**https://github.com/suvaatnbu/SKILL-TEST2.git**

Microservices Kubernetes Deployment Assessment ([Link](https://github.com/mohanDevOps-arch/Microservices-Task.git))

**Total Marks: 50  
Time Limit: 1 hour**

Objective

**Deploy a microservices application on Kubernetes using Minikube, ensuring proper service communication and configuration.**

Application Components

**You are provided with four containerized Node.js microservices:**

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

Task Requirements

1. Basic Kubernetes Deployment (30 marks)

A. Create Kubernetes Deployment manifests for all services (18 marks)

* **User Service deployment**
* **Product Service deployment**
* **Order Service deployment**
* **Gateway Service deployment  (includes additional proxy handling setup if required)**

**Each deployment must include:**

* **Correct container image reference**
* **Resource limits and requests**
* **Environment variables**
* **Liveness and readiness probes**
* **Proper labels and selectors**

      B. Create corresponding Service resources (12 marks)

* **Configure correct ports**
* **Choose proper service types**
* **Enable cluster-level service discovery using ClusterIP**

2. Minikube Setup and Validation (15 marks)

* **Initialize and configure Minikube**
* **Deploy all components successfully**
* **Validate inter-service communication using curl or logs**

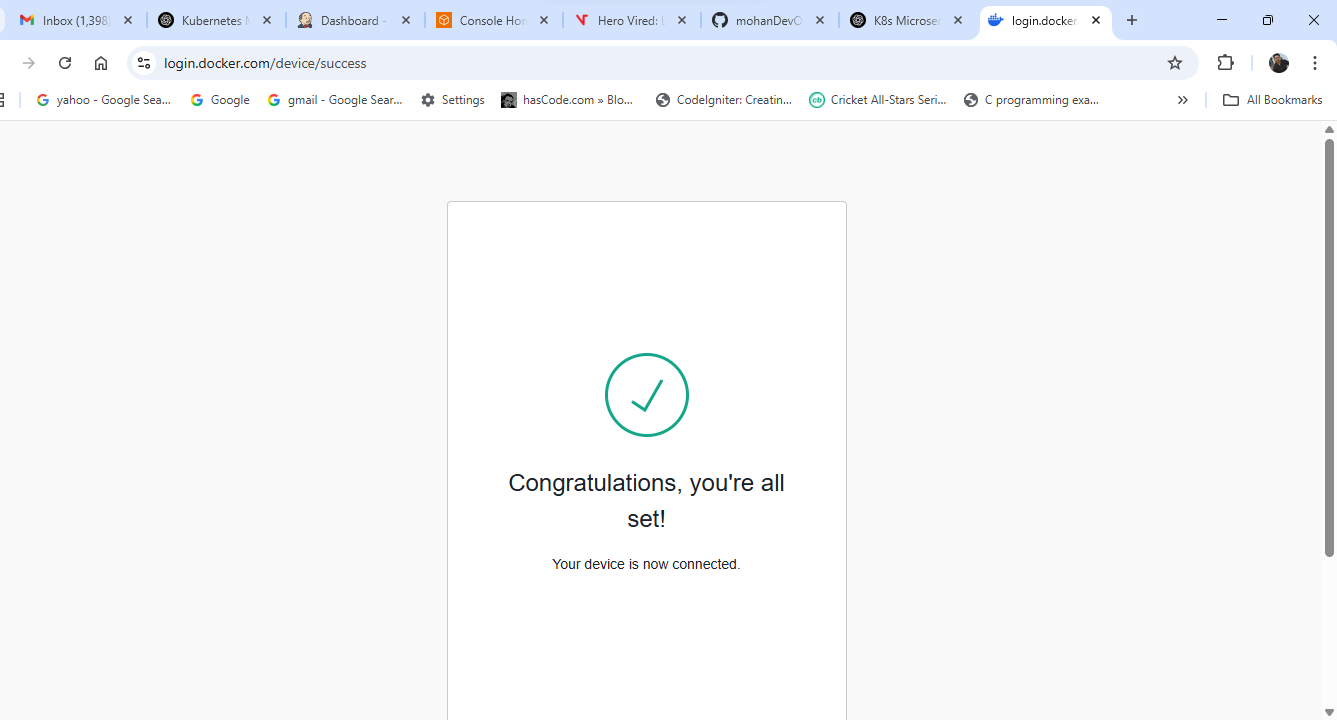
3. Documentation and Testing (5 marks)

**Provide a well-structured README.md with:**

* **Minikube setup steps**
* **Deployment process using kubectl apply -f**
* **Service testing instructions using kubectl port-forward or direct service names**
* **Troubleshooting tips**

**Include screenshots of:**

* **Running pods (kubectl get pods)**
* **Logs showing service communication**
* **Any port-forwarded test results**



**Microservices Kubernetes Deployment (Minikube)**

This document guides you through deploying a Node.js microservices application to Kubernetes using Minikube. It includes setup instructions, deployment commands, service testing, and troubleshooting tips.

**📦 Application Overview**

The application consists of four microservices:

| **Service Name** | **Port** | **Description** |
| --- | --- | --- |
| User Service | 3000 | Handles user data |
| Product Service | 3001 | Manages product information |
| Order Service | 3002 | Handles order transactions |
| Gateway Service | 3003 | API gateway / proxy router |

**🛠️ Minikube Setup**

1. **Start Minikube:**

minikube start

1. **Enable Ingress (optional for advanced routing):**

minikube addons enable ingress

**🚀 Deployment Steps**

1. **Clone this repo or copy YAML files to a directory.**
2. **Apply Deployments and Services:**

kubectl apply -f user-deployment.yaml

kubectl apply -f user-service.yaml

kubectl apply -f product-deployment.yaml

kubectl apply -f product-service.yaml

kubectl apply -f order-deployment.yaml

kubectl apply -f order-service.yaml

kubectl apply -f gateway-deployment.yaml

kubectl apply -f gateway-service.yaml

1. **Check Pod Status:**

kubectl get pods

1. **Check Services:**

kubectl get svc

**🔁 Testing Inter-Service Communication**

**Option 1: Using Port Forwarding**

kubectl port-forward service/gateway-service 3003:3003

Then test with:

curl http://localhost:3003/user

curl http://localhost:3003/product

curl http://localhost:3003/order

**Option 2: Using Pod Networking**

kubectl exec -it <pod-name> -- sh

# Inside the pod:

curl http://user-service:3000/health

curl http://product-service:3001/health

**🧪 Example Screenshots to Include**

* Output of:
* kubectl get pods
* kubectl logs <gateway-pod>
* curl results from port-forwarded endpoint

**⚠️ Troubleshooting Tips**

* **Pods not starting?** Run:
* kubectl describe pod <pod-name>
* kubectl logs <pod-name>
* **Check image pull issues** if using private DockerHub or custom images.
* **Check service name mismatches** for inter-service curl commands.
* **Ensure port numbers in deployment and service YAML match.**

**✅ Cleanup (After Test)**

kubectl delete -f .

minikube delete

