

**UNIVERSIDAD RAFAEL LANDÍVAR**

**FACULTAD DE INGENIERÍA**

**VIRTUALIZACIÓN**

**SECCIÓN 1 VESPERTINA**

**MGTR. JOSSUE LEONEL SAMAYOA PORTILLO**

# **TERCER EXAMEN PARCIAL**

**Julio Anthony Engels Ruiz Coto 1284719**

**GUATEMALA DE LA ASUNCIÓN, MAYO 14 DE 2024**

Genere usando la tecnología de su preferencia, los componentes para un sistema que contenga las siguientes capas:

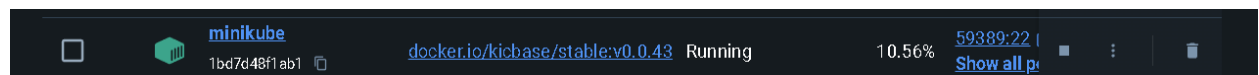
- Frontend (NGINX + React/Angular/Vue - Hello World SPA)

```
ROBLEMAS SALIDA TERMINAL PUERTOS SQL CONSOLE CONSOLA DE DEPURACION
Using the docker driver based on existing profile

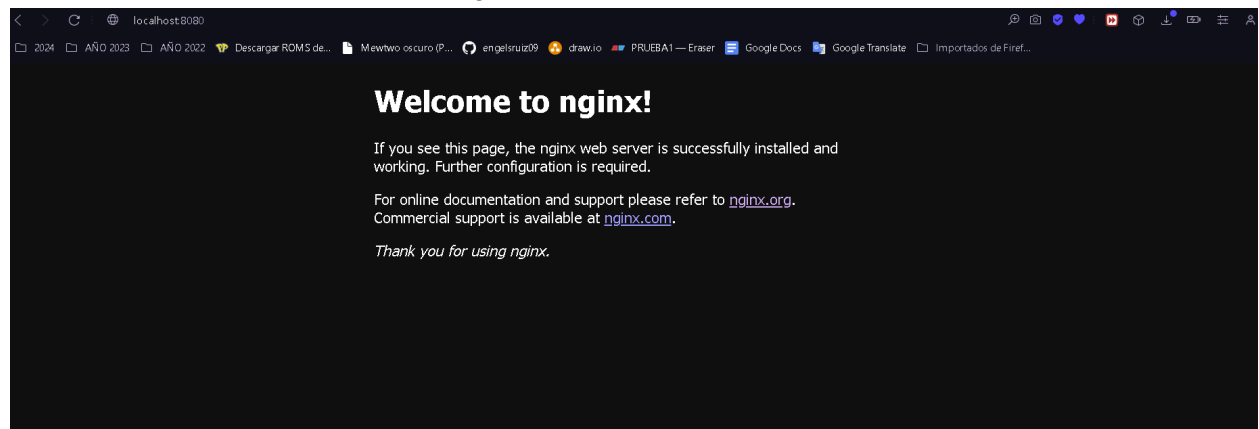
Saliendo por un error PROVIDER_DOCKER_NOT_RUNNING: "docker version --format <no value>:<no value>:<no value>" exit status 1: error during connect:
this error may indicate that the docker daemon is not running: Get "http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.45/version": open //./pipe/docker_engine:
the system cannot find the file specified.
Suggestion: Start the Docker service
Documentación: https://minikube.sigs.k8s.io/docs/drivers/docker/

minikube start
0514 20:05:56.975164 36456 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": context not found: open C:\Us
rs\julio\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: The system cannot find the path specified.
minikube v1.33.0 en Microsoft Windows 11 Pro 10.0.22631.3527 Build 22631.3527
minikube 1.33.1 is available! Download it: https://github.com/kubernetes/minikube/releases/tag/v1.33.1
To disable this notice, run: 'minikube config set WantUpdateNotification false'
```

En la presente imagen se analiza minikube iniciando con la imagen en docker creada.



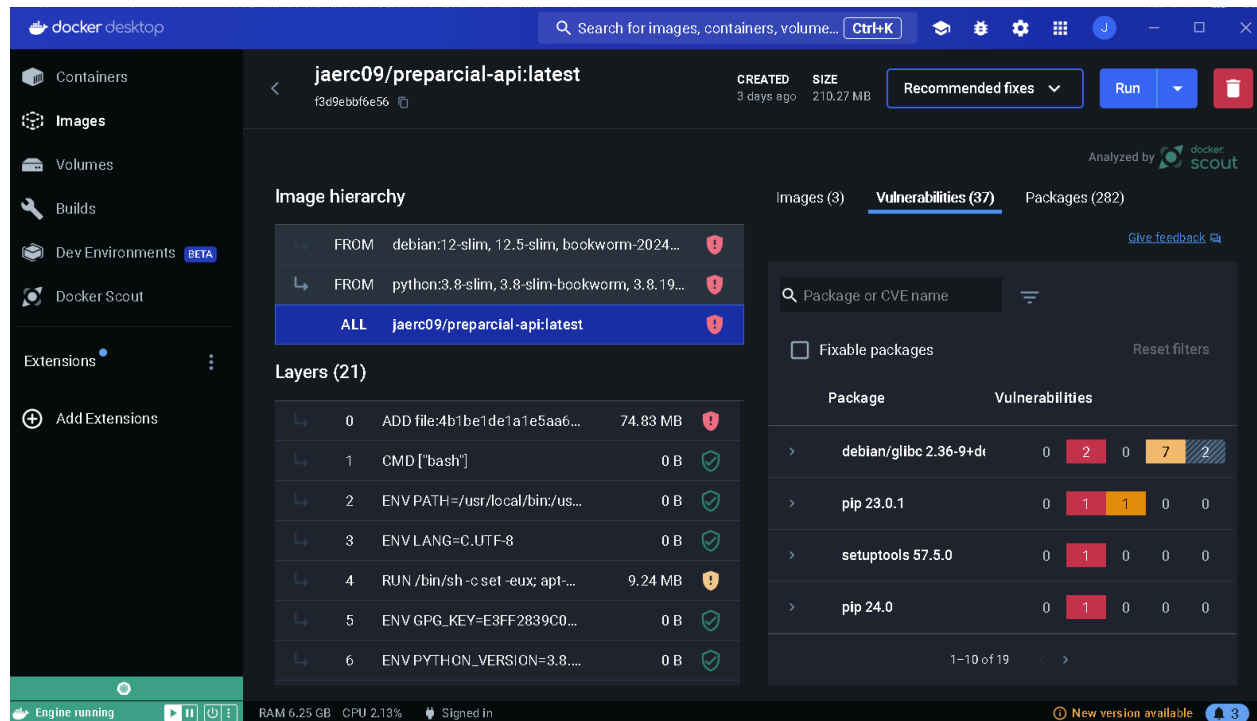
Una vez creado se visualiza de la siguiente manera el frontend:



Con este comando se realiza el port-forwarding para el deployment correspondiente, en este caso el del frontend

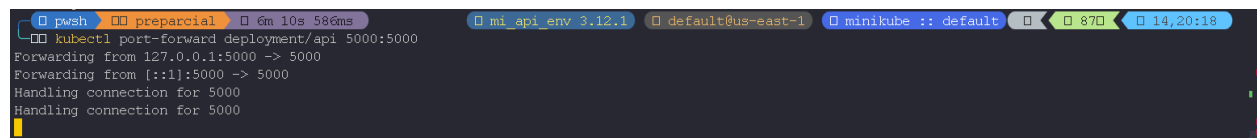
```
Done! kubect1 is now configured to use "minikube" cluster and "default" namespace by default
minikube port-forward deployment/landing-page 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
Handling connection for 8080
Handling connection for 8080
```

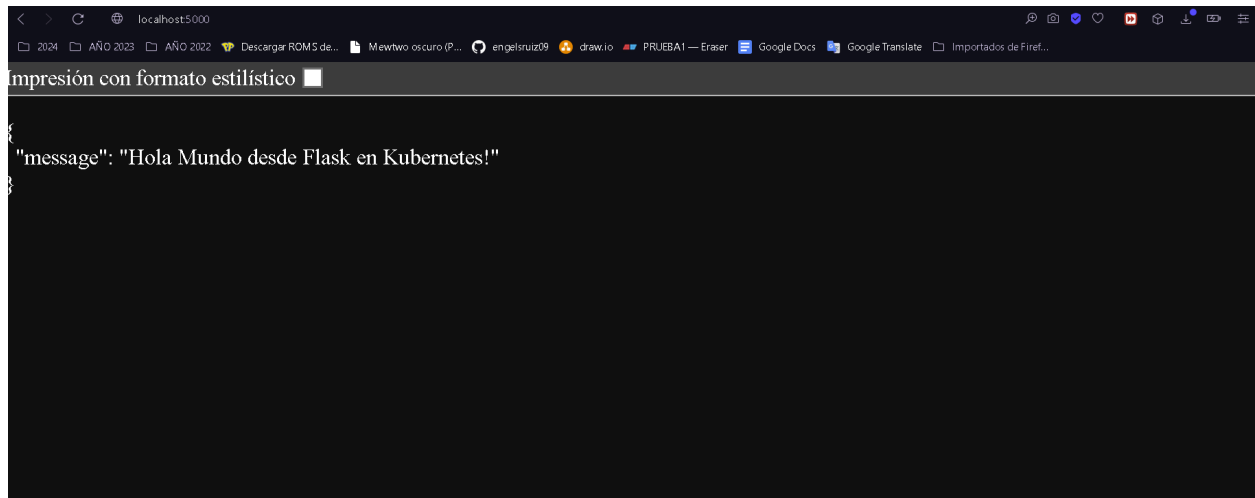
## - Backend (Apache/NGINX + Hello World API)



La imagen de mi API creada en docker que luego la subiera a docker hub y se vieran mis cambios en línea.

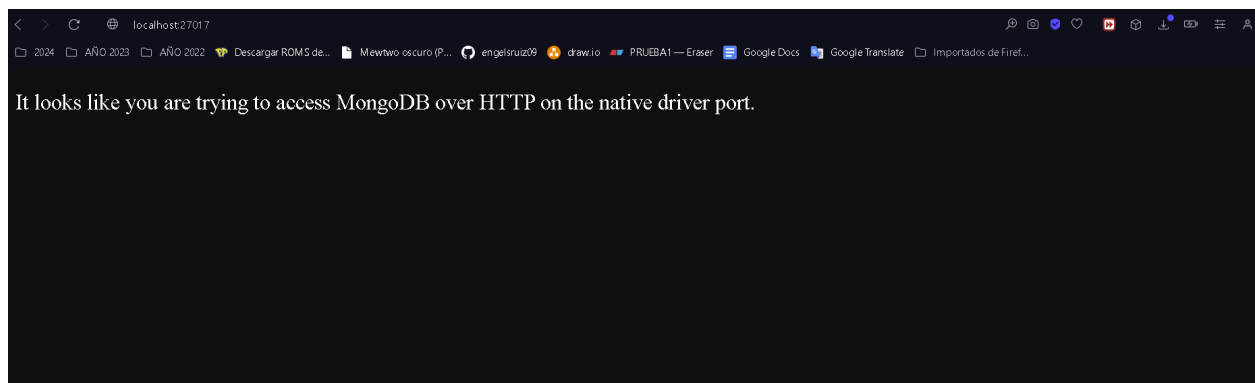
Una vez cargada en docker hub y corriendo los comandos necesarios al acceder al localhost del puerto 5000 se visualiza el contenido de esta API corriendo con toda normalidad ya que en mi caso retorna un jsonify y por eso en dicho puerto se ve de esa forma el mensaje de hola mundo,





- BDD (no relacional)

Gracias a docker hub conseguí un deployment que ya está alojado en dicha aplicación y me evito estar creando carpetas para cada una de las cosas y así generar un solo archivo main.tf, obviamente generando el archivo dockerfile para la API



Con el comando de kubectl get deployments puedo visualizar cada una de los deployments que se están corriendo y en ready salen todos con éxito.

```
❏ pwsh ❏ preparcial ❏ 3m 42s 804ms
❏ kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE
api	2/2	2	2
landing-page	2/2	2	2
mongo	1/1	1	1

Para verificar que los pods de cada uno de los servicios están levantados se puede verificar con el presente comando.

```
❏ pwsh ❏ preparcial ❏ 115ms
❏ kubectl get pods
```

NAME	READY	STATUS
api-895f47cb8-b5gvk	1/1	Running
api-895f47cb8-htk86	1/1	Running
landing-page-6477bd9468-dnbs6	1/1	Running
landing-page-6477bd9468-w2brt	1/1	Running
mongo-578b48f465-psj8v	1/1	Running