

Prueba: Docker e Integración continúa

Nombre: _____

Docker

Queremos "dockerizar" la aplicación **CRUD-PHP**. Dicha aplicación, escrita en PHP, nos permite crear, modificar y borrar datos de usuarios que se guardan en una base de datos mysql.

Descomprime el fichero `crudphp.zip` donde encontraras en el directorio `src` donde está la aplicación y el fichero `script.sh` que será el script bash que ejecuta el contenedor al crearse.

1. Crea un fichero `Dockerfile` que te permita crear una imagen con la aplicación PHP, teniendo en cuenta lo siguiente:
 - Puedes usar la imagen base que quieras.
 - Recuerda instalar la librería PHP que permite conectar a la base de datos.
 - Debes crear 4 variables de entorno con valores por defecto: `DB_HOST`, `DB_NAME`, `DB_USER` y `DB_PASSWORD`.
 - El contenedor al crearse ejecutara el script `script.sh`.
2. Crea un `docker-compose.yml` que permita la implantación de dos contenedores. Uno usando la imagen que has generado y otro con la imagen `mariadb`. Debes asegurarte que la base de datos es persistente.

¿Qué debes entregar?

1. El fichero `Dockerfile` y `docker-compose.yml` que has generado.
2. Un pantallazo de pantalla donde se vea la ejecución de la aplicación con usuario creado.
3. Después de borrar el escenario (sin borrar el volumen) un pantallazo que sigue mostrando el usuario que has creado.

IC / DC

Basándote en el pipeline que has desarrollado en la práctica. Realiza un pipeline que despliegue la aplicación en tu VPS. (Tienes que tener en cuenta que esta aplicación no tienen test, por lo tanto esa etapa la tienes que quitar).

¿Qué debes entregar?

1. El fichero `Jenkinsfile` que has desarrollado.
2. Una captura de pantalla donde se vean las etapas que se han ejecutado de forma correcta (en verde) en Jenkins.
3. Una captura de pantalla accediendo al VPS donde se vea la ejecución de la aplicación.