

DevOps Exam - Interview Task

Task: Kafka-Enabled HealthCheckService Deployment on Kubernetes

Scenario:

You are working for a company that is building a microservices architecture with Kafka as the messaging backbone. The team is responsible for deploying and managing these services on a Kubernetes cluster.

Task Requirements:

Set up Local Kubernetes Cluster:

Install and set up a local Kubernetes cluster using a tool like Minikube. Ensure that kubectl is properly configured to interact with the local cluster.

Set up Kafka Cluster on Kubernetes:

Use Helm or Kubernetes manifests to deploy a Kafka cluster with at least 3 nodes within the Kubernetes environment.

Create a topic named **health_checks_topic** with appropriate configurations.

Python HealthCheckService:

Write a Python service named HealthCheckService that periodically performs health checks on various microservices in the system.

The service should have a REST API endpoint `/check_health` that retrieves the health status of different microservices from the Kafka topic (**health_checks_topic**) and prints the results along with some text to the logs.

JSON Payload example:

```
{
  "service_name": "MyService",
  "status": "OK",
  "timestamp": "2024-01-01T12:30:45Z"
}
```

ConsumerHealthCheckService:

Write another Python service named ConsumerHealthCheckService that consumes health check messages from the **health_checks_topic**.

The service should have a REST API endpoint `/get_latest_health_check` that retrieves and prints the latest health check results from the Kafka topic.

Deployment Automation:

Create Kubernetes deployment manifests for both the HealthCheckService and ConsumerHealthCheckService. Implement a rolling deployment strategy for these services.

Use ConfigMaps/Secrets to manage any configuration that needs to be externalized. Ensure that the services can scale horizontally.

Monitoring and Logging

Set up monitoring for the Kafka cluster, HealthCheckService, and ConsumerHealthCheckService. Implement logging for both services, including printing health check results along with some text.

Share a GitHub repository with the code and Kubernetes manifests.

Upload all code files, including Python scripts, deployment manifests, and any configuration files, to the repository.

Include a directory or folder for screenshots or documentation images.

Provide a link to the repository, ensuring that it is set to public.

Good Luck!