

Digital **NAO**

Conceptos DevOps, microservicios y contenedores



PRIMER CONTACTO

Actividad

Presentación del reto titulado *Conceptos DevOps, microservicios y contenedores*.

Rubén Sánchez es un mexicano de 29 años de edad, quien se ha dedicado los últimos cuatro años de su carrera al desarrollo de software como Full Stack. Le gusta trabajar solo, ocasionalmente accede a trabajar en equipos, aunque sabe que ese no es su fuerte. Suele actualizarse con diversos cursos que toma por su cuenta, y ha ido adquiriendo experiencia en diversos los proyectos que ha desarrollado. Disfruta el trabajo desde su y atesora tener su propio horario, por eso ha decidido seguir laborando como *freelancer*. Además, siempre lo contratan ya que tiene amplia experiencia en muchos lenguajes de desarrollo y en todas las herramientas de CI/CD.

Hace algún tiempo Rubén desarrolló una página Web para una nueva transmisora de radio llamada RadioNet, que inició operaciones en la ciudad de Silao, Guanajuato. La página Web creada por Rubén sigue la arquitectura de microservicios, está montada en AWS y es perfecta... pero no cuenta con un *pipeline* adecuado de CI/CD, pues no era una prioridad en el momento en que se creó.

RadioNet es una estación innovadora y joven. Cuenta solo con doce empleados. Si bien comenzó operaciones en Silao, tiene como misión cubrir zonas en el Bajío, donde la señal es débil, y como meta, la expansión de sus servicios para brindar comunicación por Internet.

Para que RadioNet pueda lograr sus objetivos es necesario que mejore su infraestructura de TI y los propietarios lo saben. Así que, deciden contactar a Rubén de nuevo para que continúe con el proyecto, además de agregar los detalles, módulos y funcionalidades que ellos tienen en mente.

Rubén comprende la situación de RadioNet y les sugiere primero invertir en un *pipeline* de CI/CD, de modo que los cambios efectuados a su página sean más rápidos y seguros de desplegar. Les comenta que con este desarrollo será más fácil que varias personas intervengan en el proyecto, así todos los cambios que tienen en mente se puedan alcanzar más pronto y con menos errores.

En RadioNet están de acuerdo y deciden atender su sugerencia. Ahora Rubén iniciará el proyecto. Lo primero que debe hacer es elegir las herramientas que empleará para poder

hacer este trabajo. Día a día nuevas herramientas para CI/CD salen al mercado y no está por demás investigar alternativas a lo ya conocido.

Tras un estudio de las opciones disponibles y, tomando en cuenta que el servicio está montado en AWS, Rubén decide que usará las siguientes herramientas: GitHub, Docker, Jenkins y otros servicios de AWS.

De acuerdo con esta decisión, Rubén crea el repositorio de trabajo en GitHub y levanta una instancia de AWS para montar Jenkins en ella. Por tanto, se dedica a integrar GitHub con Jenkins de modo que cada vez que se haga un *merge* a la rama Master, se efectúe el despliegue. También configura GitHub para ayudar a minimizar errores, como correr las pruebas unitarias con cada *pull request* y verificar que el código cumpla con estándares de calidad como PEP8. Agrega además unos mecanismos de seguridad, como evitar que cualquier rama pueda hacer *merge* al ambiente de producción y solo pueda hacerlo la rama *master*. Por otra parte, integra que el *pull request* sea aprobado por al menos otra persona e implementa notificaciones en cada paso.

Parte del código en AWS está en Lambda y Rubén considera tener el repositorio en contenedores en un mejor acercamiento, por lo que ahora debe crear una imagen para cada *lambda* del proyecto y crear un *stack* en AWS ECR para almacenar las imágenes de Docker; de manera que se pueda escalar, de ser necesario, a otro tipo de servicios, como una ECS. Junto a esto, Rubén decide implementar CloudFormation de AWS para desplegar el código, por lo que debe crear el *template* YAML con los recursos a emplear y que estos se provisionen de manera automática. Para esto también usará AWS SAM para poder ejecutar las órdenes desde un archivo *.sh*

Ahora, el último toque es integrar todo y, de esta manera, orquestar el despliegue en automático de los recursos. Al finalizar, Rubén podrá hacer un despliegue desde su rama local con un archivo *.sh* que contiene las instrucciones, sin necesidad de usar GitHub ni Jenkins, y también el *pipeline* automático para poder subir cambios al ambiente de producción con un clic.

Anexos

Repositorio para desplegar una app en AWS.

<https://github.com/joseFranciscoHS/fullstack-app>