Situación

=========

Se requiere implementar la funcionalidad SUMAR (Backend) 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

                JAVA        https://github.com/arcmop/retobase-java.git

                PHYTON      https://github.com/arcmop/retobase-python.git

                PHP         https://github.com/arcmop/retobase-php.git

Arquitectura Solicitada

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

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

2. 01 backend, en el lenguaje que tú escojas.

4. Contenedores (sug. Docker, Podman)

Problemática (Requerido)

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

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

2. Generar un DockerFile, que permita realizar el build de una imagen con en backend.

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

    a) Nginx ---> Backend

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

3. Considerar la cláusula HEALTCHECK dentro del Dockerfile del backend

Sugerencias

===========

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

2. En caso tuvieras problemas con la instalación de docker puedes usar: "https://labs.play-with-docker.com" y recuerda guardar los archivos en tu propio repositorio

3. En caso usaras kubernetes, puedes usar "https://labs.play-with-k8s.com" y recuerda guardar los archivos en tu propio repositorio

Plus (Opcional)

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

Puedes añadir cualquiera de los siguientes puntos:

01. Realizar el aprovisionamiento de la infraestructura mediante código: Ansible, CloudFormation, Terraform, Cheff, Puppet (otro)

02. Realizar el despliegue sobre Kubernetes, Docker Swarm, Openshift

03. Implementar Idempotencia del servicio: Buscar si dos números ya fueron sumados previamente en la bd y devolver el resultado de la BD (verbo PUT)

04. Incluir integración con JMeter para realizar pruebas de estrés al backend.

05. Crear una cuenta trial en Microfocus e integrar al ALM Fortify SCA.

06. Crear una cuenta trial en Blazemeters/Saucelabs e integrar al ALM las pruebas de estrés.

07. Realizar el auto tag del respositorio, en caso la construcción (CDelivery) sea exitoso.

08. Realizar el auto incremento de la versión en el archivo de propiedades correspondiente (json, pom.xml, etc) y realizar el push de ese cambio.

09. Integrar configuraciones en cortocircuito, al backend. Por ejemplo si el CodeCoverage es menor del 70% el pipeline debe abortar.

10. Integrar tests mediante el lenguaje Gherkins (Scenario, features) y mostrar resultados en la tool de IC (Por ejemplo Plugin Serenity BDD)

11. Continuous Monitoring: Integrar alguna herramienta de monitoreo de aplicación (por ejemplo Dynatrace, New Relic)

12. Continuous Monitoring: Integrar alguna herramienta de monitoreo de salud base de datos (Prometheus)

13. Continuous Feedback: Integrar a la solución alguna herramienta de Bug Tracking System  (Bugzilla, Jira)

14. IT Alerts : Integrar Pagerdutty a la solución

15. Autoscaling: Implementar Autoescalado-Horizontal de acuerdo a demanda, defina ud el criterio.

16. IT Backup: Implemente una herramienta de backup automático para la base de datos.

17. Implementar el modelo maestro - esclavo a la base de datos.

18. Implementar caché a las consultas con Redis.

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

20. 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

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

22. ChatOps - Al construir correctamente o presentarse error el pipeline/plan debe enviar un mensaje a un grupo de Telegram/Slack creado para informar sobre los builds del proyecto. Al momento de la prueba agregaremos un número/email para comprobar la configuración.

23. AgileOps: Integrar el stack ELK o Graylog+Grafana a la solución, mostrando un "Visualization"/"Dashboard" con la cantidad de peticiones por minuto.

24. SRE: Integrar una tool de monitoreo de contenedores (por ejemplo: Portainer) a la solución.