SOFTWARE ENGINEERING PROJECT BY

ISHITA BHANDARI 22BCE2313

KRISTINE VARGESE 22BCE0383

Todo List App

A Kubernetes-deployed Todo List application with a Python FastAPI backend.

Overview

This project implements a Todo List application deployed on Kubernetes. It features a FastAPI backend service with health checks, secure endpoints, and is configured for production deployment with proper resource management and scaling.

Architecture

- **Backend**: Python FastAPI application
- **Deployment**: Kubernetes with configured resources, probes, and autoscaling
- **Configuration**: Environment variables via ConfigMaps and Secrets

Project Structure

```
todo_list_app/
 ├── kubeapp/                 # Python application
 │   ├── pyproject.toml          # Python project dependencies
 │   └── main.py              # FastAPI application code
 ├── kubernetes/              # Kubernetes manifests
 │   ├── namespace.yaml          # Namespace definition
 │   ├── configmap.yaml          # Configuration settings
 │   ├── secret.yaml             # Secure credentials
 │   ├── deployment.yaml         # Pod deployment configuration
 │   ├── service.yaml            # Service definition
 │   └── hpa.yaml              # Horizontal Pod Autoscaler
```

```
└── README.md                # Project documentation
```

API Endpoints

Method	Endpoint	Description	Authentication
GET	/health	Health check endpoint	None
GET	/secret/access	Secure endpoint requiring API key	X-API-Secret header
GET	/heavy	Endpoint simulating intensive computation	None

Tech Stack

- **Python 3.9+**
- **FastAPI** - Modern, fast web framework
- **Uvicorn** - ASGI server
- **Kubernetes** - Container orchestration

Kubernetes Features

- Namespace isolation (kubeapp-dev)
- Rolling update deployment strategy
- Resource requests and limits
- Horizontal Pod Autoscaler (2-5 replicas based on CPU usage)
- Health probes
- ConfigMaps and Secrets for configuration
- NodePort service exposure

Getting Started

Prerequisites

- Kubernetes cluster (Minikube, kind, or cloud provider)
- kubectl configured
- Docker

Deployment

1. Create the namespace:

bash

```
kubectl apply -f kubernetes/namespace.yaml
```

2. Apply ConfigMap and Secret:

bash

```
kubectl apply -f kubernetes/configmap.yaml
kubectl apply -f kubernetes/secret.yaml
```

3. Deploy the application:

bash

```
kubectl apply -f kubernetes/deployment.yaml
```

4. Create the service:

bash

```
kubectl apply -f kubernetes/service.yaml
```

5. Enable autoscaling:

bash

```
kubectl apply -f kubernetes/hpa.yaml
```

Accessing the Application

The application is exposed on port 30080 via NodePort:

bash

```
# If using Minikube
minikube service kubeapp-service -n kubeapp-dev


# If using other Kubernetes setup
# Access via NODE_IP:30080
```

Local Development

1. Install dependencies:

bash

```
pip install fastapi uvicorn
```

2. Run the application:

bash

```
uvicorn main:app --reload
```

Environment Variables

Variable	**Description**	**Default**
ENV	Environment name	"dev"
SecretKey	API key for secure routes	""
LOG_LEVEL	Logging level	"INFO"