

DevOps Engineer Position at Fz Sports

En Fz Sports ayudamos a alcanzar la mayor audiencia posible a nuestros clientes, quienes son creadores de contenidos para diversos segmentos. En esta ocasión construirás una pequeña API destinada al mundo de los deportes. Para ello detallamos a continuación los lineamientos y recursos disponibles para su creación.

Parte de tener una plataforma multi tenant que sirve a millones de dispositivos, es construir la infraestructura necesaria que permita manejar miles de requests simultáneos y con low latency. Como Ingeniero DevOps nos encontramos con el desafío de manejar los procesos CI/CD que permitan hacer CI/CD de forma eficiente y escalable.

Indicaciones Generales 🧐🧐

- El proyecto deberá estar disponible en un repositorio privado y agregar como contribuyente a leon@fzsports.com y kartigas@fzsports.com. Una vez terminado, haz click en "Submit test" en el link del correo que llegó con la prueba, de esa forma te aseguras que recibamos adecuadamente la prueba.
- La prueba no tiene un tiempo límite pero tomaremos en cuenta como parte de la evaluación el tiempo empleado. Puedes utilizar librerías de terceros según creas conveniente.
- La sección de Infraestructura como código (IaaS) debe de quedar como una subcarpeta del proyecto.
- Agregar test unitarios según considere necesario, se valorará tests de componentes.
- Agregar un archivo README.md con todas las indicaciones necesarias para ejecutar el proyecto incluidos sus test en caso de existir.

Parte Práctica

Crear semilla de datos

Usando el archivo en esta ruta:

<https://fx-nunchee-assets.s3.amazonaws.com/data/sports.xml> crear un script que sea capaz de descargarlo, procesarlo e insertarlo dentro de una base de datos MongoDB.

Es importante al momento de hacer esta carga de datos conservar una referencia al id original de cada elemento en la fuente de datos externa aun cuando Mongo use su propio formato de ids

API REST

Se debe crear un API REST en formato JSON que considere los siguientes puntos:

- Cualquier campo de tipo fecha/hora debe ser devuelto en hora UTC con formato ISO-8601
- Debemos cuidar que caracteres especiales como acentos estén encodeados de manera correcta

A continuación se listan los endpoint que formarán parte de la API:

/api/team

Debe listar todos los equipos disponibles en la base de datos

/api/teams/:idTeam/players

Debe listar los jugadores pertenecientes al equipo con id igual a idTeam

/api/teams/players/:position

Debe listar todos los jugadores de todos los equipos que juegan en la posición position (Arquero, Defensor, etc.) agrupados por su equipo correspondiente

Extra Mile

Además de los endpoints y funcionalidades descritas dejamos a tu criterio el incluir features que consideres importantes y/o útiles

en una API. Te dejamos algunos ejemplos sientente libre de seleccionar entre estos o agregar tus propias ideas:

- autenticación
- Filtros
- Paginado
- Limites de datos
- Formatos alternativos

IaaS

- Debes dockerizar tu aplicación
 - Dentro de tu proyecto debe existir un Dockerfile que construya un container Docker con tu API
 - Debes construir un pipeline que construya de forma automática tu imagen docker.
- Debes codificar un pequeño módulo en **Terraform** que permita crear los siguientes componentes de infraestructura para el provider de AWS:
 - Una VPC con 4 subnets, 2 públicas, 2 privadas
 - Un ALB
 - Un security group para el ALB que permita tráfico por el puerto 80

- Un HTTP listener que responda por default "Hola FzSports!"

Nota: El módulo de terraform no es necesario que le hagas deployment. Sin embargo si tienes la oportunidad de probarlo está ok.