Microservices Kubernetes Deployment Documentation

1. Introduction

This document provides detailed instructions for deploying a microservices-based application on Kubernetes using Minikube.

The microservices architecture consists of the following services:

- User Service (Port 3000)
- Product Service (Port 3001)
- Order Service (Port 3002)
- Gateway Service (Port 3003)

Each service is deployed on Kubernetes using appropriate configurations such as deployment, services, and ingress resources. Communication between services is managed through Kubernetes services and Ingress routing.

2. Prerequisites

Before deploying the application, ensure the following tools are installed:

- Minikube (to run Kubernetes locally)
- kubectl (the Kubernetes CLI)
- Docker (optional, if building container images locally)

Ensure that Minikube is set up and running by following the steps below.

3. Minikube Setup Instructions

3.1. Starting Minikube

1. Initialize Minikube:
   ```bash
   minikube start
   ```

2. Enable the Ingress controller:

```bash
minikube addons enable ingress
```

3. Check if the Ingress controller is running:
```bash
kubectl get pods -n kube-system
```

You should see the `nginx-ingress-controller` pod running.

4. Deployment Steps

4.1. Deploying Microservices

The application is composed of four microservices: User Service, Product Service, Order Service, and Gateway Service. Deployment YAML files are located in the `deployments/` directory.

- User Service: `deployments/user-service.yaml`
- Product Service: `deployments/product-service.yaml`
- Order Service: `deployments/order-service.yaml`
- Gateway Service: `deployments/gateway-service.yaml`

To deploy these services, run the following commands:

1. Deploy User Service:
```bash
kubectl apply -f deployments/user-service.yaml
```

2. Deploy Product Service:
```bash
kubectl apply -f deployments/product-service.yaml
```

3. Deploy Order Service:
```bash
```

kubectl apply -f deployments/order-service.yaml
```

4. Deploy Gateway Service:
```bash
kubectl apply -f deployments/gateway-service.yaml
```

4.2. Deploying Services

To expose these microservices internally, apply the corresponding Kubernetes service definitions:

```bash
kubectl apply -f services/user-service.yaml
kubectl apply -f services/product-service.yaml
kubectl apply -f services/order-service.yaml
kubectl apply -f services/gateway-service.yaml
```

5. Ingress Configuration

To configure Ingress, apply the following Ingress YAML:

kubectl apply -f ingress/ingress.yaml

The ingress.yaml should include routing rules:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: microservices-ingress
spec:
  rules:
  - host: microservices.local
    http:
      paths:
      - path: /api/users
        pathType: Prefix
        backend:
          service:
            name: user-service
            port:
```

```
          number: 3000
    - path: /api/products
      pathType: Prefix
      backend:
        service:
          name: product-service
          port:
            number: 3001
    - path: /api/orders
      pathType: Prefix
      backend:
        service:
          name: order-service
          port:
            number: 3002
    - path: /
      pathType: Prefix
      backend:
        service:
          name: gateway-service
          port:
            number: 3003
```

To test the Ingress setup:

echo "$(minikube ip) microservices.local" | sudo tee -a /etc/hosts

Now, you should be able to access the services through Ingress:

- http://microservices.local/api/users
- http://microservices.local/api/products
- http://microservices.local/api/orders
- http://microservices.local/

---

6. Testing and Validation

6.1. Validate Deployment

Check the running pods:

kubectl get pods

Ensure all services are running correctly.

6.2. Verify Service Communication

Check if services can communicate within the cluster:

kubectl exec -it <any-pod-name> -- curl http://user-service:3000
kubectl exec -it <any-pod-name> -- curl http://product-service:3001
kubectl exec -it <any-pod-name> -- curl http://order-service:3002
kubectl exec -it <any-pod-name> -- curl http://gateway-service:3003

---

7. Troubleshooting Guide

7.1. Common Issues and Fixes

1. **Pods not starting**:
   - Check logs: kubectl logs <pod-name>
   - Check status: kubectl describe pod <pod-name>
2. **Ingress not working**:
   - Ensure Ingress is enabled: minikube addons list
   - Verify Ingress configuration: kubectl get ingress
3. **Service communication issues**:
   - Test connectivity with: kubectl exec -it <pod-name> -- curl http://<service-name>:<port>

---

8. Deliverables

Ensure the following files are included in your submission:

```
submission/
├── deployments/
│   ├── user-service.yaml
│   ├── product-service.yaml
│   ├── order-service.yaml
│   └── gateway-service.yaml
├── services/
│   ├── user-service.yaml
│   ├── product-service.yaml
│   ├── order-service.yaml
│   └── gateway-service.yaml
├── ingress/
│   └── ingress.yaml
└── README.md
```

Include **screenshots** of:

- Running pods (kubectl get pods)
- Successful ingress access (curl http://microservices.local)
- Service communication tests