

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 configuration
- 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 awscli==1.36.9
2 boto3==1.35.68
3 botocore==1.35.68
4 cachetools==5.5.0
5 certifi==2024.8.30
6 charset-normalizer==3.4.0
7 colorama==0.4.6
8 docutils==0.16
9 durationpy==0.9
10 exceptiongroup==1.2.2
11 google-auth==2.36.0
12 idna==3.10
13 iniconfig==2.0.0
14 jmespath==1.0.1
15 kubernetes==31.0.0
16 numpy==2.1.3
17 oauthlib==3.2.2
18 packaging==24.2
19 pandas==2.2.3
20 pluggy==1.5.0
21 pyasn1==0.6.1
22 pyasn1_modules==0.4.1
23 pytest==8.3.3
24 python-dateutil==2.0.0, nct0
```

1.3 Kubernetes

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

```
(gsm_env) Yogeshs-MacBook-Pro:~$ kubectl version
Client Version: v1.30.5
Kustomize Version: v5.0.4-0,20230601165947-6ce0bf390ce3
Server Version: v1.31.2-eks-7f9249a
```

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

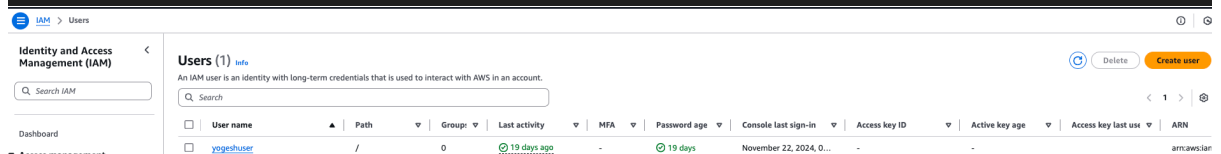
2.1 Create AWS Account

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

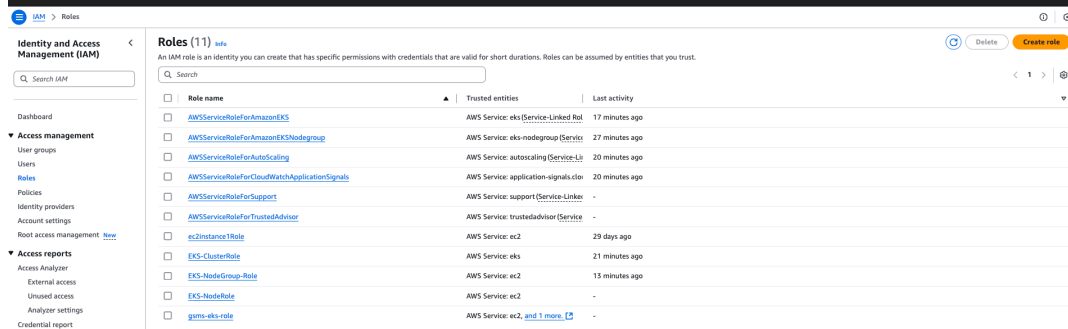
1. Navigate to IAM Console
2. Create a new IAM user role:

```
# 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)
```

The screenshot displays the AWS VPC dashboard for a VPC named 'vpc-0cad11244a8c5284b / gsms-eks-vpc'. The 'Details' tab is active, showing the VPC ID, State (Available), DNS resolution (Enabled), Main network ACL, and IPv6 CIDR. The 'Resource map' tab is also visible, showing the VPC's subnets (us-east-1a, us-east-1b), route tables (EKS-VPC-rtb-private, EKS-VPC-rtb-public), and network connections (EKS-VPC-igw, EKS-VPC-vpc-e3).

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)
```

VPC dashboard ×

Internet gateways (2) info

Search

☐	Name	Internet gateway ID	State	VPC ID	Owner
☐	-	igw-0469768ecd137aedf7	Attached	vpc-0b99208134c1cdff	842675988708
☐	EKS-VPC-igw	igw-0d61f1ba21a8392d4	Attached	vpc-0cad11244a8c5284b1goms-eks-vpc	842675988708

Virtual private cloud

2.4 S3 Bucket Creation

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

Amazon S3 > Buckets > alibaba-cluster-traces

alibaba-cluster-traces info

Objects (6) info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

☐	Name	Type	Last modified	Size	Storage class
☐	container_meta.csv	csv	November 23, 2024, 12:03:22 (UTC+00:00)	17.8 MB	Standard
☐	container_usage_small.csv	csv	November 26, 2024, 09:47:37 (UTC+00:00)	4.1 MB	Standard
☐	container_usage.csv	csv	November 22, 2024, 05:40:36 (UTC+00:00)	164.1 GB	Standard
☐	machine_meta.csv	csv	November 19, 2024, 07:08:03 (UTC+00:00)	539.6 KB	Standard
☐	machine_usage_small.csv	csv	November 26, 2024, 09:47:37 (UTC+00:00)	35.1 KB	Standard
☐	machine_usage.csv	csv	November 19, 2024, 07:08:01 (UTC+00:00)	8.4 GB	Standard

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
```

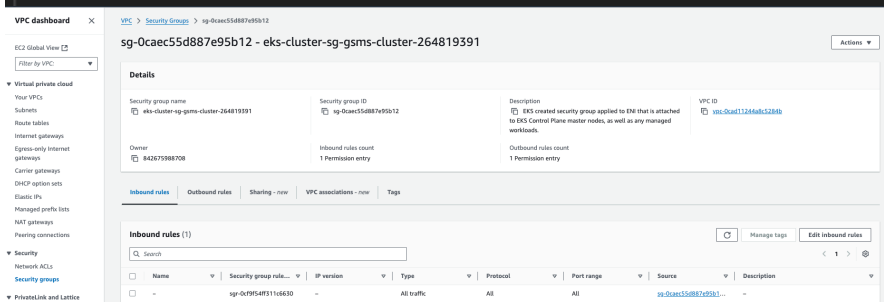
2. Create Node Security Group:

```

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

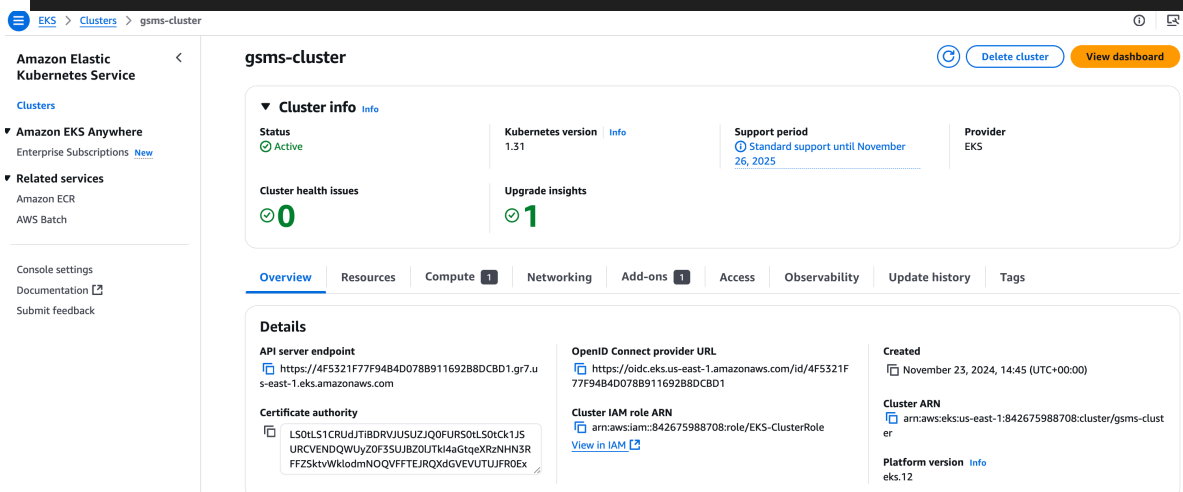
1. Create Cluster:

```

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
```

The screenshot shows the AWS Management Console for the 'gsms-nodegroup'. The left sidebar contains the navigation menu with 'Amazon Elastic Kubernetes Service' selected. The main content area shows the 'gsms-nodegroup' configuration. A notification at the top states: 'New AMI release version is available for this node group.' with a 'Learn more' link and an 'Update now' button. Below this, the 'Node group configuration' section displays the following details:

Property	Value	Status
Kubernetes version	1.31	Active
AMI type	AL2023_x86_64_STANDARD	
AMI release version	1.31.2-20241121	Disk size: 20 GiB
Instance types	t3.medium	

Below the configuration, there are tabs for 'Details', 'Nodes', 'Health issues', 'Kubernetes labels', 'Update config', 'Kubernetes taints', 'Update history', and 'Tags'. The 'Nodes' tab is selected, showing a list of 2 nodes. The table below contains the node details:

Node name	Instance type	Compute	Managed by	Created	Status
ip-10-0-15-49.ec2.internal	t3.medium	Node group	gsms-nodegroup	Created November 28, 2024, 02:30 (UTC+00:00)	Ready
ip-10-0-8-102.ec2.internal	t3.medium	Node group	gsms-nodegroup	Created November 28, 2024, 02:30 (UTC+00:00)	Ready

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-04d1fd43422fcd3c3"
      ],
      "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
```

```

(gsms_env) Yogeshs-MacBook-Pro:data yogeshpatil$ python3 GSMS_Scheduler-deployments.py
INFO:botocore.credentials:Found credentials in shared credentials file: ~/.aws/credentials
2024-12-11 19:49:28,605 - TaskExecutor - INFO - Metrics server detected and working
INFO:TaskExecutor:Metrics server detected and working
2024-12-11 19:49:28,951 - GSMSScheduler - INFO - Task executor initialized with enhanced monitoring
INFO:GSMSScheduler:Task executor initialized with enhanced monitoring
2024-12-11 19:49:28,951 - GSMSAWSImplementer - INFO - Starting to read cluster traces...
INFO:GSMSAWSImplementer:Starting to read cluster traces...
2024-12-11 19:49:28,951 - GSMSAWSImplementer - INFO - Reading machine usage data...
INFO:GSMSAWSImplementer:Reading machine usage data...
2024-12-11 19:49:29,480 - GSMSAWSImplementer - INFO - Machine data loaded: 991 rows
INFO:GSMSAWSImplementer:Machine data loaded: 991 rows
2024-12-11 19:49:29,480 - GSMSAWSImplementer - INFO - Reading container usage data...
INFO:GSMSAWSImplementer:Reading container usage data...
2024-12-11 19:49:33,284 - GSMSAWSImplementer - INFO - Container data loaded: 95533 rows
INFO:GSMSAWSImplementer:Container data loaded: 95533 rows
2024-12-11 19:49:33,299 - GSMSAWSImplementer - INFO - Converting container usage data to Task objects
INFO:GSMSAWSImplementer:Converting container usage data to Task objects
2024-12-11 19:49:33,320 - GSMSAWSImplementer - INFO - Created 105 tasks with conservative resources
INFO:GSMSAWSImplementer:Created 105 tasks with conservative resources
INFO:GSMS-Main:Collecting metrics...
2024-12-11 19:55:37,226 - TaskExecutor - INFO - Retrieved metrics for task task-0: {'cpu_usage': 4.8, 'memory_usage': 10.88671875}
INFO:TaskExecutor:Retrieved metrics for task task-0: {'cpu_usage': 4.8, 'memory_usage': 10.88671875}
2024-12-11 19:55:37,230 - TaskExecutor - INFO - Retrieved metrics for task task-1: {'cpu_usage': 4.8, 'memory_usage': 10.9140625}
INFO:TaskExecutor:Retrieved metrics for task task-1: {'cpu_usage': 4.8, 'memory_usage': 10.9140625}
2024-12-11 19:55:37,230 - TaskExecutor - INFO - Retrieved metrics for task task-2: {'cpu_usage': 4.7, 'memory_usage': 10.91796875}
INFO:TaskExecutor:Retrieved metrics for task task-2: {'cpu_usage': 4.7, 'memory_usage': 10.91796875}
2024-12-11 19:55:37,230 - TaskExecutor - INFO - Retrieved metrics for task task-3: {'cpu_usage': 4.6, 'memory_usage': 10.921875}
INFO:TaskExecutor:Retrieved metrics for task task-3: {'cpu_usage': 4.6, 'memory_usage': 10.921875}
2024-12-11 19:55:37,230 - TaskExecutor - INFO - Retrieved metrics for task task-4: {'cpu_usage': 4.7, 'memory_usage': 11.08984375}
INFO:TaskExecutor:Retrieved metrics for task task-4: {'cpu_usage': 4.7, 'memory_usage': 11.08984375}

# Monitor the deployment
kubectl get pods
kubectl get deployments
(gsms_env) Yogeshs-MacBook-Pro:data yogeshpatil$ kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
task-task-0-f4f4c5bcc-px295        1/1      Running   0           110m
task-task-1-5d98cccc98-v5xr9       1/1      Running   0           110m
task-task-2-54c4cf8699-f5lkm       1/1      Running   0           110m
task-task-3-67bd55f87b-srbcs       1/1      Running   0           110m
task-task-4-f749cb69c-85fx8        1/1      Running   0           110m
(gsms_env) Yogeshs-MacBook-Pro:data yogeshpatil$ kubectl get deployments
NAME                                READY    UP-TO-DATE    AVAILABLE   AGE
task-task-0                        1/1      1              1           110m
task-task-1                        1/1      1              1           110m
task-task-2                        1/1      1              1           110m
task-task-3                        1/1      1              1           110m
task-task-4                        1/1      1              1           110m

```

6 Metrics Collection

Metrics collected:

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

```

{} workflow_results_20241211_195537.json > {} cost_analysis
11     "memory": 10.88671875
12 },
13 "task-1": {
14     "cpu": 4.8,
15     "memory": 10.9140625
16 },
17 "task-2": {
18     "cpu": 4.7,
19     "memory": 10.91796875
20 },
21 "task-3": {
22     "cpu": 4.6,
23     "memory": 10.921875
24 },
25 "task-4": {
26     "cpu": 4.7,
27     "memory": 11.08984375
28 }
29 },
30 "completion_data": {},
31 "cost_analysis": {
32     "total_actual_cost": 0.000589,
33     "cost_per_task": {
34         "task-0": 0.000119,
35         "task-1": 0.00012,
36         "task-2": 0.000117,
37         "task-3": 0.000115,
38         "task-4": 0.000118
39     },
40     "execution_duration": {
41         "task-0": 325.91,
42         "task-1": 326.5,
43         "task-2": 326.73,
44         "task-3": 326.98,

```

References:

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