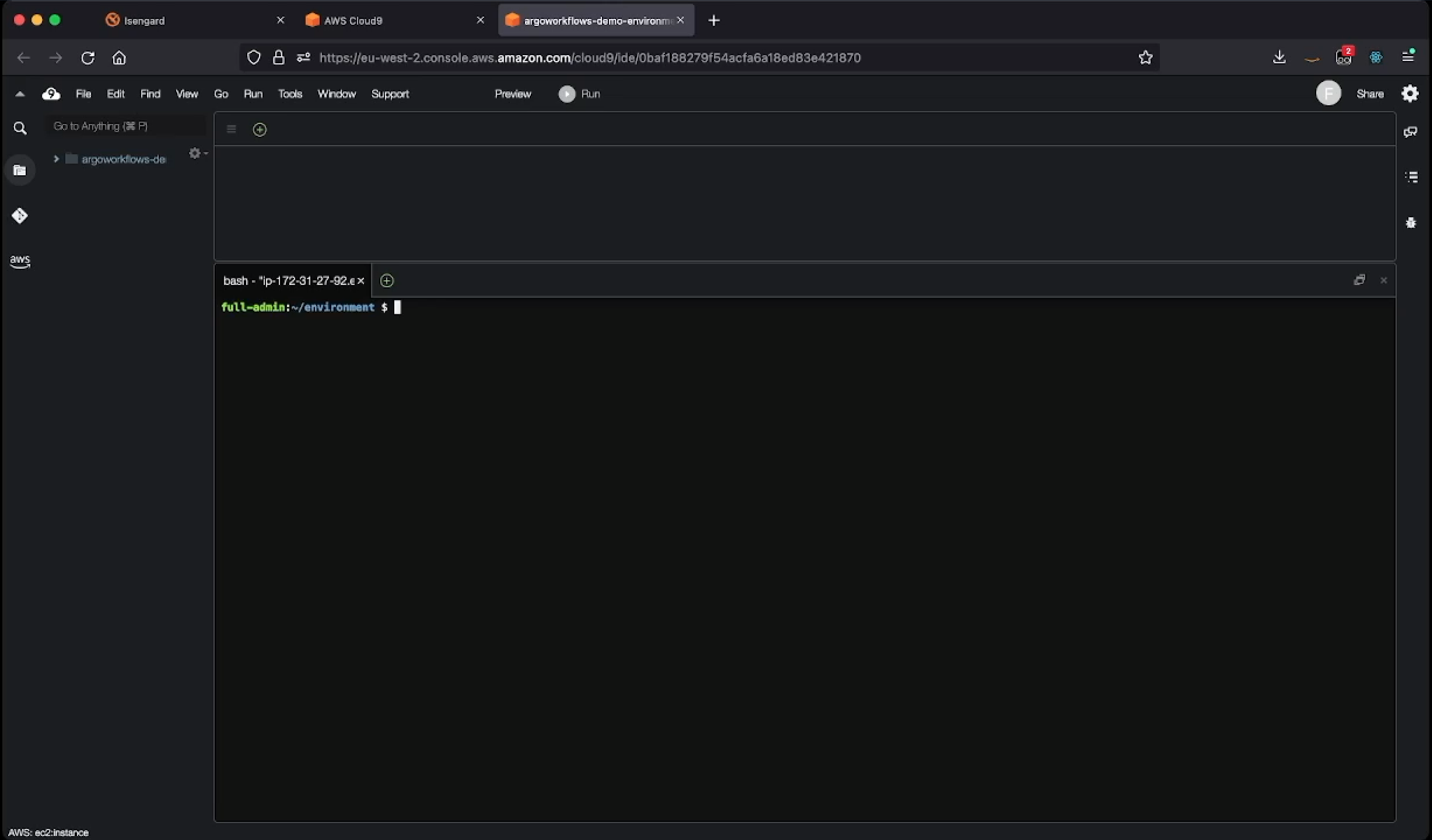
```
 _________________________________________
< But what if I wanted to share my output with the next job?! >
 -----------------------------------------
    \
     \
      \
                    ##        .
              ## ## ##       ==
           ## ## ## ##      ===
       /""""""""""""""""___/ ===
  ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
       \______ o          __/
        \    \        __/
          \____\______/
```

What is a DAG workflow?



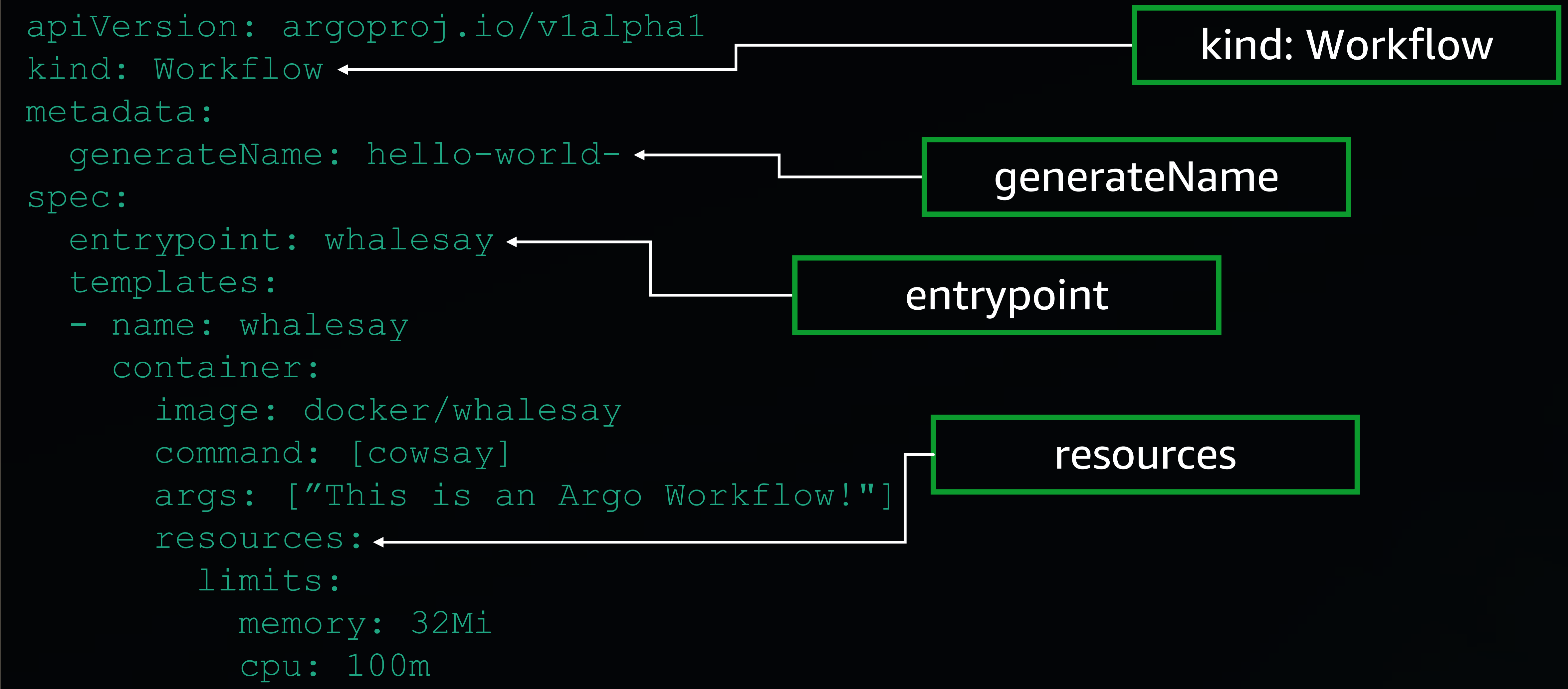
- Type of workflow that allows you to define a set of tasks and the dependencies between them.
- Tasks in a DAG are organized into a graph, where the edges between the tasks represent the dependencies.
- The graph is acyclic, and so there are no cycles or loops in the workflow, which means that the tasks will always execute in a well-defined order.

Installing Argo Workflows …


Isengard
AWS Cloud9
argoworkflows-demo-environm
https://eu-west-2.console.aws.amazon.com/cloud9/ide/0baf188279f54acfa6a18ed83e421870
File
Edit
Find
View
Go
Run
Tools
Window
Support
Preview
Run
Share
Go to Anything (⌘ P)
argoworkflows-de
bash - "ip-172-31-27-92.e
AWS: ec2:instance

```
full-admin:~/environment $
```

The anatomy of a workflow …



```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: hello-world-
spec:
  entrypoint: whalesay
  templates:
  - name: whalesay
    container:
      image: docker/whalesay
      command: [cowsay]
      args: [”This is an Argo Workflow!"]
      resources:
        limits:
          memory: 32Mi
          cpu: 100m
```

kind: Workflow

generateName

entrypoint

resources

Building a workflow (step-by-step)

Creating a workflow … [STEP 1]

1

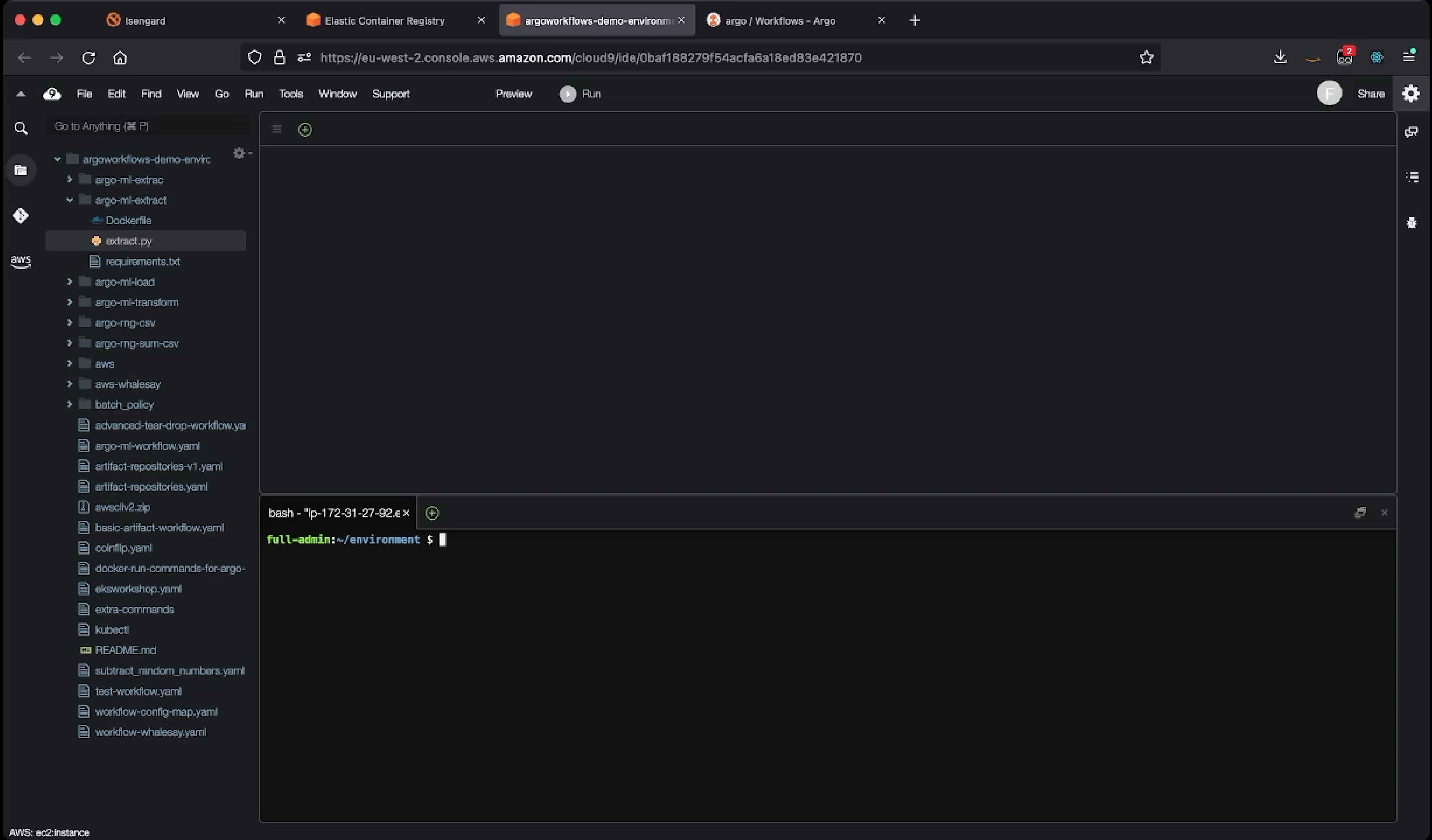
```
import requests
from datetime import datetime, timedelta
import json

# OpenAQ base URL.
BASE_URL = "https://api.openaq.org"

def get_historical_data(start ...
```

Download historical air quality data from OpenAQ API and save the results as an output …

2

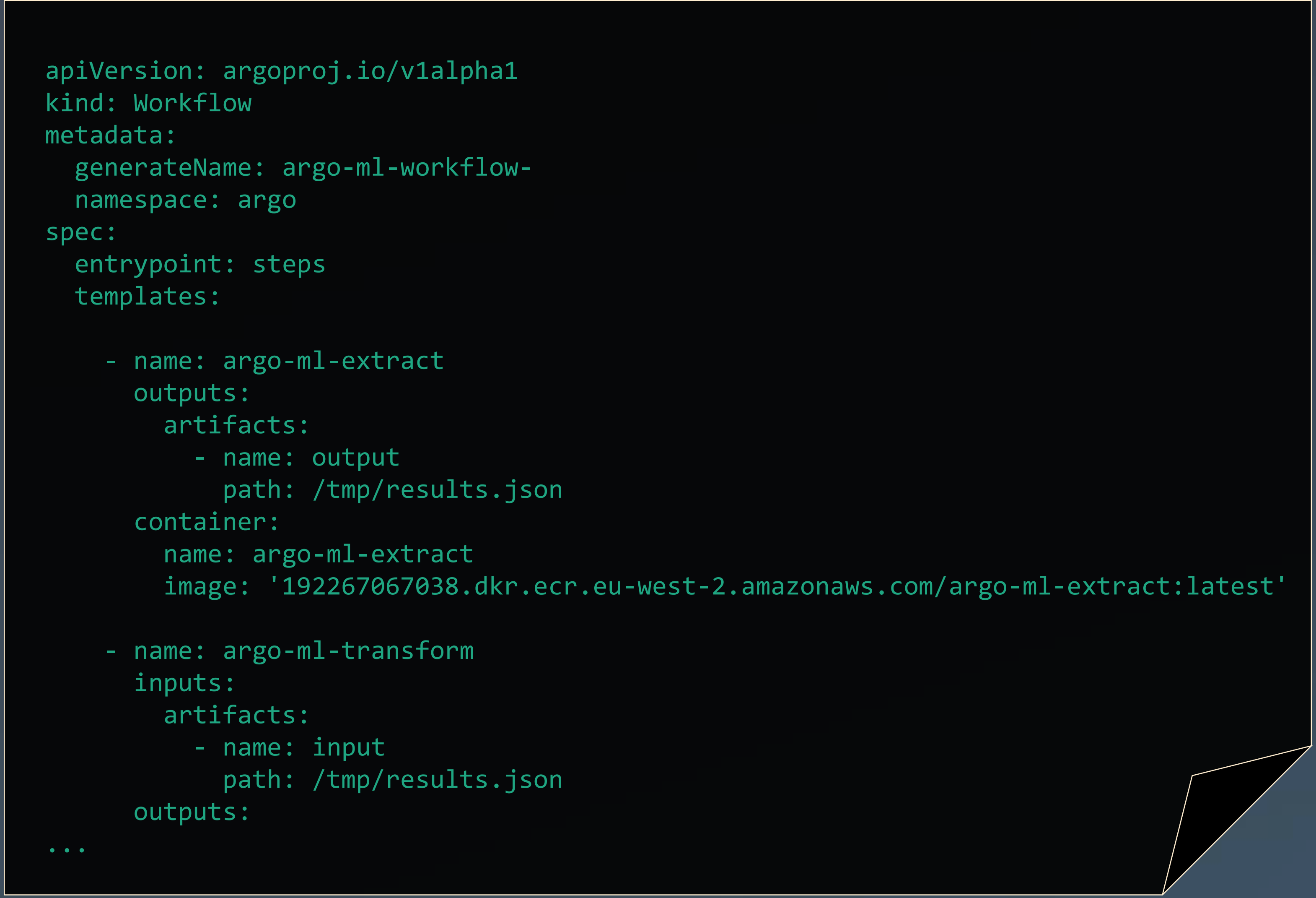
```
import pandas as pd
import json

def main():
    with open('/tmp/results.json', 'r') as f:
        response_data = json.load(f)

    df = pd.json_normalize(response_data)

...
```

Load the JSON results and transform them in to parquet format. Save the resulting parquet file to the artifact store …

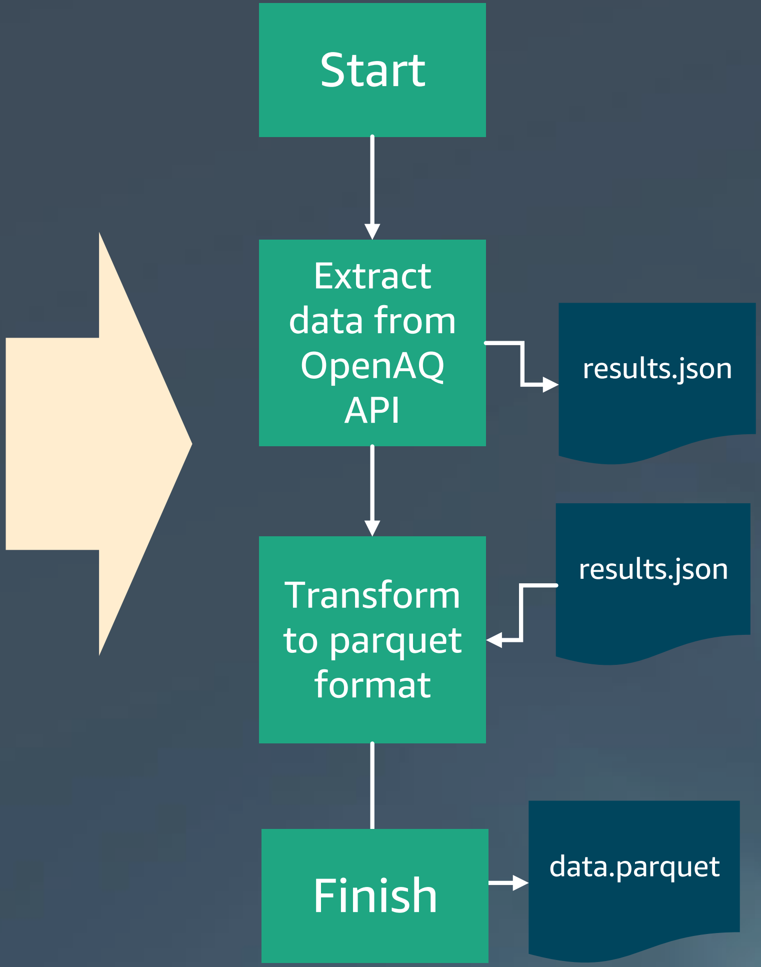


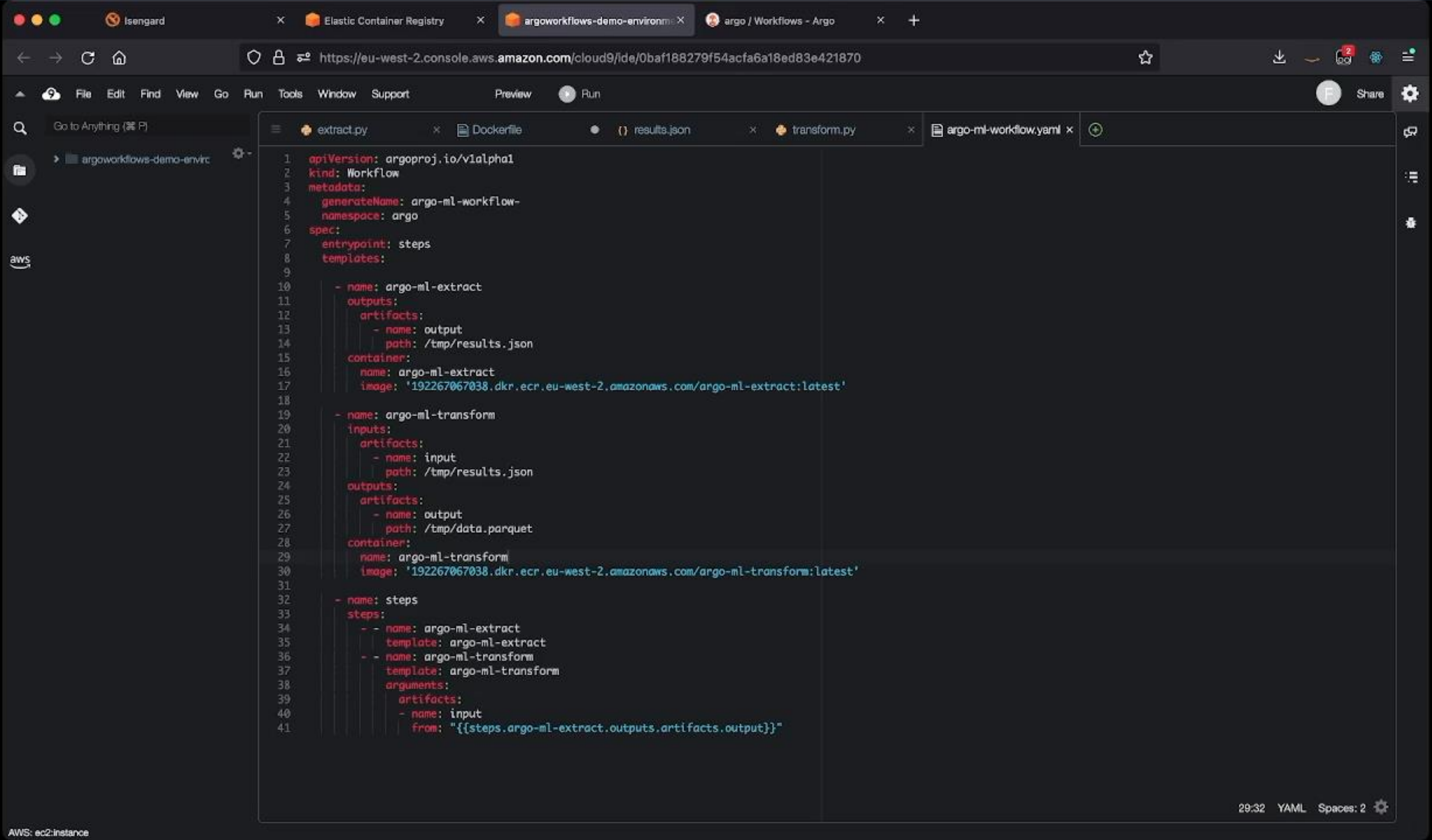
```
full-admin:~/environment $ 
```

Creating a workflow … [STEP 2]



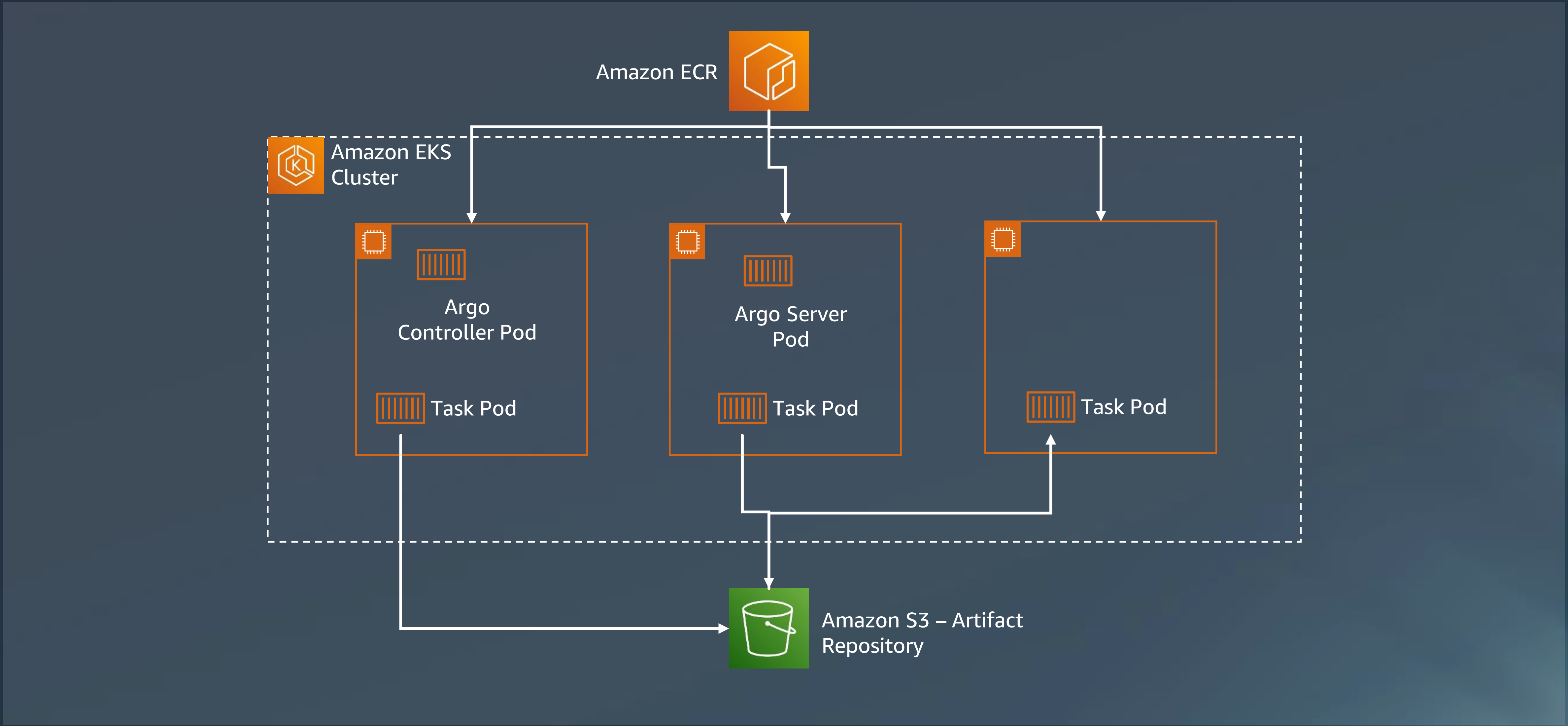
```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: argo-ml-workflow-
  namespace: argo
spec:
  entrypoint: steps
  templates:

    - name: argo-ml-extract
      outputs:
        artifacts:
          - name: output
            path: /tmp/results.json
      container:
        name: argo-ml-extract
        image: '192267067038.dkr.ecr.eu-west-2.amazonaws.com/argo-ml-extract:latest'

    - name: argo-ml-transform
      inputs:
        artifacts:
          - name: input
            path: /tmp/results.json
      outputs:
...
```



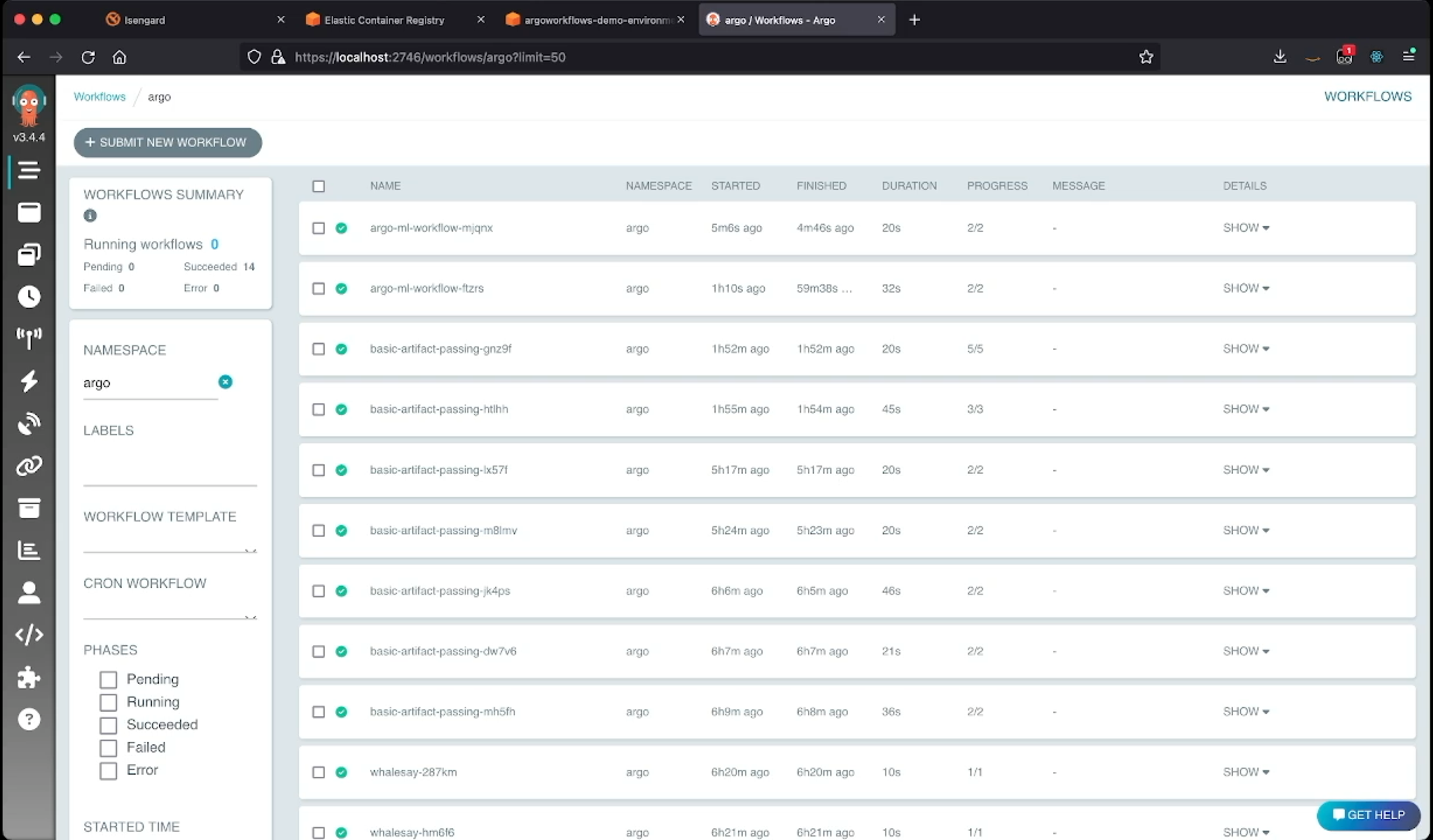


```yaml
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: argo-ml-workflow-
  namespace: argo
spec:
  entrypoint: steps
  templates:

    - name: argo-ml-extract
      outputs:
        artifacts:
          - name: output
            path: /tmp/results.json
      container:
        name: argo-ml-extract
        image: '192267067038.dkr.ecr.eu-west-2.amazonaws.com/argo-ml-extract:latest'

    - name: argo-ml-transform
      inputs:
        artifacts:
          - name: input
            path: /tmp/results.json
      outputs:
        artifacts:
          - name: output
            path: /tmp/data.parquet
      container:
        name: argo-ml-transform
        image: '192267067038.dkr.ecr.eu-west-2.amazonaws.com/argo-ml-transform:latest'

    - name: steps
      steps:
        - - name: argo-ml-extract
            template: argo-ml-extract
        - - name: argo-ml-transform
            template: argo-ml-transform
            arguments:
              artifacts:
              - name: input
                from: "{{steps.argo-ml-extract.outputs.artifacts.output}}"
```

What does this look like under-the-hood?



Reviewing our workflow … [STEP 3]


Isengard
Elastic Container Registry
argoworkflows-demo-environm
argo / Workflows - Argo
https://localhost:2746/workflows/argo?limit=50
v3.4.4
Workflows / argo
WORKFLOWS
+ SUBMIT NEW WORKFLOW
GET HELP

WORKFLOWS SUMMARY

Running workflows 0

Pending 0 Succeeded 14

Failed 0 Error 0

NAMESPACE

argo

LABELS

WORKFLOW TEMPLATE

CRON WORKFLOW

PHASES

□ Pending
□ Running
□ Succeeded
□ Failed
□ Error

STARTED TIME

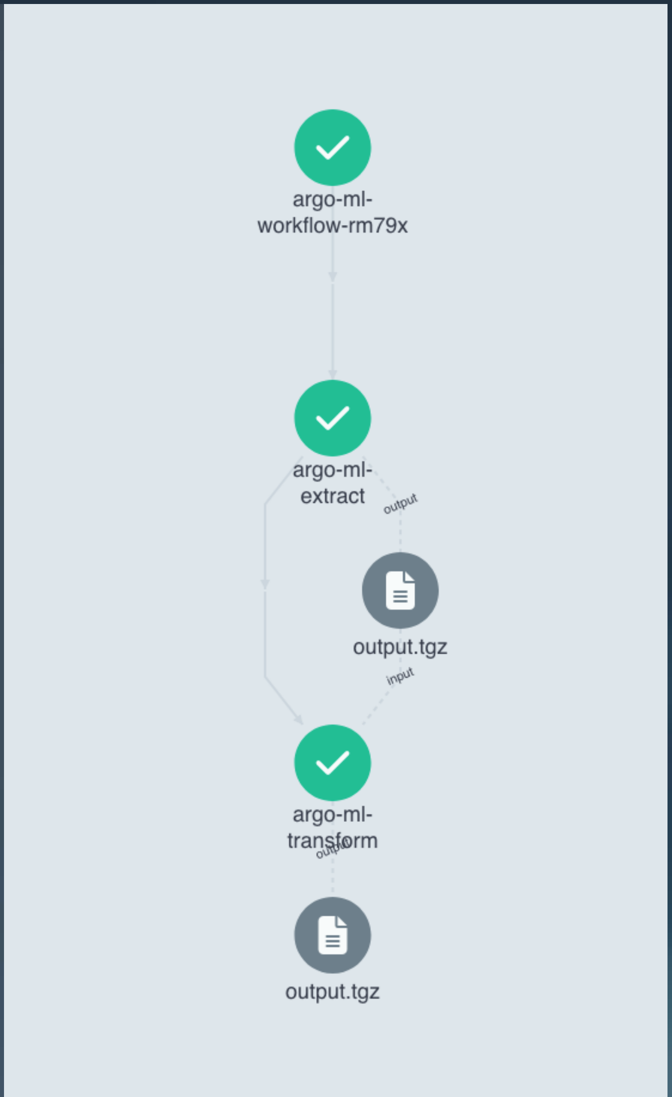
□		NAME	NAMESPACE	STARTED	FINISHED	DURATION	PROGRESS	MESSAGE	DETAILS
□	✓	argo-ml-workflow-mjqnx	argo	5m6s ago	4m46s ago	20s	2/2	-	SHOW ▾
□	✓	argo-ml-workflow-ftzrs	argo	1h10s ago	59m38s ...	32s	2/2	-	SHOW ▾
□	✓	basic-artifact-passing-gnz9f	argo	1h52m ago	1h52m ago	20s	5/5	-	SHOW ▾
□	✓	basic-artifact-passing-htlhh	argo	1h55m ago	1h54m ago	45s	3/3	-	SHOW ▾
□	✓	basic-artifact-passing-lx57f	argo	5h17m ago	5h17m ago	20s	2/2	-	SHOW ▾
□	✓	basic-artifact-passing-m8lmv	argo	5h24m ago	5h23m ago	20s	2/2	-	SHOW ▾
□	✓	basic-artifact-passing-jk4ps	argo	6h6m ago	6h5m ago	46s	2/2	-	SHOW ▾
□	✓	basic-artifact-passing-dw7v6	argo	6h7m ago	6h7m ago	21s	2/2	-	SHOW ▾
□	✓	basic-artifact-passing-mh5fh	argo	6h9m ago	6h8m ago	36s	2/2	-	SHOW ▾
□	✓	whalesay-287km	argo	6h20m ago	6h20m ago	10s	1/1	-	SHOW ▾
□	✓	whalesay-hm6f6	argo	6h21m ago	6h21m ago	10s	1/1	-	SHOW ▾

Wrap-up

- Define workflows where each step in the workflow is a container.

- Model multi-step workflows as a sequence of tasks or capture the dependencies between tasks using a directed acyclic graph (DAG).

- Easily run compute intensive jobs for machine learning or data processing in a fraction of the time using Argo Workflows on Kubernetes.



Visit the Data & AI/ML resource hub

Dive deeper into these resources, get inspired and learn how you can use AI and machine learning to accelerate your business outcomes.

- 6 steps to machine learning success e-book
- 7 leading machine learning use cases e-book
- Machine learning at scale e-book
- Achieving transformative business results with machine learning e-book
- Tackling our world’s hardest problems with machine learning e-book
- Accelerating machine learning innovation through security e-book
- … and more!

https://bityl.co/FqdC

Visit resource hub

AWS Training and Certification

Access the AI & ML learning plan courses built by AWS experts on AWS Skill Builder

- Get started with digital self-paced, on-demand training and ramp-up guides to help you grow your technical skills
- Learn how to apply machine learning, artificial intelligence, and deep learning to unlock new insights and value in your role
- Take the steps today, towards validating your expertise with an AWS Certified Machine Learning – Specialty Certification

https://bit.ly/3FnxDH7

Learn your way explore.skillbuilder.aws »

Thank you for attending AWS Innovate – Data & AI/ML Edition

We hope you found it interesting! A kind reminder to **complete the survey.**
Let us know what you thought of today's event and how we can improve the event experience for you in the future.

- aws-apj-marketing@amazon.com
- twitter.com/AWSCloud
- facebook.com/AmazonWebServices
- youtube.com/user/AmazonWebServices
- slideshare.net/AmazonWebServices
- twitch.tv/aws

Thank you!