Project Manual: Building a Scalable Microservice with Kubernetes

1. Introduction

This manual guides students through the process of designing, containerizing, and deploying a microservice on Kubernetes. By the end, students will have a working cloud-native application with scalability and monitoring features.

2. Prerequisites

Before starting, ensure you have:
✔ **Basic programming knowledge** (Python/Node.js/Java)
✔ **Docker** installed (Installation Guide)
✔ **Minikube/Kind** for local Kubernetes (Minikube Guide)
✔ **kubectl** CLI (Installation Guide)
✔ (Optional) A **cloud account** (AWS/GCP/Azure) for cloud-based Kubernetes

3. Step-by-Step Implementation

Phase 1: Application Design

Task: Develop a simple microservice (e.g., To-Do List API).

Steps:

1. Choose a backend framework:

    - **Python:** Flask/Django
    - **Node.js:** Express
    - **Java:** Spring Boot

2. Define API endpoints (example for a To-Do app):

```
GET /tasks          → List all tasks
POST /tasks         → Add a new task
GET /tasks/{id}     → Get a task by ID
DELETE /tasks/{id} → Delete a task
```

3. Test locally using `curl` or Postman.

Phase 2: Containerization with Docker

Task: Package the app into a Docker container.

Steps:

1. Write a `Dockerfile`:

```
# Example for Python/Flask
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
```

```
COPY . .
CMD ["flask", "run", "--host=0.0.0.0"]
```

2. Build and run the image:

   ```
   docker build -t todo-app .
   ```

   docker run -p 5000:5000 todo-app

3. Push to **Docker Hub**:

```
docker tag todo-app yourusername/todo-app

docker push yourusername/todo-app
```

Phase 3: Kubernetes Deployment

Task: Deploy the app on Kubernetes.

Steps:

1. Start Minikube:

   ```
   minikube start
   ```

2. Create Kubernetes manifests:

   ○ **Deployment** (`deployment.yaml`):

     ```
     apiVersion: apps/v1
     ```

   ○ `kind: Deployment`
   ○ `metadata:`
   ○ `  name: todo-app`
   ○ `spec:`
   ○ `  replicas: 2`
   ○ `  selector:`
   ○ `    matchLabels:`

```
○       app: todo
○   template:
○     metadata:
○       labels:
○         app: todo
○     spec:
○       containers:
○       - name: todo
○         image: yourusername/todo-app
○         ports:
○         - containerPort: 5000
```

- **Service** (`service.yaml`):

```
apiVersion: v1
○ kind: Service
○ metadata:
○   name: todo-service
○ spec:
○   selector:
○     app: todo
○   ports:
○     - protocol: TCP
○       port: 80
○       targetPort: 5000
○   type: LoadBalancer  # Use NodePort for Minikube
```

3. Apply the configurations:

```
kubectl apply -f deployment.yaml
kubectl apply -f service.yaml
```

4. Access the service:

```
minikube service todo-service  # For Minikube
```

Phase 4: Scaling & Monitoring

Task: Ensure the app scales and is monitored.

Steps:

1. **Manual Scaling:**

```
kubectl scale deployment todo-app --replicas=3
```

2. **Autoscaling (HPA):**

```
kubectl autoscale deployment todo-app --cpu-percent=50 --
min=2 --max=5
```

3. **Liveness/Readiness Probes** (Add to `deployment.yaml`):

```
livenessProbe:
  httpGet:
    path: /health
    port: 5000
  initialDelaySeconds: 5
  periodSeconds: 10
```

Phase 5: CI/CD Pipeline

Task: Automate deployments using GitHub Actions.

Steps:

1. Create `.github/workflows/deploy.yml`:

```yaml
name: Deploy to Kubernetes

on: [push]
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - run: kubectl apply -f deployment.yaml
```

4. Deliverables Checklist

- **Source Code** (GitHub repo with `Dockerfile`, Kubernetes manifests).
- **Documentation** (README with setup instructions).

5. Troubleshooting

Issue	Solution
Minikube not starting	Run `minikube delete && minikube start`
Image pull errors	Check Docker Hub permissions
Pods crashing	Debug logs: `kubectl logs <pod>`