The Whether App

**🌦️ WeatherAPP - Project Overview :**

**Weather** is a **cloud-native microservices application** deployed on a **local Kubernetes cluster** using **KIND (Kubernetes in Docker)**. It is designed as a modular and educational project that demonstrates how to build, deploy, and orchestrate microservices on Kubernetes in a local development environment.

This project is perfect for learning and experimenting with:

* Kubernetes resources (Pods, Services, Deployments, Stateful Sets , PV , PVC , secret , ingress)
* Service-to-service communication
* Stateless and stateful microservices
* Secure APIs using JWT (Json web token ) authentication
* External API consumption
* Persistent data storage using MySQL

**✨First building a Kubernetes cluster using KIND with Ingress controller :**

**---**

**kind delete cluster**

**cat <<EOF | kind create cluster --config=-**

**kind: Cluster**

**apiVersion: kind.x-k8s.io/v1alpha4**

**nodes:**

**- role: control-plane**

**kubeadmConfigPatches:**

**- |**

**kind: InitConfiguration**

**nodeRegistration:**

**kubeletExtraArgs:**

**node-labels: "ingress-ready=true"**

**extraPortMappings:**

**- containerPort: 80**

**hostPort: 80**

**protocol: TCP**

**- containerPort: 443**

**hostPort: 443**

**protocol: TCP**

**EOF**

**---**

// install ingress controller

# kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/main/deploy/static/provider/kind/deploy.yaml

2- project Source code :

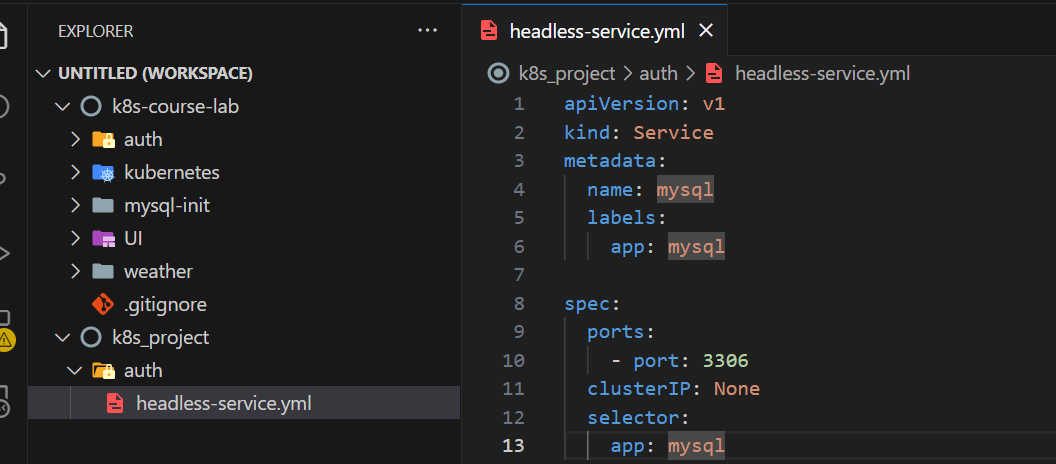
# git clone <https://github.com/abohmeed/k8s-course-lab.git>

3- create directory k8s Project :

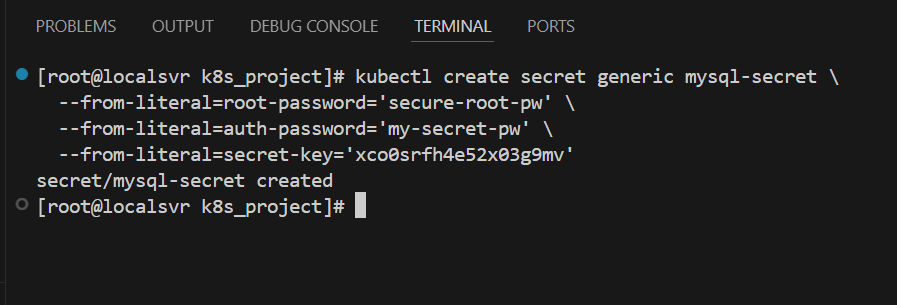
// directory structure :

1- Deploy MySQL DB [stateful set ] :

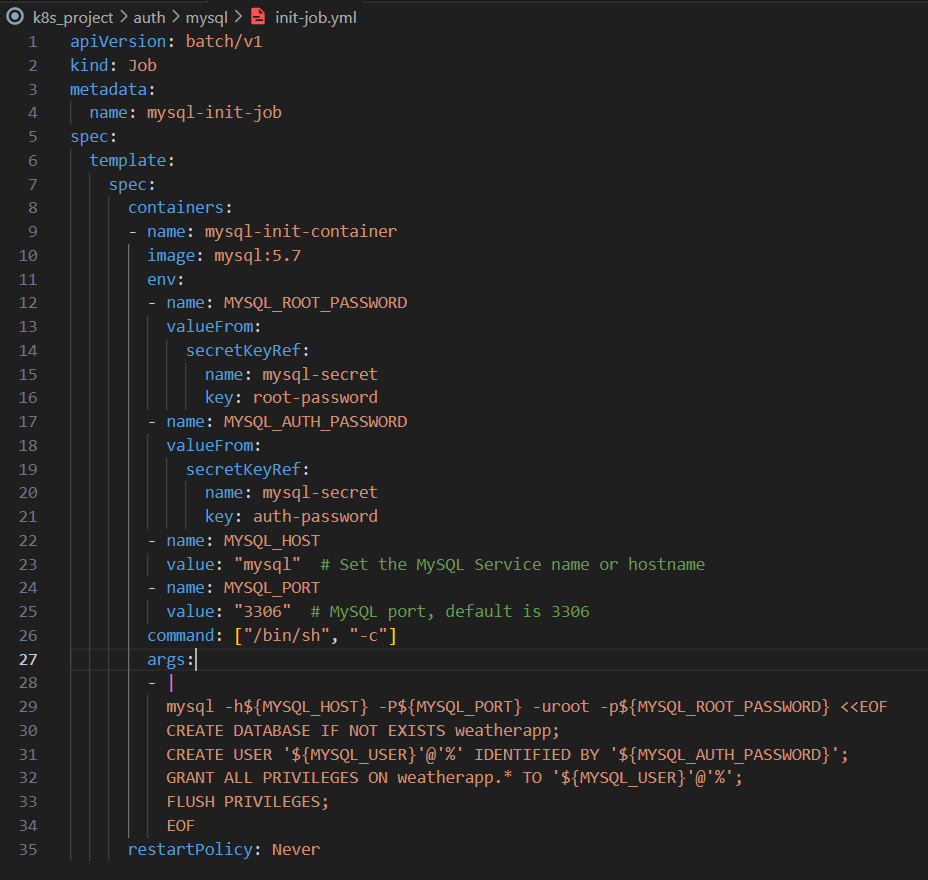
1. Create Headless service



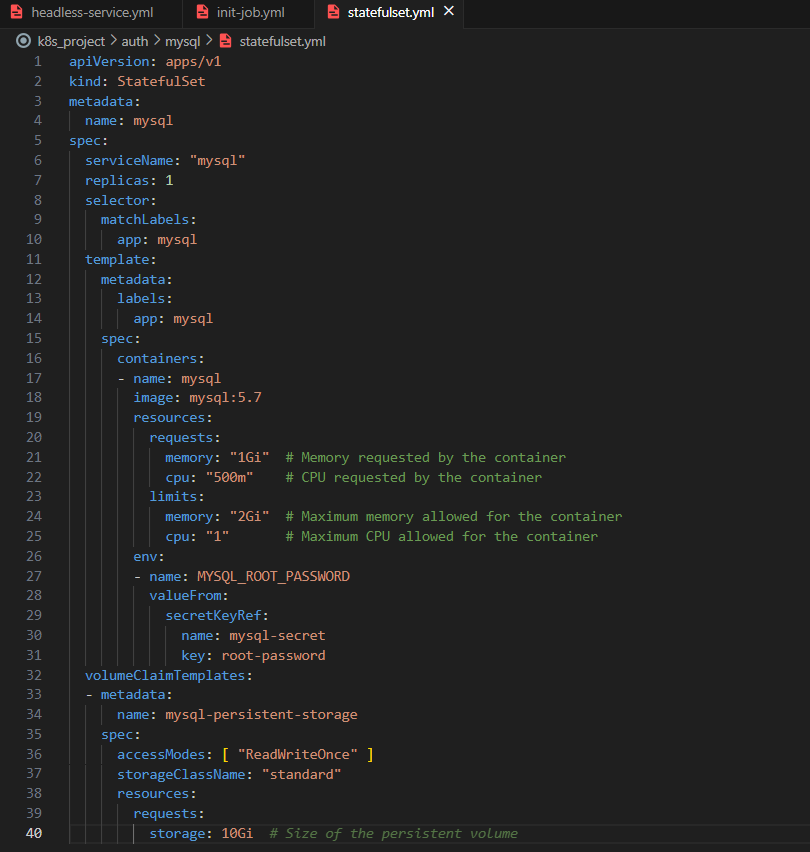
1. Create secret using imperative way (opaque secret)



1. Create init job to configure DB to the first time



1. Stateful set yaml definition file :

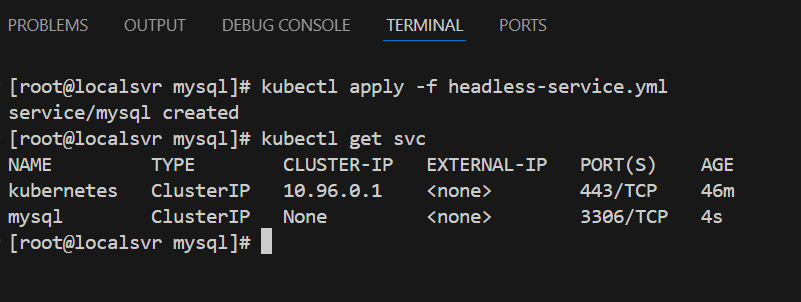


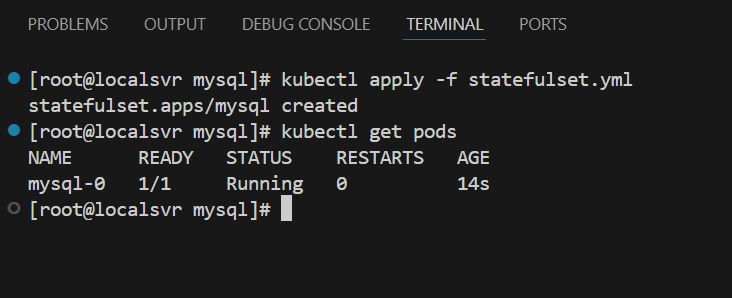
Note :

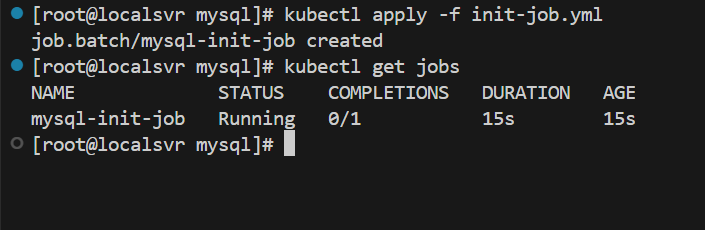
// if not have Db cluster choose replicas : 1 , if choose any number else 1 deploy a

DB with different data (inconsistent )

Apply Manifests:

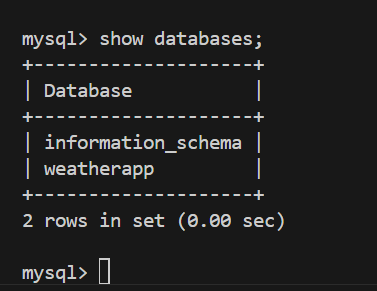






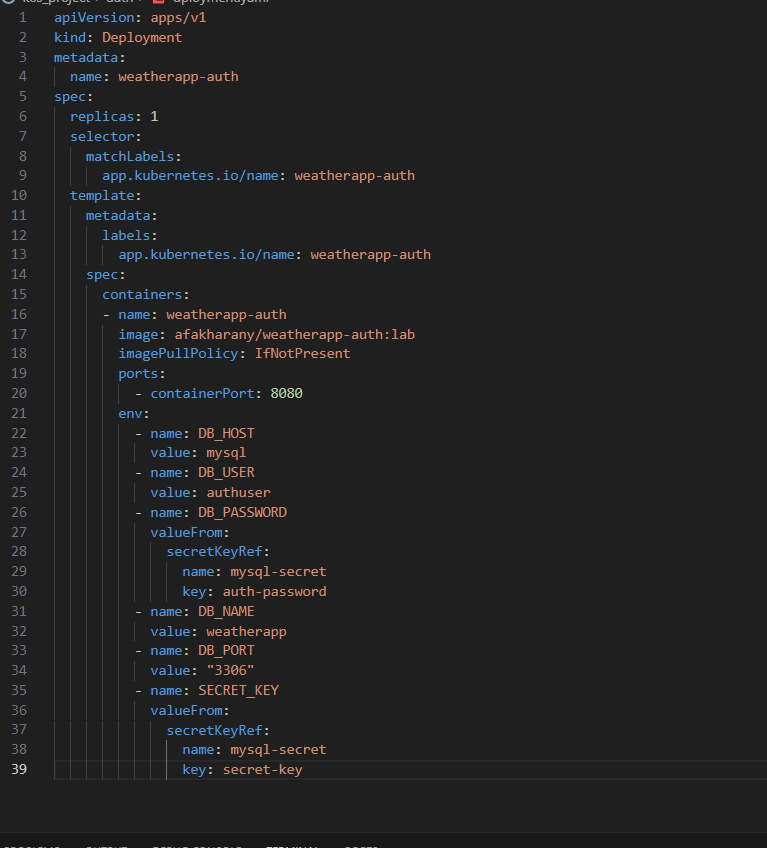
Verify :



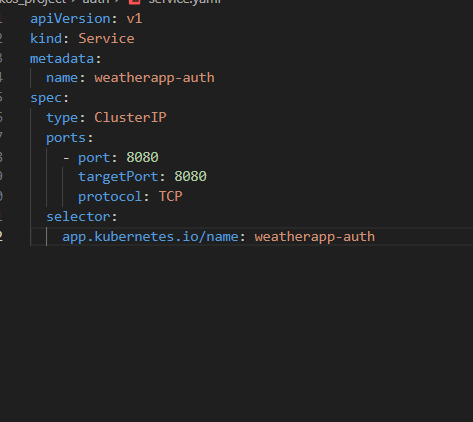


2- Authentication service deploy:

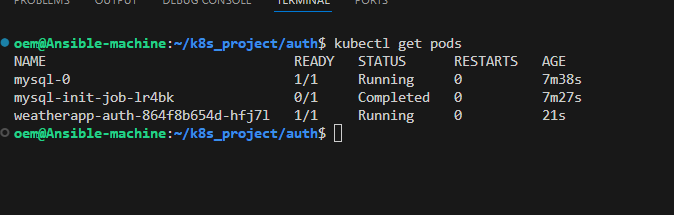
1. Deployment :

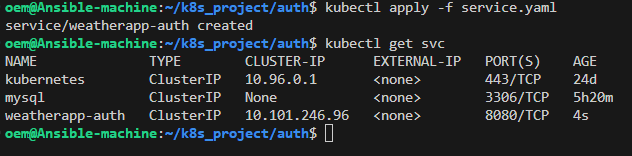


1. Service :

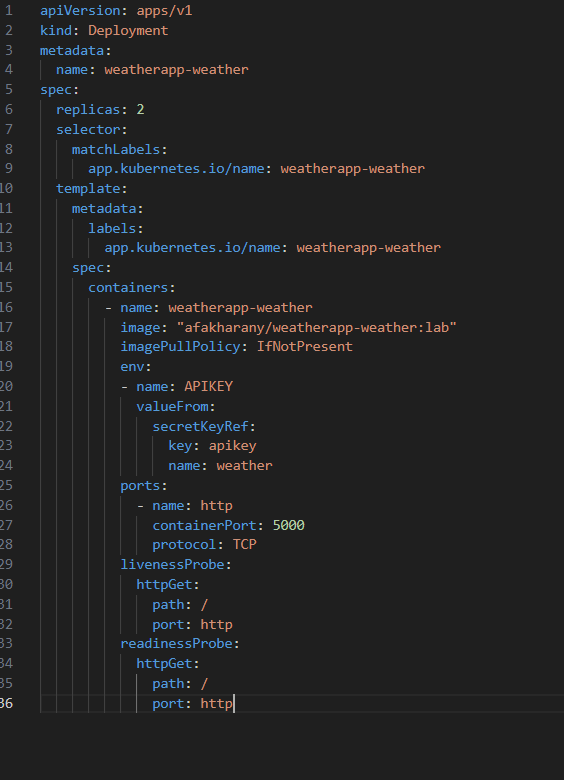


Apply :

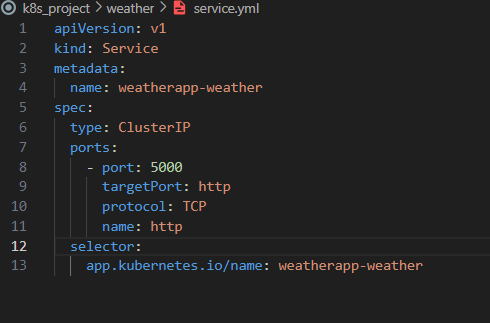




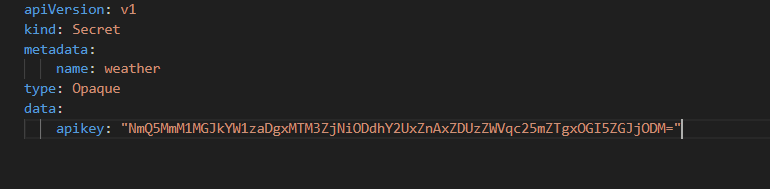
1. Weather client service :
2. Deployment :



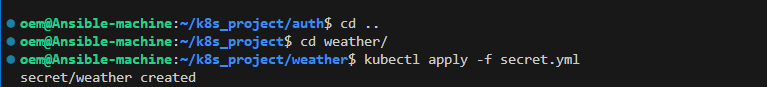
1. Service :

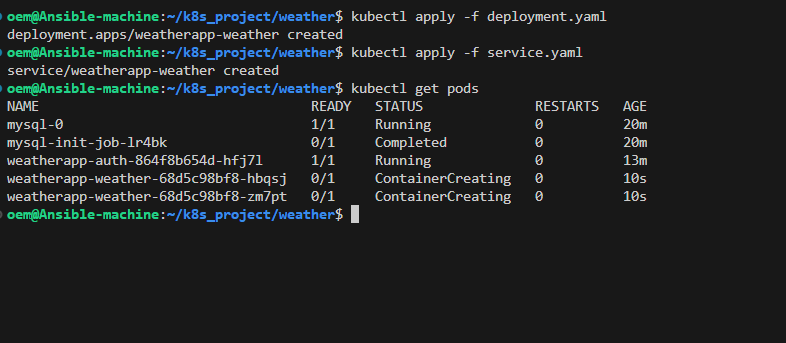


1. Secret :

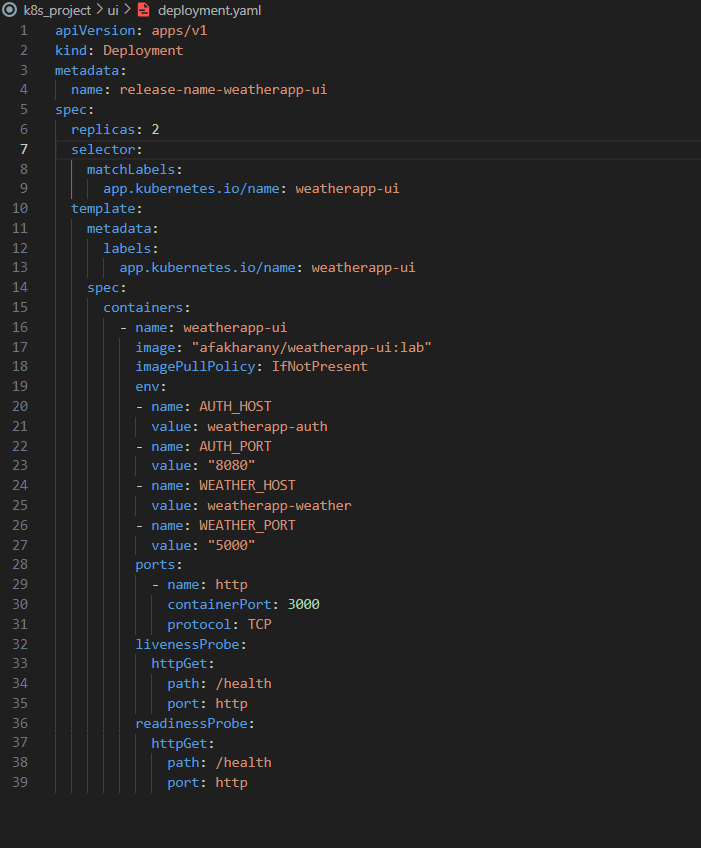


Apply :

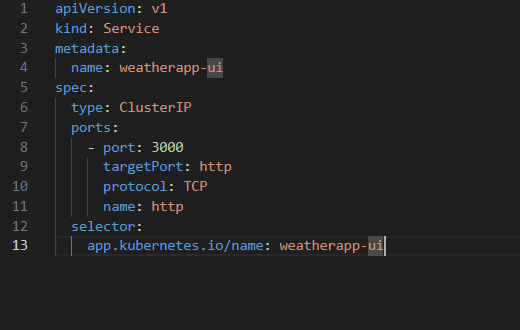




1. UI :
2. Deployment :

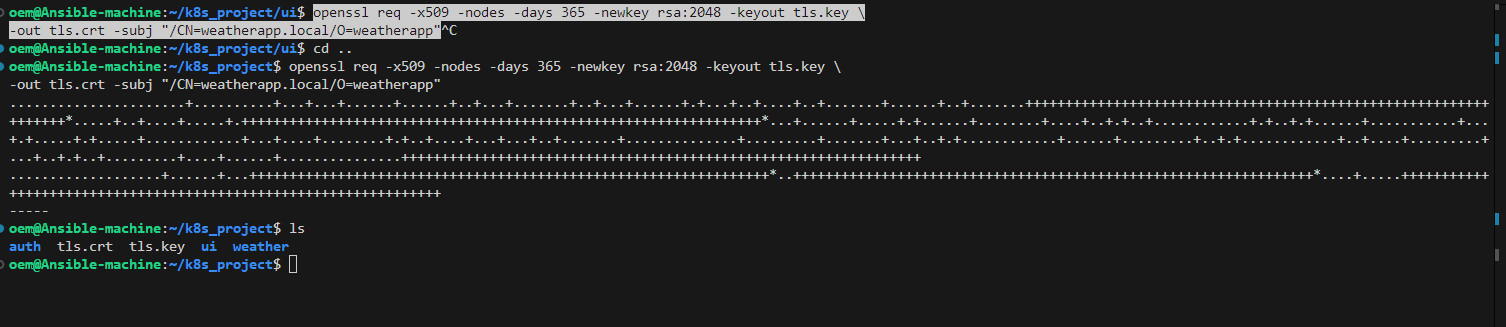


1. Service :

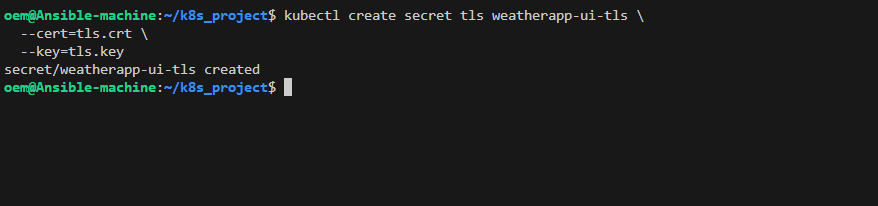


1. Ingress : single entry point
   1. Using http / https

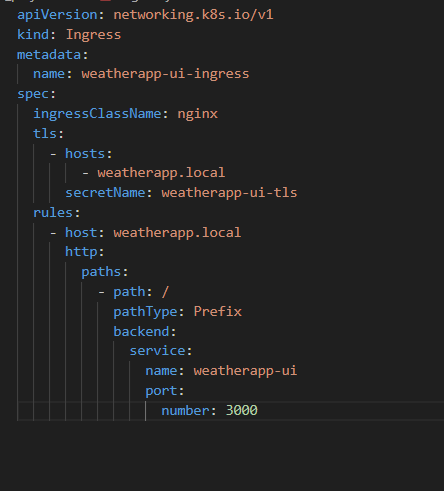
// create a certificate : .cert , .key



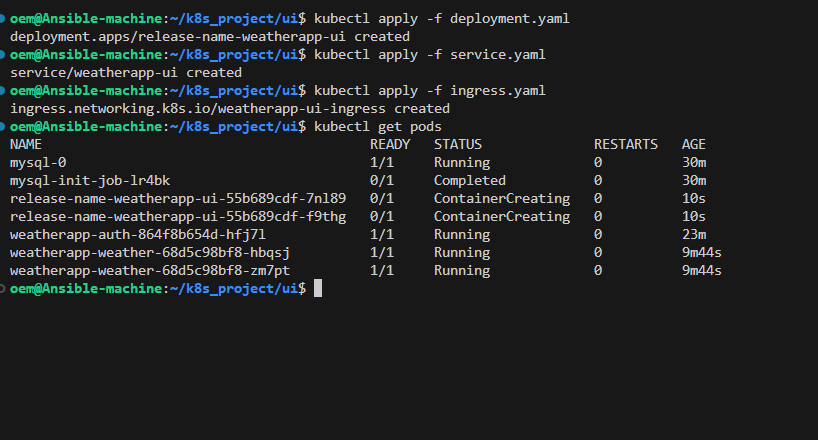
* 1. Create secret



C. ingress definition file

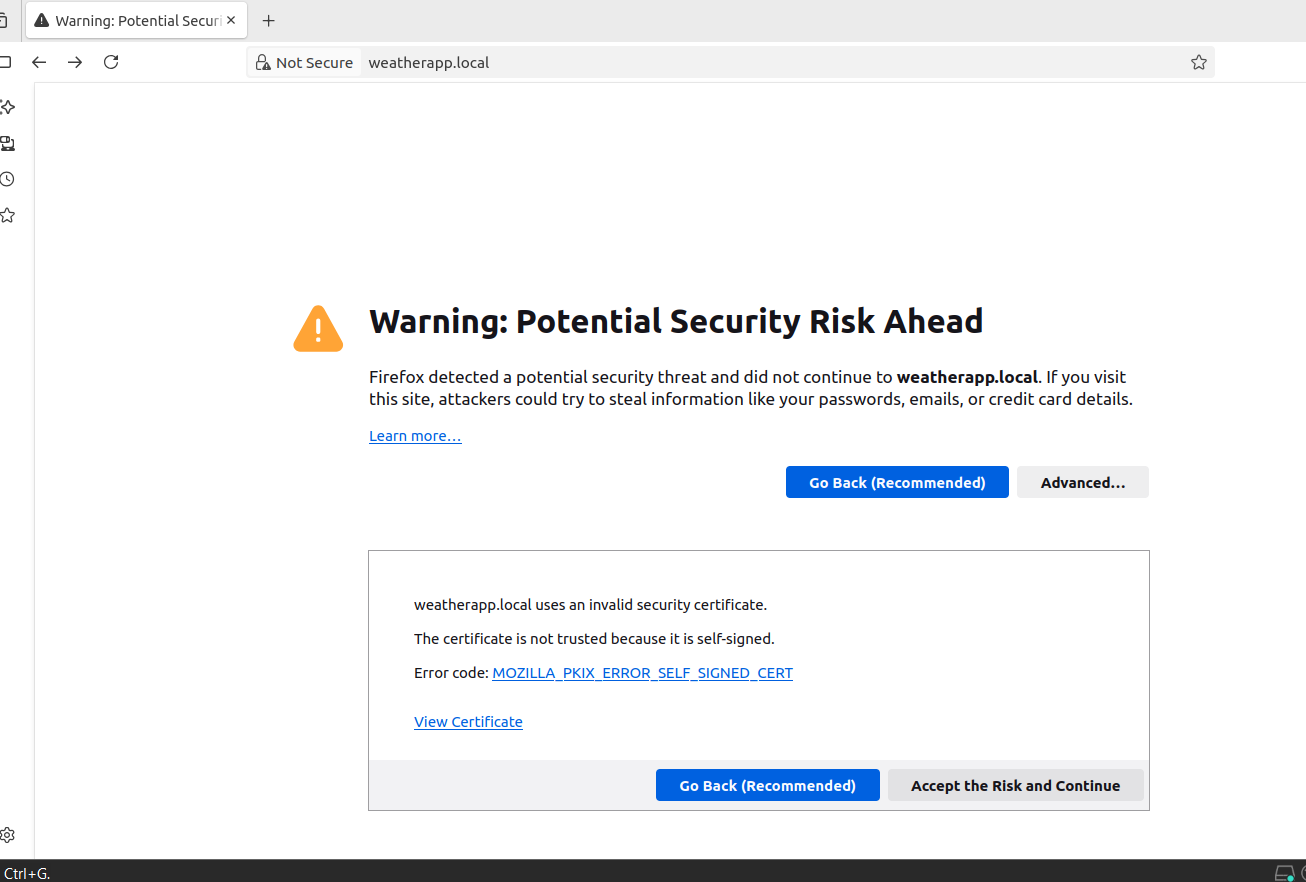


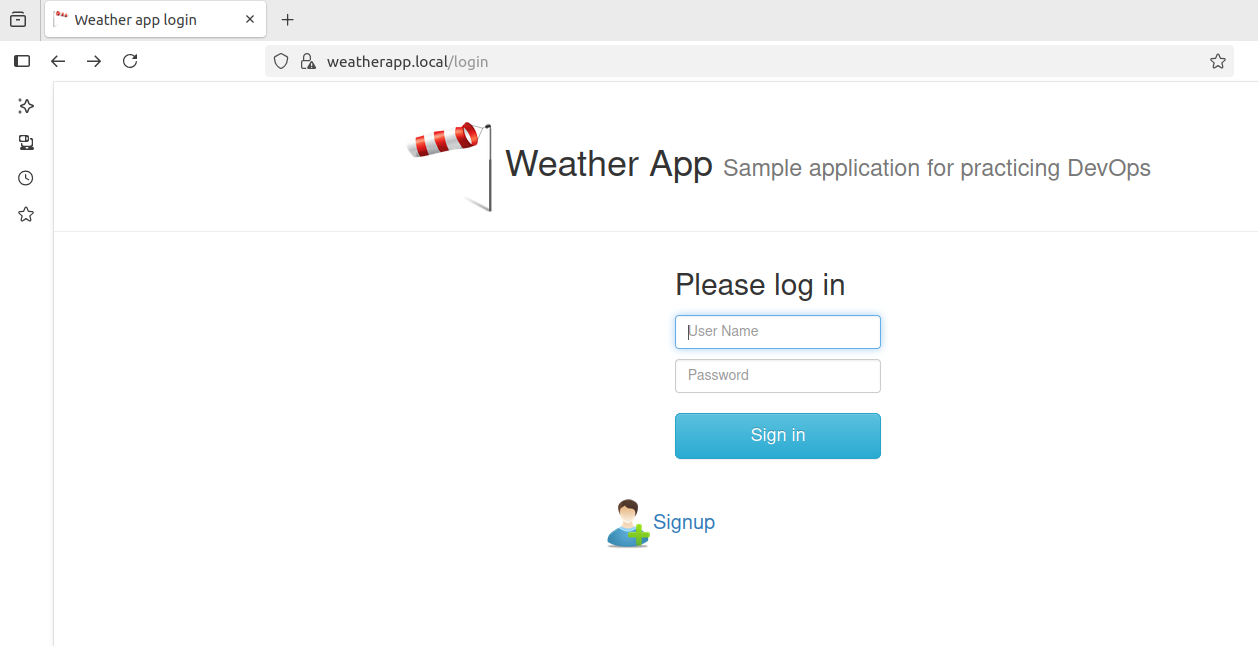
Apply:



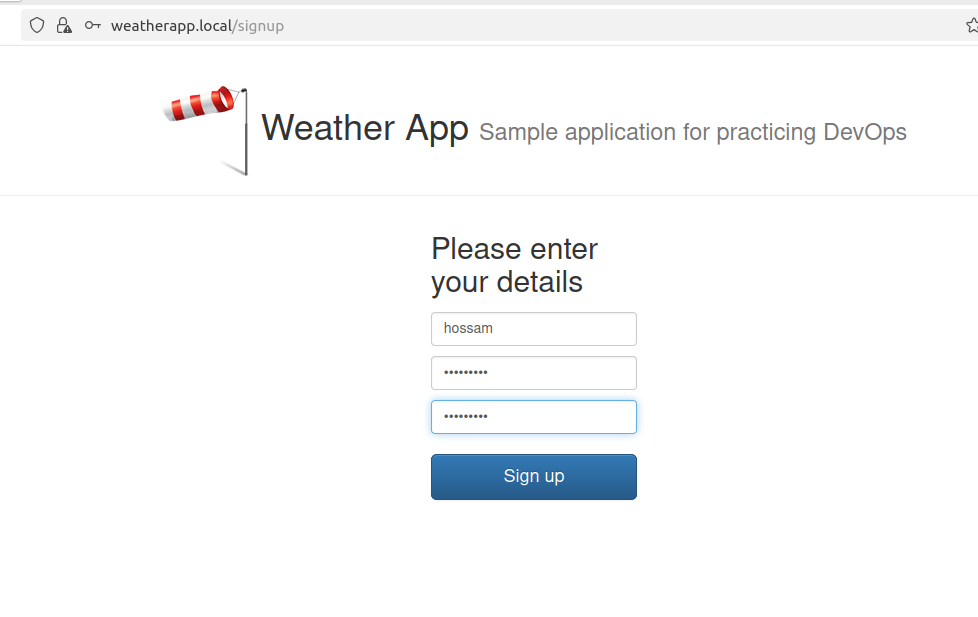
Verify MyApp :

// because a self-signed certificate not from CA

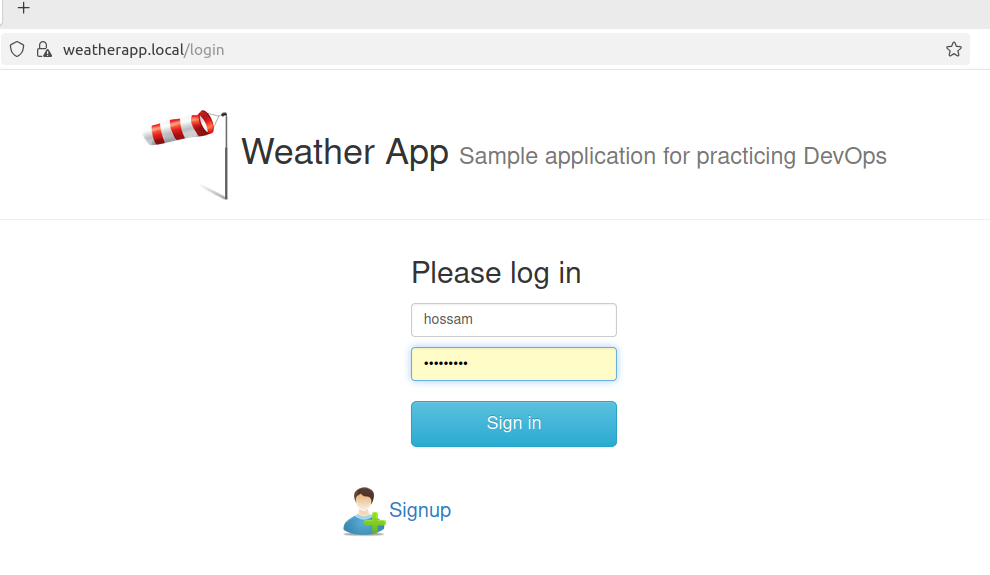




Sign up :



Sign in :



Test APP :

