

Docker

Prueba Técnica

Requisitos técnicos

1. Sistema operativo Windows, Linux o MacOS
2. Docker y docker-compose
3. NodeJs y Npm
4. Editor de texto Visual Studio Code, Notepad o similar
5. Una cuenta gratuita en GitLab, GitHub para entregar el proyecto

Objetivos

El objetivo de esta prueba es demostrar conocimientos básicos de docker, docker-compose y de algunas herramientas adicionales donde los contenedores juegan un papel importante en el día a día de un desarrollador de software.

Se asume que habrá algunos conceptos en los que la persona a evaluar necesitará investigar al respecto.

Instrucciones

Se necesita *dockerizar* (i.e: poner en un contenedor) un servidor web capaz de responder a peticiones HTTP de tipo GET en la url relativa `/health` en el puerto 5000 (e.g: GET en `http://localhost:5000/health`)

Para ello, el servidor ha de utilizar NodeJs con algún framework, como Express, que permita escuchar y responder a esas peticiones.

Entregable

En GitLab o GitHub, crear un proyecto/repositorio con lo siguiente:

1. Un archivo `Dockerfile` con todo lo necesario, incluyendo cualquier script o fichero adicional, para que se genere una imagen Docker llamada *rubiko-basic-docker* que exponga en el puerto 5000 un *endpoint* http capaz de responder a peticiones `GET`



/health con un OK. Alguien con acceso a esa imagen docker, debería poder ejecutar lo siguiente `docker run -p 5050:5000 rubiko-basic-docker` para, posteriormente, ser capaz de lanzar una petición http GET a <http://localhost:5050/health> y obtener un resultado 200 con la palabra OK.

2. Un archivo `docker-compose.yml` con dos servicios. Uno de ellos, llamado *web*, debe mapearse al puerto 5050. El otro, llamado *curl*, debe ser capaz de hacer una llamada curl al contenedor *web* y mostrar el resultado de la respuesta
3. Un archivo `README.md` explicando lo que contiene el repositorio y cómo lanzar un contenedor `rubiko-basic-docker` con docker y cómo lanzar el `docker-compose`.
4. (Avanzado) Añade soporte para una variable de entorno `GREETINGS` que el contenedor `rubiko-basic-docker` sea capaz de leer en tiempo de ejecución y sea capaz de mostrar el valor que contiene como respuesta a la llamada GET /health. En otras palabras, si se levanta un contenedor pasándole una variable de entorno con `docker run -p 5050:5000 -e GREETINGS="Hello Rubiko Tech!" rubiko-basic-docker` o, si se pasa la variable en el `docker-compose`, el endpoint GET /health debería responder con ese mismo texto "Hello Rubiko Tech!".