National College of Ireland

Fault-Tolerant Workflow Scheduling for Microservices in Cloud Environments Under Time and Cost Constraints Configuration Manual

MSc Research Project
Cloud Computing

Yogesh V. Patil
Student ID: X23219203

School of Computing
National College of Ireland

Supervisor: Prof. Abubakr Siddig

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Yogesh Vishwanath Patil		
Student ID:	X23219203		
Programme:	Cloud Computing	**Year:**	2024-2025
Module:	Msc Research Project		
Lecturer:	Prof. Abubakr Siddig		
Submission Due Date:	12 December 2024		
Project Title:	Fault-Tolerant Workflow Scheduling for Microservices in Cloud Environments Under Time and Cost Constraints.		
Word Count:			

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Yogesh Vishwanath Patil
Date:	12 December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Yogesh V Patil
Student ID: X23219203

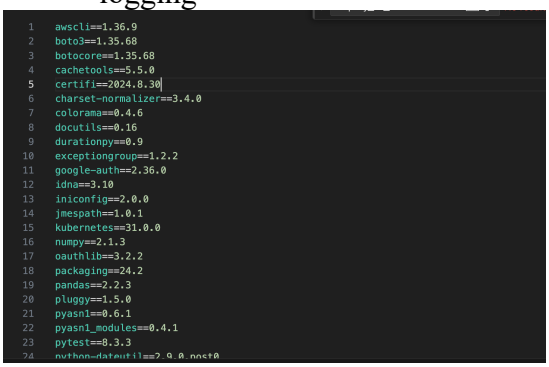
1 Prerequisites

1.1 Hardware Requirements

- Processor: Intel Core i5 or above latest confuguration
- RAM: 16 GB Minimum and max as your needs.
- Storage: SSD disk Preferred with no less than 256GB
- Network: Internet Broadband connection for package and dependency downloads

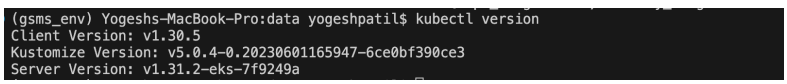
1.2 Python Setup

- Python 3.9 or higher
  - Required Python packages:
  - boto3
  - pandas
  - numpy
  - kubernetes
  - pyyaml
  - logging



1.3 Kubernetes

- Access to a Kubernetes cluster
- kubectl CLI installed and configured
- Metrics server installed on the cluster



1.4 AWS CLI

- AWS credentials with appropriate permissions

- Service account configuration for both AWS and Kubernetes

```
(gsms_env) Yogeshs-MacBook-Pro:data yogeshpatil$ aws sts get-caller-identity
{
    "UserId": "842675988708",
    "Account": "842675988708",
    "Arn": "arn:aws:iam::842675988708:root"
}
```

2 AWS Account Setup

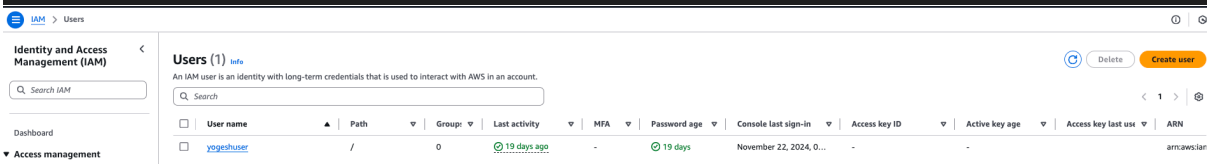
2.1 Create AWS Account

1. Go to AWS Management Console and create personal acc (https://aws.amazon.com)
2. Click "Create an AWS Account"
3. Follow the registration process
4. Select the Free Tier option

2.2 IAM User Configuration

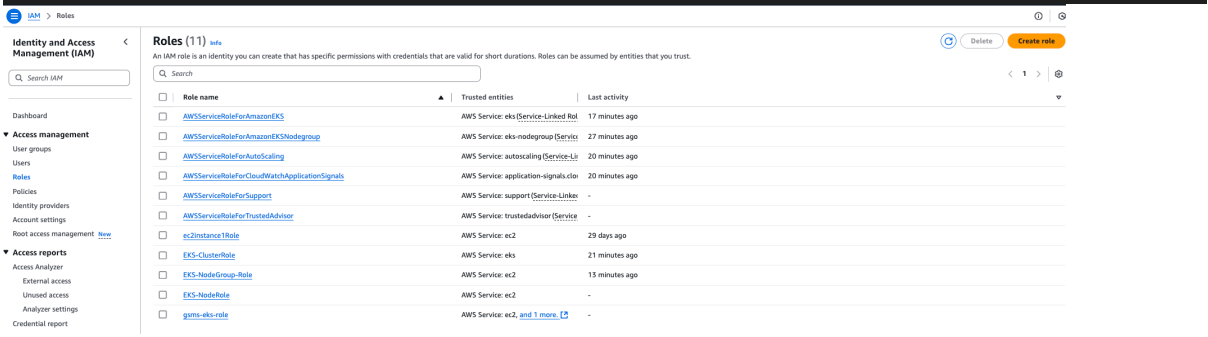
1. Navigate to IAM Console
2. Create a new IAM user role:

```
# 1. IAM Setup
# Create IAM user
aws iam create-user --user-name yogeshuser
```



3. Attach required policies:

```
# Attach required policies
aws iam attach-user-policy --user-name yogeshuser --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
aws iam attach-user-policy --user-name yogeshuser --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess
aws iam attach-user-policy --user-name yogeshuser --policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
```



2.3 Network Infrastructure

1. **Create Virtual Private Cloud:**

```
VPC_ID=$(aws ec2 create-vpc \
    --cidr-block 10.0.0.0/16 \
    --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=gsms-eks-vpc}]' \
    --query 'Vpc.VpcId' \
    --output text)
```

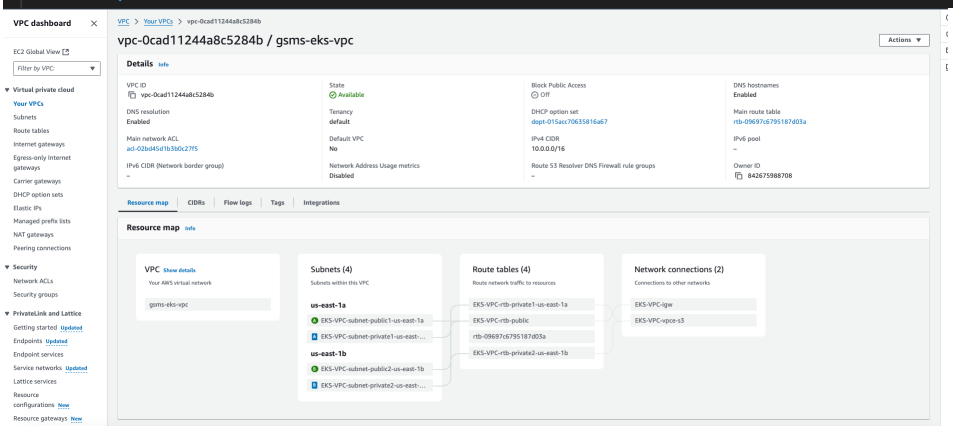
2. Create Subnets:

Public Subnet

```
PUBLIC_SUBNET_ID=$(aws ec2 create-subnet \
    --vpc-id $VPC_ID \
    --cidr-block 10.0.1.0/24 \
    --availability-zone us-east-1a \
    --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=gsms-public-1a}]' \
    --query 'Subnet.SubnetId' \
    --output text)
```

Private Subnet

```
PRIVATE_SUBNET_ID=$(aws ec2 create-subnet \
    --vpc-id $VPC_ID \
    --cidr-block 10.0.2.0/24 \
    --availability-zone us-east-1b \
    --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=gsms-private-1b}]' \
    --query 'Subnet.SubnetId' \
    --output text)
```



3. VPC Internet Gateways Configuration

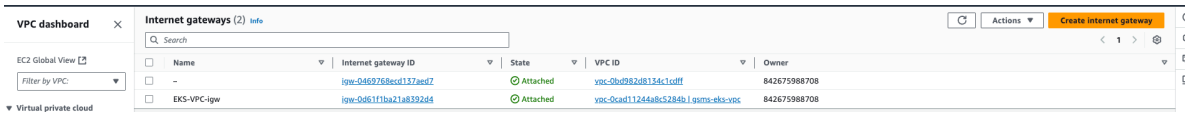
```
IGW_ID=$(aws ec2 create-internet-gateway \
    --tag-specifications 'ResourceType=internet-gateway,Tags=[{Key=Name,Value=gsms-vpc-igw}]' \
    --query 'InternetGateway.InternetGatewayId' \
    --output text)

aws ec2 attach-internet-gateway \
    --vpc-id $VPC_ID \
    --internet-gateway-id $IGW_ID
```

4. NAT Gateway Configuration

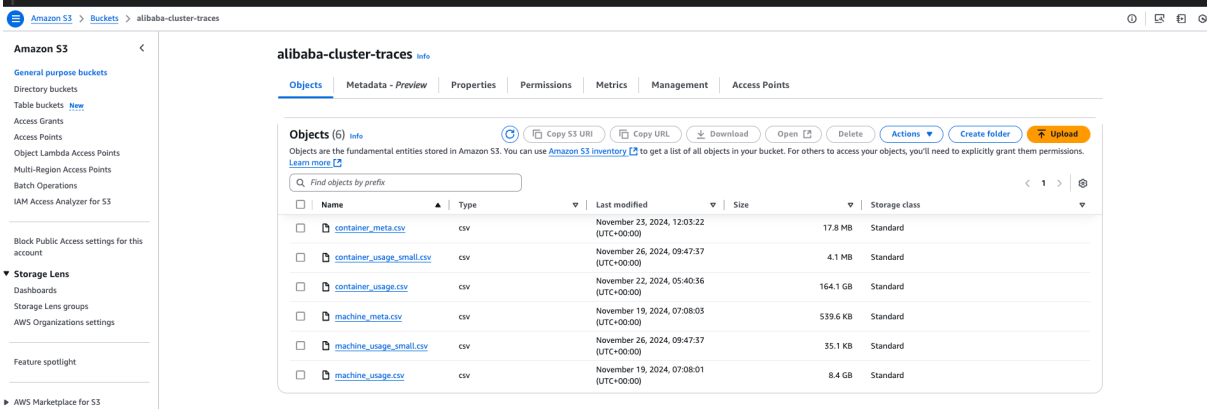
```
EIP_ID=$(aws ec2 allocate-address --domain vpc --query 'AllocationId' --output text)

NAT_GATEWAY_ID=$(aws ec2 create-nat-gateway \
    --subnet-id $PUBLIC_SUBNET_ID \
    --allocation-id $EIP_ID \
    --tag-specifications 'ResourceType=natgateway,Tags=[{Key=Name,Value=gsms-vpc-ngw}]' \
    --query 'NatGateway.NatGatewayId' \
    --output text)
```



2.4 S3 Bucket Creation

```
aws s3api create-bucket \
    --bucket alibaba-cluster-traces \
    --region us-east-1
```



2.5 Security Configuration

1. Create EKS Cluster Security Group:

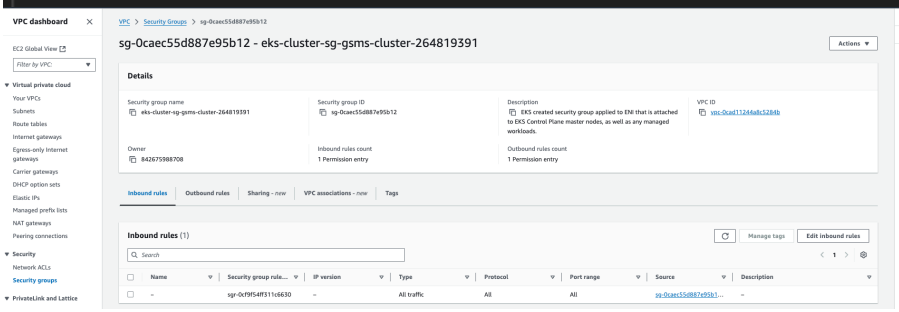
```
CLUSTER_SG_ID=$(aws ec2 create-security-group \
    --group-name eks-cluster-sg \
    --description "Security group for EKS cluster" \
    --vpc-id $VPC_ID \
    --query 'GroupId' \
    --output text)

aws ec2 authorize-security-group-ingress \
    --group-id $CLUSTER_SG_ID \
    --protocol tcp \
    --port 443 \
    --cidr 0.0.0.0/0
```

2. Create Node Security Group:

```
NODE_SG_ID=$(aws ec2 create-security-group \
    --group-name eks-node-sg \
    --description "Security group for EKS nodes" \
    --vpc-id $VPC_ID \
    --query 'GroupId' \
    --output text)

aws ec2 authorize-security-group-ingress \
    --group-id $NODE_SG_ID \
    --protocol tcp \
    --port 10250 \
    --source-group $CLUSTER_SG_ID
```

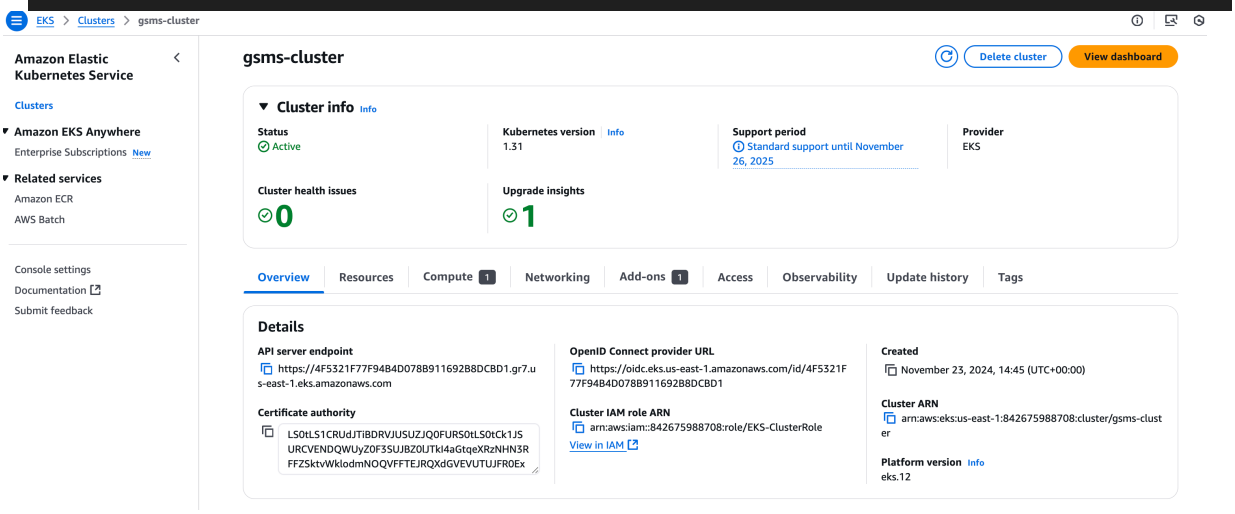


2.6 EKS Cluster Setup

1. Create Cluster:

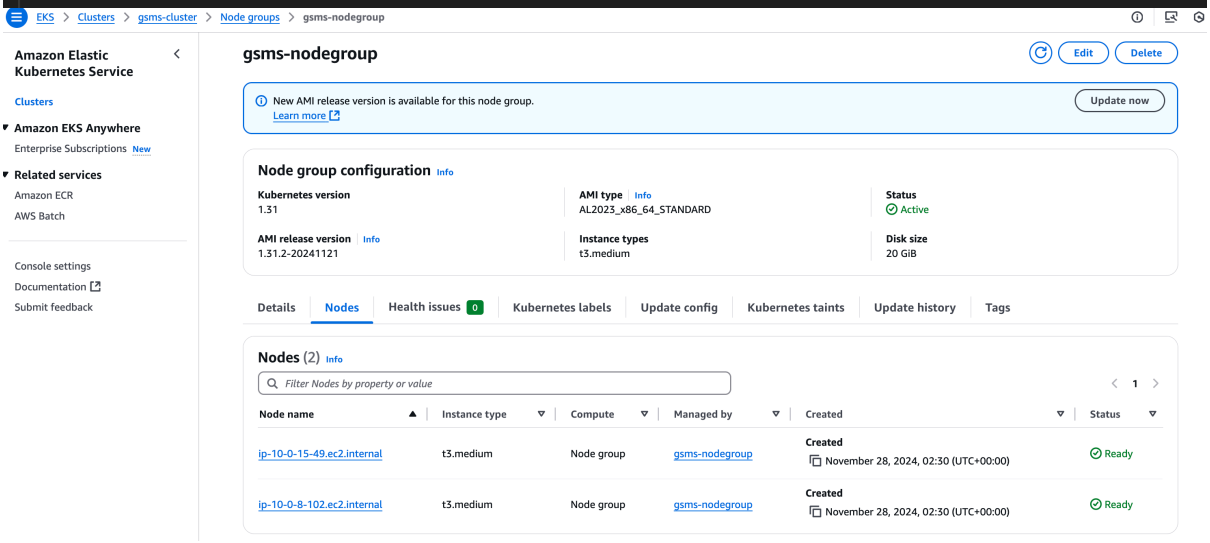
```
aws eks create-cluster \
    --name gsms-cluster \
    --role-arn arn:aws:iam::842675988708:role/EKS-ClusterRole \
    --resources-vpc-config subnetIds=$PUBLIC_SUBNET_ID,$PRIVATE_SUBNET_ID,securityGroupIds=$CLUSTER_SG_ID \
    --kubernetes-version 1.24

# Wait for cluster to be active
aws eks wait cluster-active --name gsms-cluster
```



2. Configure Node Groups

```
aws eks create-nodegroup \
    --cluster-name gsms-cluster \
    --nodegroup-name gsms-nodes \
    --scaling-config minSize=2,maxSize=4,desiredSize=2 \
    --instance-types t3.medium \
    --subnets $PRIVATE_SUBNET_ID \
    --node-role arn:aws:iam::842675988708:role/EKS-NodeRole
```



2.7 CloudWatch Monitoring

```
aws eks update-cluster-config \
    --name gsms-cluster \
    --logging '{"clusterLogging":[{"types":["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

2.8 Access Configuration

1. Configure kubectl

```
aws eks update-kubeconfig --name gsms-cluster --region us-east-1
```

2. Test Connection

Check VPC status
Verify EKS cluster status

```
# Check Cluster
aws eks describe-cluster --name gsms-cluster

# Check Node Group
aws eks describe-nodegroup --cluster-name gsms-cluster --nodegroup-name gsms-nodes
```

```
(gsms_env) Yogeshs-MacBook-Pro:data yogeshpatil$ aws eks describe-cluster --name gsms-cluster
{
    "cluster": {
        "name": "gsms-cluster",
        "arn": "arn:aws:eks:us-east-1:842675988708:cluster/gsms-cluster",
        "createdAt": 1732373122.879,
        "version": "1.31",
        "endpoint": "https://4F5321F77F94B4D078B911692B8DCBD1.gr7.us-east-1.eks.amazonaws.com",
        "roleArn": "arn:aws:iam::842675988708:role/EKS-ClusterRole",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-05d80a86e1f445056",
                "subnet-0e30d8e16372717bd",
                "subnet-0dca1649fbcec1b3d",
                "subnet-020715d70d67ec758"
            ],
            "securityGroupIds": [
                "sg-04d1fd43422fcedc3"
            ],
            "clusterSecurityGroupId": "sg-0caec55d887e95b12",
            "vpcId": "vpc-0cad11244a8c5284b",
            "endpointPublicAccess": true,
            "endpointPrivateAccess": true,
            "publicAccessCidrs": [
                "0.0.0.0/0"
            ]
        },
```

Check node group status

```
aws eks describe-nodegroup --cluster-name gsms-cluster --nodegroup-name gsms-nodes
```

#Pod Status

```
# Check Kubernetes Components
kubectl get nodes
kubectl get pods --all-namespaces
```

```
(gsms_env) Yogeshs-MacBook-Pro:data yogeshpatil$ kubectl get nodes
NAME                           STATUS   ROLES    AGE   VERSION
ip-10-0-15-49.ec2.internal     Ready    <none>   13d   v1.31.2-eks-94953ac
ip-10-0-8-102.ec2.internal     Ready    <none>   13d   v1.31.2-eks-94953ac
(gsms_env) Yogeshs-MacBook-Pro:data yogeshpatil$ kubectl get pods --all-namespaces
NAMESPACE           NAME                                                        READY   STATUS    RES
ARTS   AGE
amazon-cloudwatch   amazon-cloudwatch-observability-controller-manager-b8f4c844gwck   1/1     Running   0
       14d
amazon-cloudwatch   cloudwatch-agent-48hmm                                      1/1     Running   0
       13d
amazon-cloudwatch   cloudwatch-agent-xb482                                      1/1     Running   0
       13d
amazon-cloudwatch   fluent-bit-bw229                                            1/1     Running   0
       13d
amazon-cloudwatch   fluent-bit-k56rg                                            1/1     Running   0
       13d
default             task-task-0-f4f4c5bcc-px295                                 1/1     Running   0
       108m
default             task-task-1-5d98cccc98-v5xr9                                1/1     Running   0
       108m
default             task-task-2-54c4cf8699-f5lkm                                1/1     Running   0
       108m
default             task-task-3-67bd55f87b-srbcs                                1/1     Running   0
       108m
default             task-task-4-f749cb69c-85fx8                                 1/1     Running   0
       108m
```

3 Kubernetes Configuration

1. Metrics Server Installation

Install metrics server

Verify installation

```
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
kubectl get deployment metrics-server -n kube-system
```

```
(gsms_env) Yogeshs-MacBook-Pro:data yogeshpatil$ kubectl get deployment metrics-server -n kube-system
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
metrics-server   1/1     1            1           11d
```

2. Resource Quotas

Create namespace quota

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: gsms-quota
spec:
  hard:
    requests.cpu: "4"
    requests.memory: 8Gi
    limits.cpu: "8"
    limits.memory: 16Gi
```

4 System Configuration

1. Task Configuration - Constraints

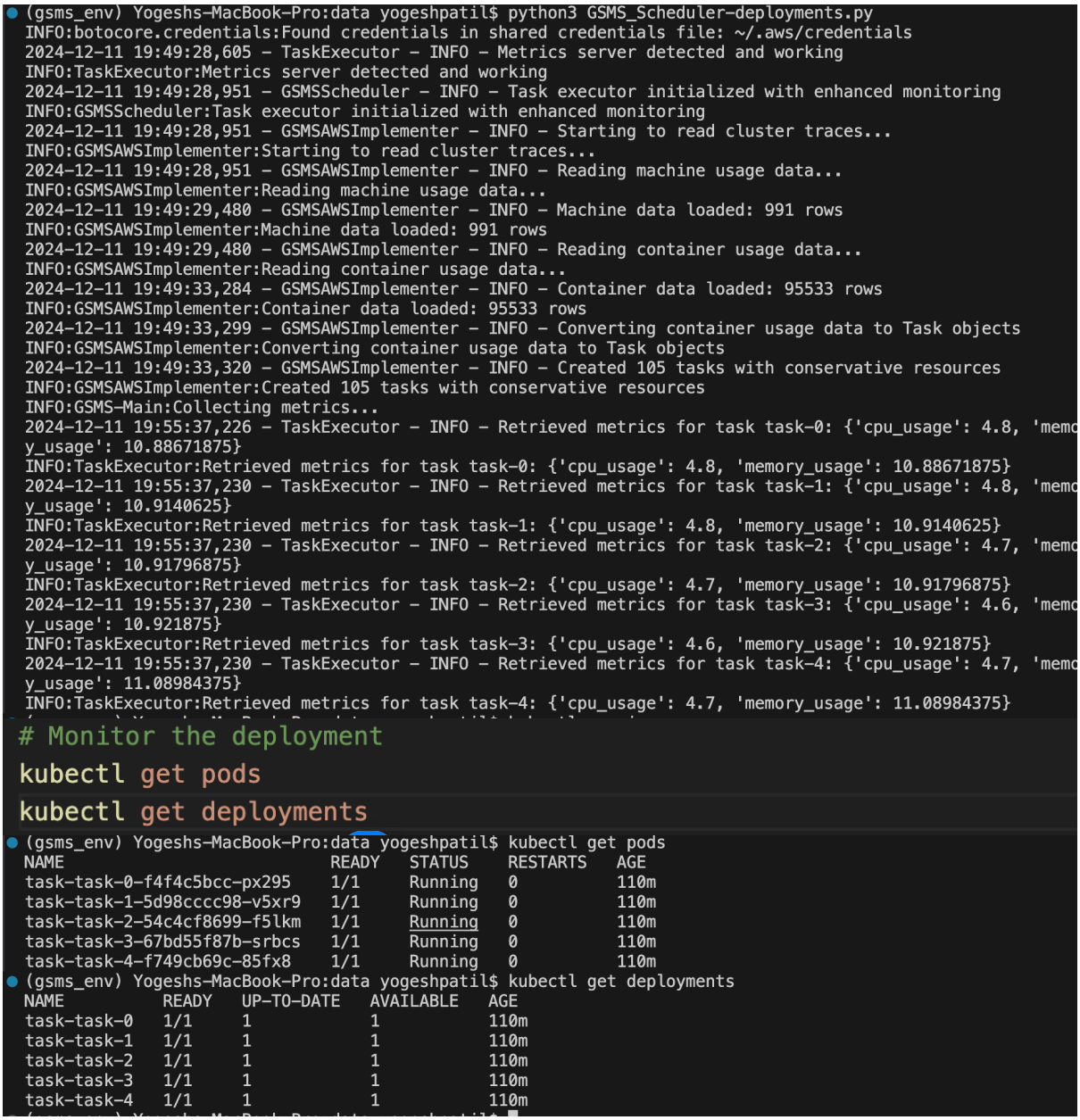
task:
 min_cpu: "100m"
 max_cpu: "500m"
 min_memory: "128Mi"
 max_memory: "512Mi"
 default_timeout: 300
 max_retries: 2

2. Scheduler Configuration

scheduler:
 max_batch_size: 5
 deadline_multiplier: 1.2
 min_reliability: 0.95
 monitoring_interval: 10

5 Running the System

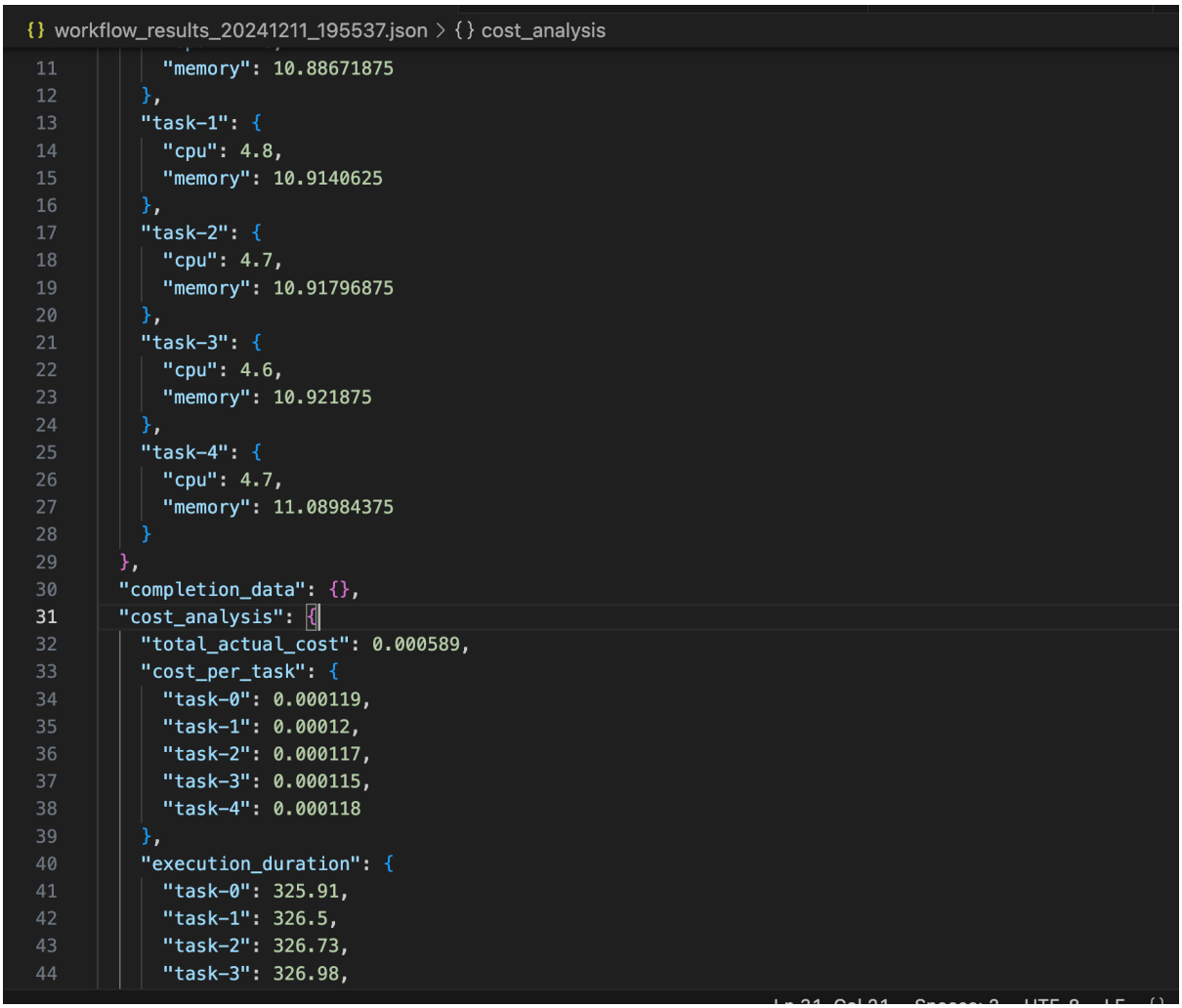
Run the scheduler
python GSMS_Scheduler-deployment.py



6 Metrics Collection

Metrics collected:

- CPU utilization
- Memory usage
- Task runtime
- Success/failure rates
- Resource efficiency



References:

https://docs.aws.amazon.com/