Deployment Task Report: AI Microservices Deployment on OpenShift

Project Overview

We have developed a set of three AI microservices designed to handle text processing tasks. Your responsibility will be to deploy these services on an OpenShift cluster, ensuring that the cluster is built from scratch on a virtual machine (VM) and that all services are properly configured, scaled, and operational.

Microservices Description

1. Sentiment Analysis Service

- **Function**: Processes text input to determine the sentiment (Positive, Negative, or Neutral).
- **Endpoint**: `/analyze_sentiment`
- **Port**: `8001`

2. Named Entity Recognition (NER) Service

- **Function**: Extracts named entities (e.g., people, organizations, dates) from the provided text.
- **Endpoint**: `/extract_entities`
- **Port**: `8002`

3. Proxy Service

- **Function**: Acts as a gateway. Accepts text input, sequentially sends it to the Sentiment Analysis and NER services, and returns a consolidated response.
- **Endpoint**: `/process_text`
- **Port**: `8003`

Task Instructions

Your task is to deploy the above services on an OpenShift cluster following the steps below. Please ensure each part of the task is completed and documented.

Step 1: OpenShift Cluster Setup

- Provision a virtual machine (VM) and set up an OpenShift cluster on it. The VM can be hosted on a cloud provider (e.g., AWS, Azure, Google Cloud) or on a local machine using a hypervisor (e.g., VirtualBox, VMWare).
- The VM should meet the following minimum specifications:
    - **CPU**: 4 vCPUs
    - **Memory**: 8GB RAM
    - **Storage**: 50GB disk space
- Ensure the VM has access to the internet to pull Docker images and OpenShift components.

Step 2: Service Deployment

- Containerize each microservice using Docker and deploy them on the OpenShift cluster.
- Configure the services to be accessible via the following endpoints:
    - **Sentiment Analysis Service**: Port `8001`
    - **NER Service**: Port `8002`
    - **Proxy Service**: Port `8003`
- Set up appropriate routes in OpenShift to expose these services to external clients.
- Verify that each service is accessible and functioning as expected.

Step 3: Implement Auto-Scaling

- Configure auto-scaling for each service to optimise resource usage.
- Ensure the services automatically scale based on CPU utilisation or request load.
- Document the configuration and triggers used for auto-scaling.

Step 4: Documentation and Reporting

- Provide a comprehensive report detailing:
    1. **Cluster Deployment**: The steps you took to build the OpenShift cluster, including screenshots and relevant commands used.
    2. **Service Deployment**: The process of containerizing and deploying each service, along with how you exposed and tested them.
    3. **Auto-Scaling Configuration**: Explanation of how auto-scaling is set up, how it triggers, and how it optimises resource usage.
    4. **Testing**: Demonstrate that all services are deployed correctly and are functional by providing test results (e.g., sample requests and responses).

Expected Deliverables

1. **Detailed Documentation** of the entire process, from cluster setup to service deployment and auto-scaling configuration.
2. **Screenshots and Logs** demonstrating that the services are up and running, including test results showing successful interactions between the services.
3. **Configuration Files** used for the deployment (e.g., YAML files for OpenShift, Dockerfiles, etc.).
4. **Auto-scaling Report** detailing how the auto-scaler is configured and the conditions under which it triggers.
5. **Recommendations and Justifications** if an alternative deployment strategy was used.

Final Note

Please note that while the initial task specifies using OpenShift for the deployment, the primary goal of this exercise is to assess how well you can optimize resources for the best performance with minimal overhead.

If you believe there is a better deployment strategy that would improve resource efficiency, sharing, and management, you are free to select another option (e.g., Kubernetes, Docker Swarm, or any cloud-native solution).

In the event you choose an alternative approach:

- Ensure you clearly document **why** you opted for the new deployment strategy.
- Provide a detailed explanation of how it better utilizes resources compared to OpenShift.
- Highlight any improvements in performance or scalability achieved by your approach.

This task is intended to evaluate your ability to optimize for **resource efficiency and performance**, so we encourage you to think critically about the best solution.

Thank you, and we look forward to seeing your implementation!