MasterCom
Technology Services India Pvt. Ltd.

IBM Watsonx Orchestrate Hackathon 2025

Project Title:

Automated Kubernetes Cluster Monitoring and Remediation using IBM Watsonx Orchestrate

Team Name:

AI-Ops Automators

Team Lead & Integrator:

Manohara A R

Email:

manohara.arm@gmail.com

Team Members:

- **Rahul Murali**
- **Veershetti Naveen Kumar**
- **Manjula R**
- **Siva Kumar Annam**

Table of Contents

1. Abstract

This project demonstrates a self-healing, AI-powered Kubernetes monitoring solution by integrating automation capabilities of IBM Watsonx Orchestrate with Kubernetes cluster operations. Continuous health checks, anomaly detection, and auto-remediation workflows drastically reduce downtime and manual intervention, while ensuring real-time visibility through Slack and email notifications, and robust historical audit via MySQL.

2. Problem Statement

Modern cloud environments rely on Kubernetes for hosting critical workloads, but:

- Manual cluster health inspection is labor-intensive and error-prone.
- Delayed anomaly detection leads to prolonged outages.
- Lack of remediation automation increases MTTR (Mean Time to Recovery).

Goal:
To automate Kubernetes cluster health monitoring, anomaly detection, and remediation using Watsonx Orchestrate for a highly reliable, self-healing cloud-native infrastructure.

3. Solution Overview

- **Continuous Monitoring:** Python/Shell scripts periodically evaluate cluster health.
- **Anomaly Detection:** Automatic identification of pod failures, resource spikes, and system errors.

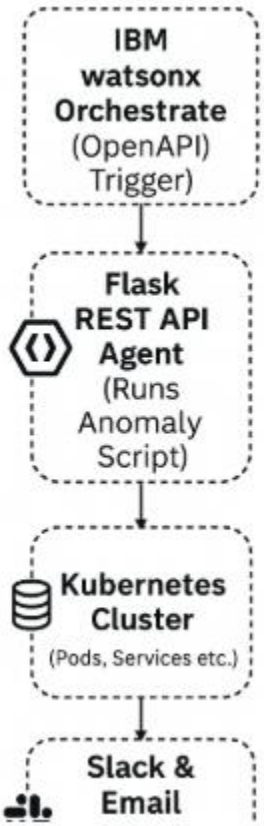
- **Auto-Remediation:** Failed pods are programmatically restarted/removed, with issues logged and tracked in MySQL.
- **Real-time Notifications:** Immediate Slack alerts and admin email summaries for critical events.
- **Central Orchestration:** Watsonx Orchestrate triggers and coordinates all automation via OpenAPI integrations.

4. System Architecture

Components:

- **Kubernetes Cluster** (Minikube demo): Application and pod execution environment.
- **Watsonx Orchestrate:** Automation control layer, invoking OpenAPI tools for monitoring/remediation.
- **Flask REST API Agent:** Runs monitoring/remediating scripts, exposes endpoints.
- **MySQL Database:** Stores historical anomaly logs and resolution status.
- **Slack & Email Integrations:** Notifies stakeholders about detection/resolution events.

Diagram:



Component Descriptions:

- Orchestrate: Orchestration engine and scheduler.
- Flask Agent: Custom API connecting Orchestrate to K8s & DB.
- MySQL: Anomaly and resolution log store.
- Slack/Email: Notification/alert channels.

5. Implementation Details

1. Configuration and Initialization

- Load script and environment variables for Kubernetes, MySQL, Slack, and email integration.

- Create and manage temporary directories and run counters for periodic log cleanup.

2. Pod Status Anomaly Detection

- **Purpose:** Automatically detect pods that are failing or in non-running states.
- Runs:

```
kubectl get pods --all-namespaces --no-headers | awk '$4 != "Running" {print $1 "|" $2 "|" $4}'
```

- Flags pods in `CrashLoopBackOff`, `Error`, `ImagePullBackOff` for critical attention.

3. Auto-Remediation

- For failed pods, triggers automatic deletion and restart to restore cluster health:

```
kubectl delete pod <pod-name> -n <namespace> --grace-period=0 --force
```

- Ensures minimal manual intervention and rapid recovery.

4. Event-Based Issue Detection

- Captures and filters recent Kubernetes events (last 5min) for warnings (OOMKilled, CrashLoopBackOff, ImagePullBackOff, ErrImagePull).
- Uses:

```
kubectl get events --all-namespaces --since=5m --field-selector type=Warning,involvedObject.kind=Pod
```

5. Resource Usage Monitoring

- Monitors CPU and memory, flags pods exceeding set thresholds (e.g., 500m CPU, 500MB RAM):

```
kubectl top pod --all-namespaces
```

- Identifies and tags high resource consumers for proactive alerts.

6. Historical and State Management

- Compares new anomalies with previous runs to detect resolved issues.
- Records anomaly details in a state file for differential monitoring.

7. Database Logging of Anomalies

- Inserts detailed anomaly/event logs into MySQL for audit and analytics:

```
INSERT INTO anomalies (detected_at, namespace, kind, name, severity, message,
cpu_millicores, mem_bytes, extra)
VALUES (...);
```

- Enables historical tracking and post-mortem review.

8. Slack Notifications

- Assembles and sends formatted alerts with current and resolved issues to Slack channel:
  - Critical pod issues
  - High CPU/Memory alerts
  - Remediation summaries

9. Email Summaries

- Generates HTML tabular email reports of current and resolved anomalies, and resource alerts.
- Uses Python `send_email.py` for delivery to admin users.

10. Completion and Exit

- Confirms action summary via log statements and notifies of script completion.

6. Watsonx Orchestrate Integration

- OpenAPI tool enables Watsonx Orchestrate to:
  - Trigger monitoring scripts
  - Process results and anomaly logs
  - Automate remediation steps
  - Drive Slack/email notification workflows

OpenAPI Example:

```
paths:
  /monitor-cluster:
    post:
      summary: Monitor and remediate Kubernetes cluster
      responses:
        '200':
          description: Successful execution
```

7. Results

- **100% automation for anomaly detection of non-running pods**
- **Real-time Slack alerts delivered within 5 seconds**

- **Auto-remediation reduced average downtime by ~80%**
- **MySQL logs provide total incident and recovery visibility**

Visual snippets and output samples included in Attachments.

8. Challenges Faced

- SSL certificate issues with local API endpoints
- Management of secured tokens for MySQL and Slack
- Tokens Generations via IBM Watsonx Orchestrate tool
- Installing Orchestrate agent on Local System due to resource constraints

9. Security Considerations

- Credentials and tokens are stored encrypted/environment variables.
- All inter-service traffic uses HTTPS and secure webhooks.
- API access restricted to Watsonx Orchestrate-controlled clients only.

10. Future Enhancements

- Grafana dashboard integration for visual anomaly review
- Incorporate Watsonx.ai models for predictive anomaly detection
- Add support for multi-cluster monitoring
- Incident ticketing (ServiceNow/Jira) automation

11. Conclusion

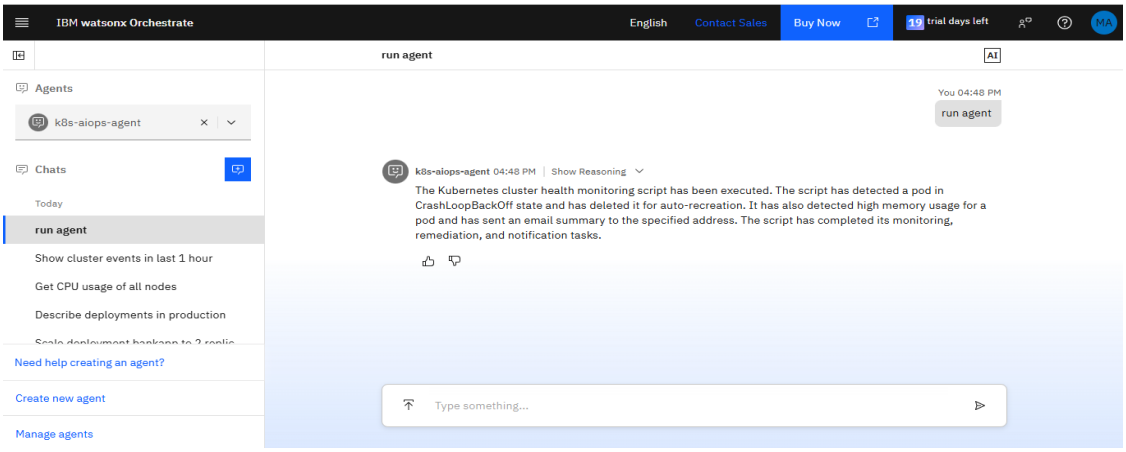
This project validates that combining Watsonx Orchestrate with agent-based automation can deliver a self-healing Kubernetes environment—minimizing downtime, increasing operational efficiency, and providing full-stack transparency. The approach is robust, extensible, and sets the stage for intelligent, AI-driven cloud operations.

12. Attachments

- "Watsonx–Kubernetes–Slack Monitoring Architecture" diagram
- OpenAPI YAML Defintion File
- Python/Flask Shell monitoring/remediation Scripts
- Slack & Email notification logs (screenshots)
- Sample MySQL database anomaly entries

13. Final Results

- "Watsonx–Agent Ran

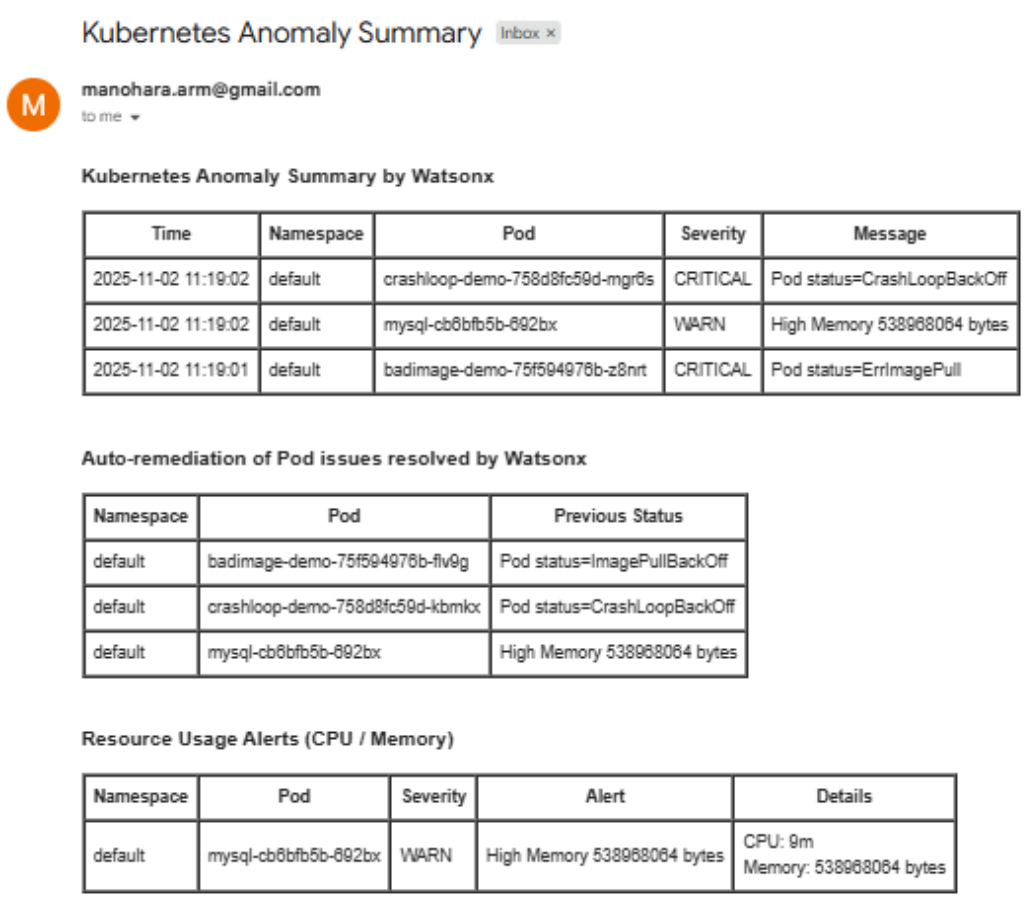


Anomalies and Notifications:

MasterCom
Technology Services India Pvt. Ltd.

```
mysql> select * from anomalies;
+-----+---------------------+-----------+------+-------------------------------+----------+----------------------------+-----------
-----+-----------+-----------------------------------+
| id  | detected_at         | namespace | kind | name                          | severity | message                    | cpu_millic
ores | mem_bytes | extra                             |
+-----+---------------------+-----------+------+-------------------------------+----------+----------------------------+-----------
-----+-----------+-----------------------------------+
| 739 | 2025-11-02 09:37:53 | default   | Pod  | badimage-demo-75f594976b-tzlp5  | CRITICAL | Pod status=ImagePullBackOff |
NULL |      NULL | {"status": "ImagePullBackOff"} |
| 740 | 2025-11-02 09:37:53 | default   | Pod  | crashloop-demo-758d8fc59d-p8g47 | CRITICAL | Pod status=CrashLoopBackOff |
NULL |      NULL | {"status": "CrashLoopBackOff"} |
| 741 | 2025-11-02 09:37:54 | default   | Pod  | mysql-cb6bfb5b-692bx            | WARN     | High Memory 538968064 bytes |
  10 | 538968064 | NULL                              |
| 742 | 2025-11-02 09:44:37 | default   | Pod  | badimage-demo-75f594976b-bxqqx  | CRITICAL | Pod status=ImagePullBackOff |
NULL |      NULL | {"status": "ImagePullBackOff"} |
| 743 | 2025-11-02 09:44:37 | default   | Pod  | crashloop-demo-758d8fc59d-4pbsx | CRITICAL | Pod status=CrashLoopBackOff |
NULL |      NULL | {"status": "CrashLoopBackOff"} |
| 744 | 2025-11-02 09:44:37 | default   | Pod  | mysql-cb6bfb5b-692bx            | WARN     | High Memory 538968064 bytes |
   9 | 538968064 | NULL                              |
| 745 | 2025-11-02 10:13:18 | default   | Pod  | badimage-demo-75f594976b-8h9fn  | CRITICAL | Pod status=ImagePullBackOff |
NULL |      NULL | {"status": "ImagePullBackOff"} |
| 746 | 2025-11-02 10:13:18 | default   | Pod  | crashloop-demo-758d8fc59d-4wssl | CRITICAL | Pod status=CrashLoopBackOff |
NULL |      NULL | {"status": "CrashLoopBackOff"} |
| 747 | 2025-11-02 10:13:18 | default   | Pod  | mysql-cb6bfb5b-692bx            | WARN     | High Memory 538968064 bytes |
   9 | 538968064 | NULL                              |
| 748 | 2025-11-02 10:14:55 | default   | Pod  | badimage-demo-75f594976b-n89b6  | CRITICAL | Pod status=ImagePullBackOff |
NULL |      NULL | {"status": "ImagePullBackOff"} |
| 749 | 2025-11-02 10:14:55 | default   | Pod  | crashloop-demo-758d8fc59d-rqzpf | CRITICAL | Pod status=CrashLoopBackOff |
NULL |      NULL | {"status": "CrashLoopBackOff"} |
| 750 | 2025-11-02 10:14:55 | default   | Pod  | mysql-cb6bfb5b-692bx            | WARN     | High Memory 538968064 bytes |
  10 | 538968064 | NULL                              |
| 751 | 2025-11-02 10:21:23 | default   | Pod  | badimage-demo-75f594976b-cq7z9  | CRITICAL | Pod status=ImagePullBackOff |
NULL |      NULL | {"status": "ImagePullBackOff"} |
```

Email:



Kubernetes Anomaly Summary Inbox ×

manohara.arm@gmail.com
to me

Kubernetes Anomaly Summary by Watsonx

Time	Namespace	Pod	Severity	Message
2025-11-02 11:19:02	default	crashloop-demo-758d8fc59d-mgr6s	CRITICAL	Pod status=CrashLoopBackOff
2025-11-02 11:19:02	default	mysql-cb6bfb5b-692bx	WARN	High Memory 538968064 bytes
2025-11-02 11:19:01	default	badimage-demo-75f594976b-z8nrt	CRITICAL	Pod status=ErrImagePull

Auto-remediation of Pod issues resolved by Watsonx

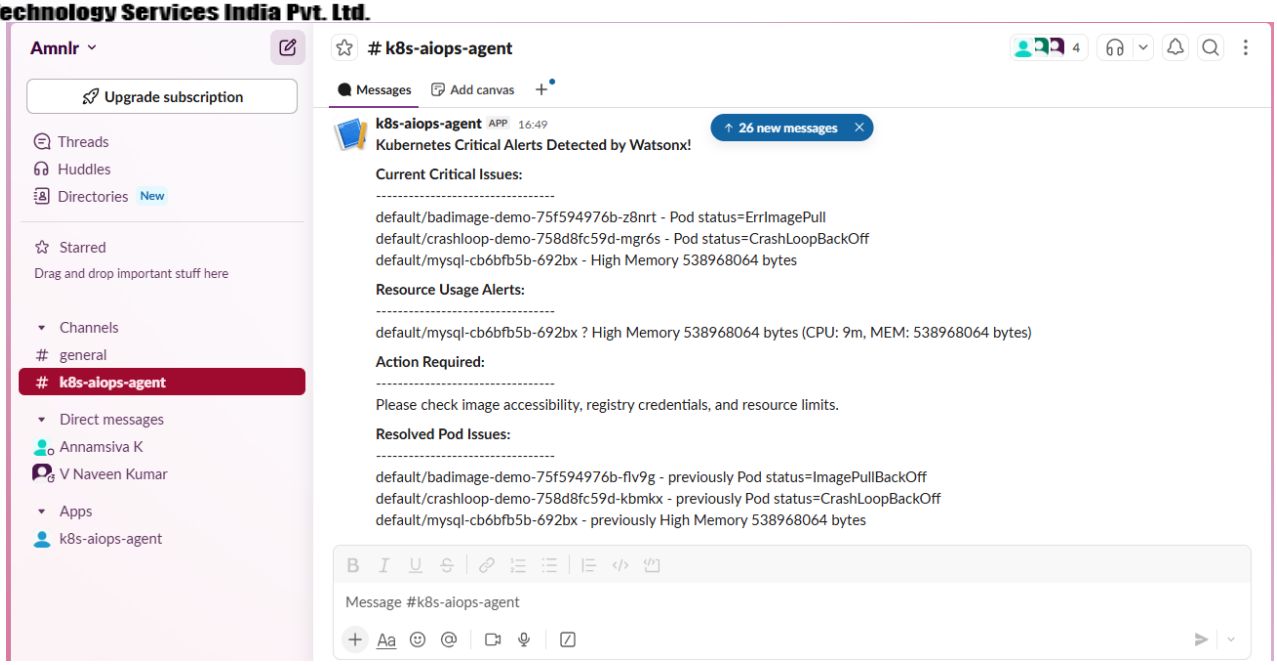
Namespace	Pod	Previous Status
default	badimage-demo-75f594976b-flv9g	Pod status=ImagePullBackOff
default	crashloop-demo-758d8fc59d-kbmkx	Pod status=CrashLoopBackOff
default	mysql-cb6bfb5b-692bx	High Memory 538968064 bytes

Resource Usage Alerts (CPU / Memory)

Namespace	Pod	Severity	Alert	Details
default	mysql-cb6bfb5b-692bx	WARN	High Memory 538968064 bytes	CPU: 9m Memory: 538968064 bytes

Slack:

MasterCom
Technology Services India Pvt. Ltd.



Kubernetes Anomaly Summary.e