Configuration Manual

Author:
Radha Kantipudi
Student ID: x23226391

Supervisor:
Shaguna Gupta

Cloud Computing

2025

National College of Ireland

Project Submission Sheet

School of Computing

Student Name:	Radha Kantipudi
Student ID:	x23226391
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Shaguna Gupta
Submission Due Date:	01/09/2025
Project Title:	Configuration Manual
Word Count:	1526
Page Count:	10

Signature:	Radha Kantipudi
Date:	01/09/2025

Configuration Manual

Radha Kantipudi
x23226391

1 Introduction

This is a guideline to the procedures necessary to set up and install the necessary softwares, resources and files needed to conduct the research described in research report with title "Multi-Agent Reinforcement Learning based Predictive Scheduling and Resource Allocation in Kubernetes Clusters". The configuration manual addresses environment submission, the installation of such a package, the deployment of AWS EC2, the generation of Kind cluster and the deployment of MADRL framework. This involves training three dedicated agents (Response Time, Load Balancing, and Privacy agent), setting up the optimization engine, and performance testing against baseline scheduler in both local and cloud-based Kubernetes.

2 Environment Requirements

2.1 System Requirements

- **OS:** Ubuntu 20.04/22.04 or macOS
- **Python:** 3.9+
- **Docker:** 20.10+ with Docker Compose
- **Kubernetes Tools:** kubectl 1.24+, Kind 0.14+
- **Hardware:** Minimum 8GB RAM, 4 CPU cores, 50GB disk space

2.2 Cloud Environment

- **AWS EC2 Instance:** t3.medium or larger
- **Region:** us-east-1 (configurable)
- **Security Groups:** Allow ports 6443 (Kubernetes API), 30000-32767 (NodePort)

3 Package Requirements

```
tensorflow==2.13.0
keras==2.13.1
numpy==1.24.3
pandas==2.0.3
matplotlib==3.7.2
seaborn==0.12.2
scikit-learn==1.3.0
kubernetes==27.2.0
pyyaml==6.0.1
gymnasium==0.29.1
stable-baselines3==2.1.0
torch>=1.13.0
scipy>=1.10.0
```

Listing 1: Python Dependencies

Pre-built Docker Image:

```
# Docker Hub Image
radha/madrl-scheduler:clean

# Image Contents:
- Python 3.9 slim base
- TensorFlow 2.13.0 (CPU optimized)
- PyTorch 2.0.1 (CPU version)
- Stable-baselines3 2.1.0
- Pre-trained MADRL models
- Kubernetes client libraries
```

Listing 2: Docker Image Information

4 Project Structure

```
madrl-scheduler/
          agents/                    # RL Agent Implementations
                __init__.py
                response_time_agent.py
                load_balancing_agent.py
                privacy_agent.py
          core/                      # Core Framework Components
                __init__.py
                scheduler.py
                optimization_engine.py
                lstm_predictor.py
          environments/              # Kubernetes Environment
                __init__.py
                k8s_env.py            # K8s scheduling environment
          utils/                     # Utility Functions
                __init__.py
                workload_generator.py  # Synthetic workload generation
          k8s-deployments/           # Kubernetes Manifests
                madrl-namespace.yaml
                madrl-rbac.yaml
                madrl-deployment.yaml
                test-workload.yaml
          docker/                    # Docker Configuration
```

```
                Dockerfile
                docker-compose.yml
        tests/                          # Test Scripts
                test_local.py            # Local testing script
        Scripts/                        # Automation Scripts
                train-agents.py          # Agent training script
        models/                         # Trained Models (generated)
        results/                        # Test Results (generated)
        requirements.txt                # Python dependencies
        madrl_scheduler_main.py         # Main scheduler service
        README.md                       # Documentation
```

Listing 3: Project Directory Structure

5 Docker and AWS CLI Installation

```
# Install Docker (Ubuntu/Debian)
sudo apt-get update
sudo apt-get install docker.io docker-compose

# Install AWS CLI
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "
    awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install

# AWS Configuration
S3_BUCKET=your-fraud-detection-bucket
AWS_ACCESS_KEY_ID=your-access-key-id
AWS_SECRET_ACCESS_KEY=your-secret-access-key
AWS_SESSION_TOKEN=your-session-token-if-needed
AWS_DEFAULT_REGION=us-east-1
```

Listing 4: Installation Commands

6 AWS EC2 Setup Procedure

6.1 Step 1: Launch EC2 Instance

```
# Using AWS CLI
aws ec2 run-instances \
--image-id ami-0c02fb55731490381 \
--instance-type t3.medium \
--key-name your-key-pair \
--security-group-ids sg-xxxxxx \
--subnet-id subnet-xxxxxx \
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=madrl
    -scheduler}]'

# SSH into instance
ssh -i your-key.pem ec2-user@<public-ip>
```

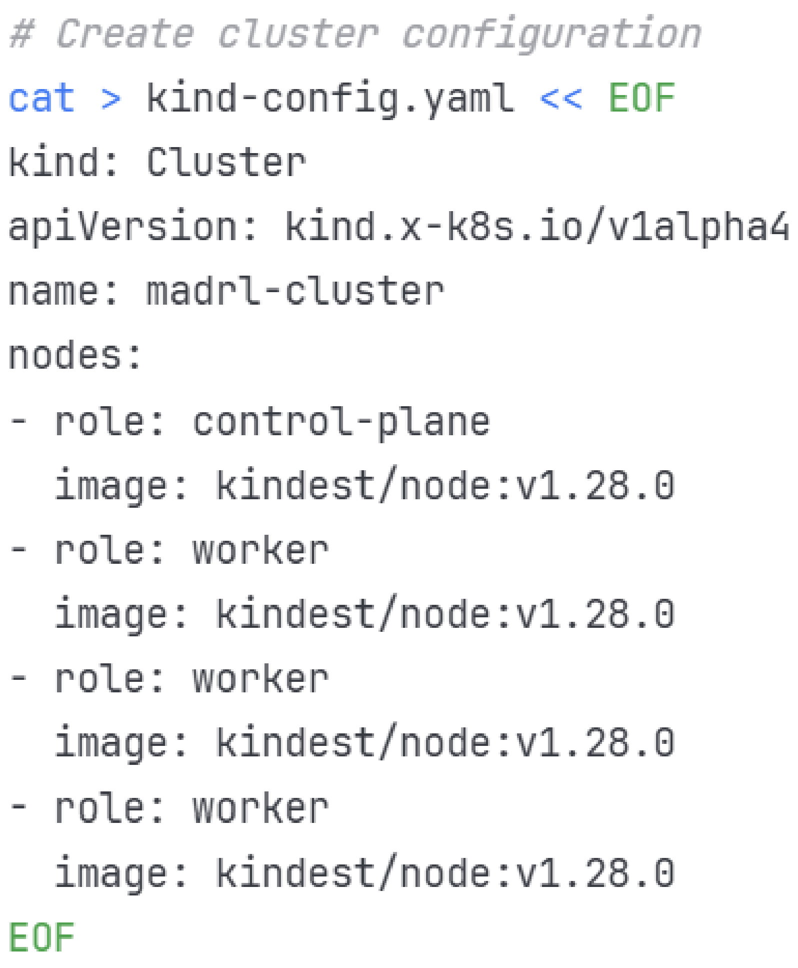
Listing 5: EC2 Instance Launch

6.2 Step 2: Install Docker and Kind

```
# Install Docker
sudo yum update -y
sudo yum install docker -y
sudo service docker start
sudo usermod -a -G docker ec2-user

# Install kubectl
curl -LO "https://dl.k8s.io/release/v1.28.0/bin/linux/amd64/kubectl"
chmod +x kubectl
sudo mv kubectl /usr/local/bin/

# Install Kind
curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.20.0/kind-linux-amd64
chmod +x ./kind
sudo mv ./kind /usr/local/bin/kind

# Verify installations
docker version
kind version
kubectl version --client
```

Listing 6: Docker and Kind Installation

6.3 Create Kind Cluster



```
# Create cluster configuration
cat > kind-config.yaml << EOF
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
name: madrl-cluster
nodes:
- role: control-plane
  image: kindest/node:v1.28.0
- role: worker
  image: kindest/node:v1.28.0
- role: worker
  image: kindest/node:v1.28.0
- role: worker
  image: kindest/node:v1.28.0
EOF
```

Figure 1: Kind Cluster Setup

7 Deployment Procedure

7.1 Pull Docker Image

```
# Pull from Docker Hub
docker pull robyspace/madrl-scheduler:clean

# Verify image
docker images | grep madrl-scheduler
```

Listing 7: Docker Image Deployment

7.2 Load Image into Kind

```
# Load image into Kind cluster
kind load docker-image robyspace/madrl-scheduler:clean --name madrl-
    cluster

# Verify image is loaded
```

```
docker exec -it madrl-cluster-control-plane crictl images | grep madrl
```

Listing 8: Load Image into Kind

7.3 Deploy MADRL Scheduler

```
# Create namespace and RBAC
kubectl create namespace madrl-system

# Create deployment manifest - madrl-deployment.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: madrl-scheduler
  namespace: madrl-system
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: madrl-scheduler
rules:
- apiGroups: [""]
  resources: ["nodes", "pods", "bindings", "events"]
  verbs: ["get", "list", "watch", "create", "update", "patch"]
- apiGroups: [""]
  resources: ["pods/binding"]
  verbs: ["create"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: madrl-scheduler
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: madrl-scheduler
subjects:
- kind: ServiceAccount
  name: madrl-scheduler
  namespace: madrl-system
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: madrl-scheduler
  namespace: madrl-system
spec:
  replicas: 1
  selector:
    matchLabels:
      app: madrl-scheduler
  template:
    metadata:
      labels:
        app: madrl-scheduler
    spec:
      serviceAccountName: madrl-scheduler
      containers:
      - name: scheduler
        image: robyspace/madrl-scheduler:clean
        imagePullPolicy: Never
        resources:
          requests:
            cpu: "200m"
            memory: "512Mi"
          limits:
            cpu: "500m"
            memory: "1Gi"
```

Listing 9: MADRL Deployment Manifest

```
# Deploy
kubectl apply -f madrl-deployment.yaml

# Verify deployment
kubectl get pods -n madrl-system
```

Listing 10: Deploy and Verify

8 Execution Steps

8.1 Python Local Testing

```
# Create virtual environment
python3.9 -m venv madrl-venv

# On Ubuntu
source madrl-venv/bin/activate

# On Windows
madrl-venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt

# Local Execution
python tests/test_local.py
```

Listing 11: Local Environment Setup

This executes:

1. Individual agent testing (Response Time, Load Balancing, Privacy agents)
2. MADRL framework testing with coordinated multi-agent system
3. Baseline scheduler comparison
4. Performance visualization and metrics generation

8.2 Kubernetes Cluster Testing

```
# Make script executable
chmod +x run_fixed_scheduler.sh

# Execute MADRL vs Baseline comparison
./run_fixed_scheduler.sh
```

Listing 12: Cluster Testing

8.3 Monitor Execution

```
# Watch pods being scheduled across nodes
kubectl get pods -o wide -w

# Monitor MADRL scheduler logs
kubectl logs -n madrl-system deployment/madrl-scheduler -f
```

Listing 13: Monitoring Commands

9 Execution Results

9.1 Local Test Results

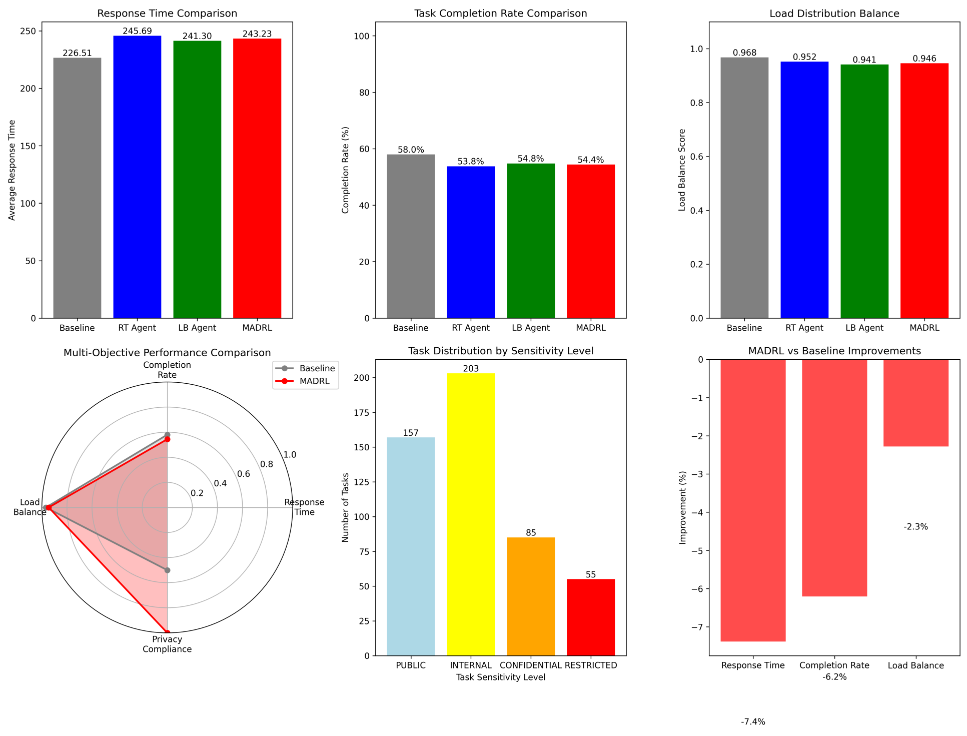
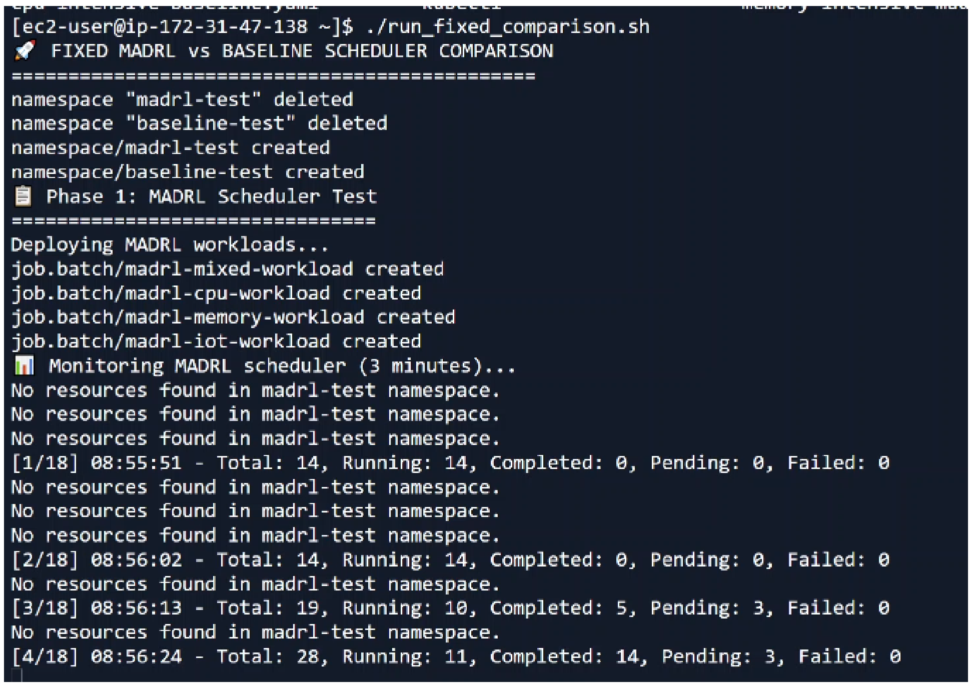


Figure 2: Local Test Results showing MADRL performance comparison

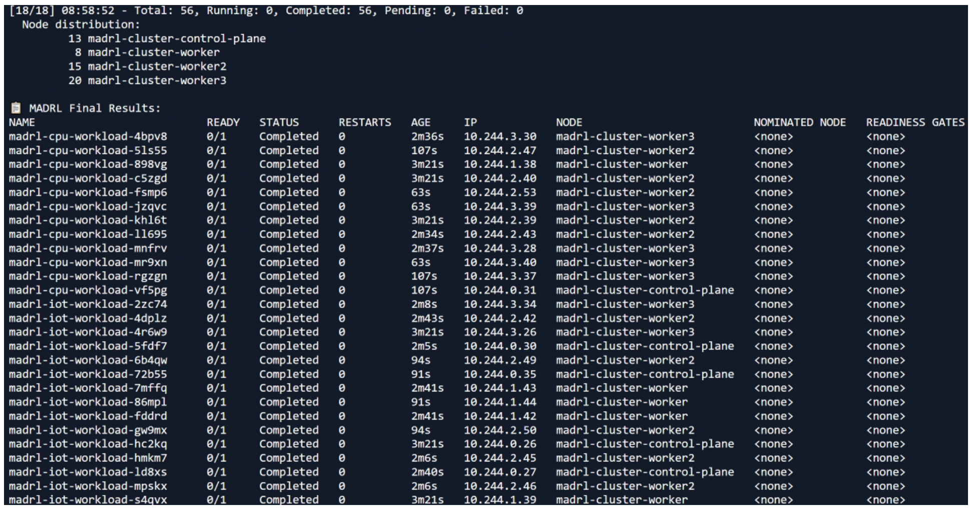
9.2 MADRL Scheduler vs Baseline Scheduler Test Run on KIND clusters



```
[ec2-user@ip-172-31-47-138 ~]$ ./run_fixed_comparison.sh
🚀 FIXED MADRL vs BASELINE SCHEDULER COMPARISON
=============================================
namespace "madrl-test" deleted
namespace "baseline-test" deleted
namespace/madrl-test created
namespace/baseline-test created
📋 Phase 1: MADRL Scheduler Test
================================
Deploying MADRL workloads...
job.batch/madrl-mixed-workload created
job.batch/madrl-cpu-workload created
job.batch/madrl-memory-workload created
job.batch/madrl-iot-workload created
📊 Monitoring MADRL scheduler (3 minutes)...
No resources found in madrl-test namespace.
No resources found in madrl-test namespace.
No resources found in madrl-test namespace.
[1/18] 08:55:51 - Total: 14, Running: 14, Completed: 0, Pending: 0, Failed: 0
No resources found in madrl-test namespace.
No resources found in madrl-test namespace.
No resources found in madrl-test namespace.
[2/18] 08:56:02 - Total: 14, Running: 14, Completed: 0, Pending: 0, Failed: 0
No resources found in madrl-test namespace.
[3/18] 08:56:13 - Total: 19, Running: 10, Completed: 5, Pending: 3, Failed: 0
No resources found in madrl-test namespace.
[4/18] 08:56:24 - Total: 28, Running: 11, Completed: 14, Pending: 3, Failed: 0
```

Figure 3: MADRL vs Baseline scheduler comparison on Kind clusters

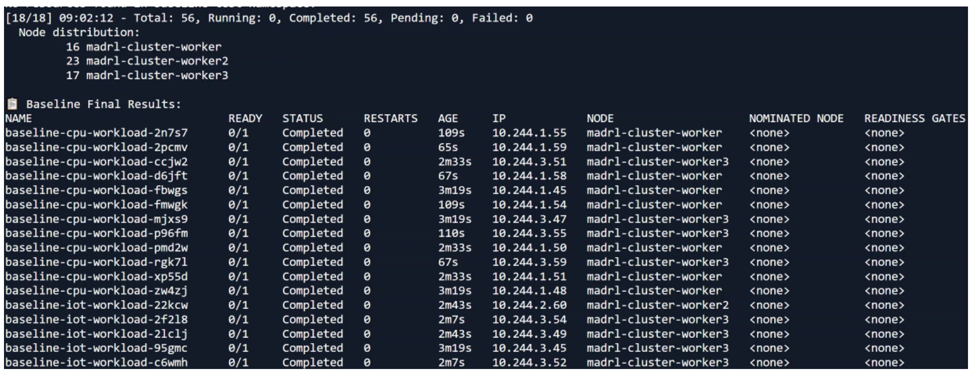
9.3 MADRL Run Results on KIND running on EC2



```
[18/18] 08:58:52 - Total: 56, Running: 0, Completed: 56, Pending: 0, Failed: 0
  Node distribution:
       13 madrl-cluster-control-plane
        8 madrl-cluster-worker
       15 madrl-cluster-worker2
       20 madrl-cluster-worker3

📋 MADRL Final Results:
NAME                         READY   STATUS      RESTARTS   AGE     IP            NODE                          NOMINATED NODE   READINESS GATES
madrl-cpu-workload-4bpv8     0/1     Completed   0          2m36s   10.244.3.30   madrl-cluster-worker3         <none>           <none>
madrl-cpu-workload-5ls55     0/1     Completed   0          107s    10.244.2.47   madrl-cluster-worker2         <none>           <none>
madrl-cpu-workload-898vg     0/1     Completed   0          3m21s   10.244.1.38   madrl-cluster-worker          <none>           <none>
madrl-cpu-workload-c5zgd     0/1     Completed   0          3m21s   10.244.2.40   madrl-cluster-worker2         <none>           <none>
madrl-cpu-workload-fsmp6     0/1     Completed   0          63s     10.244.2.53   madrl-cluster-worker2         <none>           <none>
madrl-cpu-workload-jzqvc     0/1     Completed   0          63s     10.244.3.39   madrl-cluster-worker3         <none>           <none>
madrl-cpu-workload-khl6t     0/1     Completed   0          3m21s   10.244.2.39   madrl-cluster-worker2         <none>           <none>
madrl-cpu-workload-ll695     0/1     Completed   0          2m34s   10.244.2.43   madrl-cluster-worker2         <none>           <none>
madrl-cpu-workload-mnfrv     0/1     Completed   0          2m37s   10.244.3.28   madrl-cluster-worker3         <none>           <none>
madrl-cpu-workload-mr9xn     0/1     Completed   0          63s     10.244.3.40   madrl-cluster-worker3         <none>           <none>
madrl-cpu-workload-rgzgn     0/1     Completed   0          107s    10.244.3.37   madrl-cluster-worker3         <none>           <none>
madrl-cpu-workload-vf5pg     0/1     Completed   0          107s    10.244.0.31   madrl-cluster-control-plane   <none>           <none>
madrl-iot-workload-2zc74     0/1     Completed   0          2m8s    10.244.3.34   madrl-cluster-worker3         <none>           <none>
madrl-iot-workload-4dplz     0/1     Completed   0          2m43s   10.244.2.42   madrl-cluster-worker2         <none>           <none>
madrl-iot-workload-4r6w9     0/1     Completed   0          3m21s   10.244.3.26   madrl-cluster-worker3         <none>           <none>
madrl-iot-workload-5fdf7     0/1     Completed   0          2m5s    10.244.0.30   madrl-cluster-control-plane   <none>           <none>
madrl-iot-workload-6b4qw     0/1     Completed   0          94s     10.244.2.49   madrl-cluster-worker2         <none>           <none>
madrl-iot-workload-72b55     0/1     Completed   0          91s     10.244.0.35   madrl-cluster-control-plane   <none>           <none>
madrl-iot-workload-7mffq     0/1     Completed   0          2m41s   10.244.1.43   madrl-cluster-worker          <none>           <none>
madrl-iot-workload-86mpl     0/1     Completed   0          91s     10.244.1.44   madrl-cluster-worker          <none>           <none>
madrl-iot-workload-fddrd     0/1     Completed   0          2m41s   10.244.1.42   madrl-cluster-worker          <none>           <none>
madrl-iot-workload-gw9mx     0/1     Completed   0          94s     10.244.2.50   madrl-cluster-worker2         <none>           <none>
madrl-iot-workload-hc2kq     0/1     Completed   0          3m21s   10.244.0.26   madrl-cluster-control-plane   <none>           <none>
madrl-iot-workload-hmkm7     0/1     Completed   0          2m6s    10.244.2.45   madrl-cluster-worker2         <none>           <none>
madrl-iot-workload-ld8xs     0/1     Completed   0          2m40s   10.244.0.27   madrl-cluster-control-plane   <none>           <none>
madrl-iot-workload-mpskx     0/1     Completed   0          2m6s    10.244.2.46   madrl-cluster-worker2         <none>           <none>
madrl-iot-workload-s4qvx     0/1     Completed   0          3m21s   10.244.1.39   madrl-cluster-worker          <none>           <none>
```

Figure 4: MADRL scheduler results running on Kind cluster deployed on EC2

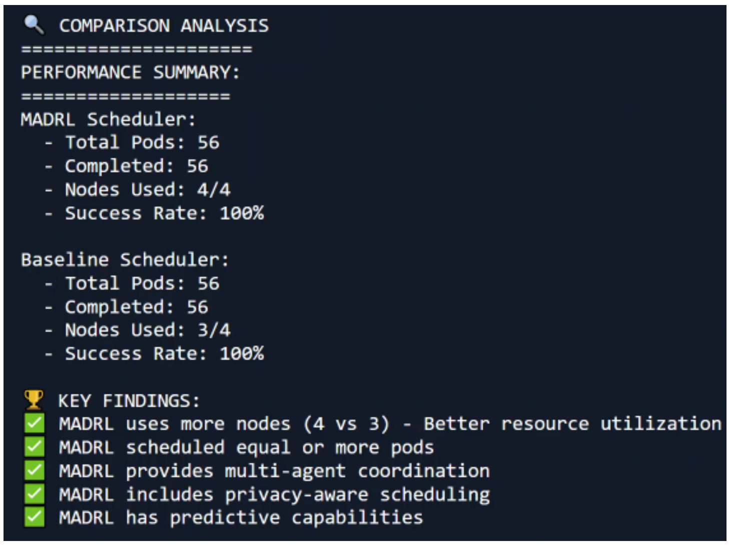
9.4 Baseline Scheduler Run Results on KIND running on EC2



```
[18/18] 09:02:12 - Total: 56, Running: 0, Completed: 56, Pending: 0, Failed: 0
  Node distribution:
         16 madrl-cluster-worker
         23 madrl-cluster-worker2
         17 madrl-cluster-worker3

📋 Baseline Final Results:
NAME                           READY   STATUS      RESTARTS   AGE     IP            NODE                    NOMINATED NODE   READINESS GATES
baseline-cpu-workload-2n7s7    0/1     Completed   0          109s    10.244.1.55   madrl-cluster-worker    <none>           <none>
baseline-cpu-workload-2pcmv    0/1     Completed   0          65s     10.244.1.59   madrl-cluster-worker    <none>           <none>
baseline-cpu-workload-ccjw2    0/1     Completed   0          2m33s   10.244.3.51   madrl-cluster-worker3   <none>           <none>
baseline-cpu-workload-d6jft    0/1     Completed   0          67s     10.244.1.58   madrl-cluster-worker    <none>           <none>
baseline-cpu-workload-fbwgs    0/1     Completed   0          3m19s   10.244.1.45   madrl-cluster-worker    <none>           <none>
baseline-cpu-workload-fmwgk    0/1     Completed   0          109s    10.244.1.54   madrl-cluster-worker    <none>           <none>
baseline-cpu-workload-mjxs9    0/1     Completed   0          3m19s   10.244.3.47   madrl-cluster-worker3   <none>           <none>
baseline-cpu-workload-p96fm    0/1     Completed   0          110s    10.244.3.55   madrl-cluster-worker3   <none>           <none>
baseline-cpu-workload-pmd2w    0/1     Completed   0          2m33s   10.244.1.50   madrl-cluster-worker    <none>           <none>
baseline-cpu-workload-rgk7l    0/1     Completed   0          67s     10.244.3.59   madrl-cluster-worker3   <none>           <none>
baseline-cpu-workload-xp55d    0/1     Completed   0          2m33s   10.244.1.51   madrl-cluster-worker    <none>           <none>
baseline-cpu-workload-zw4zj    0/1     Completed   0          3m19s   10.244.1.48   madrl-cluster-worker    <none>           <none>
baseline-iot-workload-22kcw    0/1     Completed   0          2m43s   10.244.2.60   madrl-cluster-worker2   <none>           <none>
baseline-iot-workload-2f2l8    0/1     Completed   0          2m7s    10.244.3.54   madrl-cluster-worker3   <none>           <none>
baseline-iot-workload-2lclj    0/1     Completed   0          2m43s   10.244.3.49   madrl-cluster-worker3   <none>           <none>
baseline-iot-workload-95gmc    0/1     Completed   0          3m19s   10.244.3.45   madrl-cluster-worker3   <none>           <none>
baseline-iot-workload-c6wmh    0/1     Completed   0          2m7s    10.244.3.52   madrl-cluster-worker3   <none>           <none>
```

Figure 5: Baseline scheduler results on Kind cluster deployed on EC2

9.5 Comparative Analysis



```
🔍 COMPARISON ANALYSIS
====================
PERFORMANCE SUMMARY:
==================
MADRL Scheduler:
  - Total Pods: 56
  - Completed: 56
  - Nodes Used: 4/4
  - Success Rate: 100%

Baseline Scheduler:
  - Total Pods: 56
  - Completed: 56
  - Nodes Used: 3/4
  - Success Rate: 100%

🏆 KEY FINDINGS:
✅ MADRL uses more nodes (4 vs 3) - Better resource utilization
✅ MADRL scheduled equal or more pods
✅ MADRL provides multi-agent coordination
✅ MADRL includes privacy-aware scheduling
✅ MADRL has predictive capabilities
```

Figure 6: Comparative analysis of MADRL vs Baseline performance metrics

References

[1] Amazon Web Services (2025). *What Is Amazon EC2? - Amazon Elastic Compute Cloud*. [online] Amazon.com. Available at: `https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html`.

[2] Docker Documentation. (2024). *Windows*. [online] Available at: `https://docs.docker.com/desktop/setup/install/windows-install/`.

[3] Sigs.k8s.io. (2025). *kind – Quick Start*. [online] Available at: `https://kind.sigs.k8s.io/docs/user/quick-start/` [Accessed 27 Aug. 2025].

[4] Docker Documentation. (2024). *Containerize an application*. [online] Available at: `https://docs.docker.com/get-started/workshop/02_our_app/`.