

PROYECTO FINAL BOOTCAMP DEVOPS2

CÓDIGO FACILITO – PDF DESCRIPTIVO



INICIO

BUSCAR DUEÑOS

VETERINARIOS

ERROR

Bienvenido al Desafío DevOps



códigofacilito

PetClinic

Cuidando de tus mascotas con amor y profesionalidad



DATOS TÉCNICOS

- NOMBRE: Proyecto Final DevOps2 (CódigoFacilito.com)
- DESCRIPCIÓN: Todo el ciclo de desarrollo de aplicación PetClinic.
- REPOSITORIO: <https://github.com/gsmx64/cf-devops2>
- DOCKER HUB: <https://hub.docker.com/repositories/gsmx64>
- PROMETHEUS: <http://34.139.84.184:30000/>
- GRAFANA: <http://34.139.84.184:30001/>

ÍNDICE

Inicio - Datos técnicos.....	1
Presentación y Motivación.....	5
Retos y Elaboración del Proyecto.....	5
Infraestructura cm código.....	7
-.....	8
-.....	9
-.....	11
-.....	15
-.....	20
-.....	21
Referencias y Fuentes.....	21

PRESENTACIÓN Y MOTIVACIÓN

Hola soy Gonzalo Mahserdjian, soy de Córdoba, Argentina. Ya había realizado el Bootcamp de DevOps parte 1, ahora completando DevOps parte 2.

RETOS Y ELABORACIÓN DEL PROYECTO

Opté por realizar el proyecto 1 denominado Petclinic, el cual cloné desde el repositorio original de Roxs, citado en <https://gitlab.com/training-devops-cf/cf-devops-challenge/cf-devops-petclinic>.

Mi repositorio para la elaboración del desafío será <https://github.com/gsmx64/cf-devops2>.

Como primera medida, debíamos iniciar con los archivos ya definidos, la aplicación con los docker compose, pero como era de esperarse, algo escondido iba a haber y no iniciaría a la primera instancia.

```

o clean it up.
[+] Running 2/2
  ✓ Container cf-devops2-mysqlserver-1   Created
  ✓ Container petclinic                   Created
Attaching to mysqlserver-1, petclinic
mysqlserver-1 | 2025-05-17 18:24:08+00:00 [Note] [Entrypoint]: E
petclinic      | /__cacert_entrypoint.sh: line 114: /app/mvnw: Pe
mysqlserver-1 | 2025-05-17 18:24:08+00:00 [Note] [Entrypoint]: S
mysqlserver-1 | 2025-05-17 18:24:08+00:00 [Note] [Entrypoint]: E
petclinic exited with code 126
mysqlserver-1 | 2025-05-17 18:24:09+00:00 [Note] [Entrypoint]: I

```

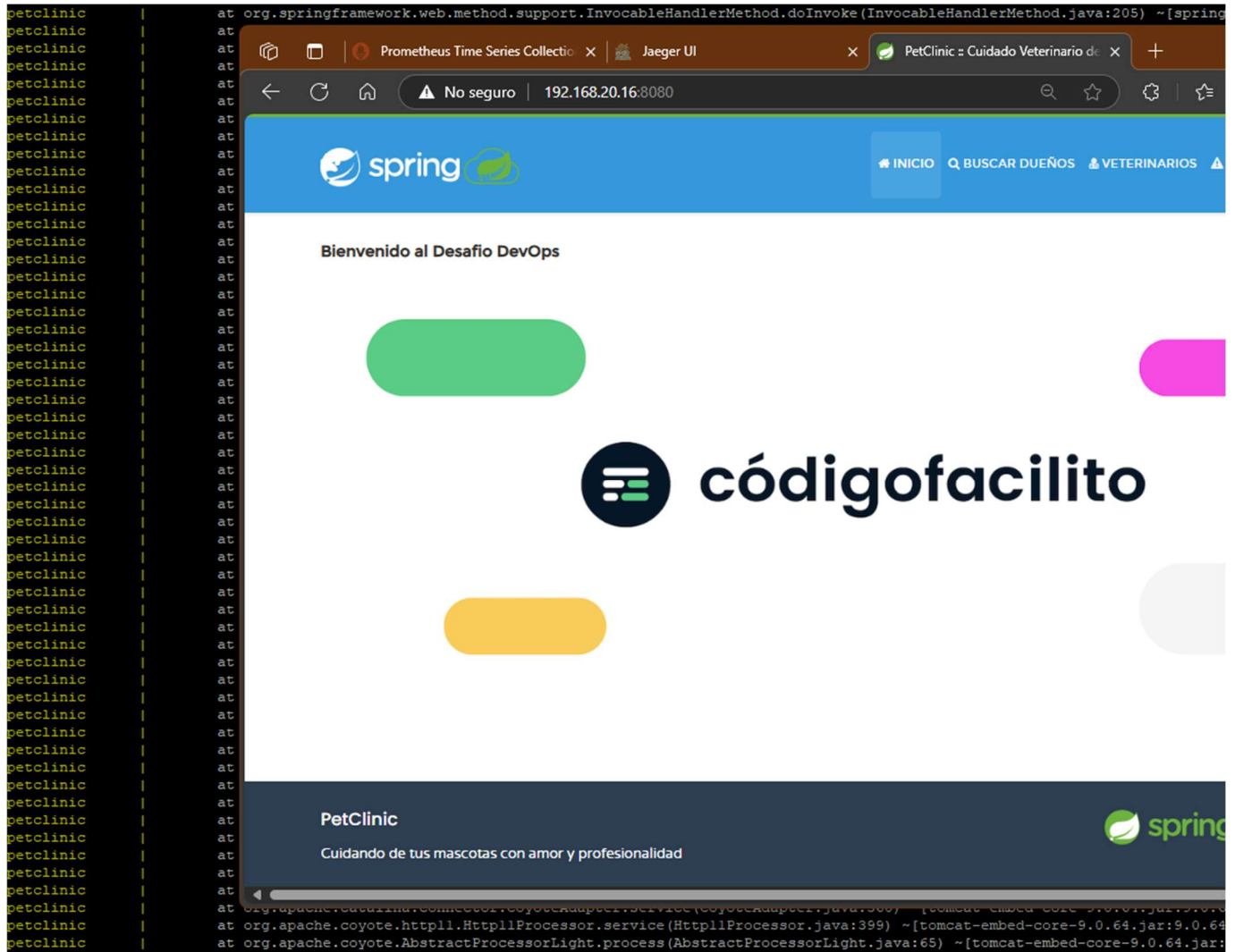
Modifiqué el archivo Dockerfile adicionándole permisos de ejecución al script mvnw, pero luego me encontré que al final no terminaba de iniciar, para lo cual tuve que cambiar el shebang de ejecución, iniciándolo con "sh".

```

Dockerfile M X
Dockerfile > ...
1  FROM eclipse-temurin:21-jdk-jammy AS base
2  WORKDIR /app
3  COPY .mvn/ .mvn
4  COPY mvnw pom.xml ./
5  RUN chmod +x ./mvnw && ./mvnw dependency:resolve
6  COPY src ./src
7
8  # Instalar el agente OpenTelemetry en base para compartirlo
9  ARG OTEL_AGENT_VERSION=1.32.0
10 RUN curl -L https://github.com/open-telemetry/opentelemetry-java-instrumentation/
11    -o /opt/opentelemetry-javaagent.jar
12
13 FROM base AS development
14 # Configura el agente OpenTelemetry y el debugger
15 ENV JAVA_OPTS="-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=5005"
16 ENV JAVA_TOOL_OPTIONS="-javaagent:/opt/opentelemetry-javaagent.jar"
17 CMD ["sh", "./mvnw", "spring-boot:run", "-Dspring-boot.run.profiles=mysql"]
18

```

Finalmente inicia:



I - INFRAESTRUCTURA COMO CÓDIGO

En la carpeta <https://github.com/gsmx64/cf-devops2/terraform> del repositorio están generados los archivos .tf para la creación de toda la infraestructura automatizada para el proveedor GCP, con los environments dev y prod, y separados por módulos.

El tf state se genera local, y como primer paso genera un bucket y modifica el archivo backend.tf para guardar el estado de forma remota, en la carpeta dev o prod según corresponda el workspace.

Luego levanta las variables del proyecto de GCP, crea la VPC, la subnet, reglas de firewall, la base de datos mysql 8.0 dentro del servicio Cloud SQL, la creación de los repositorios para las imágenes docker y finalmente finaliza con la creación del clúster GKE.

```

terraform > environments > prod > main.tf > module "project" > google_apis
1  # Configure the Google Cloud provider
2
3  module "bucket" {
4      source = "../../modules/bucket"
5      tfstate_filename = var.tfstate_filename
6      tfstate_remote_bucket_intro = var.tfstate_remote_bucket_intro
7      tfstate_folder_prefix = var.tfstate_folder_prefix
8      location = var.region
9  }
10
11 module "project" {
12     source = "../../modules/project"
13     billing_account = var.billing_account
14     project_id = var.project_id
15     project_name = var.project_name
16     google_apis = var.google_apis
17 }
18
19 module "vpc" {
20     source = "../../modules/vpc"
21     vpc_name = "${module.vpc_name}-${terraform.workspace}"
22     auto_create_subnetworks = "${module.auto_create_subnetworks}"
23 }

```

...

.