# Documentacion Proyecto 3: despliegue de aplicación contenerizada con kubernetes

En este documento se presentará el paso a paso de como se desplegar el proyecto 3 de Tópicos especiales de telemática

## Arquitectura escalable y alta disponibilidad

Diagrama

Descripción generada automáticamente con confianza media

* VPC
* 2 Availability zones
* 2 public subnets
* 2 private subnets
* 1 Nat Gateway para la conexión a internet
* 2 Nat Gateway para cada availability zone
* 1 base de datos en MongoDB Atlas

## Despliegue local webapp en cluster kubernetes usando microk8s

### Prerequisitos

* Tener instalado Docker
* Tener instalado Microk8s
* O tener Docker-Desktop con kubernetes habilitado

### Paso a paso

1. Importaremos el repositorio donde se encuentra nuestro proyecto
   1. Git clone https://github.com/JJCanon/JJCanon-CDavilad-ST0263.git
2. Nos dirijimos a la carpeta JJCanon-CDavila-ST0263/Project3/bookstore2
   1. Cd JJCanon-CDavila-ST0263/Project3/bookstore2
3. Abrimos el archive secret.yaml y editamos el valor de MONGODB\_URI por el enlace que nos proporciona nuestra propia base de datos mongoDB pero codificada en base64.
4. Creamos la imágenes del frontend y backend:
   1. docker build -t frontend:latest ./frontend
   2. docker build -t backend:latest ./backend
5. Ejecutamos los manifiestos en Microk8s:
   1. microk8s kubectl apply -f configmap.yaml
   2. microk8s kubectl apply -f secret.yaml
   3. microk8s kubectl apply -f backend-deployment.yaml
   4. microk8s kubectl apply -f backend-service.yaml
   5. microk8s kubectl apply -f frontend-deployment.yaml
   6. microk8s kubectl apply -f frontend-service.yaml
   7. microk8s kubectl apply -f ingress.yaml

**Nota: en caso de usar Docker desktop, utilice los mismos comandos pero sin la palabra “microk8s”.**

1. probamos nuestra pagina en el navegador : <http://localhost>