MLOPS ASSIGNMENT 2

November 1,2024

Name: Asim Sharif
Roll No: 19i-0518
Section: AI-J

Table of Contents

1. System Architecture

1.1 Overview

The system is a microservices-based authentication system consisting of four main components:

- **Frontend Service**
- **Backend Service**
- **Authentication Service**
- **Database Service**

1.2 Component Details

1.2.1 Frontend Service

- **Technology**: React.js
- **Container**: Nginx
- **Scale**: 3 replicas
- **Purpose**: Manages user interface and interaction

1.2.2 Backend Service

- **Technology**: Express.js
- **Scale**: 3 replicas
- **Features**:
    - REST API endpoints
    - Handles business logic
    - Database communication
    - Processes authentication

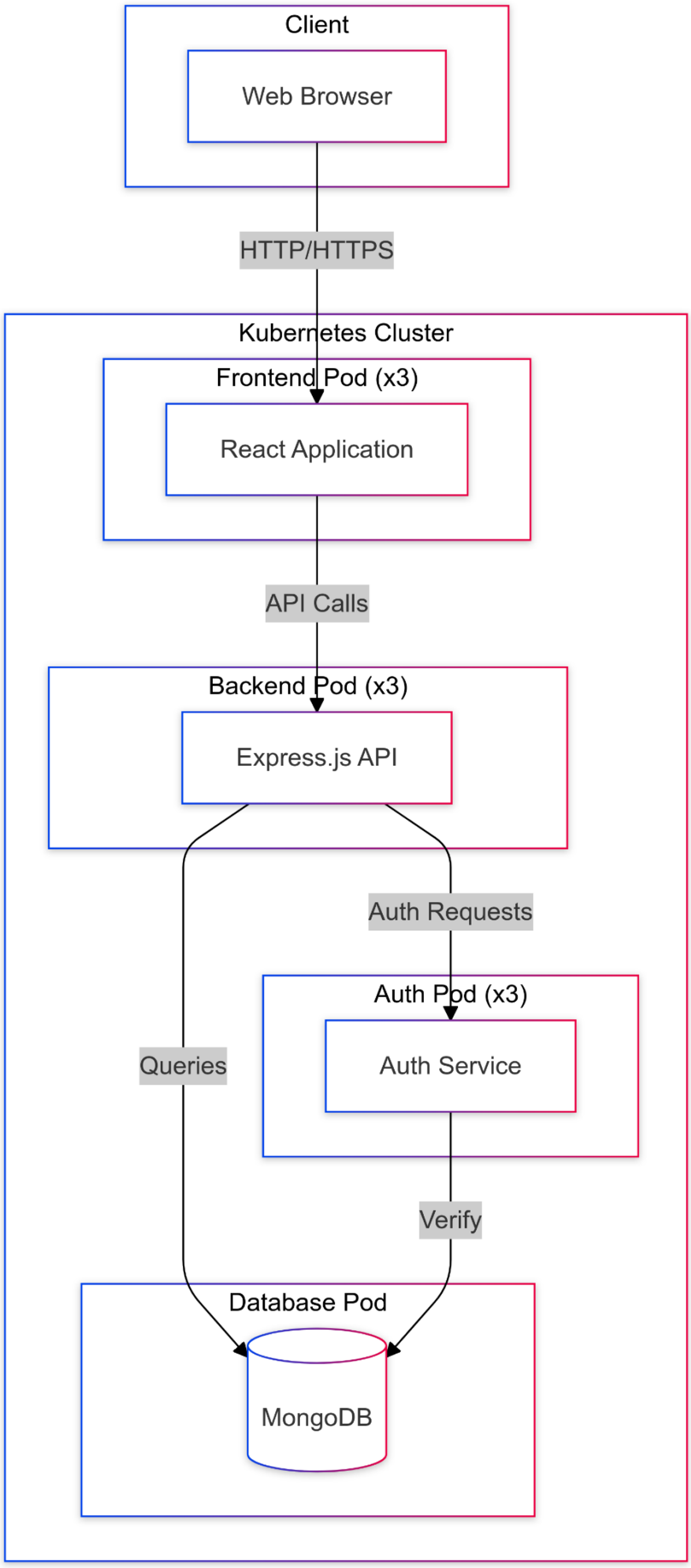
1.2.3 Authentication Service

- **Features**:
    - JWT token management
    - Password encryption
    - Session handling
- **Security**: bcrypt hashing

1.2.4 Database Service

- **Technology**: MongoDB
- **Persistence**: Kubernetes PVC
- **Data**: Stores user information and authentication data

1.2.5 Architecture Diagram


Client
Web Browser
HTTP/HTTPS
Kubernetes Cluster
Frontend Pod (x3)
React Application
API Calls
Backend Pod (x3)
Express.js API
Auth Requests
Auth Pod (x3)
Auth Service
Queries
Verify
Database Pod
MongoDB

2. Prerequisites

2.1 Required Software

- Docker Desktop
- Minikube v1.34.0 or higher
- kubectl CLI tool
- Node.js v16 or higher

2.2 System Requirements

- Minimum 8GB RAM
- 20GB free disk space
- x86_64 processor

---

3. Deployment Process

3.1 Initial Setup

```
# Clone repository
git clone <repository-url>
cd mlops-auth-app
```

3.2 Building Images

```
# Build all services
docker-compose build
```

3.3 Kubernetes Deployment

```
# Start Minikube
minikube start --driver=docker

# Load images
minikube image load mlops-auth-app-frontend:latest
minikube image load mlops-auth-app-backend:latest

# Deploy services
kubectl apply -f k8s/mongodb-deployment.yaml
kubectl apply -f k8s/backend-deployment.yaml
kubectl apply -f k8s/frontend-deployment.yaml
```

---

4. Access Instructions

4.1 Frontend Access

minikube service frontend-service

- **Default URL**: http://localhost:30000

4.2 Backend Service

- **Service Name**: backend-service
- **Port**: 3001
- **URL**: http://backend-service:3001

4.3 MongoDB Access

- **Service**: mongodb-service
- **Port**: 27017
- **Connection String**: mongodb://mongodb-service:27017/auth-db

---

5. Troubleshooting Guide

5.1 Common Issues

Pod Startup Issues

```
# Check pod status
kubectl get pods
kubectl describe pod <pod-name>
```

Service Connection Issues

```
# Verify services
kubectl get services
kubectl describe service <service-name>
```

Application Errors

```
# View logs
kubectl logs -f deployment/frontend
kubectl logs -f deployment/backend
```

5.2 Health Monitoring

```
# System status
kubectl get deployments
kubectl get pods -o wide
kubectl get services
```

6. Maintenance and Scaling

6.1 Scaling Services

```
# Scale deployments
kubectl scale deployment frontend --replicas=5
kubectl scale deployment backend --replicas=5
```

6.2 Updates and Upgrades

```
# Update deployments
kubectl set image deployment/frontend frontend=mlops-auth-app-frontend:new-version
kubectl set image deployment/backend backend=mlops-auth-app-backend:new-version
```

6.3 Backup Procedures

- **MongoDB PVC backup recommended**
- **Schedule regular backups**
- **Verify backup integrity**

---

7. Security Considerations

7.1 Authentication Security

- **JWT token expiration**: 1 hour
- **Password hashing**: bcrypt
- **Protected endpoints**
- **CORS configuration**

7.2 Environment Security

- Use environment variables
- Secure secrets management
- Regular security updates
- Access control implementation

---

8. Monitoring and Logging

8.1 Kubernetes Dashboard

```
minikube dashboard
```

8.2 Resource Monitoring

```
# Real-time monitoring
kubectl get pods -w
kubectl get services -w
```

8.3 Log Management

```
# Access logs
kubectl logs -f deployment/frontend
kubectl logs -f deployment/backend
```

---

9. Quick Reference Commands

9.1 Deployment Commands

```
# Start system
minikube start --driver=docker
kubectl apply -f k8s/

# Stop system
kubectl delete -f k8s/
minikube stop
```

9.2 Monitoring Commands

```
# Status checks
kubectl get pods
kubectl get services
kubectl get deployments
```

9.3 Troubleshooting Commands

```
# Debugging
kubectl describe pod <pod-name>
kubectl logs <pod-name>
kubectl get events
```