-

A continuación el reto técnico.

Situación

=========

Se requiere implementar la funcionalidad SUMAR mediante un servicio REST, que consiste en:

1. Recibir sólo dos números y sumarlos. (ejemplo: http:///xxxxxxxx:yy/zzzzzz/sumar/10/20 )

2. Devolver un json como respuesta.

      El código (backend) ya está desarrollado en:

      NODE        [https://github.com/arcmop/retobase-nodejs.git](https://github.com/arcmop/retobase-nodejs.git" \t "/home/miguel/Documents\\x/_blank)

      JAVA        [https://github.com/arcmop/retobase-java.git](https://github.com/arcmop/retobase-java.git" \t "/home/miguel/Documents\\x/_blank)

      PHYTON      [https://github.com/arcmop/retobase-python.git](https://github.com/arcmop/retobase-python.git" \t "/home/miguel/Documents\\x/_blank)

      PHP         [https://github.com/arcmop/retobase-php.git](https://github.com/arcmop/retobase-php.git" \t "/home/miguel/Documents\\x/_blank)

Arquitectura Solicitada

=======================

1. 01 nginx como puerta de entrada (reverse proxy)

2. 01 backend, en el lenguaje que tú escojas. (sug. java,node)

3. 01 base de datos, la que tú escojas (sug. postgres)

4. Contenedores (sug. Docker, Podman)

Problemática

============

0. Almacenar los números y el resultado en una base de datos.

1. Generar un bashero/yml que realice la compilación del backend de forma automática, ejecutando los test unitarios.

2. Generar un DockerFile, que permita construir el build de una imagen de Nginx como reverse-proxy.

3. Generar un DockerFile, que permita construir el build de una imagen con en backend.

4. Generar un DockerFile, que permita construir el build de una imagen personalizada con la base de datos.

5. Generar un bashero/yml que realice la construcción automática de las imágenes mediante los archivos DockerFile respectivos.

6. Generar un bashero/yml que permita ejecutar las imágenes de Nginx y el Backend conectando

    a) Nginx ---> Backend ----> Base Datos

7. Continuous Integration  : Realizar el despliegue mediante un toolchain de ALM por ejemplo Github/Gitlab/Bitbucket + Jenkins/Bamboo + Sonarqube

8. Continuous Delivery     : Implementar Docker Registry en local o Cloud Provider (AWS, Azure, GCP, Ibm Cloud, Ali Cloud) para las imágenes de docker, Artifactory, Nexus Sonatype

8. Continuous Deployment   : Realizar un Pipeline (Jenkins, vía código) / Plan (vía código \*Bamboo Specs\*) para el deploy automático

Consideraciones

===============

1. Subir los archivos solicitados a Github/Gitlab/Bitbucket en un repositorio en modo público, de manera que todos los archivos puedan ser descargados para su revisión.

2. Considerar al menos un test unitario para el backend.

3. Considerar el build automático del backend mediante un gestor de dependencias (maven, gradle, etc...)

4. Para la base de datos, considerar solo una tabla simple (sumando01, sumando02, resultado)

5. Considerar la cláusula HEALTCHECK dentro del Dockerfile del backend

6. En caso tuvieras problemas con la instalación de docker puedes usar: "[https://labs.play-with-docker.com](https://labs.play-with-docker.com/" \t "/home/miguel/Documents\\x/_blank)" y recuerda guardar los archivos en tu propio repositorio

7. En caso usaras kubernetes, puedes usar "[https://labs.play-with-k8s.com](https://labs.play-with-k8s.com/" \t "/home/miguel/Documents\\x/_blank)" y recuerda guardar los archivos en tu propio repositorio

Sugerencias

===========

1. Puedes usar Springboot o Micronaut.io, con los cuales tendrás el backend en custión de minutos.

2. En caso algún punto te resulte complicado puedes avanzar con los demás.

3. El Backend debe poder ser multiréplicas, ten cuidado de no exponer el puerto como external.

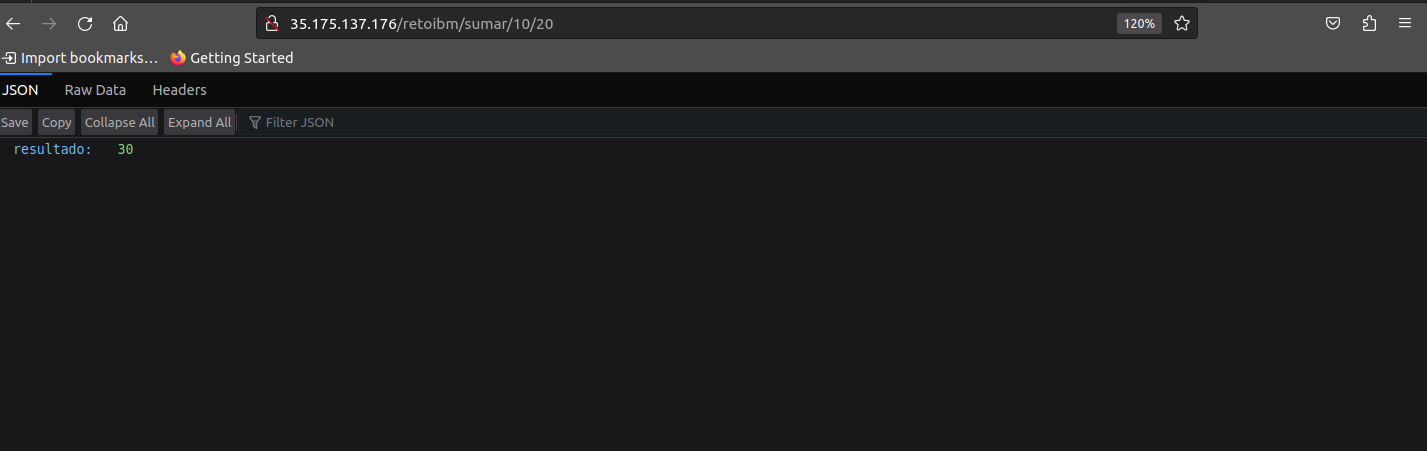
Quedo al pendiente ante cualquier duda, confío en que tendrás éxito en el mismo.

Tienes hasta el día domingo 02 de Abril para entregar el reto.

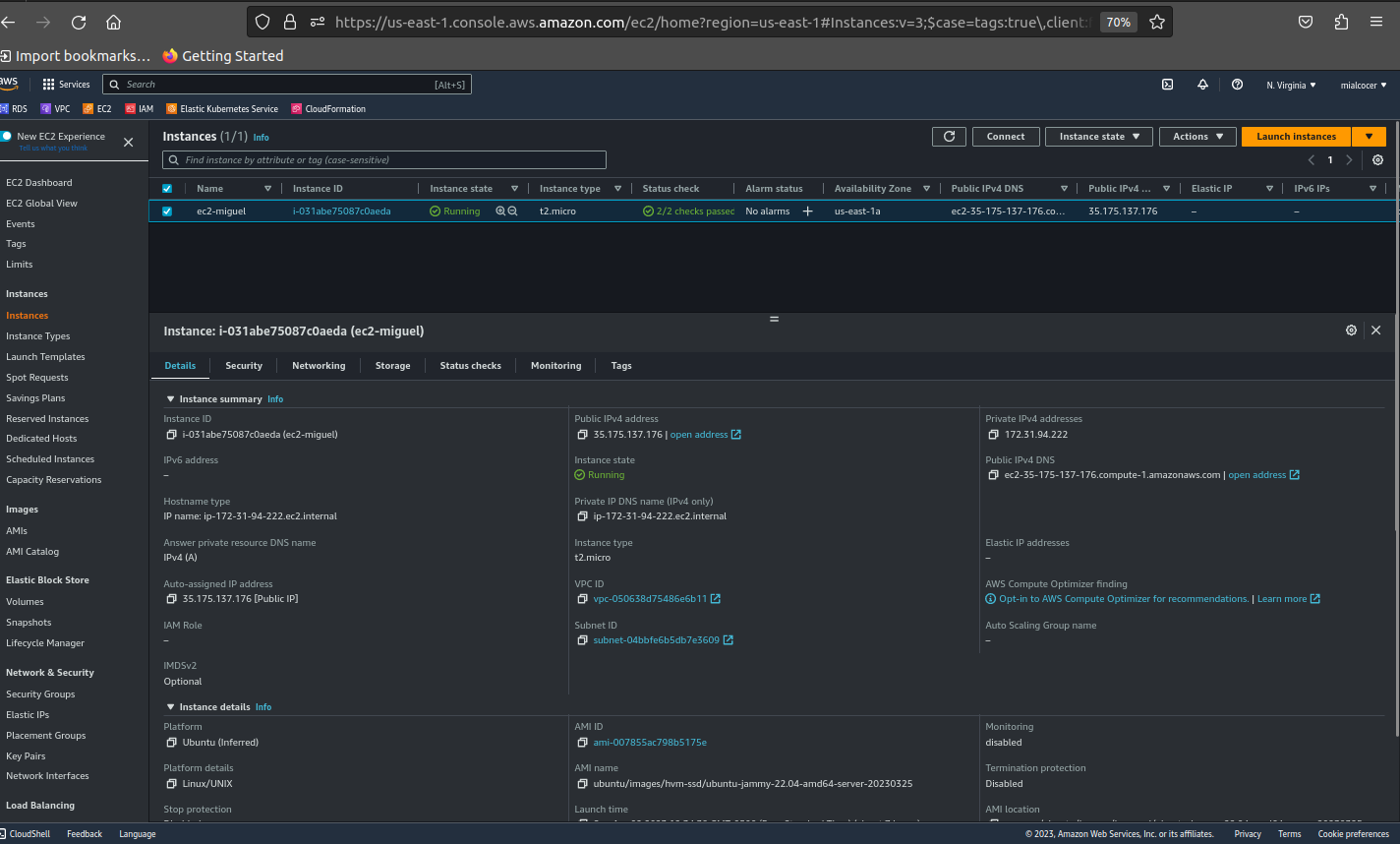
-

**EVIDENCIAS:**

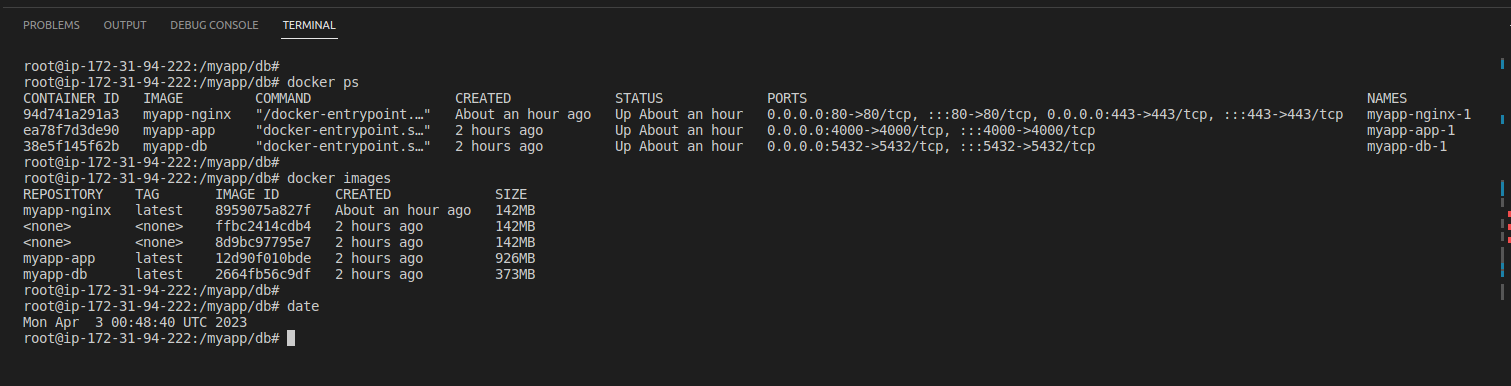
-Link: <http://35.175.137.176/retoibm/sumar/10/20>



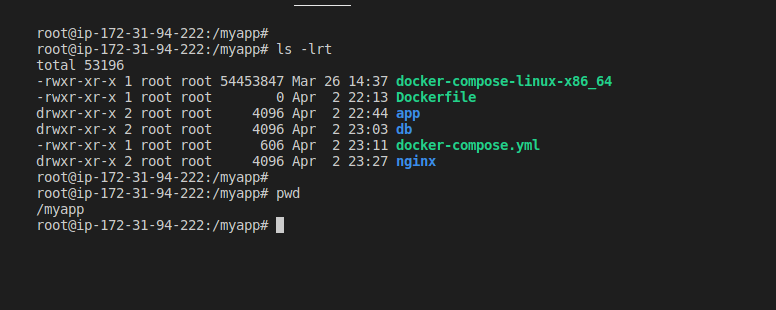
-



-



-



-

