

## **Documentación – Prueba Técnica**

### **Ingeniero de Infraestructura**

#### **1. Introducción**

Este documento presenta la implementación solicitada en la Prueba Técnica para el puesto de Ingeniero de Infraestructura, siguiendo prácticas profesionales de automatización, seguridad, despliegue continuo y diseño cloud.

#### **2. Arquitectura General**

La solución combina Next.js SSR, Docker, AWS EC2/ECR, Terraform, GitHub Actions, Nginx y Trivy, garantizando un flujo moderno de despliegue y actualización continua.

#### **3. Infraestructura como Código Terraform**

Terraform crea EC2, ECR, IAM Roles y Security Groups. Permite reproducibilidad, control de versiones y despliegue automatizado.

#### **4. Aplicación Next.js SSR**

Aplicación con basePath /prueba-tecnica, renderizado SSR, TailwindCSS y lectura de variables de entorno en runtime.

#### **5. Docker Multistage**

El Dockerfile multistage genera una imagen optimizada, pequeña y adecuada para producción.

#### **6. Configuración de Nginx**

Nginx se configuró como reverse proxy, manejando tráfico HTTP/HTTPS, TLS y headers de seguridad solicitados por la prueba.

#### **7. Seguridad TLS**

Se utiliza certificado autofirmado para esta prueba. En producción se usaría Let's Encrypt + dominio real (No utilicé Let's Encrypt porque este servicio requiere un dominio público propio para validar el certificado, y actualmente solo cuento con una IP pública, la cual no es compatible con la verificación de Let's Encrypt. Por esa razón, el despliegue se dejó funcionando únicamente con HTTP y con la configuración preparada para añadir TLS en cuanto se disponga de un dominio real.)

#### **8. CI/CD – GitHub Actions**

Cada push a main construye la imagen, la sube a ECR, escanea vulnerabilidades y despliega automáticamente en EC2 por SSH.

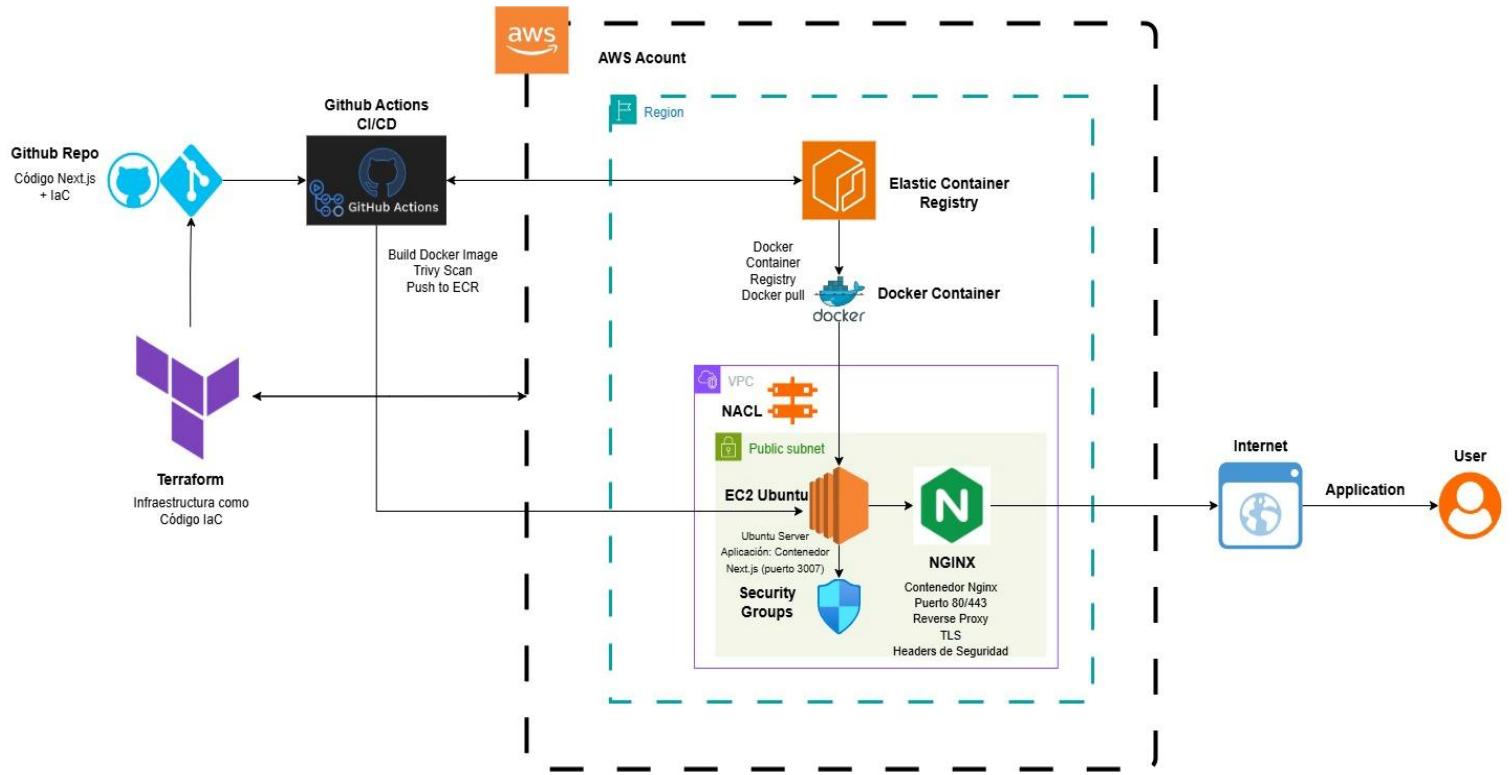
## 9. Evidencia de Escaneo Trivy

Se incluye el archivo trivy-report.txt como evidencia de seguridad.

## 10. URL Pública

- HTTP: <http://44.213.82.171/prueba-tecnica>
- HTTPS: <https://44.213.82.171/prueba-tecnica>

## 11. Diagrama de Infraestructura



### 1. Código Fuente y Terraform (IaC)

El repositorio en GitHub contiene tanto el código de la aplicación Next.js como la infraestructura definida en Terraform (IaC).

Terraform se utiliza para crear los recursos principales en AWS, como la instancia EC2 y el repositorio ECR.

## 2. Pipeline de CI/CD – GitHub Actions

Cada vez que se realiza un push a la rama main, GitHub Actions ejecuta un pipeline que:

- Construye la imagen Docker de la aplicación.
- Ejecuta un escaneo de vulnerabilidades con Trivy.
- Publica la imagen resultante en Amazon ECR.
- Finalmente realiza un despliegue automático por SSH hacia la instancia EC2.

Este proceso garantiza integración continua, despliegue automático y auditoría de seguridad.

## 3. AWS ECR – Registro de Imágenes

La imagen Docker generada se almacena en Amazon Elastic Container Registry (ECR).

Posteriormente, la instancia EC2 realiza un docker pull para obtener siempre la versión más reciente de la aplicación.

## 4. AWS EC2 – Servidor de Aplicación

La instancia EC2 Ubuntu es la máquina donde se ejecuta el contenedor.

La arquitectura incluye:

- VPC
- Subred Pública
- NACL
- Security Groups

Estas capas controlan el tráfico entrante/saliente y garantizan la seguridad de la infraestructura.

En la EC2 se ejecuta:

- Docker Engine
- Contenedor Next.js (puerto interno 3007)
- Servidor NGINX como reverse proxy

## 5. Nginx como Reverse Proxy

NGINX recibe las peticiones del usuario en los puertos 80/443 y las redirige (proxy\_pass) al contenedor Next.js.

Además, aplica:

- Headers de seguridad
- Manejo TLS
- Validación del tráfico entrante
- Esto mejora el rendimiento y protege el servicio.

## 6. Acceso del Usuario Final

El usuario accede a la aplicación a través de la URL pública de la instancia EC2.

El flujo completo es:

- Usuario → Nginx → Contenedor Next.js → Respuesta SSR

Con esto se garantiza una experiencia completa de Server-Side Rendering funcionando desde AWS.

## 12. Conclusiones

La solución cumple con los requisitos indicados en la prueba: SSR, Docker multistage, IaC, CI/CD, seguridad, Nginx, AWS y entrega profesional.