

Práctica: Contenedores, más que VMs

Descripción general

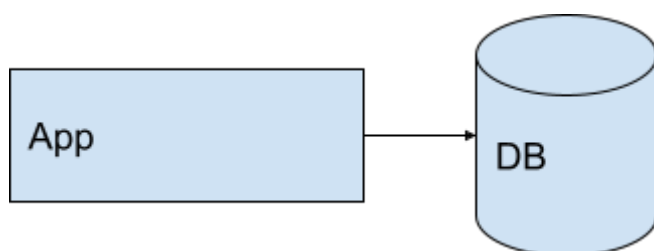
En esta práctica debemos implementar una aplicación consistente en un micro servicio que sea capaz de leer y escribir en una base de datos.

La aplicación tendrá que poderse desplegar de tres formas distintas: en local mediante la herramienta docker-compose, en un cluster Kubernetes mediante los manifiestos entregados y a través de la herramienta de gestión de paquetes helm.

El microservicio / aplicación y la base de datos son de tu elección.

Puedes basarte en el flask-counter que hemos estudiado durante el curso, desarrollar tu propia aplicación sencilla o basarte en alguna aplicación existente de tu elección (NodeJS, Java, Python, Go, ...).

Puedes usar cualquier lenguaje de programación y cualquier framework.



Hitos

1. Crear **repositorio GIT** con la aplicación.
2. Crear un fichero **README.md** con todas las instrucciones necesarias para poder comprender el proyecto y ponerlo en funcionamiento.
 - a. Descripción de la aplicación.
 - b. Funcionamiento de la aplicación.
 - c. Requisitos para hacerla funcionar.
 - d. Instrucciones para ejecutarla en local y verificar el funcionamiento correcto.
 - e. Instrucciones para desplegarla en Kubernetes, incluyendo requerimientos, dependencias, tipos de Kubernetes soportados, etc.
3. **Dockerfile** que construya el / los contenedores necesarios:
 - a. Que compile, instale dependencias, etc.
 - b. Que la empaque con los requisitos mínimos (usar **Multistage**)

4. Docker **compose**
 - a. Que permita ejecutar la aplicación completa en local, considerando la persistencia de datos y comunicaciones entre distintos contenedores.
 - b. Instrucciones para verificar el funcionamiento correcto.
5. **Logs de la aplicación:**
 - a. Formato JSON a ser posible **[OPCIONAL]**
 - b. Asegurarse de que todos los componentes (aplicación y base de datos) mandan sus logs por la salida estándar y salida de error (STDOUT / STDERR).
6. **Configurabilidad de la aplicación:**
 - a. La aplicación debe de poder ser configurable, por ejemplo, host de la base de datos, puerto, usuario, contraseña, etc. **Mediante fichero de configuración y/o variables de entorno.**
 - b. Trata de dotar de la mayor flexibilidad posible al microservicio y documenta todas las opciones de configuración en el README.
7. Generación de **manifiestos de Kubernetes** en **directorio k8s/**, mínimo:
 - a. Has de usar las imágenes creadas anteriormente y subidas a internet (Docker Hub).
 - b. Utiliza un Deployment para la aplicación.
 - c. Crea Servicios para todo lo que necesite estar expuesto (tanto al interior como al exterior del cluster).
 - d. Utiliza un StatefulSet para la base de datos (con persistentvolumeclaim).
 - e. Añade un Ingress para exponer la aplicación al exterior a través de un Ingress Controller.
 - f. ConfigMap(s) para datos de configuración NO sensibles.
 - g. Secret(s) explicando en la documentación cómo hay que generarlos.
 - h. **README K8s:**
 - i. Describe los recursos creados y documenta todo lo que sea configurable.
 - ii. Incluye instrucciones de despliegue en kubernetes y cómo probar la aplicación y el correcto funcionamiento.
8. Generación de **chart de Helm** en directorio **charts/**, mínimo:
 - a. deployment
 - b. service
 - c. persistentvolumeclaim (para la BBDD)
 - d. ingress
 - e. configmap
 - f. secret
 - g. **README Helm Chart:** Trata de dotar de flexibilidad y configurabilidad al chart de helm, explicando todo lo que se pueda modificar en el fichero values.yaml. Por ejemplo, habilitar o deshabilitar Ingress, permitir exponer la aplicación con un servicio del tipo LoadBalancer, etc.

REQUISITOS KUBERNETES / HELM:

9. Asegurar que los PODs de la base de datos y la aplicación permanezcan lo más juntos posibles al desplegarse en Kubernetes.
10. Asegurar que los PODs de las réplicas de la aplicación permanezcan lo más separados posibles entre ellos.
11. Autoescalar la aplicación (no la base de datos) cuando pase de un umbral de uso de CPU del 70% (**hacerlo opcional en helm mediante variable**), asegurando siempre una alta disponibilidad.
12. Usar almacenamiento persistente provisionado de forma dinámica para la base de datos (en el StatefulSet).
13. Configuración externa mediante Secret y ConfigMap (nada "hardcoded"), esto es **FUNDAMENTAL**.
14. Instalar Ingress Controller: Nginx (<https://kubernetes.github.io/ingress-nginx/deploy/>) incluyendo los manifiestos y/o instrucciones necesarias en el proyecto (README)
15. Exponer la aplicación públicamente mediante Ingress. Puede usarse el servicio de nip.io si no se dispone de DNS (<https://nip.io/>).

EXTRAS OPCIONALES:

16. **Centralización de Logs** [OPCIONAL] : Proporciona una solución de centralización de todos los logs de tu aplicación (o de todo el cluster) utilizando el stack de Elastic (Elasticsearch, Kibana y Filebeat). Puedes usar sidecar containers con Filebeat en tus workloads o desplegar Filebeat como DaemonSet en todos los nodos de Kubernetes.
17. Exposición de **métricas** [OPCIONAL]: Proporciona una solución de métricas para algún componente del ecosistema. Para ello puedes utilizar el Elastic Stack (Elasticsearch, Kibana, Metricbeat) ó una solución basada en Prometheus (Prometheus, Grafana e instalar algún exporter). Puedes centrarte en las métricas de la base de datos, de la aplicación o a nivel de Kubernetes (monitorización de Kubernetes).
18. Creación de algún **dashboard** para la visualización de los logs y/o métricas, ó explica en las instrucciones cómo verificar que los logs y las métricas están llegando al destino [OPCIONAL].
19. Instala y utiliza algún **Operador**, por ejemplo: el operador de Elastic (ECK), ó Prometheus Operator [OPCIONAL].
20. Despliegue de Kubernetes a través de [Kops](#) en Google Cloud o AWS. Para ello añade un directorio **kops** al proyecto con las instrucciones y ficheros necesarios. [OPCIONAL].



Formato de entrega

- La práctica ha de entregarse como repositorio GitHub dentro de la organización Keepcoding, no se aceptarán entregas en cuentas de GitHub personales, ya sean los repositorios públicos o privados.
- La entrega ha de comunicarse mediante el formulario oficial de entrega, no se aceptan o revisan entregas comunicadas por ningún otro medio.