🚀 Oberon App Deployment with Docker, Kubernetes & LocalTunnel

\*by dcjoncas\*

# 📄 Overview

This document summarizes how the Oberon Pro app was containerized, deployed on Kubernetes with two pods for scalability and high availability, and exposed securely to the internet using LocalTunnel. It also documents the troubleshooting steps and key learnings throughout the process.

# 📋 Deployment Steps

## ✅ Step 1 — Containerize your app with Docker

• Created a Dockerfile for the Flask app.  
• Built the Docker image:

docker build -t oberon\_pro .

• Tested locally:

docker run -p 5000:5000 oberon\_pro

• App was accessible at: http://localhost:5000

## ✅ Step 2 — Push your image to Docker Hub

• Logged in to Docker Hub (dcjoncas) using a Personal Access Token.  
• Tagged the image:

docker tag oberon\_pro dcjoncas/oberon\_pro:latest

• Pushed to Docker Hub:

docker push dcjoncas/oberon\_pro:latest

• Verified the image at: https://hub.docker.com/r/dcjoncas/oberon\_pro

## ✅ Step 3 — Deploy to Kubernetes

• Enabled Kubernetes in Docker Desktop.  
• Created two manifests:

* deployment.yaml

apiVersion: apps/v1  
kind: Deployment  
metadata:  
 name: oberon-deployment  
spec:  
 replicas: 2  
 selector:  
 matchLabels:  
 app: oberon  
 template:  
 metadata:  
 labels:  
 app: oberon  
 spec:  
 containers:  
 - name: oberon  
 image: dcjoncas/oberon\_pro:latest  
 ports:  
 - containerPort: 5000

* service.yaml

apiVersion: v1  
kind: Service  
metadata:  
 name: oberon-service  
spec:  
 type: NodePort  
 selector:  
 app: oberon  
 ports:  
 - protocol: TCP  
 port: 5000  
 targetPort: 5000  
 nodePort: 30001

• Applied both:

kubectl apply -f deployment.yaml

kubectl apply -f service.yaml

• Confirmed with: kubectl get pods, kubectl get service oberon-service  
• Local address: http://localhost:30001

## ✅ Step 4 — Expose publicly using LocalTunnel

• Ran LocalTunnel:

lt --port 30001

• Got public URL like: https://fancy-name.loca.lt

# 🔷 Troubleshooting Notes

* 🚫 Pods failed with ErrImagePull

Cause: Kubernetes couldn’t find your local Docker image.  
Solution: Pushed the image to Docker Hub and updated deployment.yaml.

* 🚫 Docker login failed (401 Unauthorized)

Cause: CLI device login flow buggy on Windows.  
Solution: Used Personal Access Token with username & token as password.

* 🚫 Service not accessible

Cause: Confusion between internal port and NodePort.  
Solution: Checked NodePort in kubectl get service and used correct URL.

* 🚫 PowerShell errors on pod names

Cause: Pod names are not commands.  
Solution: Use pod names with kubectl commands like logs, describe, delete.

* 🚫 Specific user to pod?

Not by default. Kubernetes load-balances requests (round-robin).

# 🌟 Key Learnings

|  |  |
| --- | --- |
| Component | Purpose |
| Docker | Packages app to run anywhere |
| Docker Hub | Stores image in cloud |
| Kubernetes | Runs, manages, and scales app |
| NodePort | Exposes pods to host network |
| LocalTunnel | Makes app public on the internet |

# 📝 Final Summary

✅ Containerized the app with Docker  
✅ Pushed to Docker Hub (dcjoncas/oberon\_pro:latest)  
✅ Deployed two pods on Kubernetes for high availability  
✅ Verified pods & Service are running  
✅ Used LocalTunnel to expose app securely online

# 🔗 Access

Local: http://localhost:30001  
LAN: http://<your-lan-ip>:30001  
Internet: https://fancy-name.loca.lt