

Proyecto Final Bootcamp DevOps 2023

Automatización Web Voto Telemático

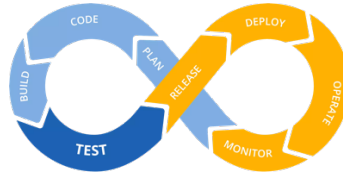
Realizado por Carlos Santaella Pérez

Índice

1. Objetivo y alcance
2. Aplicación
3. Arquitectura
4. Pipelines CD/CI
5. Repositorio Github

1. Objetivo y Alcance

El proyecto plantea la la implantación de proceso CD/CI enmarcado en la cultura DevOps.



Se plantea un APP simple que ayude a la votación telemática a través de un portal web que capturará los datos del votante y los guardará en una bases de datos.

Se plantea un pipeline para la automatización de las siguientes etapas:

- Desarrollo
- Build
- Test-Staging
- Linting
- Sec
- Deploy

Se incluye el registro de Log y Monitorización del entorno de Staging y Produccion

No se incluyen en el proyecto las siguientes elementos por falta de tiempo:

- Uni Test
- Test de rendimiento

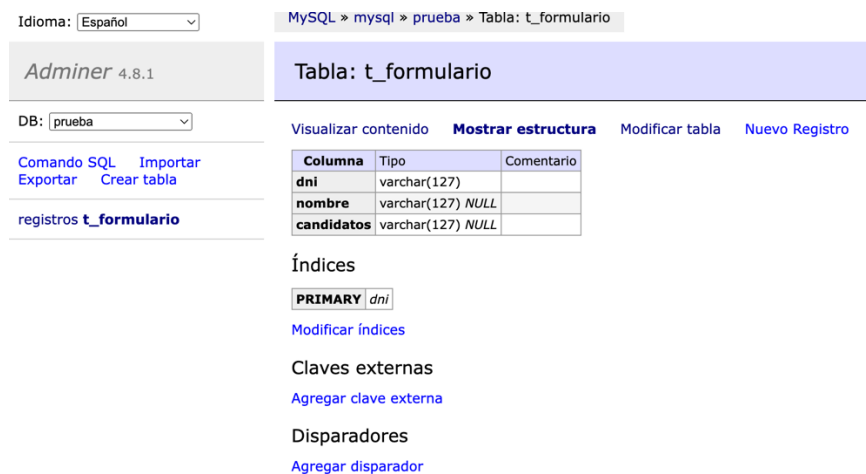
2. Aplicación

El funcionamiento básico de la aplicación es la gestión del voto electrónico en la que se solicitará al votante su nombre, dni y opción de voto que será recogido en una base de datos para su posterior contabilización.



The screenshot shows a web browser window with the address bar set to 'localhost'. The page title is 'Formulario'. The form contains three input fields: 'Nombre:', 'DNI:', and 'Selecciona un candidato:'. The 'Selecciona un candidato:' field is a dropdown menu currently showing 'Candidato 1'. Below the fields is a green button labeled 'Enviar'.

Estructura de la Base de Datos



The screenshot shows the Adminer 4.8.1 interface. The language is set to 'Español' and the database is 'prueba'. The table 't_formulario' is selected. The table structure is displayed as follows:

Columna	Tipo	Comentario
dni	varchar(127)	
nombre	varchar(127) NULL	
candidatos	varchar(127) NULL	

Below the table structure, the 'Índices' section shows a primary key on the 'dni' column. Other sections like 'Claves externas', 'Disparadores', and 'registros t_formulario' are also visible.

3. Arquitectura

Stack tecnológico

En el proyecto se han utilizado las siguientes tecnologías:

- Desarrollo PHP + html + CSS
- Base de datos: MySQL en AWS RDS
- Servidor Web: Nginx
- Repositorio de fuentes: Github
- Automatización de procesos CD/CI: Github Action
- Entorno de despliegue de Staging y Producción: AWS ECS + Fargate
- Container Registry: AWS ECS
- Monitorización y Log: AWS CloudWatch
- Despliegue de Infraestructura (IaC): Terraform
- AWS Route53 para publicación de la Web

Las pautas de decisión en la elección de los diferentes componentes del proyecto han sido las siguientes:

- AWS ECS + Fargate
 - o Al tratarse de una aplicación Web de uso masivo en las fechas previas a la elecciones, se ha elegido un entorno que permitiese el crecimiento horizontal para adaptarse a los picos de demanda
 - o Se utiliza Fargate para simplificar la gestión de provisión de contenedores y no requerir de un especialista
- Container Registry AWS ECR
 - o Nos provee un contenedor unido al entorno de despliegue
 - o Incluye detección de vulnerabilidades de los contenedores lo que nos permite añadir una capa de seguridad
- Github Action para automatización.
 - o Permite simplificar el proceso de despliegue consolidando código, acciones CD/CI e integración con todas las tecnologías empleadas en el proyecto.
 - o Sin coste. En caso de un entorno en producción real, el coste por usuario es competitivo.
 - o Facilita el trabajo de un equipo de desarrollo y DevOps
- Monitorización y Log con CloudWatch
 - o Simplifica las tareas de registro de log y monitorización en un entorno AWS
- Despliegue de infraestructura por código con Terraform
 - o Nos permite integrar todas las tecnologías incluidas en el proyecto
 - o Permite evitar el vendor lock dando la posibilidad de diversificar plataformas sin necesitar perfiles SRE con otras capacidades

Entornos Dev, Staging y Producción

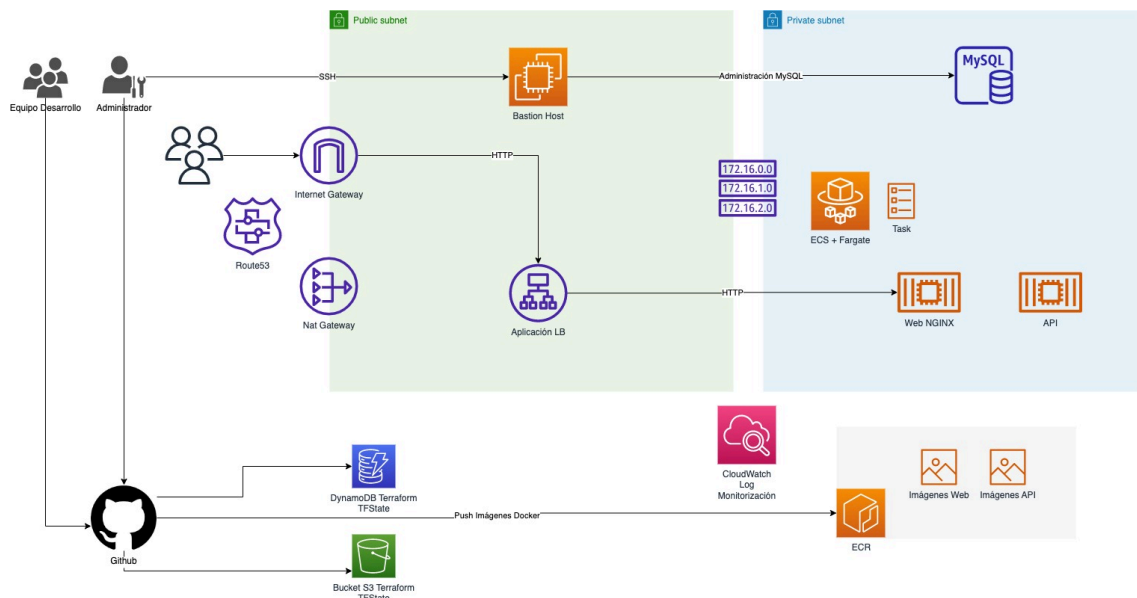
Desarrollo

El desarrollador cuenta con un entorno local basado en Docker para estandarizar el entorno sobre el que desarrollaría un equipo incluyendo:

- Docker Compose para:
 - o Entorno de Desarrollo
 - Contenedor Nginx
 - Contenedor PHP
 - Contenedor MySQL
 - o Infraestructura por código
 - Contenedor que ejecuta el código Terraform

Staging y Producción

Infraestructura basada en AWS con los siguientes elementos.



Estructura de Carpetas

```

+- deploy                               Infraestructura como código
|   +- templates                       Archivos Terraform
|       +- Bastion                     Scrips de personalización Bastion Host
|       +- ECS                         Scrips de parametrización de ECS
|
+- src                                  Entorno de desarrollo y Docker para desarrollo
|                                     local de los programadores
|                                     Script Docker Compose para creación de entorno
|                                     local
|       +- api                         Dockerfile para creación del contenedor API-PHP
|       +- web                         Fuentes del desarrollo web y php
+- .github
|   +- workflows                       Pipelines CD/CI

```

4. Pipeline CD/CI

Ramas

El proyecto utiliza las siguientes ramas:

- **Rama Producción.** Donde se guarda el código que está en Producción
- **Rama Master.** Es el eje central de consolidación del desarrollo de los programadores
- **Features/*.** Donde se desarrollan las funcionalidades nuevas que se integrarán en Master

Pipelines

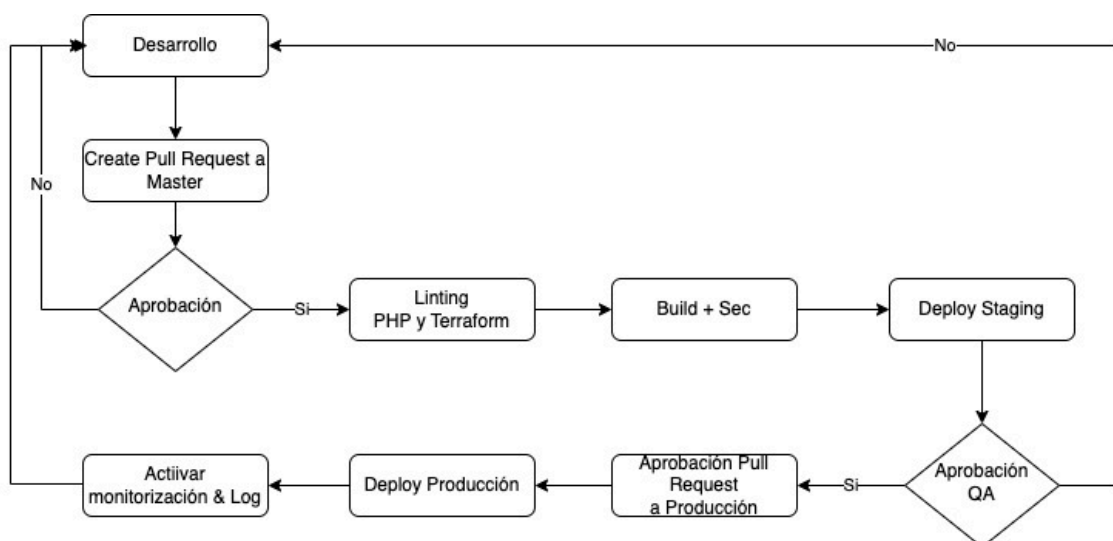
Se utiliza Github para:

- Repositorio de código compartido con equipo de desarrollo y operaciones
- Gestión de secretos AWS
- Implementación de CI/CD

El proyecto cuenta con 4 archivos pipeline para el proceso de automatización.

- pipeline_principal.yml
 - o Contiene el pipeline completo
 - o Lo desencadena los Pull Request a Master y Producción
- terraform_destroy_produccion.yml
 - o Contiene el pipeline que borra la infraestructura de Producción
 - o Se ejecuta de forma manual
- terraform_destroy_staging.yml
 - o Contiene el pipeline que borra la infraestructura de Staging
 - o Se ejecuta de forma manual
- terraform_validate.yml
 - o Pipeline para validación de formatos y estructura Terraform
 - o Se ejecuta de forma manual

Pipeline Principal



5. Repositorio

`https://github.com/csantaella/proyecto_devops_publico`